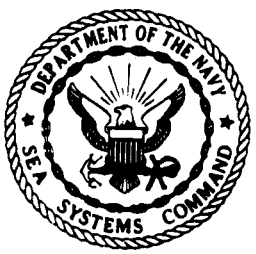
NAVSEA OP 3514 (PMS/SMS) VOLUME 1

FIRST REVISION
CHANGE 7

DIGITAL COMPUTER MK 152 SERIES

DESCRIPTION, OPERATION, AND MAINTENANCE



THIS PUBLICATION SUPERSEDES OP 3514 (PMS/SMS)
DATED 1 JANUARY 1972

PUBLISHED BY DIRECTION OF
COMMANDER, NAVAL SEA SYSTEMS COMMAND

1 JANUARY 1973
CHANGED 1 JULY 1978

LIST OF EFFECTIVE PAGES

Page No.	*Change No.	Page No.	*Change No.
Title and A	7	2-120	5
B/(C blank)	7	2-121 thru 2-129	0
Flyleaf/Flyleaf-2	0	2-130 and 2-131	7
Foreword and Foreword-2	7	2-132 thru 2-139	0
Foreword-3/(Foreword-4 blank)	7	2-140	5
i thru vii	0	2-141 thru 2-150	0
viii and ix	7	2-151/(2-152 blank)	7
x	0	2-153 thru 2-158	0
xi	7	2-159	7
xii	0	2-160	0
xiii	7	2-161	7
xiv thru xvi	0	2-162 thru 2-173	0
xvii thru xix	7	2-174	5
xx	0	2-175	0
xxi	7	2-176	5
xxii	0	2-177 thru 2-198	0
xxiii thru xxvii	7	2-199	7
xxviii	0	2-200 thru 2-213	0
xxix/(xxx blank)	7	2-214	5
xxxi and xxxii	0	2-215 thru 2-267	0
1-1 thru 1-9	0	2-268	5
1-10	5	2-269 thru 2-286	0
1-11 thru 1-13	0	2-287	2
1-14 and 1-15	5	2-288 thru 2-291	0
1-16 and 1-17/(1-18 blank)	7	2-292	7
2-1	5	2-293 thru 2-295	0
2-2 thru 2-5	0	2-296	5
2-6	7	2-297	0
2-7 thru 2-12	0	2-298	5
2-13	5	2-299	7
2-14	7	2-300	5
2-15 thru 2-17	5	2-301 thru 2-305	0
2-18	7	2-306	5
2-19 thru 2-21/(2-22 blank)	0	2-307 thru 2-315	0
2-23	5	2-316	5
2-24 thru 2-26	0	2-317 and 2-318	0
2-27	5	2-319	5
2-28 thru 2-76	0	2-320 thru 2-322	0
2-77	5	2-323	5
2-78 thru 2-101	0	2-324 and 2-325	0
2-102	5	2-326 and 2-327	7
2-103	0	2-328	0
2-104 and 2-105	5	2-329	5
2-106	0	2-330	0
2-107 and 2-108	5	2-331	5
2-109	7	2-332 and 2-333/(2-334 blank)	0
2-110 and 2-111	5	2-335 and 2-336	0
2-112 thru 2-118	0	2-337 and 2-338	5
2-119	7	2-339 and 2-340	0

*Zero in this column indicates an original page.

LIST OF EFFECTIVE PAGES (Continued)

Page No.	*Change No.
2-341	5
2-342 thru 2-344	0
2-345	5
2-346 thru 2-368	0
2-369/(2-370 blank)	5
2-371 thru 2-376	0
2-377	5
2-378 thru 2-397	0
2-398	5
2-399	0
2-400 thru 2-448	0
2-449	7
2-450	0
2-451	5
2-452 and 2-453	0
2-454 thru 2-456	7
2-457	7
2-458	5
2-459 and 2-460	0
2-461	5
2-462	0
2-463 and 2-464	5
2-465 thru 2-473	0
2-474 thru 2-476	7
2-477 and 2-478	0
2-479 and 2-480	5
2-481 thru 2-483	0
2-484 and 2-485	5
2-486 thru 2-493/(2-494 blank)	0
3-1 thru 3-26	0
4-1 thru 4-15	0
4-16	7
5-1	6
5-2	7
5-3	0
5-4 and 5-5	6
5-6 thru 5-10	0
5-11	6
5-12	0
5-13 and 5-14	7
5-14A/(5-14B blank)	7
5-15 and 5-16	7
5-16A/(5-16B blank)	6
5-17	0
5-18 and 5-19	0
5-20 and 5-21	7
5-22 and 5-23/(5-24 blank)	6

*Zero in this column indicates an original page.

NAVORD OP 3514 (PMS/SMS) VOLUME 1 FIRST REVISION

CHANGE RECORD

CHANGE NO.	DATE	TITLE AND/OR BRIEF DESCRIPTION	SIGNATURE OF VALIDATING OFFICER
CHANGE #6	26 AUG 86	VOL. 1 FIRST REV. CH. 6	Mark a. Lewelle

FOREWORD

NAVSEA OP 3514 (PMS/SMS) First Revision physically and functionally describes the Digital Computer Mark 152 Series and provides all other information required by shipboard personnel for operation, installation, and maintenance of the equipment.

OP 3514 (PMS/SMS) is one of a family of Ordnance Publications (OP's) which provide comprehensive coverage of all modifications of Digital Computer Mk 152 and associated peripheral equipments. These manuals include the following:

Manual	Nomenclature of Equipment
OP 3514 (PMS/SMS) First Revision	Digital Computer Mk 152 Series
OP 3515 (PMS/SMS)	Input/Output Console Mk 77 Series
OP 3516 (PMS/SMS)	Digital Data Recorder Mk 19 Series
OP 3517 (PMS/SMS)	Control Panel Mk 298 Series
OP 3518 (PMS/SMS)	Motor Generator Set Mk 9 Series
OP 4245 (PMS/SMS)	Input/Output Console Mk 95 Series

Although not included in the foregoing list, a Signal Data Converter (SDC) is an essential part of any digital Fire Control System (FCS) which employs component parts originally designed for analog operation. The TARTAR, TALOS, and TERRIER weapon systems use SDC's Mk 72, Mk 66, and Mk 75, respectively. Gun Fire Control System (GFCS) Mk 86 uses SDC Mk 69/Mk 70 and Signal Data Translator (SDT) Mk 1 for data conversion.

The digital computer complex equipment configuration found on various gun and guided missile ships depends principally on the shipboard weapons system(s). While other factors can influence the actual configuration on board a particular ship, the following equipments are generally utilized with the missile/gun systems indicated:

Equipment	TARTAR	TERRIER	TALOS	GFCS Mk 86
Digital Computer	Mk 152	Mk 152	Mk 152	Mk 152
I/O Console	Mk 77	Mk 77	Mk 77	Mk 77
I/O Console	Mk 95	-------	-------	-------
Digital Data Recorder	Mk 19	Mk 19	Mk 19	-------
Control Panel	Mk 298	Mk 298	Mk 298	-------
Motor Generator Set	Mk 9	Mk 9	Mk 9	-------
Signal Data Converter	Mk 72	Mk 75	Mk 66	Mk 69/Mk 70
Signal Data Translator	-------	-------	-------	SDT Mk 1

This manual is applicable to Digital Computers Mk 152 Series configured as follows:

MOD NO.	IL NUMBER	UNIVAC NUMBER	MODULES (DRAWERS)	MEMORY SIZE	I/O CHANNELS
0	2652595	7049747-00	3(6)	32K	16 slow interface
1	2525383	7049747-05	2(4)	32K	8 slow interface
2	2536007	7049747-08	3(6)	32K	8 slow interface 8 fast interface
3	2687278	7049747-11	3(6)	40K	16 slow interface
4	2923042	7049747-13	3(6)	40K	8 slow interface 8 fast interface
5	3140291	7049747-14	2(4)	48K	8 slow interface

This publication consists of three volumes structured as follows:

VOLUME 1 - Description, Operation, and Maintenance
- Chapter 1. Introduction
- Chapter 2. Description
  - Section 2-1. Physical Description
  - Section 2-2. General Functional Description
  - Section 2-3. Detailed Functional Description
- Chapter 3. Operation
- Chapter 4. Installation
- Chapter 5. Maintenance

VOLUME 2 Part 1 - Trouble Isolation
- Chapter 6. Introduction
- Chapter 7. Supporting Maintenance Data
  - Section 7-1. General Troubleshooting Information
  - Section 7-2. Manual Troubleshooting
  - Section 7-3. Troubleshooting Charts
  - Section 7-4. Schematic Diagrams of Printed Circuit Cards

VOLUME 2 Part 2 and Part 2A-Trouble Isolation
- Chapter 8. Diagnostic Program

VOLUME 2 Part 3 - Trouble Isolation
- Chapter 9. Functional Schematics

VOLUME 3 - Parts List
- Chapter 10. Introduction
- Chapter 11. Replaceable Parts List

Ships, training activities, supply points, depots, Naval shipyards and supervisors of shipbuilding are requested to arrange for the maximum practical use and evaluation of NAVSEA technical manuals. All errors, omissions, discrepancies, and suggestions for improvements to NAVSEA technical manuals shall be reported to the Naval Sea Data Support Activity (NSDSA), Naval Ship Weapon Systems Engineering Station (Code 5740), Port Hueneme, CA 93043 on NAVSEA Technical Manual Deficiency/Evaluation Report,

NAVSEA Form 5600/2. To facilitate such reporting, three copies of Form NAVSEA 5600/2 are included at the end of each unclassified bound part of this technical manual being changed. All feedback comments will be thoroughly investigated and originators will be advised of action resulting therefrom. Extra copies of Form NAVSEA 5600/2 may be requisitioned from the Naval Publications and Forms Center (NPFC), Philadelphia, PA 19120.

The technical content of the manual covers Digital Computer Mk 152 Series as modified by ORDALTs 7732, 7889, 8008, 8331 Change 1, 30023, 8339, 8286, 8409, 8490, and 30094.

TABLE OF CONTENTS

Chapter Page

LIST OF ILLUSTRATIONS xviii

LIST OF TABLES xxiii

SAFETY SUMMARY xxxi

1. INTRODUCTION 1-1

1-1. Purpose 1-1
1-3. Functions 1-1
1-4. Capabilities 1-4
1-6. General Characteristics 1-4
1-10. Operational Characteristics 1-5
1-11. System Description 1-5
1-12. Weapons System Relationship 1-5
1-13. Computer Complex 1-5
1-14. Computer Group Equipments 1-5
1-15. Model Differences 1-14
1-17. Reference Data 1-14
1-19. Equipment Supplied 1-14
1-20. Special Tools and Test Equipment 1-14
1-21. Equipment Nomenclature 1-14
1-22. Reference Publications 1-16

2. DESCRIPTION 2-1

Section 2-1. Physical Description

2-1. Introduction 2-1
2-2. Computer Cabinet 2-1
2-3. Basic Cabinet 2-1
2-4. Expanded Cabinet 2-1
2-5. Remote Control Console 2-1
2-6. Approximate Size and Weight 2-1
2-7. Cooling 2-1
2-8. Fuse Compliment 2-1
2-9. Environmental Requirements 2-1
2-10. Cabinet Drawers and Chassis 2-3

TABLE OF CONTENTS (Cont)

Chapter			Page
	2-11.	Logic Drawers	2-3
	2-12.	Mounting	2-3
	2-13.	Drawer Locking	2-3
	2-14.	Logic Chassis	2-3
	2-15.	Memory Drawer	2-3
	2-16.	Control and Bootstrap Memories	2-3
	2-17.	Power Supplies	2-3
	2-18.	Main Memory Power Supply	2-3
	2-19.	Panels	2-10
2-20.	Computer Assemblies		2-10
2-22.	Module Compliment		2-10
2-23.	Physical Description of Basic Modules		2-10

Section 2-2. General Functional Description

2-27.	General Description		2-19
2-29.	Control Section		2-19
2-31.	Console Control		2-19
2-32.	Timing Circuits		2-23
2-34.	Program Translation and Control		2-23
2-35.	Registers		2-25
2-36.	Special Circuits		2-27
2-37.	Arithmetic Section		2-27
2-38.	Memory Section		2-28
2-39.	Main Memory		2-28
2-40.	Control Memory		2-29
2-41.	Bootstrap Memory		2-29
2-42.	Bootstrap Mode		2-29
2-43.	I/O Section		2-32
2-44.	Major I/O Operations		2-35
2-45.	Data Transfers		2-35
2-46.	I/O Communication Modes		2-35
2-47.	Input/Output Signals		2-36
2-48.	Data Request Signals		2-36
2-49.	Real Time Clock (RTC) Request Signals		2-36
2-50.	Interrupt Signals		2-36
2-51.	Priorities		2-37

TABLE OF CONTENTS (Cont)

Chapter			Page
		2-52. I/O Circuitry	2-37
2-53.	Instructions Analysis		2-38
	2-54.	Instructions Repertoire	2-38
		2-55. Symbols Used	2-38
		2-56. Instruction Word Formats	2-38
		2-57. List of Instructions	2-38
2-58.	Logic		2-73
	2-59.	Internal Logic	2-73
	2-60.	External Logic	2-73
2-61.	Symbology		2-73
	2-62.	Circuit Symbology	2-73
		2-63. Inverter	2-73
		2-64. AND Inverters	2-73
		2-65. AND-OR Inverters	2-73
		2-66. OR Inverter	2-73
		2-67. Flip-Flops	2-73
		2-68. Amplifier Driver	2-77
		2-69. Input Amplifiers	2-77
		2-70. Output Amplifiers and Line Drivers	2-77
		2-71. Indicator Driver	2-77
	2-72.	Reference Designators	2-77
		2-73. Unique Designators	2-77
		2-74. Alpha Assignments	2-78
		2-75. Numeric Assignments	2-79
		2-76. Card Location Designators	2-79
		2-77. Test Block Designators	2-79
		2-78. Connector Designators	2-81
		2-79. Special Symbology	2-81
	2-80.	Duplicate Circuits	2-81
	2-81.	Chassis Maps	2-81

Section 2-3. Detailed Functional Description (Control Section)

2-82.	Introduction		2-87
2-83.	Control Section		2-87
	2-85.	Console Control	2-87
		2-86. Start/Stop Logic	2-87
		2-87. Start Circuit	2-87

TABLE OF CONTENTS (Cont)

Chapter			Page
		2-93. Stop Circuit	2-91
	2-94.	Master Clear	2-91
		2-99. I/O Clear	2-91
	2-100.	Mode Select Logic	2-92
	2-101.	Run Mode	2-92
	2-103.	Phase Step Mode	2-92
	2-108.	OP Step Mode	2-94
		2-111. Load Mode	2-94
2-113.		Timing Circuits	2-94
	2-114.	Master Clock	2-94
		2-118. Phase 1 (ø 1)	2-95
		2-119. Phase 2 (ø 2)	2-95
		2-120. Phase 3 (ø 3)	2-95
		2-121. Phase 4 (ø 4)	2-95
	2-123.	Clock Distribution	2-96
	2-125.	Timing Chains	2-96
2-131.		Program Translation and Control	2-100
	2-132.	F-Register	2-100
	2-136.	Function Code Subtranslators	2-102
		2-137. Subtranslator 1	2-102
		2-138. Subtranslator 2	2-102
		2-139. Subtranslator 3	2-102
	2-140.	Function Translators	2-103
	2-141.	Sequencer	2-103
		2-142. I-Sequence	2-103
		2-143. R-Sequence	2-103
		2-144. W-Sequence	2-103
		2-145. B-Sequence	2-103
		2-146. I/O Sequence	2-103
		2-147. Interrupt Sequence	2-103
		2-148. Upper-Rank Sequence Designator	2-103
		2-153. Lower-Rank Sequence Designator	2-104
2-158.		Major Sequence Analysis	2-105
	2-159.	I (Initiate) Sequence	2-105
	2-164.	Advance-P Subsequence	2-106
	2-165.	R (Read) Sequence	2-106
	2-167.	W (Write) Sequence	2-122
2-169.		Command Instructions	2-125
	2-170.	Execution of Stop Instruction (f= 50:56)	2-125

TABLE OF CONTENTS (Cont)

Chapter		Page
2-172.	Execution of ENTALK, ADDALK Instructions (f= 70, 71)	2-125
2-174.	Execution of ENTICR, ENTSR Instruction (f= 50:72, 50:73)	2-128
2-175.	I-Sequence Data Flow for f= 50:72, 50:73	2-128
2-176.	Execution of ENTBK, ENTBKB Instructions (f= 36, 37)	2-128
2-177.	I and Next I-Sequence Data Flow for f= 36, 37	2-128
2-178.	Execution of ENTB, ENTBB Instructions (f= 32, 33)	2-137
2-179.	R1 and Next I-Sequence Data Flow for f= 32, 33	2-137
2-180.	Execution of ENTAU, ENTAUB, ENTAL, ENTALB Instructions (f= 10, 11, 12, 13)	2-137
2-181.	ENTAU (f= 10)	2-137
2-182.	ENTAUB (f= 11)	2-137
2-183.	ENTAL (f= 12)	2-137
2-184.	ENTALB (f= 13)	2-137
2-185.	R1-Sequence Data Flow for (f= 10-13)	2-137
2-186.	Execution of CL, CLB, STRB, STRBB, STRAL, STRALB, STRAU, STRAUB Instructions (f= 40, 41, 42, 43, 44, 45, 46, 47)	2-138
2-187.	CL (f= 40)	2-138
2-188.	CLB (f= 41)	2-138
2-189.	STRB (f= 42)	2-138
2-190.	STRBB (f= 43)	2-138
2-191.	STRAL (f= 44)	2-138
2-192.	STRALB (f= 45)	2-138
2-193.	STRAU (f= 46)	2-138
2-194.	STRAUB (f= 47)	2-138
2-195.	W-Sequence Data Flow for f= 40-47	2-138
2-196.	Execution of STRICR, STRADR, STRSR Instructions (f= 72, 74, 75)	2-145

TABLE OF CONTENTS (Cont)

Chapter			Page
	2-197.	STRICR (f= 72)	2-145
	2-198.	STRADR (f= 74)	2-145
	2-199.	STRSR (f= 75)	2-145
	2-200.	W-Sequence Data Flow for f= 72, 74, 75	2-145
2-201.		Execution of Jump (JP, JPB, JPAUZ, JPALZ, JPAUNZ, JPALNZ, JPAUP, JPALP, JPAUNG, JPALNG, AND BJP) Instructions (f= 34, 35, 60-67, and 73)	2-145
	2-202.	JP (f= 34)	2-145
	2-203.	JPB (f= 35)	2-153
	2-204.	JPAUZ (f= 60)	2-153
	2-205.	JPALZ (f= 61)	2-153
	2-206.	JPAUNZ (f= 62)	2-153
	2-207.	JPALNZ (f= 63)	2-153
	2-208.	JPAUP (f= 64)	2-153
	2-209.	JPALP (f= 65)	2-153
	2-210.	JPAUNG (f= 66)	2-153
	2-211.	JPALNG (f= 67)	2-153
	2-212.	BJP (f= 73)	2-153
	2-213.	I-Sequence Data Flow for f= 34, 35, and 60 through 67, and I and Next I-Sequence Data Flow for f= 73	2-153
2-214.		Execution of Jump (IJPEI, IJP, RJP, IRJP, and IRJPB) Instructions for (f= 54, 55, 76, 30, and 31)	2-161
	2-215.	IJPEI (f= 54)	2-161
	2-216.	IJP (f= 55)	2-161
	2-217.	RJP (f= 76)	2-161
	2-218.	IRJP (f= 30)	2-161
	2-219.	IRJPB (f= 31)	2-161
	2-220.	R1-Sequence Data Flow for f= 54, 55, W-Sequence Data Flow for f= 76, and R1 and W-Sequence Data Flow for f= 30, 31	2-161
2-221.		Execution of Skip (SKP, SKPNBO, SKPOV, and SKPNOV) Instructions (f= 50:50, 50:51, 50:52, and 50:53)	2-161

TABLE OF CONTENTS (Cont)

Chapter			Page
		2-222. SKP (f= 50:50)	2-167
		2-223. SKPNBO (f= 50:51)	2-167
		2-224. SKPOV (f= 50:52)	2-167
		2-225. SKPNOV (f= 50:53)	2-167
		2-226. I-Sequence Data Flow for f= 50:50 through 50:53	2-167
	2-227.	Execution of Skip (BSK) Instruction (f= 56)	2-167
		2-228. R1 and Next I-Sequence Data Flow for f= 56	2-167
	2-229.	Execution of SKP (ISK) Instruction (f= 57)	2-175
		2-230. R1 and W-Sequence Data Flow for f= 57	2-175
	2-231.	Execution of Skip (SKPODD, SKPEVN) Instructions (f= 50:54, 50:55)	2-175
		2-232. I-Sequence Data Flow for f= 50:54, 50:55	2-175
Section 2-3. Detailed Functional Description (Arithmetic Section)			
2-233.	Arithmetic Section		2-183
	2-234. Adders		2-183
		2-235. X-D' Adder	2-183
		2-236. Half-Subtractor	2-183
		2-237. Borrow Generator	2-183
		2-240. Full Adder	2-184
	2-244.	B Adder	2-185
	2-248.	KO-1 Adder	2-187
	2-251. Logic Networks		2-189
	2-252.	AU, AL Positive, Negative	2-189
	2-253.	AU, AL Zero	2-189
	2-254.	Borrow	2-189
	2-255.	Overflow	2-189
	2-257.	Parity	2-191
		2-258. Group Parity	2-191
		2-259. Section Parity	2-193
		2-260. Total Parity	2-193
	2-261.	K-Zero	2-195
	2-262.	B-Zero	2-195

TABLE OF CONTENTS (Cont)

Chapter			Page
2-263.	Arithmetic and Logical Instructions		2-196
	2-264.	Add-Subtract Instructions	2-196
		2-265. ADDAL (f= 14)	2-196
		2-266. ADDALB (f= 15)	2-196
		2-267. SUBAL (f= 16)	2-196
		2-268. SUBALB (f= 17)	2-196
	2-269.	R1-Sequence Data Flow for f= 14 through 17	2-196
	2-270.	ADDA (f= 20)	2-196
	2-274.	ADDAB (f= 21)	2-200
	2-275.	SUBA (f= 22)	2-200
	2-276.	SUBAB (f= 23)	2-200
	2-277.	R1 and R1/R2 - Sequence Data Flow for f= 20 through 23	2-200
	2-279.	RND (f= 50:60)	2-204
	2-280.	I-Sequence Data Flow for f= 50:60	2-204
	2-281.	Logical Instructions	2-209
	2-282.	CMAL (f= 02)	2-209
	2-283.	CMALB (f= 03)	2-209
	2-284.	CMSK (f= 06)	2-209
	2-285.	CMSKB (f= 07)	2-209
	2-286.	R1-Sequence Data Flow for f= 02, 03, 06, and 07	2-209
	2-287.	Clearing of Comparison Flip-Flops	2-209
	2-288.	SLSU (f= 04)	2-213
	2-289.	SLSUB (f= 05)	2-213
	2-290.	SLSET (f= 51)	2-213
	2-291.	SLCL (f= 52)	2-213
	2-292.	SLCP (f= 53)	2-213
	2-293.	R1-Sequence Data Flow for f= 04, and 05	2-213
	2-294.	R1-Sequence Data Flow for f= 51 and 52	2-215
	2-295.	R1-Sequence Data Flow for f= 53	2-215
	2-296.	R1-Sequence Essential Commands	2-218
	2-297.	CPAL (f= 50:61)	2-218
	2-298.	CPAU (f= 50:62)	2-218
	2-299.	CPA (f= 50:63)	2-218
	2-300.	I and Next I-Sequence Data Flow for f= 50:61, 50:62, and 50:63	2-218
	2-301.	Shift Instructions	2-222

TABLE OF CONTENTS (Cont)

Chapter Page

2-302. RSHAU (f= 50:41) 2-222
2-303. RSHAL (f= 50:42) 2-222
2-304. RSHA (f= 50:43) 2-222
2-305. SF (f= 50:44) 2-222
2-306. LSHAU (f= 50:45) 2-222
2-307. LSHAL (f= 50:46) 2-225
2-308. LSHA (f= 50:47) 2-225
2-309. Instruction Sequence 2-225
2-312. Initial Shift Count Equal to Zero or $A_{35i} \neq A_{34i}$ 2-226
2-313. Initial Shift Count Not Equal to Zero or $A_{35i} = A_{34i}$ for f= 50:44 2-231
2-314. W-Sequence Data Flow for SF Instruction (f= 50:44) 2-231
2-315. Multiply Instructions 2-231
2-316. MULAL (f= 24). 2-232
2-317. MULALB (f= 25) 2-232
2-319. R1 and Multiply Sequence Data Flow 2-242
2-321. Divide Instructions 2-245
2-322. DIVA (f= 26) 2-245
2-323. DIVAB (f= 27) 2-245
2-324. R1 and Divide Sequence Data Flow. . . 2-245

Section 2-3. Detailed Functional Description (Memory Section)

2-326. Memory Section 2-259
2-329. 16-BIT S_1-Register 2-259
2-330. Address Translators 2-259
2-331. 8-BIT S_0-Register 2-261
2-332. Main Memory 2-261
2-335. Main Memory Addressing 2-261
2-338. Core Magnetization 2-263
2-341. Memory Stack Operation 2-263
2-344. X and Y-Line Selection 2-267
2-355. Sense/Inhibit Line Operation 2-274
2-369. Main Memory Timing 2-282
2-376. Control Memory 2-285
2-377. Core Pair Operation 2-285

TABLE OF CONTENTS (Cont)

Chapter			Page
		2-380. Physical Arrangement	2-287
		2-382. Control Memory Addressing	2-287
		2-390. Selected Circuit Analysis	2-292
		2-392. Digit-Line Current	2-294
		2-396. Sense Line Output	2-295
		2-397. Control Memory Timing	2-295
	2-401.	Bootstrap Memory	2-298
		2-402. Storage Element	2-298
		2-403. Readout Operation	2-298
		2-404. Physical Arrangement	2-298
		2-405. Group and Line Selection	2-300
		2-410. Sense-Line Analysis	2-304
		2-411. Initiate Bootstrap Memory	2-304
	Section 2-3. Detailed Functional Description (I/O Section)		
	2-413.	Input/Output Section	2-307
		2-415. Interface	2-307
		2-416. Data Exchange Rate	2-307
		2-417. I/O Signals	2-307
		2-418. Data Request Signal	2-307
		2-419. Real-Time-Clock Request (RTC Request) Signals	2-309
		2-420. Interrupt Signals	2-309
		2-421. Buffers and Address Control Words	2-309
		2-422. Terminal Address Control Word (TACW)	2-309
		2-423. Initial Address Control Word (IACW)	2-309
		2-425. Special I/O Logic	2-310
		2-426. Active Flip-Flop	2-310
		2-427. Priority Logic	2-310
		2-428. I/O Translator	2-310
		2-429. Data Request Operations	2-310
		2-430. Dual-Channel Operation	2-312
		2-431. Continuous-Data Mode (CDM)	2-312
		2-432. Externally Specified Index (ESI)	2-312
		2-433. Externally Specified Address (ESA)	2-313
		2-434. Execution of SIN, SOUT, SEXF, INSTP, OUTSTP, EXFSTP Instructions	2-313

TABLE OF CONTENTS (Cont)

Chapter			Page
	2-435.	SIN (f= 50:01)	2-313
	2-436.	SOUT (f= 50:02)	2-313
	2-437.	SEXF (f= 50:03)	2-313
	2-438.	INSTP (f= 50:15)	2-313
	2-439.	OUTSTP (f= 50:16)	2-313
	2-440.	EXFSTP (f= 50:17)	2-313
	2-441.	I-Sequence Data Flow for f= 50:01 through 50:03, and 50:15 through 50:17	2-313
2-444.		Execution of IN, OUT, and EXF Instructions (f=50:11 through 50:13)	2-329
	2-445.	IN (f= 50:11)	2-329
	2-446.	OUT (f= 50:12)	2-329
	2-447.	EXF (f= 50:13)	2-329
	2-448.	Execution Sequences	2-329
	2-449.	I-Sequence Data Flow for f= 50:11 through 50:13	2-329
	2-450.	IB1 Sequence	2-329
	2-451.	IB2 Sequence	2-329
	2-452.	IB1 and IB2 Sequence Data Flow for f = 50:11 through 50:13	2-330
	2-453.	IB1 and IB2 Sequence Essential Commands for f= 50:11 through 50:13	2-330
2-454.		Priority Signal Detection and Selection	2-335
	2-457.	Data Request and Monitor Interrupt Priority	2-336
	2-458.	Data Request Scan	2-336
	2-459.	Interrupt Request Scan	2-336
	2-460.	Request Signal Entrance Logic	2-336
	2-461.	IDR Signal Detection	2-336
2-463.		I/O Signal Selection	2-337
	2-464.	Data-Request Scan	2-337
	2-465.	Interrupt Request Scan	2-341
	2-466.	I/O Signal Selection Essential Commands	2-341
2-467.		Channel Priority Selection	2-351
2-470.		Function Priority Selection	2-351
2-473.		Group Priority Selection	2-353
2-474.		Group Translator (I/O Translator 1)	2-353
2-475.		Mode Translator (I/O Translator 1)	2-355
2-476.		Channel Translator	2-355

TABLE OF CONTENTS (Cont)

Chapter		Page
2-477.	Function Translator	2-355
2-478.	Data-Request Sequence Timing	2-357
2-484.	Input-Data-Request Operations	2-365
2-485.	Single-Channel Operation	2-365
2-488.	Dual-Channel Operation	2-365
2-490.	Scan and I/O Sequence Single-Channel ID Operation Data Flow	2-371
2-491.	Scan, I/O1, and I/O2 Sequence Dual-Channel ID Operation Data Flow	2-371
2-492.	Scan and I/O Sequence Essential Commands for Single-or Dual-Channel ID Operations	2-371
2-494.	ID Acknowledge Timing	2-383
2-495.	Scan, I/O1, and I/O2 Sequence Essential Commands (Fast Interface)	2-383
2-496.	Scan, I/O1, and I/O2 Sequence Essential Commands (Slow Interface)	2-383
2-497.	Input Data Operations with ESI	2-383
2-499.	Scan, I/O1, and I/O2 Sequence Essential Commands for ID Operations with ESI	2-384
2-500.	Input Data Operations with ESA	2-393
2-502.	Scan, I/O1, and I/O2 Sequence Data Flow Block Diagram for ID Operations with ESA	2-393
2-503.	Scan, I/O1 and I/O2 Essential Commands for ID Operations with ESA	2-393
2-504.	Output Data Request Operations without ESI or ESA	2-397
2-505.	Single-Channel Operation	2-397
2-508.	Dual-Channel Operation	2-397
2-510.	Scan and I/O Sequence Data Flow for Single-Channel Operation	2-399
2-511.	Scan and I/O-Sequence Data Flow for Dual Channel Operation	2-400
2-512.	Scan, I/O1, I/O, and I/O2 Sequence Essential Commands for Dual-Channel OD Operations	2-400
2-513.	OD Acknowledge Timing	2-409
2-515.	Output Data Request Operations with ESI	2-417

TABLE OF CONTENTS (Cont)

Chapter			Page
	2-517.	Scan and I/O-Sequence Data Flow for OD Operations with ESI (Not Buffer Termination)	2-417
	2-519.	I/O2 Sequence Data Flow (Buffer Termination)	2-418
2-521.		Output Data Request Operations with ESA	2-423
	2-522.	Scan, I/O1, and I/O2 Sequence Data Flow for OD Operations with ESA	2-423
	2-523.	Scan, I/O1 and I/O2 Sequence Essential Commands for OD Operations with ESA	2-424
2-524.		External Function Request Operations	2-429
	2-525.	Single-Channel EFR Operation without ESI or ESA	2-429
	2-526.	Dual-Channel EFR Operation without ESI or ESA	2-429
	2-527.	Scan and I/O1 Sequence Data Flow for Single-Channel Operation	2-429
	2-528.	Scan, I/01 and I/02 Sequence Data Flow for Dual-Channel Operation	2-429
	2-529.	Scan, I/O1, and I/O2 Sequence Essential Commands for Single or Dual-Channel EF Operations	2-429
	2-530.	EF Acknowledge Timing	2-437
	2-531.	Normal Mode EF Acknowledge Timing	2-437
	2-532.	Intercomputer Mode EF Acknowledge Timing	2-437
	2-533.	EF Operations with ESI	2-437
	2-534.	I/O2 Sequence Data Flow for Terminate EF Operations with ESI	2-437
	2-535.	Scan, I/O1 and I/O2 Sequence Data Flow for EF Operations with ESA	2-437
2-536.		Continuous Data Mode Operations	2-445
	2-538.	Continuous Data Mode Sequence Data Flow	2-445
	2-539.	CDM Sequence Essential Commands	2-446
2-540.		Execution of OUTOV and EXFOV Instructions (f= 50:26, 50:27)	2-451
	2-541.	I-Sequence Data Flow	2-451
	2-542.	First I-Sequence Essential Commands	2-451
	2-543.	Remaining Sequences with Output Available and No Resume-Fault	2-451

TABLE OF CONTENTS (Cont)

Chapter			Page
	2-544.	Sequences After First I-Sequence Essential Commands with Output Available and No Resume-Fault	2-451
	2-545.	Remaining Sequences with Output Not Available and No Resume-Fault	2-451
	2-546.	Sequence After First I-Sequence Essential Commands with Output Not Available and No Resume-Fault	2-452
	2-547.	Remaining Sequences with Resume-Fault	2-452
	2-548.	Sequences After First I-Sequence Essential Commands with Resume-Fault	2-452
	2-549.	Decrement P Data Flow	2-459
2-550.		Execution of SKPIIN, SKPOIN, SKPFIN, SKPNR, SRSM Instructions	2-459
	2-551.	SKPIIN (f= 50:21)	2-459
	2-552.	SKPOIN (f= 50:22)	2-459
	2-553.	SKPFIN (f= 50:23)	2-459
	2-554.	SKPNR (f= 50:57)	2-459
	2-555.	SRSM (f=50:20)	2-460
	2-556.	I-Sequence Data Flow for f= 50:20 through 50:23 and 50:57	2-460
	2-557.	I-Sequence Essential Commands for f= 50:20 through 50:23 and 50:57	2-460
2-558.		External Interrupt Request Operations	2-465
	2-559.	Single-Channel Operation	2-465
	2-560.	Dual-Channel, ESI, or ESA Operation	2-465
	2-561.	I/O1 and I/O2 Sequence Data Flow for EIR Operations	2-465
	2-562.	Scan and I/O2 Sequence Essential Commands for EIR Operations	2-465
2-563.		Real-Time-Clock (RTC) Request Operations, Generation of RTC Monitor Interrupts, and Execution of RTC Instructions	2-471
	2-565.	RTC Overflow Interrupt	2-471
	2-566.	RTC Request Operations Data Flow	2-471
	2-567.	RTC (f= 50:14) Instruction	2-472
	2-568.	I/O1, RTC and Next Sequence Essential Commands for RTC Operations	2-472

TABLE OF CONTENTS (Cont)

Chapter			Page
2-569.	Execution of RIL, EXL, SIL, SXL, WTFI, and Special and Monitor Interrupt Operations Instructions. .		2-477
	2-570.	RIL (f= 50:30)	2-477
	2-571.	EXL (f= 50:32)	2-477
	2-572.	SIL (f= 50:34)	2-477
	2-573.	SXL (f= 50:36)	2-477
	2-574.	WTFI (f= 50:24)	2-477
	2-575.	Execution Sequences	2-477
	2-576.	RIL, EXL, SIL, SXL Operations	2-477
	2-577.	WTFI Instruction	2-478
2-578.	Special and Monitor Interrupt Operations		2-478
	2-579.	Instruction Fault	2-478
	2-580.	Resume-Fault (Intercomputer Time-Out Fault)	2-478
	2-581.	RTC Monitor	2-478
	2-582.	External Sync	2-478
	2-583.	RTC Overflow	2-478
	2-584.	External Interrupt Monitor (EI Mon)	2-478
	2-585.	External Function Monitor (EF Mon)	2-478
	2-586.	Output Data Monitor (OD Mon)	2-478
	2-587.	Input Data Monitor (ID Mon)	2-478
	2-588.	Honoring Operations	2-478
	2-592.	Special Interrupt Translator	2-480
	2-593.	Interrupt/I-Sequence Data Flow	2-480
	2-594.	Scan and Interrupt/I-Sequence Essential Commands	2-480
	2-595.	EI Monitor Operations	2-487
	2-596.	EI Monitor Operation Essential Commands. . .	2-487
2-597.	Intercomputer Operations		2-487
	2-598.	Intercomputer Channels	2-487
	2-599.	External Function/External Interrupt Exchange	2-487
	2-600.	Data Exchange	2-487
	2-601.	ODR Logic	2-488
	2-602.	Resume Fault Interrupt	2-488

Section 2-3. Detailed Functional Description (Power Distribution)

2-603. Power Distribution 2-491

TABLE OF CONTENTS (Cont)

Chapter			Page
	2-604.	General	2-491
	2-605.	Power Control Circuit	2-491
	2-606.	Main DC Power Supply and Distribution	2-491
	2-607.	-15 VDC Supply	2-491
	2-608.	+15 VDC Supply	2-492
	2-611.	±4.5 VDC Supply	2-492
	2-612.	Main Memory Power Supply	2-492
	2-613.	+6 VDC Power Supply	2-492
	2-614.	+3 VDC Supply	2-492
	2-616.	±15 VDC Supply	2-493
	2-617.	-4.5/-3 VDC Supply	2-493
	2-618.	Miscellaneous Circuits	2-493
3.	OPERATION		3-1
	3-1.	General Information	3-1
	3-5.	Controls and Indicators	3-1
	3-9.	Daily Turn-On	3-25
4.	INSTALLATION		4-1
	4-1.	Unpacking Information	4-1
	4-3.	Unpacking Procedure	4-1
	4-4.	Placement of the Computer	4-1
	4-6.	Cabling Connections	4-2
	4-8.	Computer Ground	4-15
	4-9.	Preoperational Inspection and Preliminary Testing	4-15
	4-10.	Visual Inspection	4-15
	4-11.	Power Check	4-16
	4-12.	Fan Assembly A12 Rotation Check	4-16
	4-13.	Operational Check	4-16
5.	MAINTENANCE		5-1
	5-1.	Introduction to the Diagnostic Program	5-1
	5-3.	Detection Phase	5-1
	5-4.	Isolation Phase	5-1
	5-5.	Testing Philosophy	5-1

TABLE OF CONTENTS (Cont)

Chapter			Page
	5-6.	Limitations of the Diagnostic	5-1
5-7.	Basic Computer Programming		5-2
	5-8.	Scope	5-2
	5-9.	Program Development	5-2
	5-11.	Programming Language	5-3
	5-12.	Assembling Process	5-4
		5-13. Inspect and Change Program	5-4
		5-15. Bootstrap Program (Channel 0)	5-5
5-17.	Diagnostic Program Description		5-5
	5-19.	CONTBASI	5-5
	5-20.	AIMAT	5-13
	5-21.	MEDIA	5-14A
	5-22.	IOTA	5-16
	5-23.	ZPAC Utility Package	5-21
5-24.	Scheduled Maintenance		5-21
	5-25.	General Information	5-21
	5-28.	Scheduled Tests	5-21
5-30.	Test and Servicing Procedures		5-22
	5-35.	Maintenance Requirements	5-22
		5-37. Unscheduled (UMRC) Maintenance	5-23
		5-38. Unscheduled MRC's	5-23
		5-39. Background Information	5-23
5-40.	Yard/Tender Corrective Maintenance		5-23

APPENDIX A A-1

A-1 Digital Computer Terminology A-1

LIST OF ILLUSTRATIONS

Number	Title	Page
1-1.	Computer with Maximum of Eight I/O Channels	1-2
1-2.	Computer with Maximum of 16 I/O Channels	1-3
1-3.	Block Diagram of Typical Computer Complex	1-6
1-4.	Computer Group Equipments in Common Usage	1-7
2-1.	Computer Drawer Locations	2-4
2-2.	Computer with Logic Drawer A2 Extended and Logic Chassis A2A1 Partially Opened	2-5
2-3.	Typical Memory Stack in Memory Drawer	2-7
2-4.	Control and Bootstrap Memories in Logic Drawer A4	2-8
2-5.	Power Supply PS1	2-9
2-6.	Computer Mk 152 Series, Major Units and Assemblies	2-11
2-7.	Printed Circuit Modules	2-12
2-8.	Computer Simplified Block Diagram	2-20
2-9.	Computer Block Diagram	2-21
2-10.	Main Timing Cycle Outputs	2-24
2-11.	Typical I/O Cable Configurations	2-33
2-12.	Typical Inverter Symbols	2-74
2-13.	Typical Flip-Flop Symbols	2-75
2-14.	Typical Amplifier and Driver Symbols	2-76
2-15.	Examples of Logic Notations	2-78
2-16.	Card Location, Test Block, and Connector Designators	2-80
2-17.	Signal Line Symbology	2-83
2-18.	Special Symbology	2-84
2-19.	Simplified Chassis Map	2-85
2-20.	Control Section, Block Diagram	2-88
2-21.	Operational Controls, Control Panel 1 (A2)	2-93
2-22.	Main Timing Cycle Outputs	2-97
2-23.	Master Clock Logic, Simplified Diagram	2-98
2-24.	Master Clock Output Signals	2-99
2-25.	I-Sequence Data Flow	2-115
2-26.	Basic R-Sequence	2-121
2-27.	Basic W-Sequence	2-122
2-28.	I-Sequence Data Flow for f= 50:56	2-127
2-29.	I-Sequence Data Flow for f= 70, 71	2-129
2-30.	I-Sequence Data Flow for f= 50:72, 50:73	2-132
2-31.	I and Next I-Sequence Data Flow for f= 36, 37	2-134
2-32.	R1 and Next I-Sequence for f= 32, 33	2-139
2-33.	R-Sequence Data Flow for f= 10-13	2-142

LIST OF ILLUSTRATIONS (Cont)

Number	Title	Page
2-34.	W-Sequence Data Flow for f= 40-47	2-144
2-35.	W-Sequence Data Flow for f=72, 74, 75	2-148
2-36.	I-Sequence Data Flow for f= 34, 35, 60-67	2-154
2-37.	I and Next I-Sequence Data Flow for f= 73	2-156
2-38.	R1-Sequence Data Flow for f= 54, 55	2-160
2-39.	W-Sequence Data Flow for f= 76	2-162
2-40.	R-1 and W-Sequence Data Flow for f= 30, 31	2-163
2-41.	I-Sequence Data Flow for f= 50:50 through 50:53	2-168
2-42.	R1 and Next I-Sequence Data Flow for f=56	2-171
2-43.	R1 and W-Sequence Data Flow for f= 57	2-176
2-44.	I-Sequence Data Flow for f= 50:54, 50:55	2-180
2-45.	Section and Subsection Borrow Capability	2-184
2-46.	Method of Incrementing or Decrementing the Contents of the B-Register	2-187
2-47.	KO-1 Adder Block Diagram	2-188
2-48.	KO-1 Adder Borrow Request Generation Logic	2-190
2-49.	Parity Evaluation Example	2-191
2-50.	Parity Evaluator, Section 5-0	2-192
2-51.	Parity Evaluated, Final Stage	2-194
2-52.	R1-Sequence Data Flow for f= 14 through 17	2-198
2-53.	R1 and R1/R2-Sequence Data Flow for f= 20 through 23	2-201
2-54.	I-Sequence Data Flow for f= 50:60	2-205
2-55.	R1-Sequence Data Flow for f= 02, 03, 06, and 07	2-210
2-56.	R1-Sequence Data Flow for f= 04, 05	2-214
2-57.	R1-Sequence Data Flow for f= 51, 52	2-216
2-58.	R1-Sequence Data Flow for f= 53	2-217
2-59.	I and Next I-Sequence Data Flow for f= 50:61, 50:62, 50:63	2-221
2-60.	Shift Sequence Simplified Logic	2-227
2-61.	Shift Operations Involving the AL-Register	2-228
2-62.	Final Scale W and Last W-Sequence Data Flow	2-231
2-63.	Scale Sequence, One Place Shift	2-232
2-64.	Simplified Multiply Sequence	2-243
2-65.	R1 and Multiply Sequence Data Flow	2-244
2-66.	Multiply Final Sign Correction Data Flow	2-247
2-67.	Simplified Divide Sequence	2-248
2-68.	R1 and Divide Sequence Data Flow	2-249
2-69.	Divide Final Sign Correction Data Flow	2-250
2-70.	Memory Section Simplified Block Diagram	2-260

LIST OF ILLUSTRATIONS (Cont)

Number	Title	Page
2-71.	Main Memory Block Diagram	2-262
2-72.	Main Memory Addressing	2-264
2-73.	S_1 Bit Translation	2-264
2-74.	Core Magnetization	2-265
2-75.	Memory Stack Operation	2-266
2-76.	X or Y-Line Selection Circuit	2-269
2-77.	X and Y-Line Selection	2-271
2-78.	Typical Core Matrix	2-274
2-79.	Sense-Inhibit Line Selections	2-277
2-80.	Signal Detection through Rejection Transformer	2-276
2-81.	Sense-Inhibit Line Circuitry	2-279
2-82.	Operation of Saturable Reactor T1	2-281
2-83.	Main Memory Timing	2-283
2-84.	Control Memory Core Pair Operation	2-286
2-85.	Memory Arrangement for 128 Word Locations	2-288
2-86.	Control Memory Address Translation from SO	2-289
2-87.	Simplified Diagram of Complete Control Memory Word Line Array (Read Operation)	2-290
2-88.	Detailed Diagram of Partial Control Memory Word Line Connection (Read Operation)	2-291
2-89.	Simplified Diagram of Control Word Drive Circuitry for Read Operation (SO= 000_8)	2-293
2-90.	Simplified Diagram of Control Memory Digit Line Circuitry (Bit Zero)	2-296
2-91.	I-Sequence Data Flow Referencing Control Memory	2-297
2-92.	W-Sequence Data Flow for f= 46, 47 with Storage Address in Control Memory	2-297
2-93.	Control Memory Timing Waveforms	2-299
2-94.	Core Transformer	2-300
2-95.	Example of Two Bootstrap Addresses	2-301
2-96.	Bootstrap Address Translation From SO	2-301
2-97.	Simplified Diagrams of Bootstrap Drive Lines	2-302
2-98.	Simplified Circuit Diagram of Bootstrap Drive Circuitry (Group O and Line O)	2-303
2-99.	I-Sequence Data Flow Referencing Bootstrap	2-305
2-100.	Bootstrap Memory Theoretical Timing	2-306
2-101.	Terminal Address Control Word Format	2-310
2-102.	Initial Address Control Word Format	2-310

LIST OF ILLUSTRATIONS (Cont)

Number	Title	Page
2-103.	Buffer Program Examples	2-311
2-104.	I-Sequence Data Flow for f= 50:01 - 50:03, 50:15-50:17	2-314
2-105.	IB1 and IB2-Sequence Data Flow for f= 50:11-50:13	2-331
2-106.	Scan Sequence Timing Cycle	2-338
2-107.	Input/Output Signal Selection Flow (Sheet 1 of 2)	2-339
2-108.	Scan and I, R1, R2, W, B1 or B2 Sequence Timing Example, (No Request Signal Detected)	2-358
2-109.	Single Request, Single Channel-Timing Example	2-359
2-110.	Single Data Request, Dual Channel-Timing Example	2-361
2-111.	Multiple Data Request, Single Channel-Timing Example	2-363
2-112.	Input-Data Operations, Single Channel without ESI and ESA	2-366
2-113.	Dual-Channel I/O Interface	2-368
2-114.	Scan and I/O1-Sequence Data Flow for Single- or Dual- Channel ID Operations	2-372
2-115.	I/O2-Sequence Data Flow for Dual-Channel ID Operations	2-373
2-116.	ESI Input Word Format for IDR Operations	2-384
2-117.	Scan and I/O1-Sequence Data Flow for ID Operations with ESI	2-387
2-118.	Store Select L1→SO Transfer for ESI	2-388
2-119.	I/O2-Sequence Data Flow for Terminate ID Operations with ESI	2-388
2-120.	ESA Input Word Format for IDR Operations	2-393
2-121.	Scan, I/O1, and I/O2-Sequence Data Flow for ID Operations with ESA	2-394
2-122.	Output Data Operations, Single Channel without ESI and ESA	2-398
2-123.	Scan and I/O1-Sequence Data Flow for Single- or Dual- Channel OD Operations	2-401
2-124.	I/O2 Sequence Data Flow for Dual-Channel OD Operations	2-402
2-125.	OD Acknowledge Simplified Logic	2-410
2-126.	ESI Word Formats for ODR Operations	2-418
2-127.	Scan and I/O1-Sequence Data Flow for OD Operations with ESI	2-419
2-128.	Store Select L1→SO Transfer for ES1	2-420
2-129.	I/O2-Sequence Data Flow for Terminate OD Operations with ESI	2-420
2-130.	ESA Word Formats for ODR Operations	2-423
2-131.	Scan, I/O1, and I/O2-Sequence Data Flow for OD Operations with ESA	2-425
2-132.	External Function Operations, Single Channel without ESI and ESA	2-430

LIST OF ILLUSTRATIONS (Cont)

Number	Title	Page
2-133.	Scan and I/O1-Sequence Data Flow for Single or Dual Channel EF Operations	2-431
2-134.	I/O2-Sequence Data Flow for Dual-Channel EF Operations	2-432
2-135.	EF-Acknowledge Simplified Logic	2-438
2-136.	Scan and I/O1-Sequence Data Flow for EF Operations with ESI	2-439
2-137.	I/O2-Sequence Data Flow for Terminate EF Operations with ESI	2-440
2-138.	Scan, I/O1, and I/O2-Sequence Data Flow for EF (OD) Operations with ESA	2-443
2-139.	Continuous Data Mode Sequence Data Flow	2-447
2-140.	I-Sequence Data Flow for f= 50:26, 50:27	2-452
2-141.	P Decrement for f=50:26, 50:27 with Resume-Fault	2-459
2-142.	I-Sequence Data Flow for f=50:20-50:23, 50:57	2-461
2-143.	External Interrupt Operations, Single Channel	2-466
2-144.	I/O1 and I/O2-Sequence Data Flow for EIR Operations	2-467
2-145.	RTC Address 00015_8 Bit Weights in Seconds	2-471
2-146.	I/O1 and RTC and Next Sequence Data Flow for RTC Operations	2-473
2-147.	Interrupt/I-Sequence Data Flow	2-481
2-148.	EIR and EI Mon Interaction, Simplified Logic Diagram	2-489
2-149.	Intercomputer I/O Cabling	2-490
3-1.	Controls and Indicators, Input/Output Panel (A1 or Optional A8)	3-2
3-2.	Controls and Indicators, Control Panel 1, (A2)	3-7
3-3.	Controls and Indicators, Control Panel 2, (A4)	3-15
3-4.	Controls and Indicators, Power Panel (A5)	3-21
4-1.	Computer Clearance Requirements	4-3
4-2.	Location and Access to Cable Entry Panel	4-4
4-3.	Location of Cabinet Mounting Holes	4-5

LIST OF TABLES

Number	Title	Page
1-1.	Operational Characteristics	1-8
1-2.	Modifications of Computer Mk 152 Series	1-14
1-3.	General Characteristics	1-15
1-4.	Computer Group Equipment Nomenclature	1-16
1-5.	Reference Publications	1-17
2-1.	List of Fuses	2-2
2-2.	Computer Power Control Switches and Relays	2-6
2-3.	DC Power Supply Outputs	2-6
2-4.	Main Memory Power Supply Outputs	2-10
2-5.	Designating Nomenclature, and Applicability of Major Units and Assemblies	2-13
2-6.	Printed Circuit Module Complement, Logic Circuits	2-14
2-7.	Printed Circuit Module Complement, Memory Chassis	2-18
2-8.	Major Command Sequences	2-26
2-9.	Memory Address Assignments	2-30
2-10.	Description of Input/Output Control Signals	2-34
2-11.	I/O Chassis Channel Assignments	2-34
2-12.	Symbols Used	2-39
2-13.	Instruction Word Format	2-41
2-14.	List of Instructions	2-43
2-15.	Control Panel Switch Settings and Related Operation	2-89
2-16.	Normal Master Clock Operation	2-96
2-17.	Main Timing	2-101
2-18.	Initial Sequence Flip-Flop Setting Conditions (NON-I/O)	2-105
2-19.	I-Sequence	2-107
2-20.	Advance-P Subsequence	2-116
2-21.	R-Sequence	2-118
2-22.	W-Sequence	2-123
2-23.	Control Section Command Instructions	2-126
2-24.	I-Sequence Essential Commands for f= 50:56	2-127
2-25.	I-Sequence Essential Commands for f= 70, 71	2-130
2-26.	Commands/Descriptions for Execution of ENTALK, ADDALK Instructions (f= 70, 71)	2-131
2-27.	I-Sequence Essential Commands for f= 50:72, 50:73	2-133
2-28.	Commands/Descriptions for Execution of ENTICR, ENTSR Instructions (f= 50:72, 50:73)	2-133
2-29.	I and Next I-Sequence Essential Commands for f=36, 37	2-135

LIST OF TABLES (Cont)

Number	Title	Page
2-30.	Commands/Descriptions for Execution of ENTBK, ENTBKB Instructions (f= 36, 37)	2-136
2-31.	R1 and Next I-Sequence Essential Commands for f=32, 33	2-140
2-32.	Commands/Descriptions for Execution of ENTB, ENTBB Instructions (f= 32, 33)	2-141
2-33.	R1-Sequence Essential Commands for f= 10 through 13	2-143
2-34.	Commands/Descriptions for Execution of ENTAU, ENTAUB, ENTAL, ENTALB Instructions (f=10, 11, 12, 13)	2-143
2-35.	W-Sequence Essential Commands for f= 40-47	2-146
2-36.	Commands/Descriptions for Execution of CL, CLB, STRB, STRBB, STRAL, STRALB, STRAU, STRAUB, (f= 40, 41, 42, 43, 44, 45, 46, 47)	2-147
2-37.	W-Sequence Essential Commands for f= 72, 74, 75	2-149
2-38.	Commands/Descriptions for Execution of STRICR, STRADR, STRSR, (f= 72, 74, 75)	2-150
2-39.	I-Sequence Essential Commands for f= 34, 35, 60-67	2-155
2-40.	I and Next I-Sequence Essential Commands for f= 73	2-157
2-41.	Instructions/Descriptions for f= 34, 35, 60-67 and 73	2-158
2-42.	R1-Sequence Essential Commands for f= 54, 55	2-164
2-43.	W-Sequence Essential Commands for f= 76	2-164
2-44.	R1 and W-Sequence Essential Commands for f= 30, 31	2-165
2-45.	Instructions/Descriptions for f= 54, 55, 76, 30 and 31	2-166
2-46.	I-Sequence Essential Commands for f= 50:50 through 50:53	2-169
2-47.	Instructions/Descriptions for f= 50:50 through 50:53	2-170
2-48.	R1 and Next I-Sequence Essential Commands for f= 56	2-172
2-49.	Commands/Descriptions for the Execution of BSK Instruction (f= 56)	2-173
2-50.	R1-Sequence Essential Commands for f= 57	2-177
2-51.	W-Sequence Essential Commands for f= 57	2-178
2-52.	Commands/Descriptions for the Execution of ISK Instruction (f= 57)	2-179
2-53.	I-Sequence Essential Commands for f= 50:54, 50:55	2-181
2-54.	Commands/Descriptions for the Execution of SKPODD, SKPEVN Instructions (f= 50:54, 50:55)	2-182
2-55.	Intersection Borrows to Section 05-00	2-186
2-56.	Intersection Borrows to Section 11-06	2-186
2-57.	Intersection Borrows to Section 17-12	2-186
2-58.	KO-1 Adder Borrow Request Conditions	2-189

LIST OF TABLES (Cont)

Number	Title	Page
2-59.	Parity Evaluator, Group 1 and O Conditions	2-193
2-60.	Parity Evaluator, Section 5-0 Conditions	2-193
2-61.	Parity Evaluator, Total Parity Conditions	2-195
2-62.	Arithmetic Section-Arithmetic and Logical Instructions	2-197
2-63.	R1-Sequence Essential Commands for f= 14 through 17	2-199
2-64.	Commands/Descriptions for f= 14 through 17	2-200
2-65.	R1 and R1/R2 Essential Commands for f= 20 through 23	2-202
2-66.	Commands/Descriptions for f= 20 through 23	2-203
2-67.	X-D' Adder Conditions for f= 50:60	2-204
2-68.	I-Sequence Essential Commands for f= 50:60	2-206
2-69.	Commands/Descriptions for f= 50:60	2-207
2-70.	R1-Sequence Essential Commands for f= 02, 03, 06, and 07	2-211
2-71.	Commands/Descriptions for f= 02, 03, 06, and 07	2-211
2-72.	Conditions for Low Level Output of 10A00	2-212
2-73.	Conditions for Low Level Output of 70A17 (Set Greater FF)	2-212
2-74.	Truth Table of (AU) v $(AL)_i$ - L(AU) (Y)' = AL_f	2-215
2-75.	Truth Table of (Y) v (AL_i) - L(Y) $(AL)_i$ = AL_f	2-219
2-76.	R1-Sequence Essential Commands for f= 04, 05, 51 through 53	2-219
2-77.	Commands/Descriptions for f= 04, 05, 51, 52 and 53	2-220
2-78.	I and Next I-Sequence Essential Commands for f= 50:61 50:62, 50:63	2-223
2-79.	Commands/Descriptions for f= 50:61, 50:62, 50:63	2-224
2-80.	I and Shift Sequence Essential Commands with Initial Shift Count = $O/A_{35i} \neq A_{34i}$	2-229
2-81.	Commands/Descriptions for Initial Shift Count Equal to Zero	2-229
2-82.	I and Shift Sequence Essential Commands with Initial Shift Count $\neq 0$ or $A_{35} = A_{34}$ for A f= 50:44	2-233
2-83.	Commands/Descriptions for Initial Shift Count Not Equal to Zero or $A_{35} = A_{34}$ for A 50:44	2-236
2-84.	Termination of Scale Sequence by $AU_{34} \neq AU_{33}$ (W-Sequence)	2-241
2-85.	Commands/Descriptions for Termination of Scale Sequence by $AU_{34} \neq AU_{33}$	2-242
2-86.	R1, Multiply and Next I-Sequence Essential Commands	2-251
2-87.	Commands/Descriptions for Execution of f= 24, 25	2-253
2-88.	Divide Sequence Sign Correction	2-255
2-89.	R1, Divide, and Next I-Sequence Essential Commands	2-256
2-90.	Commands/Descriptions for Execution of f= 26, 27	2-258

LIST OF TABLES (Cont)

Number	Title	Page
2-91.	Computer Memory Sizes	2-259
2-92.	Current Diverter Designators	2-273
2-93.	Read/Write Select Transformer Designators	2-273
2-94.	Current Regulator Designators	2-273
2-95.	I/O Interface Characteristics	2-308
2-96.	Data Transfer Rates (Maximum)	2-308
2-97.	Assigned Addresses for Address Control Words	2-312
2-98.	I-Sequence Essential Commands for f= 50:01 through 50:03, 50:15 through 50:17	2-315
2-99.	Commands/Descriptions for Execution of SIN, SOUT, SEXF, INSTP, OUTSP, EXFSTP Instructions (f= 50:01 through 50:03 and 50:15 through 50:17)	2-316
2-100.	Channel Translator Indications	2-319
2-101.	Group Translator Indications	2-319
2-102.	Function Translator Indications	2-319
2-103.	IB1 and IB2-Sequence Essential Commands	2-321
2-104.	IB1 and IB2-Sequence Commands/Descriptions	2-325
2-105.	IB1, IB2, and Next I-Sequence Essential Commands for f= 50:11 - 50:13	2-332
2-106.	Function Translator Indications	2-333
2-107.	Function Priority for Signals Presented to Input/Output Logic	2-335
2-108.	Scan Sequence Essential Commands	2-342
2-109.	I/O Signal Selection Commands/Descriptions	2-343
2-110.	List of Commands and Tables where Detailed Description of Command can be Found	2-346
2-111.	Channel Priority Gates, Chassis 1 and 2	2-352
2-112.	Function Priority Selection Indications	2-352
2-113.	Group Priority Selection Indications	2-354
2-114.	Group Translator Indications	2-355
2-115.	Channel Translator Indications	2-356
2-116.	Function Translator Indications	2-356
2-117.	Single Channel Forward ID Buffer IACW=005000, TACW=005003	2-367
2-118.	Single Channel Backward ID Buffer IACW=405003, TACW=405000	2-367
2-119.	Dual Channel Forward ID Buffer on Channels 2, 3 IACW=005000, TACW=005003	2-369

LIST OF TABLES (Cont)

Number	Title	Page
2-120.	Dual Channel Backward ID Buffer on Channels 2, 3 IACW=405003, TACW=405000	2-369
2-121.	Scan, I/O1, and I/O2-Sequence Essential Commands for Single or Dual-Channel ID Operations	2-374
2-122.	Scan I/O1 and I/O2-Sequence Commands/Descriptions for Single or Dual-Channel ID Operations	2-377
2-123.	ID-Acknowledge Timing for Fast Interface	2-385
2-124.	ID-Acknowledge Timing for Slow Interface	2-385
2-125.	Scan, I/O1 and I/O2-Sequence Commands/Descriptions for ID-Acknowledge Timing (Fast Interface)	2-386
2-126.	Scan, I/O1 and I/O2-Sequence Commands/Descriptions for ID-Acknowledge (Slow Interface)	2-386
2-127.	Scan, I/O1 and I/O2-Sequence Essential Commands for ID Operations with ESI	2-389
2-128.	Scan, I/O1 and I/O2 Commands/Descriptions for ID Operations with ESI	2-391
2-129.	Scan, I/O1 and I/O2-Sequence Essential Commands for ID Operations with ESA	2-395
2-130.	Scan, I/O1 and I/O2-Sequence Commands/Descriptions for ID Operations with ESA	2-396
2-131.	Single-Channel Forward OD or EF Buffer IACW=005000, TACW=005003	2-399
2-132.	Single-Channel Backward OD or EF Buffer IACW=405003, TACW=405000	2-399
2-133.	Dual-Channel Forward OD Buffer on Channels 2 and 3 IACW=005000, TACW=005003	2-400
2-134.	Dual-Channel Backward OD Buffer on Channels 2 and 3 IACW=405003, TACW=405000	2-400
2-135.	Scan, I/O1 and I/O2-Sequence Essential Commands for Single-or Dual-Channel OD Operations	2-403
2-136.	Scan, I/O1, I/O, and I/O2-Sequence Commands/Descriptions for Dual-Channel OD Operations	2-406
2-137.	OD Acknowledge Timing for Fast Interface, Not Intercomputer	2-411
2-138.	OD Acknowledge Timing for Slow Interface, Not Intercomputer	2-412
2-139.	OD Acknowledge Timing Commands/Descriptions for Fast Interface Not-Intercomputer	2-413
2-140.	OD Acknowledge Timing Commands/Descriptions for Slow Interface, Not-Intercomputer	2-414

LIST OF TABLES (Cont)

Number	Title	Page
2-141.	OD Acknowledge Timing for Fast Interface, Intercomputer	2-414
2-142.	OD Acknowledge Timing for Slow Interface, Intercomputer	2-415
2-143.	OD Acknowledge Timing Commands/Descriptions for Fast Interface, Intercomputer	2-416
2-144.	Scan, I/O1, and I/O2-Sequence Essential Commands for OD Operations with ESI	2-421
2-145.	Scan, I/O1, and I/O2-Sequence Essential Commands for OD Operations with ESA	2-426
2-146.	Scan, I/O1 and I/O2-Sequence Essential Commands for Single-or Dual-Channel EF Commands	2-433
2-147.	Scan, I/O1, and I/O2-Sequence Commands/Descriptions for Single-or Dual-Channel EF Operations	2-435
2-148.	Scan, I/O1 and I/O2-Sequence Essential Commands for EF Operations with ESI	2-441
2-149.	Continuous Data Mode Sequence Essential Commands	2-448
2-150.	Commands/Descriptions for CDM Sequence	2-449
2-151.	I-Sequence Essential Commands for f= 50:26, 50:27	2-453
2-152.	Commands/Descriptions for First I-Sequence (f= 50:26 and 50:27)	2-454
2-153.	Sequences After First I-Sequence Essential Commands for f= 50:26 and 50:27 with Output Available and No Resume-Fault	2-454
2-154.	Commands/Descriptions for Sequences After First I-Sequence with Output Available and No Resume-Fault	2-455
2-155.	Sequences after First I-Sequence Essential Commands for f= 50:26, 50:27 with Output Not Available and No Resume-Fault	2-456
2-156.	Sequence after First I-Sequence Essential Commands for f= 50:26, 50:27 with Resume-Fault	2-457
2-157.	Commands/Descriptions for Sequences after First I-Sequence with Resume-Fault	2-458
2-158.	I-Sequence Data Flow for f= 50:20 - 50:23, 50:57	2-462
2-159.	I-Sequence Commands/Descriptions for f= 50:20 through 50:23 and 50:57	2-463
2-160.	Scan, I/O1, and I/O2-Sequence Essential Commands for EI Operations	2-468
2-161.	Scan and I/O2-Sequence Commands/Descriptions for EIR Operations	2-470

LIST OF TABLES (Cont)

Number	Title	Page
2-162.	I/O1, RTC, and Next Sequence Essential Commands for RTC Operations	2-474
2-163.	I/O1, RTC and Next Sequence Commands/Descriptions for RTC Operations	2-475
2-164.	Special and Monitor Interrupt Jump Addresses	2-479
2-165.	Special Interrupt Translator Conditions	2-480
2-166.	Scan and Interrupt in Parallel with I-Sequence Essential Commands for Special and Monitor Interrupts	2-482
2-167.	Scan and Interrupt/I-Sequence Commands/Descriptions	2-484
2-168.	EI Monitor Operation Command/Description	2-490
3-1.	Control/Indicator Functions, Input/Output Panel (A1 or Optional A8)	3-3
3-2.	Control/Indicator Functions, Control Panel 1 (A2)	3-8
3-3.	Control/Indicator Functions, Control Panel 2 (A4)	3-16
3-4.	Control/Indicator Functions, Power Control Panel (A5)	3-22
3-5.	Initial Switch Positions	3-25
4-1.	Computer Clearance Requirement	4-3
4-2.	I/O Cable Connector Pin Assignments (J1-J6, J21-J36)	4-7
4-3.	Pin Assignments for Remote Console Cable Connector (J17)	4-11
4-4.	Pin Assignments for External Sync Connector (J18)	4-14
5-1.	Inspect and Change Program Instructions	5-4
5-2.	Bootstrap Program Instructions	5-6

SAFETY SUMMARY

The following are general safety precautions that are not related to any specific procedures and therefore do not appear elsewhere in this publication. These are recommended precautions that personnel must understand and apply during many phases of operation and maintenance.

KEEP AWAY FROM LIVE CIRCUITS

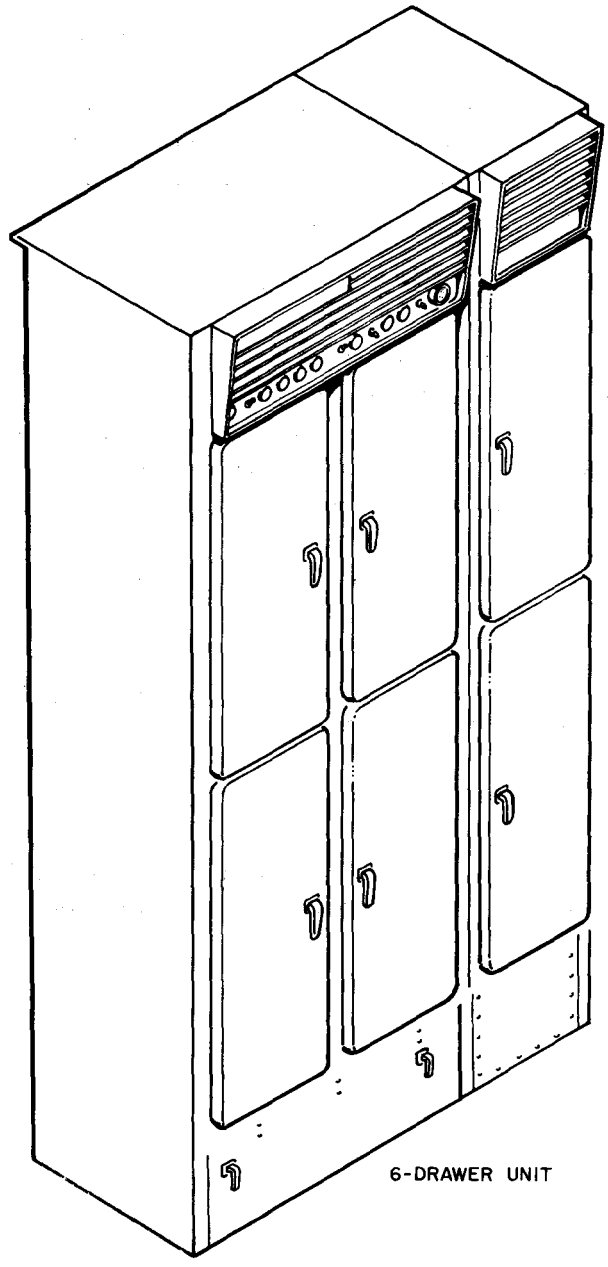
Operating personnel must at all times observe all safety regulations. Do not replace components or make adjustments inside the equipment with the high voltage supply turned on. Under certain conditions, dangerous potentials may exist when the power control is in the off position, due to charges retained by capacitors. To avoid casualties, always remove power and discharge and ground a circuit before touching it.

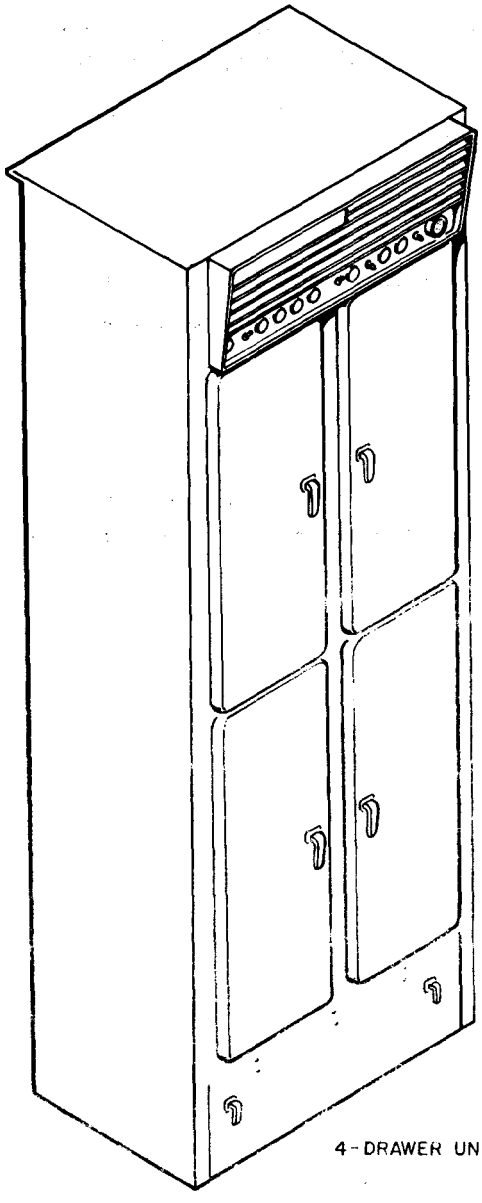
DO NOT SERVICE OR ADJUST ALONE

Under no circumstances should any person reach into or enter an enclosure for the purpose of servicing or adjusting the equipment except in the presence of someone who is capable of rendering aid.

RESUSCITATION

Personnel working with or near high voltages should be familiar with modern methods of resuscitation. Such information may be obtained from the Bureau of Medicine and Surgery.





Frontispiece. Digital Computer Mk 152 Series

CHAPTER 1

INTRODUCTION

1-1. PURPOSE

1-2. Digital Computer Mk 152 is a general-purpose stored-program real-time digital data-processing device used to solve fire control problems and to control operation of the weapons system. Computer Mk 152 Series (hereinafter referred to as the computer) is housed in one or more cabinets which contain a power supply, logic circuits, a core memory, a maintenance and control panel, and a cooling system. The physical characteristics of the computer depend upon the size of memory and the number of Input/Output (I/O) channels. The frontispiece (opposite page) illustrates both 4-drawer and 6-drawer configurations of the computer with protective cover doors closed. Located behind these doors are all control panels except the power control panel, which is located at the top of the unit. Figures 1-1 and 1-2 show the computers with doors open to expose the control panels.

1-3. FUNCTIONS. The computer uses a stored program from an extremely flexible repertoire of instructions to perform mathematical operations, control other equipment, provide communications, solve real-time problems, and perform data processing operations. Computer operation must be initiated manually; thereafter operation may be automatic for a given function. Computer operation for a specific function is initiated, controlled, and terminated by the program stored in the memory section of the computer. This program contains instructions, constants, decision-making capabilities, and an input/output capability. The instructions initiate and control specific operations. The decision-making capability is accomplished by instructions which compare quantities and make real-time decisions. The input/output capability allows data to be entered into or extracted from the computer. The instructions for a program are usually stored in memory at sequential addresses, and are performed in sequential order unless a program decision causes some instructions to be skipped. The program or routine is terminated when a pre-determined event or conclusion is reached or when all instructed operations have been performed. Principal functions performed by the computer in a typical shipboard installation are summarized as follows:

1. Communication with other elements of the weapons system in time-multiplexing fashion through an interfacing device.
2. Operation in conjunction with ancillary or peripheral equipment as, for example, the Input/Output Console Mk 77; acceptance of data loaded through a keyboard or tape reader and output of data which may be monitored through a perforator and printer.
3. Performance of logical, sequential, and arithmetic operations required to solve fire control problems and to control weapons system units.
4. Utilization of a priority scheme having decision-making capabilities to meet

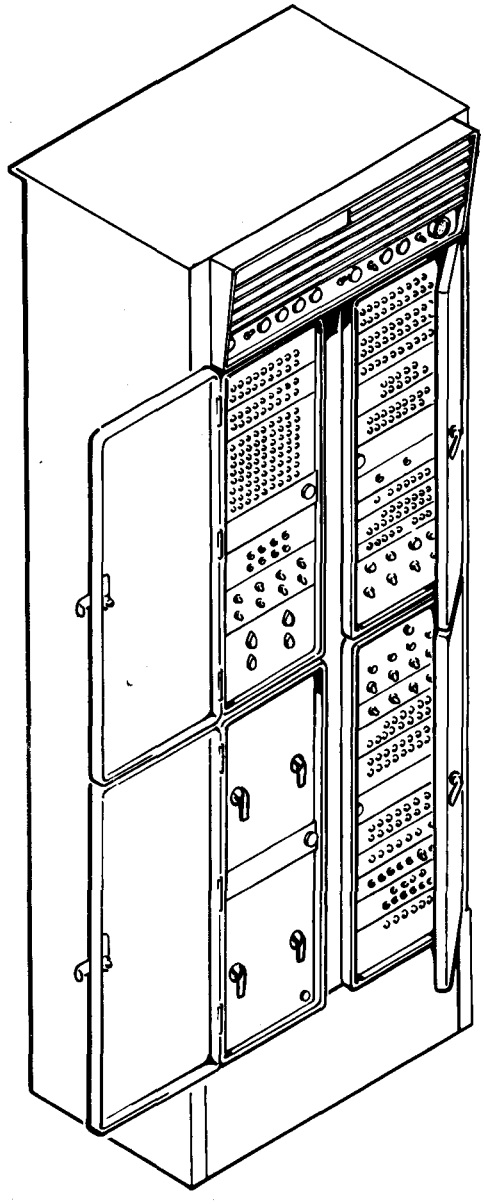


Figure 1-1. Computer with Maximum of Eight I/O Channels

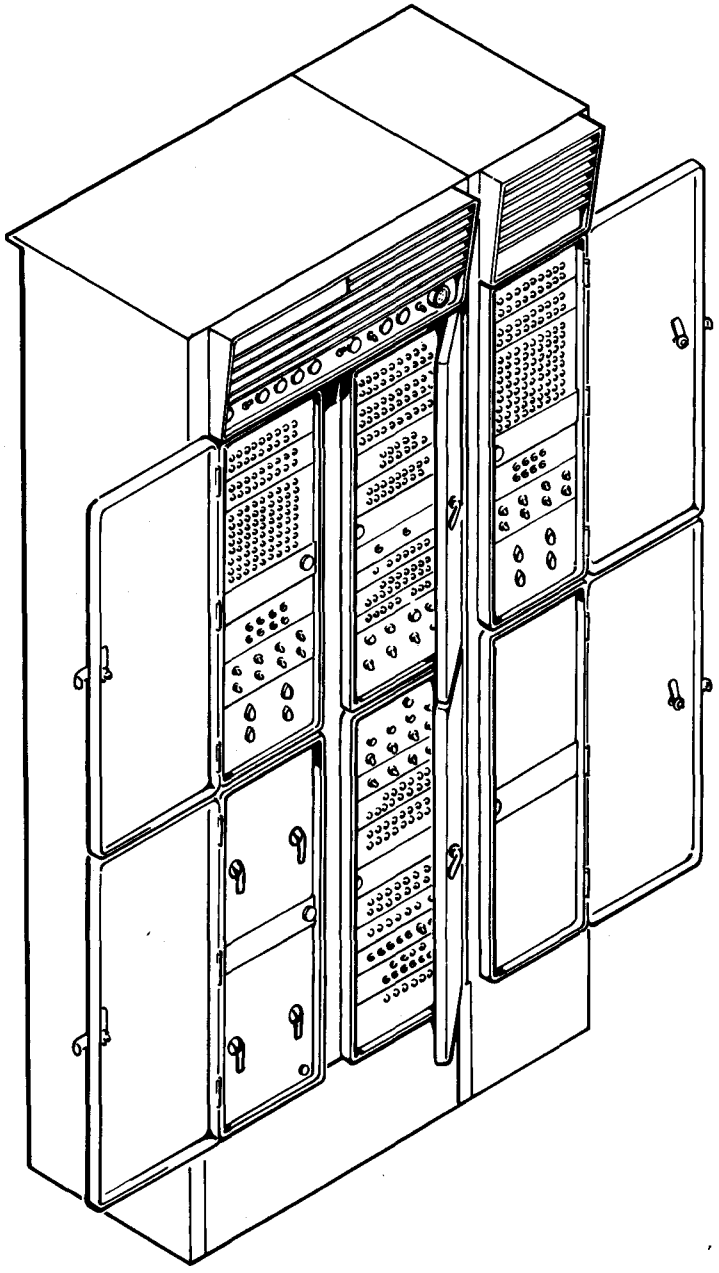


Figure 1-2. Computer with Maximum of 16 I/O Channels

extreme requirements of real-time and concurrent batch-processing operations through implementation of a program interruption system.

5. Performance of self-test functions manually and automatically for itself and for peripheral equipments under the control of computer test programs.

1-4. CAPABILITIES

1-5. The Digital Computer Mk 152 is a military version of UNIVAC 1219B, a member of the UNIVAC family of computers. The Mk 152 has greater operational speed than the widely used UNIVAC 1218 Computer and is more versatile than its immediate predecessor, the UNIVAC 1219. It is functionally compatible with either of these machines and is capable of executing programs written for either.

1-6. GENERAL CHARACTERISTICS. The computer is fully capable of meeting the extreme requirements of real-time and concurrent batch processing operations because it is equipped with a 2-microsecond internal random-access core memory in sizes of either 32,768 or 40,960 18-bit words with a read-access time of 0.75 microsecond and a fast, 500-nanosecond control memory. In addition to these capabilities, other random-access storage devices connected to I/O channels provide unlimited memory capacities. A portion (32 word locations) of core memory, which has a characteristic nondestructible feature, stores constants and instructions for automatic recovery from fault situations and for an initial load of routines.

1-7. The computer is designed with a modular I/O section which provides the option of 8 or 16 I/O channels for communication with peripheral equipments or other computers. Each 8-channel module is available in either of two types of interface design (slow or fast). I/O communication is normally accomplished in an 18-bit parallel mode; however, adjacent single channels can be combined into one 36-bit I/O channel by switch setting.

1-8. Because of its high internal operating speed, core memory, and 500-nanosecond control memory, the computer is capable of transferring 500,000 words per second. Arithmetic operations can be performed on the basis of a single-length 18-bit word, or a double-length 36-bit word if greater precision is required. The repertoire of 102 instructions allows complete programming freedom in mathematical and logical computations, as well as full control of I/O buffer transfers and of real-time on-line operations. The computer features buffered parallel transfers, one's-complement binary arithmetic, direct addressing, and program-controlled automatic address or operand modification via eight control-memory-contained index registers.

1-9. The ability of the computer to process various applications concurrently is implemented by a program-intervention system called Interrupts. Requests for program intervention may originate at some remote external device (External Interrupts) or they may originate within the computer (Internal Interrupts). Because requests for program intervention may occur simultaneously, the computer has the capability of selecting for solution the problem request requiring the most urgent attention. Under program control the other requests may be honored in turn according to the next highest priority, or they may be ignored. With this Interrupts feature, real-time problem solution is possible and maximum processing potential of the system is realized, since less important

routines can be processed during the computer's surplus time.

1-10. Operational Characteristics. Operational characteristics of Digital Computer Mk 2 Series are summarized in table 1-1.

1-11. SYSTEM DESCRIPTION

1-12. WEAPONS SYSTEM RELATIONSHIP. A typical shipboard weapons system comprises several subsystems including (1) a Fire Control System (FCS), (2) a Weapon Direction System (WDS), (3) a Guided Missile Launching System (GMLS) and/or guns, and (4) guided missiles, STANDARD missiles, and/or other kinds of projectiles. Each of these subsystems is itself composed of one or more major equipments. The computer is a major equipment that functions within the FCS along with other major equipments including the director and fire control radar set. Fire control systems utilizing various modifications of Digital Computer Mk 152 Series are installed on many different ship classes. Detailed information pertaining to a specific weapon system relationship is available in the appropriate weapon system Ordnance Pamphlet (OP).

1-13. COMPUTER COMPLEX. Figure 1-3 is a block diagram of a typical computer complex. Other ship classes have similar configurations. The components which comprise the computer complex along with a brief description of their functions are as follows:

1. Digital Computer Mk 152 Series is the heart of the computer complex and interfaces directly with all other units.

2. Input/Output (I/O) Console Mk 77 is used primarily to load data into the computer through a keyboard or tape reader and to monitor data outputs from the computer through a tape perforator and printer.

3. A Signal Data Converter (SDC) is shown in figure 1-3. Some type of converter is used on all digital ships to allow communications between the digital computer and weapons subsystem units.

4. Digital Data Recorder (DDR) Mk 19 is a large-capacity medium-speed storage system with capability either of receiving data from the computer and recording it on magnetic tape or of retrieving data which has been previously recorded on magnetic tape and transferring it to the computer. The data is recorded on 1/2-inch mylar-backed magnetic tape.

5. Motor Generator Set Mk 9 converts 208-VAC 60-Hz 3-phase power or 416-VAC 60-Hz 3-phase power to 115-VAC 400-Hz 3-phase power for operating equipments comprising the computer complex.

1-14. COMPUTER GROUP EQUIPMENTS. The computer may be connected to a variety of military or commercial peripheral equipments including any of the following devices:

1. Paper-tape reader-punch units
2. Magnetic-tape systems
3. High-speed printer units
4. Card reader-punch units
5. Teletype printer units
6. Display and display interface units
7. Radars and radar adapter units
8. Manual entry devices

It is impractical to describe in this publication all possible peripheral equipments which are, or could be, connected to the computer. Consequently, figure 1-4 depicts only the most commonly used computer group equipments. For detailed information, reference should be made to the appropriate equipment manual. (See paragraph 1-22, Reference Publications.) OPNAV 43P1 contains an explanation of the basic concepts and implementation of the Planned Maintenance System/Surface Missile Systems (PMS/SMS) at the equipment level. The appropriate preventive/corrective Maintenance Requirement Cards (MRC's), when used in conjunction with the corresponding

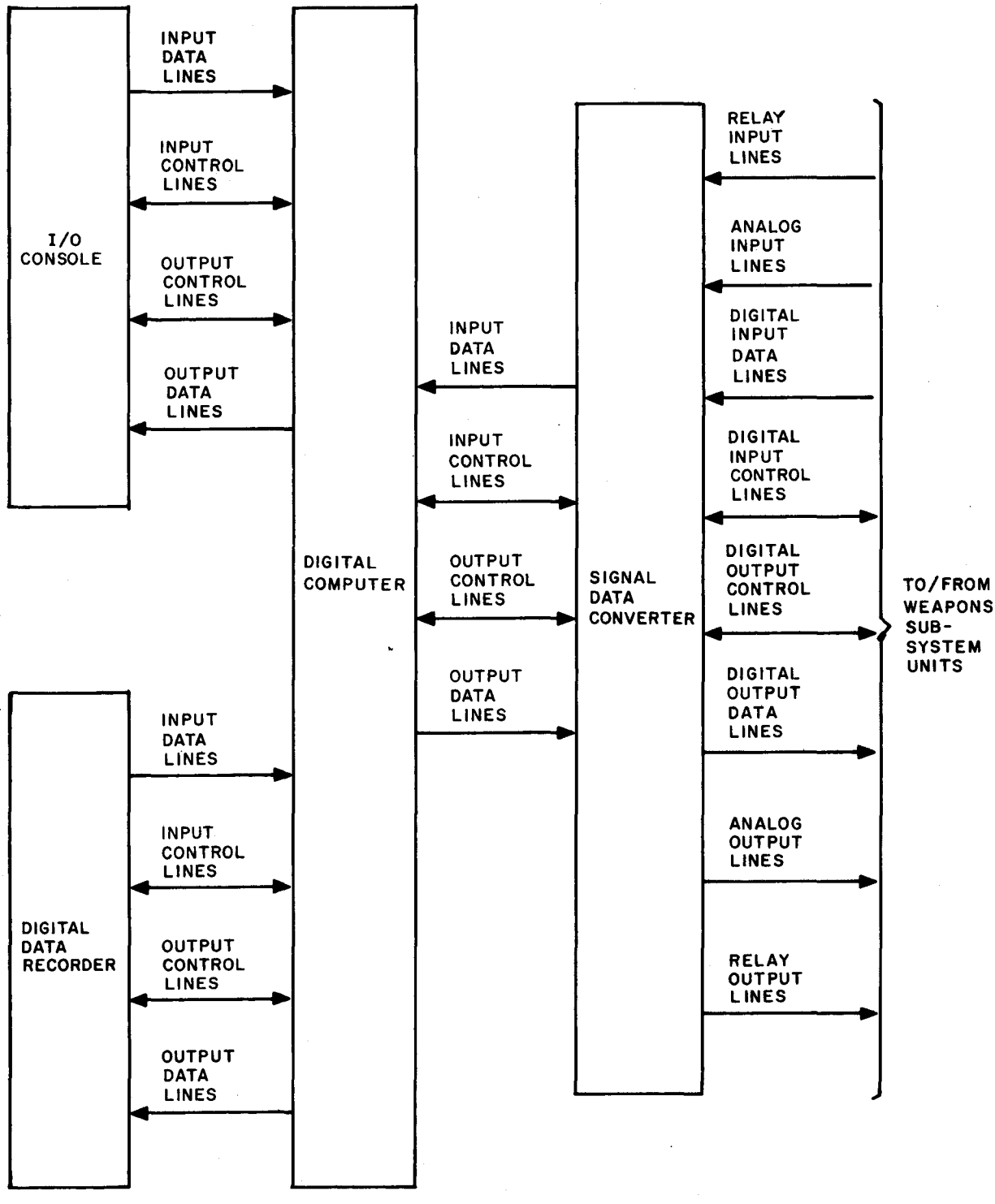


Figure 1-3. Block Diagram of Typical Computer Complex

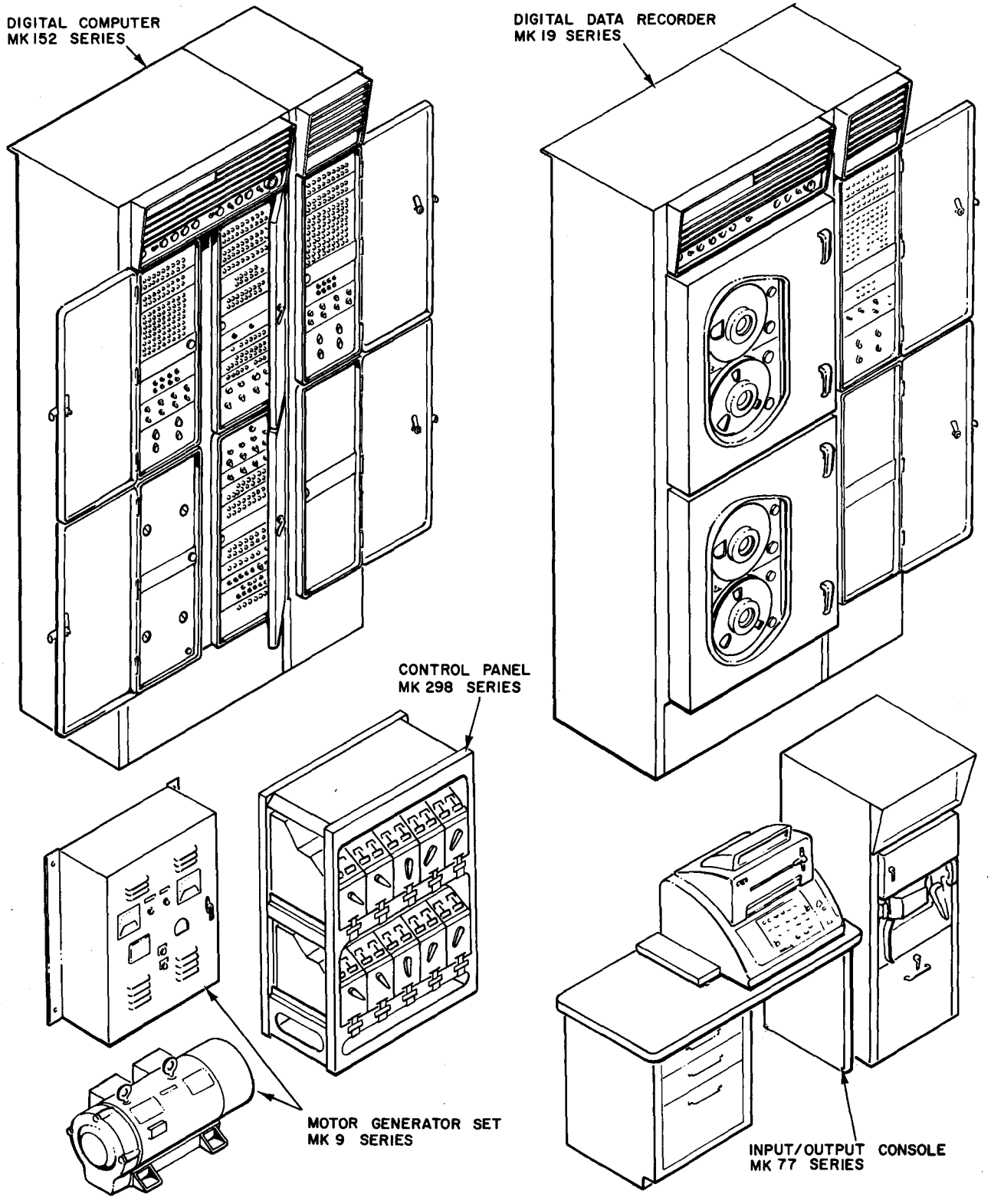


Figure 1-4. Computer Group Equipments in Common Usage

TABLE 1-1. OPERATIONAL CHARACTERISTICS

Item	Description
CONTROL	
Logic Levels:	Low = -4.5 VDC High = 0 VDC
Instruction Repertoire:	102 single address flexible instructions with provision for address modification by eight index registers.
Clock:	
Type	Automatic, additive, under program control
Location	Control memory
Duration	Established under program control
Granularity	Least significant bit represents 1/1024 second; other on request
Interrupt	Interrupt occurs when program preset value is reached or at overflow.
Synchronizing:	
Interrupt	Interrupt occurs whenever the non-I/O synchronizing control line is set to logical one by an external device.
Purpose	To allow a variable granularity clock function or to provide a high priority, alarm recognition capability
ARITHMETIC	
Organization:	Parallel subtractive one's complement either single length (18 bits) or double length (36 bits).

TABLE 1-1. OPERATIONAL CHARACTERISTICS (cont)

Item	Description
ARITHMETIC (cont)	
Execution Times:	Typical execution times, including instruction and data fetch plus indexing Add, subtract (single length) 4 μsec Multiply/divide 14 μsec Add, subtract (double length) 6 μsec Compare/masked compare and branch 6μsec Register shifts: right, left, single, double (n = shift count) 2+.5n μsec
MEMORY	
Control Memory:	
Word Length	18 bits
Cycle Time	500 nanoseconds
Access Time	300 nanoseconds
Capacity	128 or 256 18-bit words
Type	Word organized, magnetic core
Purpose	Index registers, clock cells, I/O buffer control, and interrupt registers

TABLE 1-1. OPERATIONAL CHARACTERISTICS (cont)

Item	Description
MEMORY (cont)	
Main Memory:	
Logic Levels	Low = 0 VDC = 0 High = +3 VDC = 1
Cycle Time	2 microseconds
Access Time	750 nanoseconds
Capacity	32,768, 40,960, or 49,152
Word Length	18 bits
Type	Coincident current, magnetic core
Purpose	Program and data storage
Nondestructive Readout Memory:	
Cycle Time	2 microseconds
Access Time	300 nanoseconds
Capacity	32 18-bit words
Type	Word organized, transformer core; unalterable
Purpose	Bootstrap (initial load) program storage. Paper tape or magnetic tape load is used.
INPUT/OUTPUT	
I/O Channels:	
Type	Simplex, 18-bit parallel

TABLE 1-1. OPERATIONAL CHARACTERISTICS (cont)

Item	Description
INPUT/OUTPUT (cont)	
Number	4, 8, 12, or 16; each input and output
Signal Level:	Fast Interface High = 0.0 VDC = 1 Low = -3.0 VDC = 0 Slow Interface High = 0.0 VDC = 1 Low = -15.0 VDC = 0
Operation:	Each channel fully buffered and, once activated, operates without program attention asynchronously at the rate of the peripheral unit.
Maximum Data Transfer Rate: (words per second)	Fast Interface Single channel: 166,667 18-bit words Dual channel: (1 odd and 1 even) 333,334 18-bit words (166,667 36-bit words) Multi-channel: (1 or more odd and 1 or more even) 500,000 18-bit words (max. main memory rate) Slow Interface Single channel: 41,667 18-bit words Dual channel: 83,334 18-bit words (41,667 36-bit words) Multi-channel: (2 or more odd and 2 or more even) 166,667 18-bit words (max. -15 VDC interface rate)

TABLE 1-1. OPERATIONAL CHARACTERISTICS (cont)

Item	Description
INPUT/OUTPUT (cont)	
Information Transfers:	
Input Channels	Input data; interrupt data
Output Channels	Output data; external command data
External Interrupts	From peripheral on each channel
Internal Interrupts	At buffer termination (under program control)
Processing Time Required	2 microseconds/word transferred 0 microsecond during extended sequence instructions
Delay due to Program	2 microseconds maximum
Operating Modes Standard:	
Normal Single Channel	18-bit parallel transfers
Normal Dual Channel	Consecutive, even/odd ascending numbered, channels may be paired to form a single 36-bit parallel channel on odd numbered channel.
Externally Specified Index	18-bit parallel data transfers with storage address indirectly specified by external device
Externally Specified Address	18-bit parallel data transfers with storage address directly specified by external device
Continuous Data Mode	Program-controlled automatic re-initiation of previously established buffers. Program-controlled termination of CDM. 18-bit parallel or 36-bit parallel I/O word transfers

TABLE 1-1. OPERATIONAL CHARACTERISTICS (cont)

Item	Description
INPUT/OUTPUT (cont)	
Intercomputer Single Channel	Direct 18-bit parallel data transfers with other computers. No interface adapters required for intercomputer communication.
Intercomputer Dual Channels	Direct 36-bit parallel data transfer with other computers. No interface adapters required for intercomputer communication.

equipment and weapon system manuals, provide all information required to maintain equipments of the computer group in proper operating condition.

1-15. MODEL DIFFERENCES

1-16. Several modifications of Digital Computer Mk 152 Series are identified in table 1-2. The number and speed of I/O channels and memory size are the two functional differences between these computers. The basic functional and physical descriptions and installation procedures described in this manual apply to all modifications.

1-17. REFERENCE DATA

1-18. Table 1-3, General Characteristics, contains quick reference data including physical dimensions and weight, required environmental conditions, and power requirements for optional computer configurations.

1-19. EQUIPMENT SUPPLIED. Since the computer is self-contained, no additional equipment is supplied other than a chassis wrench, which is used for engaging and disengaging the computer chassis during maintenance. Power and signal cabling is not supplied with the computer and must be provided in the proper length as Government Furnished Equipment (GFE). The cabling and end-connectors must be installed at the installation site.

1-20. SPECIAL TOOLS AND TEST EQUIPMENT. Reference should be made to NAVORD OD 24651, Weapons System Test Equipment Index, for information concerning available equipment to support testing, troubleshooting, and corrective maintenance of the computer and associated equipment in a shipboard environment. Chapter 7, in Volume 2 of this manual describes special tools and test equipment used in maintaining the computer.

1-21. EQUIPMENT NOMENCLATURE. Official nomenclature, equivalent UNIVAC designations, and common names are given in table 1-4 for proper identification of selected units which compose all or part of the computer complex on one or more ship classes. Designations, nomenclature, and common names of computer major units and

TABLE 1-2. MODIFICATIONS OF COMPUTER MK 152 SERIES

Mod No.	Number of Drawers	Size of Memory	Input/Output Channels
0	6	32k	16 Slow Interface
1	4	32k	8 Slow Interface
2	6	32k	8 Slow Interface 8 Fast Interface
3	6	40k	16 Slow Interface
4		40k	8 Slow Interface 8 Fast Interface
5	4	48k	8 Slow Interface

TABLE 1-3. GENERAL CHARACTERISTICS

Characteristic	Description	
Physical Description	4-Drawer Unit (Mod 1)	6-Drawer Unit (All Other Mods)
Weight (Max)	1050 pounds	1460 pounds
Overall Dimensions (Max):		
Height	72 inches	72 inches
Width	26 inches	38 inches
Depth	29 inches	29 inches
Cubic Content	54,288 cubic inches	79,344 cubic inches
Clearance Requirements:		
Top	12 inches	12 inches
Front	39 inches	39 inches
Rear	10 inches	10 inches
Sides	8 inches	8 inches
Required Environmental Conditions		
Operating Temperature Range	32°F (0°C) to 122°F (50°C)	
Overtemperature Warning	115°F (46°C)	
Overtemperature Shutdown	140°F (60°C)	
Humidity, Maximum Relative	95%	
Heat Dissipation	Approx. 5000 BTU's/hour for Mod 1	
AC Power Requirements		
Logic Power	115 VAC ±1%, 400 Hz ±5%, 3-phase, 3-wire	
Blower Power	115 VAC ±10%, 400 Hz ±5%, 3-phase, 3-wire	

AC Power Requirements Per Option

Computer Configuration		Max. Input Power Required
Memory Size	I/O Channels	at 115 Volts (Watts)
32k (Mod 1)	8	1860
32k (Mod 0, 2)	16	2080
40k (Mod 3, 4)	16	2300
48k (Mod 5)	8	2100

TABLE 1-4. COMPUTER GROUP EQUIPMENT NOMENCLATURE

Official Nomenclature	Mark (NAVORD)	Univac Designation	Common Name
Digital Computer	Mk 152 Series	1219B Family	Computer
Input/Output (I/O) Console	Mk 77 Series	1532 Family	I/O Console
Input/Output (I/O) Console	Mk 95 Series	1870 Family	I/O Console
Digital Data Recorder	Mk 19	1540 Family	Mk 19 Recorder
Control Panel	Mk 298	1299	Mk 298 Control Panel
Motor Generator	Mk 9	8503-1	Mk 9 MG Set

assemblies are discussed in Section 2-1.

1-22. REFERENCE PUBLICATIONS

1-23. Table 1-5 is a tabulation of documents covering system/equipment with which the computer is used, associated equipment, and support equipment, and includes various official special directives and manuals used by computer maintenance personnel and operators.

TABLE 1-5. REFERENCE PUBLICATIONS

Publication	Title
NAVSEA OP 3498 (PMS/SMS)	Signal Data Converter Mk 66
NAVSEA OP 3515 (PMS/SMS)	Input/Output Console Mk 77 Series
NAVSEA OP 3516 (PMS/SMS)	Digital Data Recorder Mk 19 Series
NAVSEA OP 3517 (PMS/SMS)	Control Panel Mk 298 Series
NAVSEA OP 3518 (PMS/SMS)	Motor Generator Set Mk 9 Series
NAVSEA OP 3974 (PMS/SMS)	Signal Data Converter Mk 72 Mod 0
NAVSEA OP 4002 (PMS/SMS)	Signal Data Converter Mk 75
NAVSEA OP 4245 (PMS/SMS)	Input/Output Console Mk 95 Series
NAVSEA OP 4324 (PMS/SMS)	Signal Data Converter Mk 72 Mod 13
ORDALT 00	Index of Alterations to Ordnance (Less Aviation) Numerical Listing of Ordnance Alterations
NAVSEA OP 1700	Standard Fire Control Symbols, Volumes I and III
NAVSANDA 2002	Navy Stock List of Forms and Publications
NAVORD OD 24651	Weapons System Test Equipment Index Volumes I and II
NAVPERS 10088	Digital Computer Basics
	Multimeter CTO-630-A (SCAT 4245) Instruction Manual
	Differential Voltmeter CCUH-803B/AG
	(SCAT 4208) Instruction Manual
	Oscilloscope CBTV-545-B (SCAT 4316) Instruction Manual

CHAPTER 2

DESCRIPTION

Section 2-1. Physical Description

2-1. INTRODUCTION

This section provides a physical description of major assemblies of the computer. See Volume 3, Parts List for detailed description of minor assemblies and components.

2-2. COMPUTER CABINET

2-3. BASIC CABINET. The computer is housed in a single deck-mounted air-cooled drip-proof cabinet and is self-sufficient except for required input power and data signal lines. The basic computer cabinet (figure 1-1) is the Mod 1 Computer configuration and is divided into four drawers, a power supply, two fan assemblies, and various panels necessary for operation/maintenance, control, and input/output connections. All drawers, chassis, and assemblies mounted within the cabinet frame assembly may be removed for maintenance or repair when required.

2-4. EXPANDED CABINET. An expanded cabinet (figure 1-2), used with all mods other than the Mod 1, has 16 input/output channels. Drawers A1 and A8 are identical, and an additional fan assembly is provided when this cabinet is used.

2-5. Remote Control Console. In some installations, the computer has a remote control console that duplicates several of the control switches. The remote console is connected to the computer by a standard input/output cable which is plugged into a special remote console jack.

2-6. Approximate Size and Weight. Approximate dimensions of the computer cabinet(s) are as follows:

1. Height - 72 inches (All mods)
2. Width - 26 inches, Mod 1 and 5
   38 inches, Mod 0, 2, 3, & 4
3. Depth - 29 inches (All mods)
4. Weight - 1050 pounds, Mod 1
   1460 pounds, Mod 0, 2, 3,4
   1150 pounds, Mod 5

2-7. Cooling. Three fan assemblies provide forced air cooling for the cabinet interior. Air enters through a grill at the top of the cabinet, flows through an air filter, circulates throughout the cabinet, and is exhausted through a grill at the lower rear of the cabinet.

2-8. Fuse Complement. Table 2-1 is a list of fuses used to protect computer circuits. Drawer locations and amperage ratings are also given.

2-9. Environmental Requirements. Computer operating temperature ranges are from 0°C (32°F) to 50°C (122°F). Allowable non-operating temperature ranges are from -62°C to +75°C. The computer can operate with a maximum relative humidity of 95 percent.

TABLE 2-1. LIST OF FUSES

Drawer	Fuse	Rating (Amps)
A1, A8 (I/O drawer)	F1, F4	5
	F2, F5	12
	F3, F6	4
A2 (Control drawer)	F1	8
	F2	10
	F3	5
	F4	6
	F5	12
	F6	4
A3 (Memory drawer)	F1, F2, F3, F6	1
	F4, F5	20
	F7	15
	F8	8
	F9	0.5
A4 (Control drawer)	F1, F4	6
	F2	12
	F3	3
	F5	8
	F6	5
PS1 (Power supply)	F1, F2, F3	8
	F4, F5, F6	2
	F7, F11	6
	F8, F9, F10	20
	F12, F13, F14	5
	F15, F16, F17	12
A5 (Power supply)	F1	12

2-10. CABINET DRAWERS AND CHASSIS. Figure 2-1 shows the locations of logic/memory drawers A1 through A4 contained in the basic cabinet, as well as the additional drawer A8 (and A9) furnished with the expanded cabinet. Each logic drawer contains two logic chassis as illustrated in figure 2-2. Volume 3, Parts List provides identification of units. The locations of certain other major assemblies are also shown in figure 2-1. Drawers A1, A2, A4, and ~~A8~~ contain logic circuitry needed for control of various computer instructions and transfers. Drawer A3 contains computer core memory and associated circuitry. Power supply PS1 produces AC and DC voltages used throughout the computer cabinet. Fan assemblies A6, A12 and ~~A13~~ provide forced air cooling for the cabinet drawers. Power Control Panel A5 allows complete control of the computer from a single panel during performance of turn on and turn off sequential procedures. Abnormal conditions are indicated by audible and visual warnings. Table 2-2 is a list of power control switches and relays.

2-11. Logic Drawers. The logic drawers (see figure 2-2) contain circuitry required to perform control and logic functions designated by the computer program. Indicators and switches located on the front of each drawer provide a visual display and manual control of circuit conditions.

2-12. Mounting. Each drawer is slide-mounted to allow easy access to the cabinet interior and to facilitate inspection and maintenance of drawer assemblies.

2-13. Drawer Locking. Each logic drawer is held in place by a locking screw which, when loosened, permits the drawer to be extended approximately 26 inches from the cabinet. Extension of the drawer breaks the electrical connections by disconnecting the interassembly cable connectors from their mating receptacles on the rear of the cabinet frame.

2-14. Logic Chassis. Each logic drawer contains two vertically mounted logic chassis, a front panel, and twelve connectors for interassembly cables. One side of each chassis contains printed circuit modules and the reverse side contains intrachassis wiring. The indicators and manual controls associated with the circuits on the chassis are mounted on the front panel of the drawer. The front panel is hinged to open and expose circuit test blocks.

2-15. Memory Drawer. Memory drawer assembly A3 contains core memory stacks and associated circuitry to provide 18-bit word storage in the computer. A single memory drawer can house 16 core memory stacks for a maximum storage capacity of over 65 thousand 18-bit words. Figure 2-3 shows the location of memory stacks on an extended memory drawer.

2-16. Control and Bootstrap Memories. Control and bootstrap memories are located in logic drawer A4 of the standard cabinet. Figure 2-4 illustrates the control memory stack and the bootstrap memory assembly and their locations in the logic drawer.

2-17. Power Supplies. Power supply PS1 (see figure 2-5) contains the main DC power supplies which furnish DC voltage required for computer logic functions. Table 2-3 lists the DC power supply outputs and their characteristics.

2-18. Main Memory Power Supply. An additional power supply is located in memory drawer A3. This power supply provides the outputs (listed in table 2-4) specifically required for main memory operation.

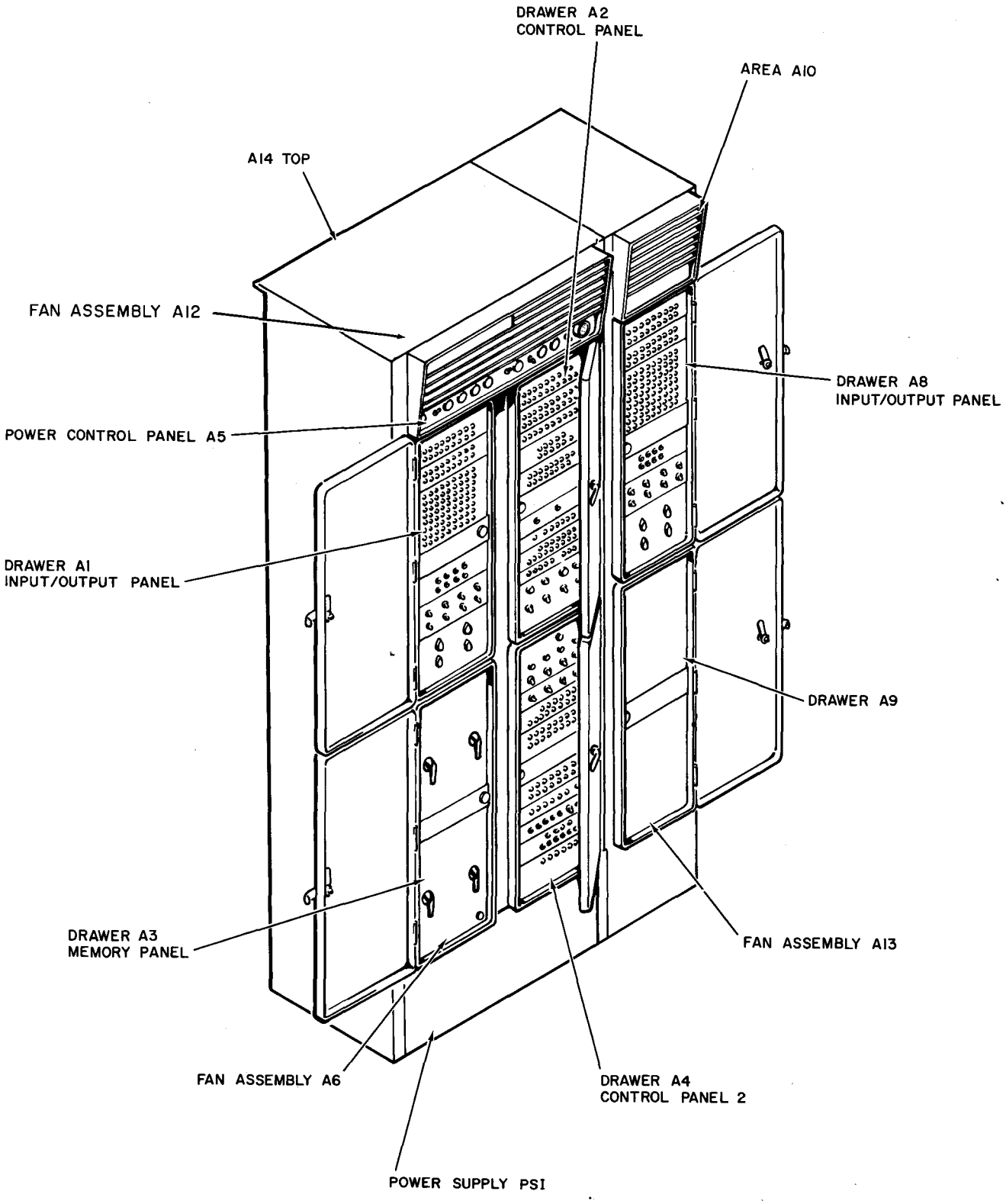


Figure 2-1. Computer Drawer Locations

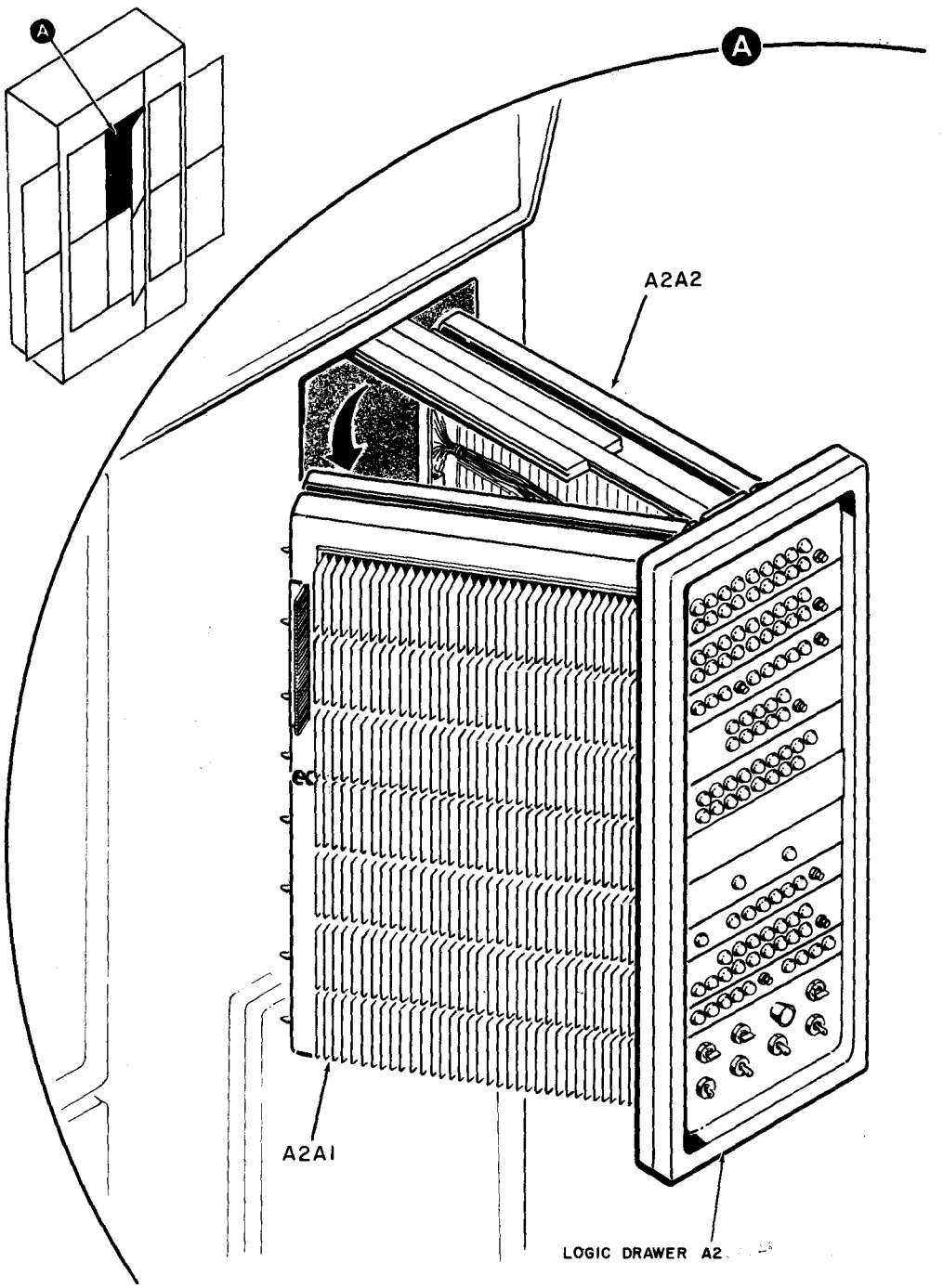


Figure 2-2. Computer with Logic Drawer A2 Extended and Logic Chassis A2A1 Partially Opened

TABLE 2-2. COMPUTER POWER CONTROL SWITCHES AND RELAYS

Relay or Switch Designator	Name	Plate Reference
PS1K1	Power control relay	P-178
A5S1	INDICATE-OFF-INDICATE/SET switch	P-176
A5S2	BATTLE SHORT switch	P-176
A5S3	DISC ALARM-RESET ALARM switch	P-176
A5S4	POWER ON-OFF switch	P-176
A12S1	Power supply interlock switch	P-177
A12S2	115° Temperature alarm switch	P-177
A12S3	140° Overtemperature switch	P-177
A12S4	115° Temperature alarm switch	P-177
A12S5	140° Overtemperature switch	P-177

TABLE 2-3. DC POWER SUPPLY OUTPUTS

DC Voltage	Tolerance	Current (Max. Amp)
+15.0*	+13.5 to 16.5 Volts	40.0
-15.0*	-13.5 to -16.5 Volts	48.0
-4.50	-4.0 to -4.5 Volts	29.0
±10.0 (Main Memory)	Adjustable from ±9.0 to ±11.0	0.2 amp for each 4K memory bank
±10.0 (Control Memory)	Adjustable from ±7.0 to ±11.0	6 amp peak
±15.0 (Control Memory)	Adjustable from ±13.5 to ±16.5	6 amp peak

* Absolute values of these two voltages may differ by not more than 6% of the lower voltage.

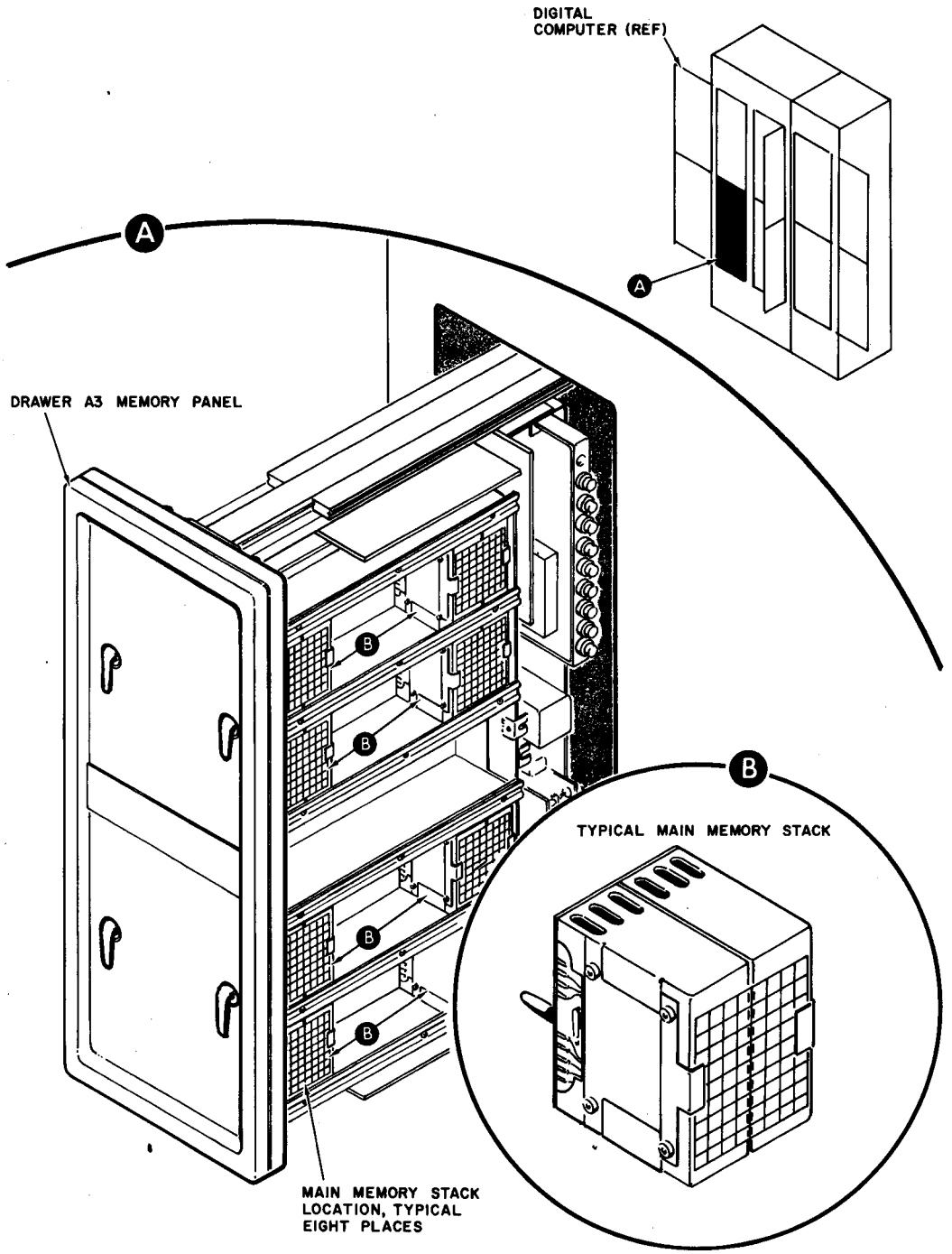


Figure 2-3. Typical Memory Stack in Memory Drawer

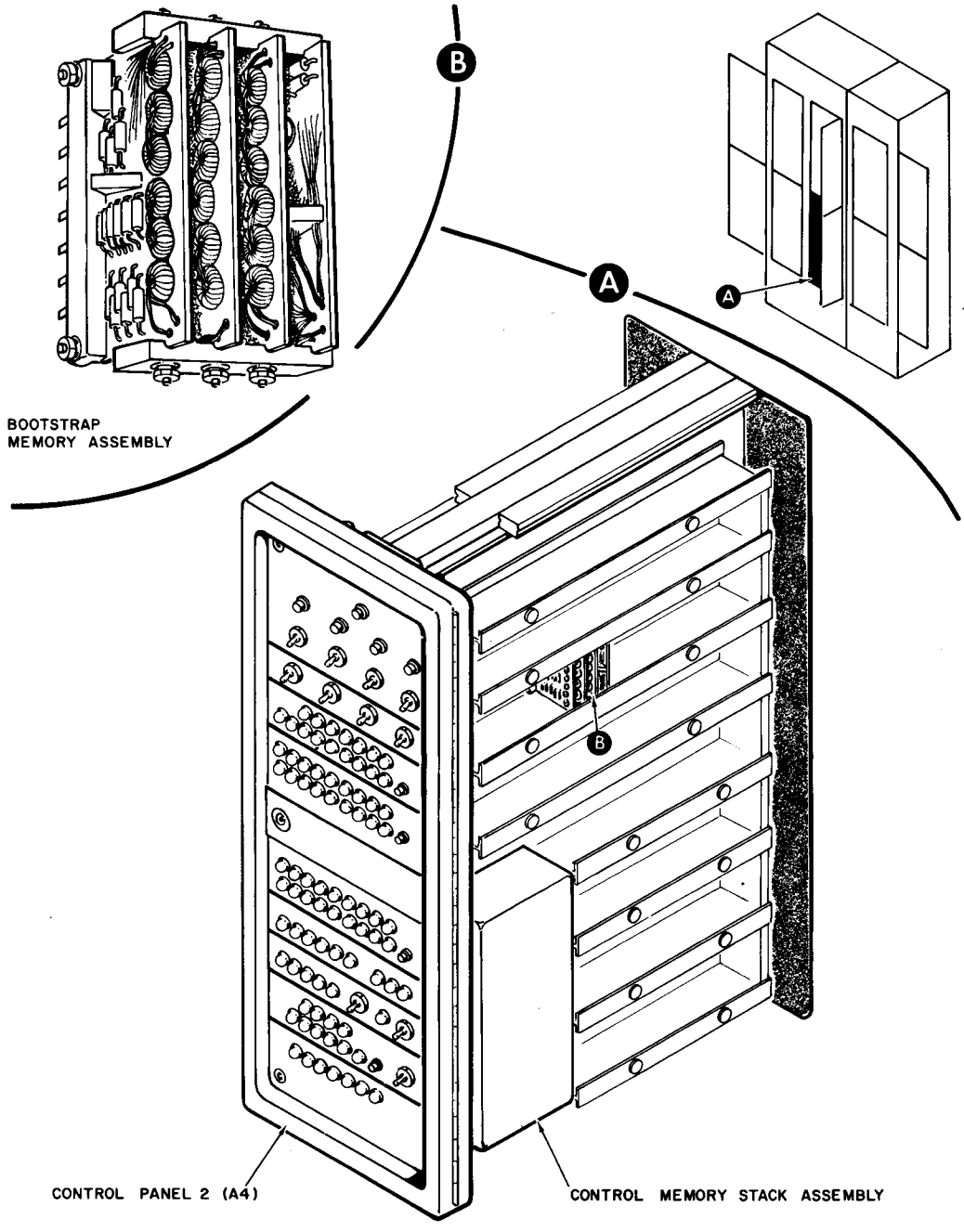


Figure 2-4. Control and Bootstrap Memories in Logic Drawer A4

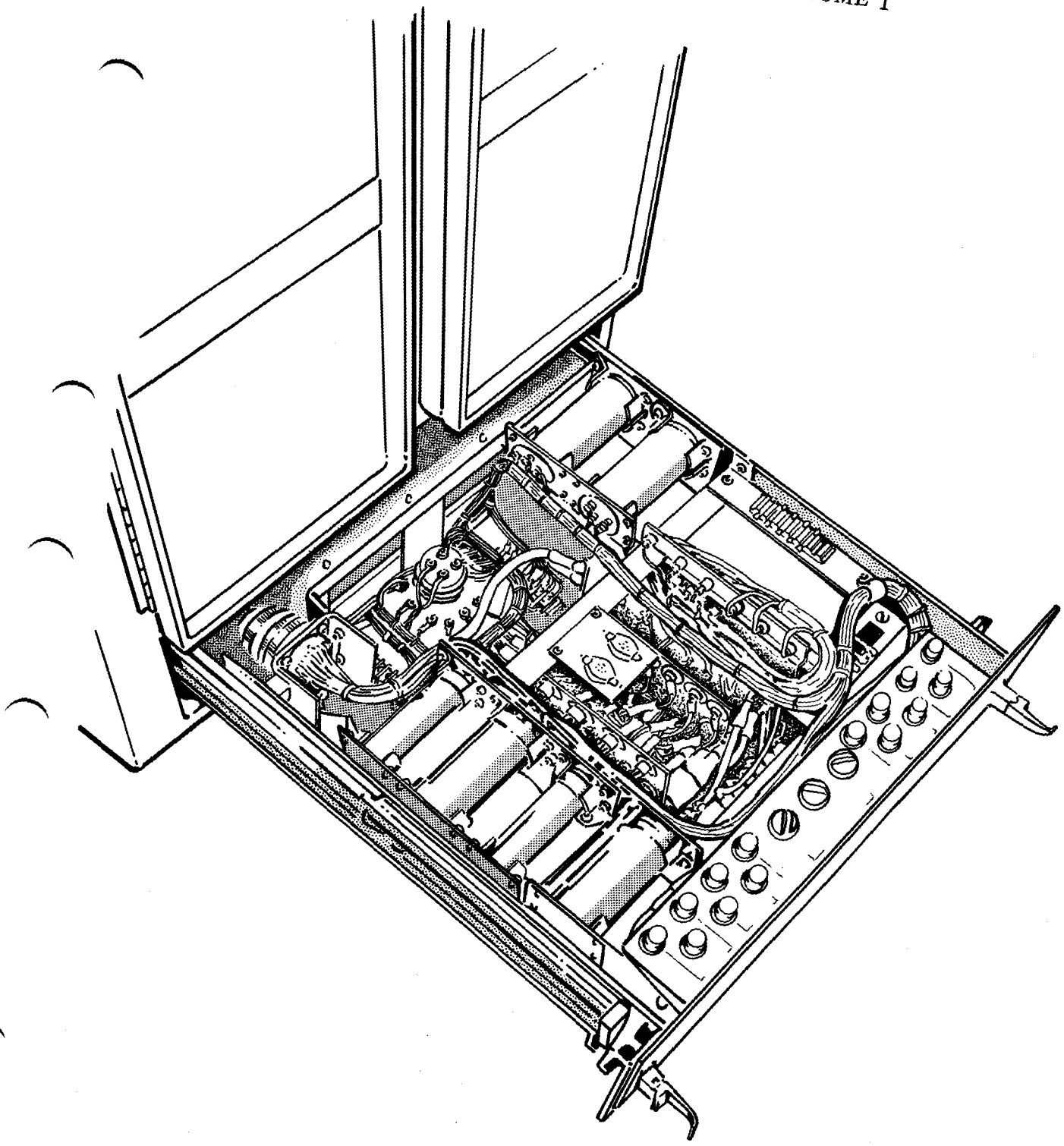


Figure 2-5. Power Supply PS1

2-19. Panels. Power Control Panel A5 provides control of computer power, a visual display of computer operating conditions, and an audible alarm for signaling temperature/voltage/program irregularities. Switches and indicators for logic control are mounted on the drawer-front panels, and provide manual control and allow proper sequencing during initial power applications. The cable entry panel is located on the cabinet top and contains input and output connectors, a remote control connector (para 2-5), and an external synchronizing jack.

2-20. COMPUTER ASSEMBLIES

2-21. Figure 2-6 identifies various computer units and assemblies that are contained within both electrical equipment cabinets. Table 2-5 provides the official nomenclature and applicability of the major units and assemblies.

2-22. MODULE COMPLEMENT. The printed circuit module is the basic replaceable unit of the computer and provides a mounting board for circuit components. The computer uses two basic types of modules (figure 2-7). Refer to Chapter 7 for schematic diagrams.

2-23. Physical Description of Basic Modules. In general, there are two types of printed circuit cards to supply the logic circuitry within the computer. The two types differ in the number of contact connections and circuit content. With few exceptions type A contains from one to five circuits and terminates in a 15-pin connector. Type B contains from 1 to 16 circuits and terminates in a 56-pin connector. The type B cards are used in memory logic and control to minimize space and provide maximum memory storage capability.

2-24. Card-type numbers listed in the functional schematics of Chapter 9 are the last four digits of the actual part number. For the complete number, refer to tables 2-6 and 2-7.

2-25. Each card is color coded, using the standard resistor color coding scheme according to the last four digits as shown in the functional schematics. The 2013 card (7002013) is color coded red, black, brown, and orange and is an example of the coding system.

2-26. The least significant digit in the card type number indicates the revision of the basic card. For example, card type 7002015 indicates the third revision of the basic 7002010 card.

TABLE 2-4. MAIN MEMORY POWER SUPPLY OUTPUTS

DC Voltage	Tolerance	Current (Max. Amp)
+3.0	+2.9 to +3.1	8.0
-3.0	-2.9 to -3.1	0.5
-15.0	-13.5 to -16.5	20.0
+15.0	+13.5 to +16.5	20.0
+6.0	+5.7 to +6.3	15.0

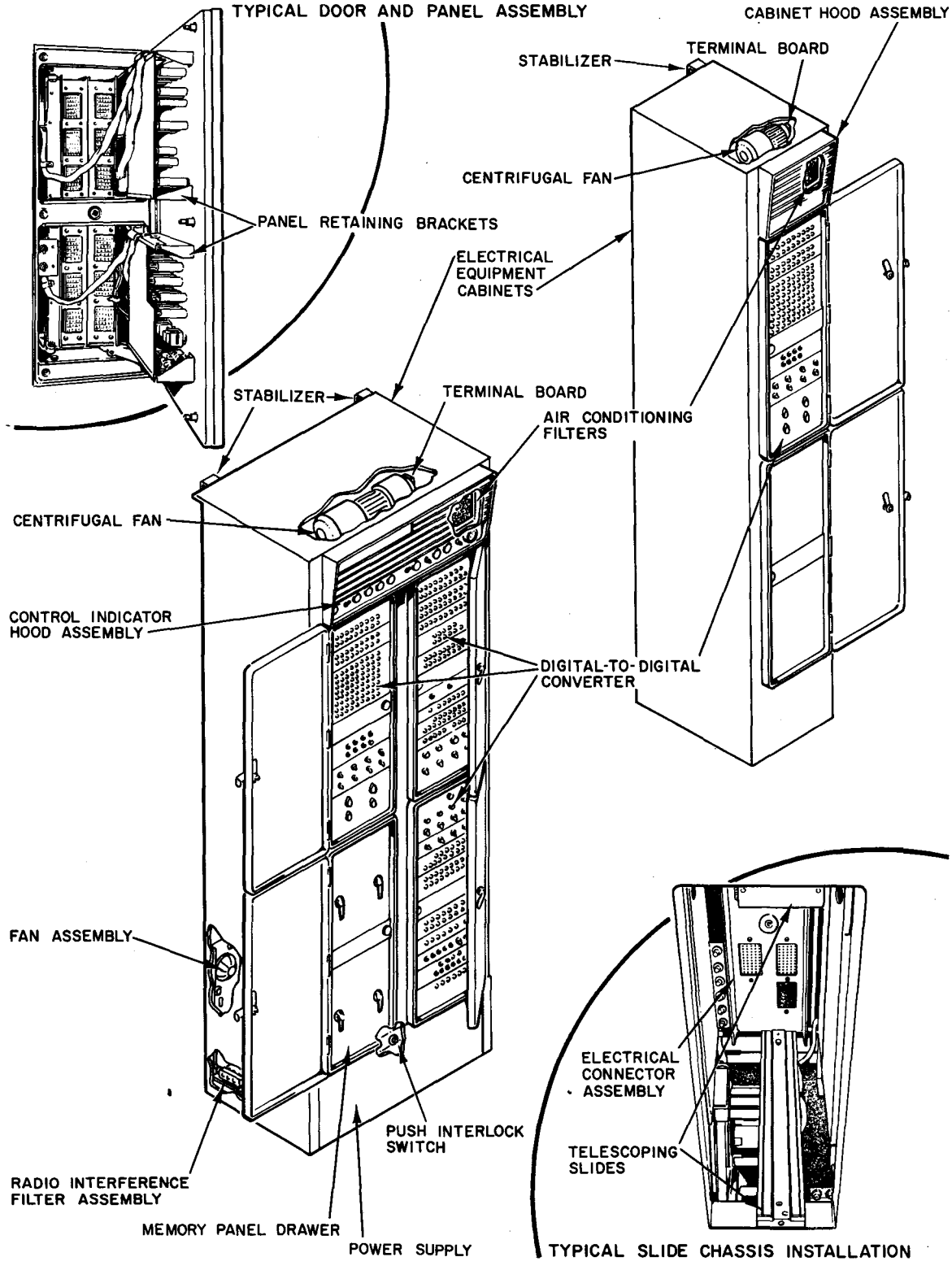


Figure 2-6. . Computer Mk 152 Series, Major Units and Assemblies

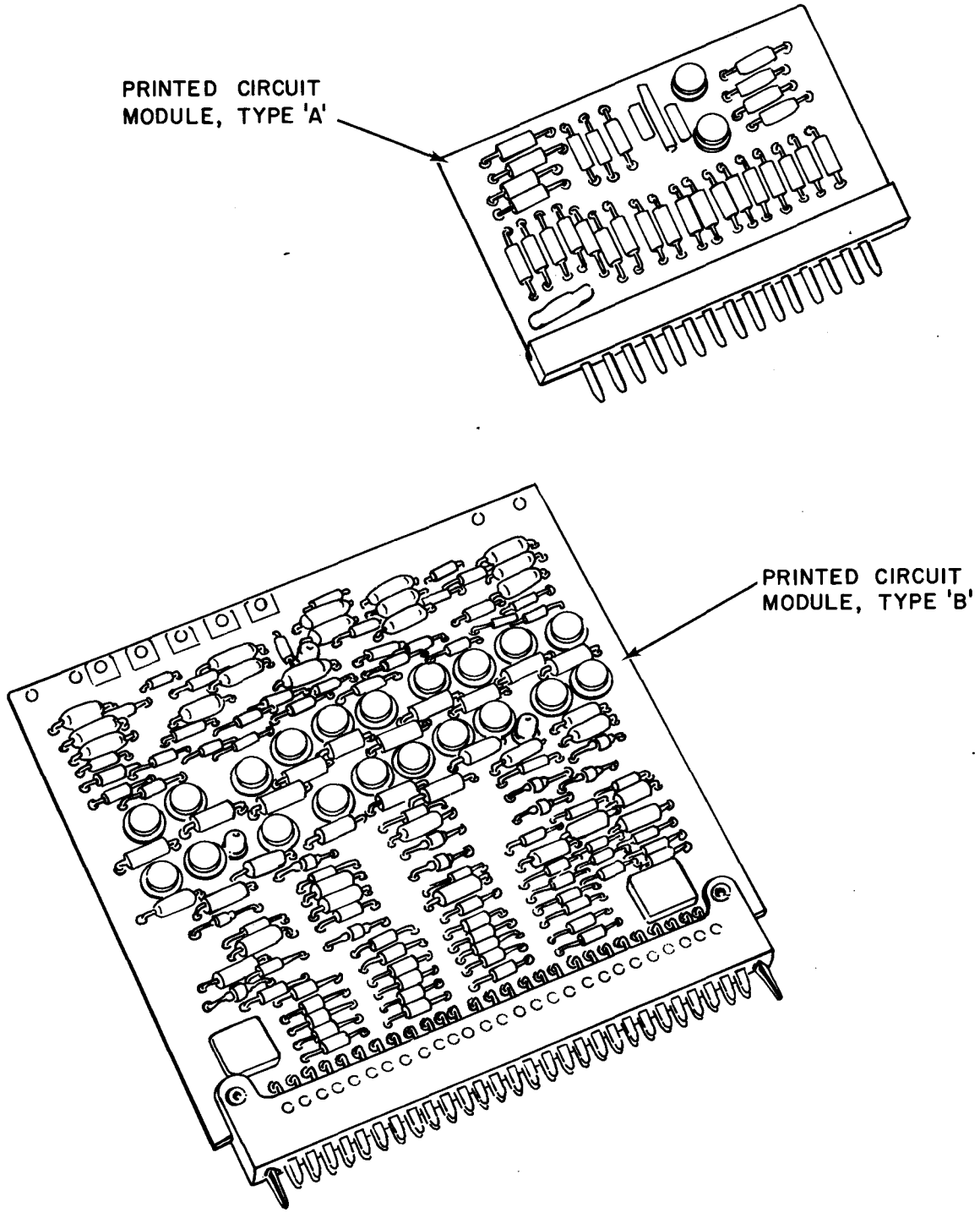


Figure 2-7. Printed Circuit Modules

TABLE 2-5. DESIGNATIONS, NOMENCLATURE, AND APPLICABILITY OF MAJOR UNITS AND ASSEMBLIES

Nomenclature	Applicability	Ref. Des.	Part No. **	EIC
Computer, Digital Mk 152 Series	Mod 0	Unit*	2652595-00 (7049747-00)	5CZ1000
	Mod 1		2525383-00 (7049747-05)	5CZ2000
	Mod 2		2536007-00 (7049747-08)	5CZ3000
	Mod 3		2687278-00 (7049747-11)	5CZ4000
	Mod 4		2923042-00 (7049747-13)	5CZ5000
	Mod 5		3140291-00 (7049747-14)	5CZ6000
Converter, Digital-to-Digital		A1	2685361-01 (7053795-01)	
		A2	2685361-02 (7053795-02)	
		A4	2685361-03 (7053795-03)	
	Mods 0, 2, 3, Only	A8	2685361-00 (7053795-00)	
Memory Drawer Assembly	Mods 0, 1, 2,	A3	2685331-00 (7053750-00)	
	Mods 3, 4, and 5	A3	2685331-01 (7053750-04)	
Hood Assembly, Control-Indicator		A5	2685418-00 (7050307-01)	
Fan Assembly, Axial		A6	2685256-00 (7033221-01)	
Connector Assembly, Electrical		A12	2685365-00 (7053799-00)	
	Mods 0, 2, 3 Only	A13	2685195-00 (7025669-06)	
Power Supply		PS1	2685289-00 (7038864-02)	
Filter Assembly, Radio Interference		A14	2685262-00 (7033257-00)	

* Refer to applicable system OP for specific computer configurations

** NAVORD Part Numbers are given followed by UNIVAC Part Numbers in parentheses

TABLE 2-6. PRINTED CIRCUIT MODULE COMPLEMENT, LOGIC CIRCUITS

NAVORD (UNIVAC) Part No.	Description	Mod 1, 5	Mods 0, 3	Mods 2, 4
2684511 (7000210)	Pulse-Delay Oscillator	1	1	1
2684519 (7002000)	Flip-Flop	103	103	103
2684520 (7002013)	Driver-Amplifier	124	156	156
2684521 (7002020)	Flip-Flop	86	118	118
2684522 (7002030)	Inverter	79	144	144
2684523 (7002040)	Inverter	66	75	75
2684524 (7002050)	Inverter	124	144	144
2684525 (7002060)	Inverter	145	183	183
2684526 (7002070)	Inverter	96	125	125
2684527 (7002080)	Inverter	10	12	12
2684528 (7002090)	Input Amplifier	97	185	97
2685118 (7002100)	Driver Amplifier	3	3	3
2685119 (7002120)	Driver Amplifier	2	2	2
2684529 (7002130)	Driver Amplifier	24	48	24
2684530 (7002141)	Driver Amplifier	36	72	36

TABLE 2-6. PRINTED CIRCUIT MODULE COMPLEMENT, LOGIC CIRCUITS (Cont)

NAVORD (UNIVAC) Part No.	Description	Mod 1, 5	Mods 0, 3	Mods 2, 4
2684531 (7002160)	Inverter	7	7	7
2684532 (7002220)	Inverter	4	4	4
2684533 (7002321)	Differential Amplifier			88
2684534 (7002332)	Control Line Amplifier			24
2686823 (7002342)	Data Line Amplifier			36
2686825 (7002730)	Jumper Board			2
2685120 (7002861)	Voltage Regulator	1	1	1
2685121 (7002880)	Voltage Sensor	1	1	1
2684539 (7002900)	Flip-Flop	109	143	143
2684540 (7002920)	Inverter	65	73	73
2684541 (7002930)	Flip-Flop	62	104	102
2684545 (7003180)	Capacitor Assembly	38	48	48
2684547 (7003480)	Time Delay	8	16	8
2685122 (7003490)	Driver Amplifier	3	3	3
2685123 (7003600)	Regulator Amplifier	1	1	1

TABLE 2-6. PRINTED CIRCUIT MODULE COMPLEMENT, LOGIC CIRCUITS (Cont)

NAVORD (UNIVAC) Part No.	Description	Mod 1, 5	Mods 0, 3	Mods 2, 4
2685124 (7003621)	Memory Driver	9	9	9
2685125 (7003630)	Pulse Delay Network	3	3	3
2685126 (7003640)	Emitter Follower	5	5	5
2685127 (7003670)	Driver Amplifier	4	4	4
2685128 (7003680)	Transformer Driver	4	4	4
2685129 (7003710)	Pulse Delay Network	2	2	2
2685130 (7003720)	Voltage Sensor	1	1	1
2685131 (7003730)	Regulator Amplifier	1	1	1
2685132 (7003740)	Control Amplifier	6	6	6
2685133 (7003760)	Current Diverter	2	2	2
2685134 (7003771)	Driver Amplifier	6	6	6
2685135 (7003780)	Driver Amplifier	18	18	18
2685136 (7003850)	Sense Amplifier	6	6	6
2685137 (7104010)	Selector Mode	1	1	1

TABLE 2-6. PRINTED CIRCUIT MODULE COMPLEMENT, LOGIC CIRCUITS (Cont)

NAVORD (UNIVAC) Part No.	Description	Mod 1, 5	Mods 0, 3	Mods 2, 4
(7109000)	Resistor Assembly	3	4	4
(7109010)	Resistor-Capacitor Assembly	3	4	4
	Total (See Note 1)	1369	1955	1947

NOTE 1 - Nominal Value Because of Customer Options

TABLE 2-7. PRINTED CIRCUIT MODULE COMPLEMENT, MEMORY CHASSIS

NAVORD (UNIVAC) Part No.	Description	Memory Capacity		
		32K (Mod 0, 1, 2)	40K (Mod 3, 4)	48K (Mod 5)
2685419 (7500040)	Driver Amplifier	3	5	5
2685420 (7500260)	Oscillator, Delay Line Amplifier	1	2	2
2685421 (7500280)	Emitter Follower	1	2	2
2685422 (7500320)	Voltage Regulator	1	2	2
2685423 (7500340)	Voltage Sensor	1	2	2
2685424 (7500400)	Memory Switch, Bipolar	11	17	17
2685425 (7500421)	Drive Diverter	4	6	6
2686800 (7500431)	Transformer Assembly	8	10	12
3137506 (7500651)	Sense Amplifier	36	54	54
2685428 (7500660)	Capacitor-Diode Assembly	4	6	6
2685429 (7500671)	Capacitor Assembly	1	2	2
2685430 (7500761)	Level Change Amplifier	6	12	12
2686801 (7500781)	Voltage-Current Regulator	2	3	3
2685432 (7500900)	Driver Amplifier	2	4	4

Section 2-2. General Functional Description

2-27. GENERAL DESCRIPTION

2-28. The computer is a general-purpose stored-program real-time digital machine fully capable of solving any problem amenable to solution by standard mathematical techniques; it can also perform most data processing functions. Computer logic is divided into four functionally definable sections: control, arithmetic, memory, and input/output (figure 2-8).

2-29. CONTROL SECTION. The control section provides the timing, instruction translation, and operational sequencing required for performance under either program control or manual operation. Under program control, the computer performs the instructions of an entire program at a high rate of speed, stopping only where programmed. For manual operation, two control panels contain the switches and indicators by which the computer may be sequenced through the functions of an instruction, allowing each operation and its results to be visually displayed and examined. These two are control panel 1, located on drawer A2, and control panel 2, located on drawer A4.

2-30. The control section of the computer (figure 2-9) consists of five logically definable areas: console control, timing, program translation and control, registers, and special circuits.

2-31. Console Control. The console control area contains the controls, indicators, and logic circuitry required by an operator for program control and maintenance. Switches in the console control area provide for master clearing of the computer, starting and stopping of operations, operational mode selection, controlling of programmed stops and skips, and manual control of computer logic as follows:

1. MASTER CLEAR sets all affected registers to zero, whereas the I/O CLEAR switch position operates selectively on the input/output section of the computer.

2. Computer start logic consists of start and restart. Start is a manual keying of the computer to initiate program run; restart provides an automatic keying useful in troubleshooting.

3. Stops may be initiated by manual switching or by programmed instructions. The program-stop logic detects programmed stops and compares these instructions with the switch settings that effect the stops. A programmed stop is an instruction which causes the computer to cease operation until manual intervention again returns it to a run status. There are five switch-controlled stops and one unconditional stop.

4. The program-skip logic functions during the execution of skip instructions to effect the omission of the next sequential instruction if the necessary conditions have been met. Skip conditions include switch settings similar to program stops and other conditions reflecting the results of certain computer operations, such as compare instructions.

5. The mode selection logic determines the mode of operation of the computer. The four modes are as follows:

a. Load Mode. The Load Mode causes the computer logic to address the first instruction of the Bootstrap memory. The computer operates at high speed while the addressing function is being performed and then reverts to normal. b. Phase Step. Phase Step Mode provides normal computer operation at a controlled rate of speed and is used principally for isolation of computer logic malfunctions. c. OP Step. In this mode the computer operates at a high rate of speed but its operation is stopped after each instruction, or each sequence, depending on position of SEQ/STOP switch. d. Run Mode. This is the normal mode for operating the computer at high speed. All operations that do not involve trouble isolation or program debugging are performed in this mode. Each of the four modes is selectable by one of the indicating pushbutto. located on the logic panel.

6. Clock phases may be set for trouble-shooting purposes with a similar set of indicating pushbuttons. Additionally, clock phases may be continuously repeated through use of the PHASE REPEAT switch. The FUNCTION REPEAT switch forces the

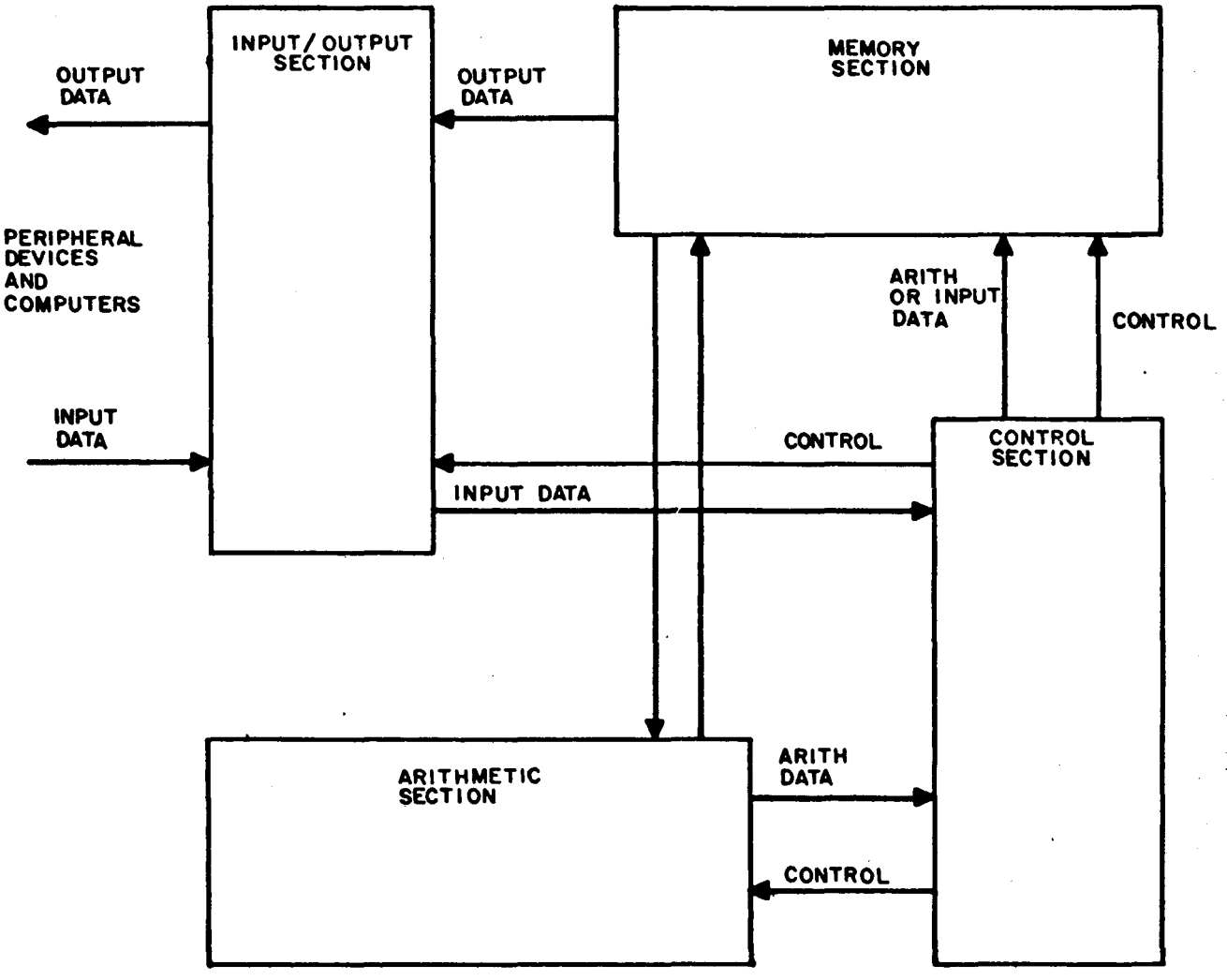


Figure 2-8. Computer Simplified Block Diagram

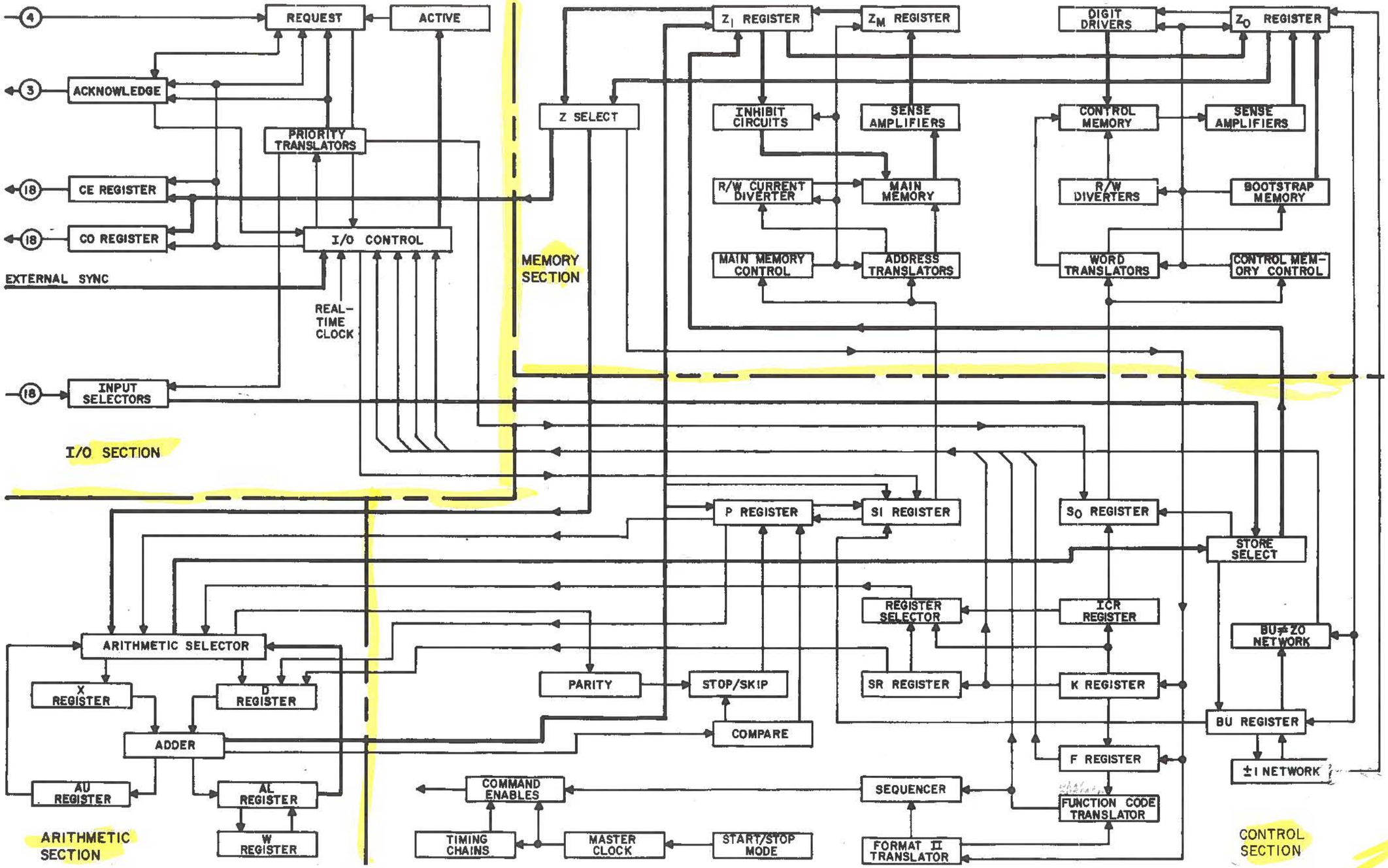


Figure 2-9. Computer Block Diagram

computer to perform the same instruction operation repeatedly. AUTO RECOVERY switch controls fault recovery logic and the DISC ADV P switch inhibits incrementing of the program counter.

2-32. Timing Circuits. Timing commands within the computer are a function of the master clock and the main timing cycle circuits. The master clock circuit produces four basic timing pulses used to establish the operation of the main timing cycle circuits. It also provides command-enable pulses through gating by the main timing chain. The master clock generates and distributes four basic timing pulses for each 500 nanoseconds of operation. These four pulses, called phases one through four, constitute one complete clock cycle. Four clock cycles represent one computer cycle. Figure 2-10 shows the relationship between the master clock phase pulse outputs and the outputs from the main timing chain flip-flops.

2-33. The main timing cycle circuits supply enables to the computer logic for operation of command timing sequences. Timing pulses developed by command timing sequences are numerically identified by the clock cycle and phase with which they are associated. The only exception to this rule is T52, used only during complement A and complement AL instructions. In the term T34, the 3 represents the third clock cycle and the 4 indicates that phase four is the last usable phase occurring during that timing pulse. Phase pulses are generally not used during the first half (shaded area, figure 2-10) of the timing pulses because of the instability of the pulse during this time. The second half of the pulse is used to ensure the proper voltage level and stability of the signal when used for gating purposes. Each phase pulse duration time is approximately 125 nanoseconds; the clock cycle is 500 nanoseconds; and the computer memory cycle is 2 microseconds.

2-34. Program Translation and Control. Each of the program instructions contains a coded command for specific computer sequencing. These commands must be decoded to perform the specific function indicated by the instruction. The program translation and control circuitry, consisting of five logically definable areas, interprets the coded command and establishes the sequence of events to be performed by supplying the command enables for the operation. The five areas of the circuitry are the following:

1. F-Register - A seven-bit flip-flop register which stores the function code during execution of an instruction.

2. Function Code Translator - Decodes the function code contained in the F-register and provides enable signals to initiate the necessary computer operation. Format I instructions (refer to paragraph 2-56) are identified by a two-digit octal code.

3. Format II Translator - Supplies the necessary enables to alter the normal sequence of events to perform specific instructions. Format II instructions (refer to paragraph 2-56) are identified by a four-digit octal code and are indicated by the FII light on the F-register.

4. Sequencer - Influences the main timing sequence to ensure the proper execution of each instruction. To perform this function, the sequencer selects individual major command sequences in the proper order to execute the instruction. By selectively gating the main timing chain enables, only the operations of the selected command sequence are performed. Command sequences include:

a. The major command sequences

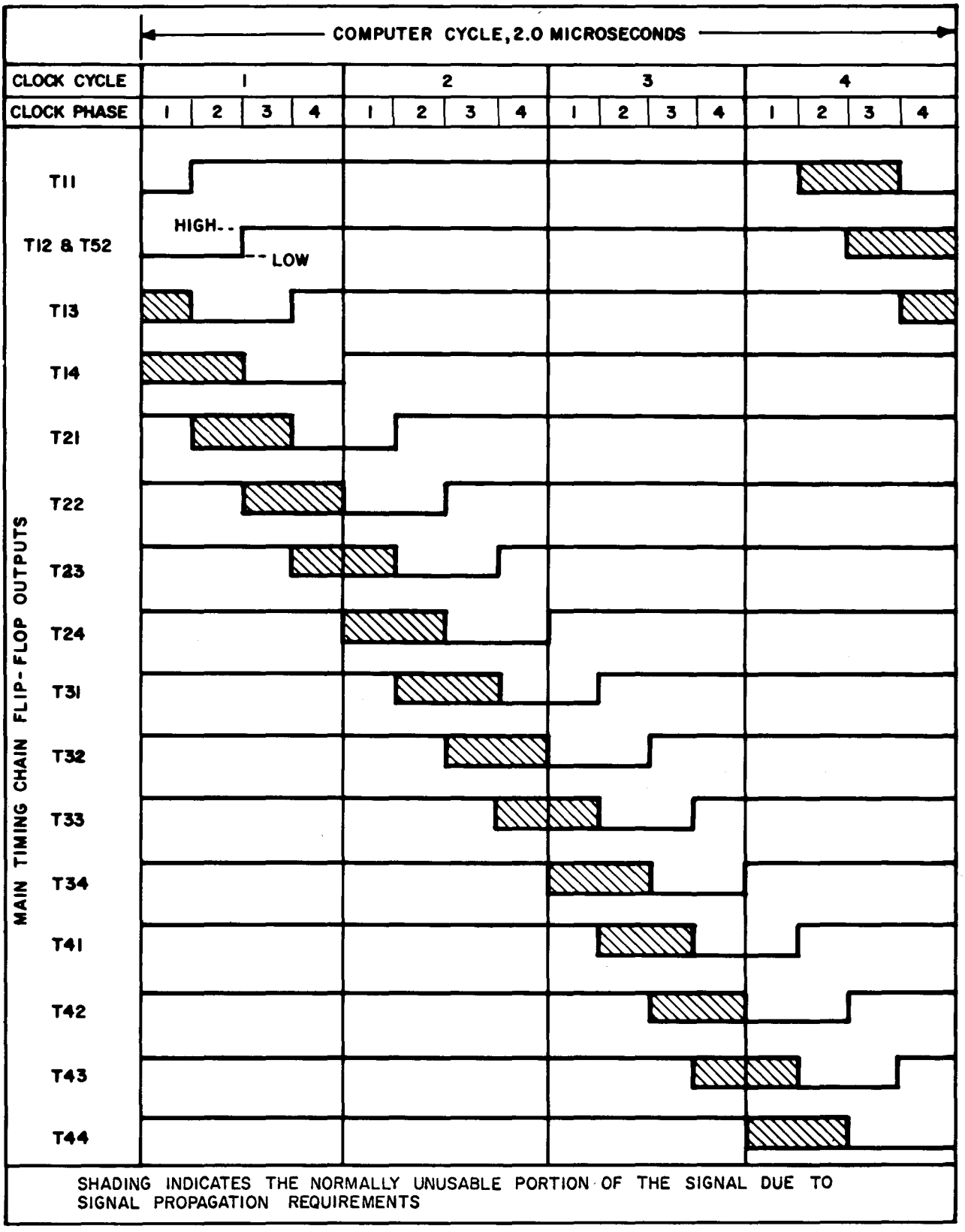


Figure 2-10. Main Timing Cycle Outputs

available for selection by the sequencer are listed in table 2-8. As indicated, they are divided into three general categories; control, input/output, and wait. Not all major command sequences are required to perform an instruction. It is the function of the sequencer to select those which will fulfill the requirements of the instruction. In several instructions, only one sequence is required; others require several sequence combinations.

b. In addition to the major command sequences, two minor command sequences are initiated by the main timing chain under the influence of the appropriate major sequence. These sequences are ADVANCE P sequence and MULTIPLY/DIVIDE/SHIFT sequence and they operate independently from the major sequence enables.

5. Command Enable Circuitry - Provides the required enabling signals for transfer, shifting, scaling, and storage of words, instructions, or operands, in or between the several registers within the computer functional logic. Command enables are generated by sequencer outputs or by individual instructions themselves, depending upon the operation to be performed.

2-35. Registers. The control section of the computer contains several registers used for the performance of control functions. These are the index registers, index control register, special register, program address register, two memory address registers, and K-register. Each of the following registers has a specific logic function to perform during the manipulation of instructions and commands.

1. The index registers are eight memory locations, the contents of which may be used to modify odd-numbered instructions from 03 to 47. Only one of the eight registers may be used during a given operation, and selection of the active index register is determined by the contents of the Index Control Register (ICR).

2. The three-bit ICR contains the programmed address of the currently active index register to be used for instruction modification. Any one of the eight index registers may be selected by the numerical value entered into the ICR through program control.

3. The five-bit Special Register (SR), when active, is used to supply the four most significant bits of memory address in the S-register. The maximum bit content of a programmed address is 12 bits. The memory address consists of 16 bits. The additional four bits will be supplied by either the SR or program address register. When SR is active (bit 3 from the K-register set) it will supply the necessary four bits, otherwise they will be supplied by the program address register.

4. The program address (P) register is used to store the address of the instruction currently being entered for execution. The contents of the P-register are incremented by one when an instruction is transferred from its memory location and thus it normally contains the next sequential address to be entered. The bits of the address supplied by P-register to the memory address register during those instructions when the SR register is not active are the four most significant bits of the P-register.

5. The memory address (S) registers receive the address of a memory location at the start of a memory cycle and retain this address throughout read/write cycle to control the translation circuitry. The S-registers may receive the address from the I/O section (which generates certain assigned addresses), from control or arithmetic section, or from an I/O channel connected to a peripheral device capable of specifying an address. The S1 register is a 16-bit register associated with main memory. The 8-bit S_0 register is associated with control memory.

TABLE 2-8. MAJOR COMMAND SEQUENCES

Type	Sequence	Function
Control	I	Reads instruction word from memory.
	R1	Reads normal 18-bit operands from memory and performs data manipulations as required.
	R2	Reads second 18 bits of a 36-bit operand and performs data manipulations as required.
	W	Writes an 18-bit word in memory and performs data manipulations as required.
Input/ Output	Interrupt	Reads contents of interrupt entrance registers from memory.
	B1	Reads terminal address control word and stores it in appropriate control memory address.
	B2	Reads initial address control word and stores it in appropriate control memory address.
	I/O 1	Reads from or writes into memory, either an 18-bit word or first 18 bits of a 36-bit word.
	I/O 2	Reads from or writes into memory, second 18 bits of a 36-bit word.
Wait	Wait	Inhibits performance of all control sequences but permits input/output operations to continue until an interrupt occurs.

6. The K-register is basically a six-bit double-rank counter used for both storage and control. Its primary use is to store and record shift and scale factor counts during execution of multiply, divide, shift, and scale factor instructions. It is also used to store selective skip and stop instructions. During operation, a transfer of bits from the upper rank to the lower rank will decrement the contents by one. The return transfer from lower rank to upper rank prepares the register for the next sequence.

2-36. Special Circuits. Included in the control section are several circuits which perform unique functions in operation of the computer. These special circuits are:

1. The Parity Circuits complete operation of the parity check, which is originated by the computer arithmetic section. The number of ones in the X-register is counted in a pair-compare scheme to insure an odd parity content. If the resulting signal indicates an even parity, parity flip-flop is set to notify associated computer logic of the parity status.

2. The B-Register and the ±1 Network are used to increment or decrement contents of a memory address during input and output operations and to update currently active index register during either B skip or B jump instructions. The B-register normally contains the value of the active index register.

3. The Compare Circuitry is used in conjunction with the various jump commands. It determines the condition present in the computer and furnishes enables for corresponding jump instructions.

4. The Overflow Circuit performs a test to determine whether an overflow condition exists during an arithmetic operation and notifies computer control logic of results.

5. Stop/Skip Logic is used to test for various conditions under which a stop or skip instruction would be performed. Used in conjunction with the STOP and SKIP switches on Control Panel 2 (A4), the stop/skip logic compares setting of these switches with the contents of the instruction to determine if this operation should be performed. Skips can also be generated with the outputs of the parity or compare circuits and the proper instructions.

2-37. ARITHMETIC SECTION. The arithmetic section of the computer performs arithmetic and logic functions to present a solution to a given problem. Arithmetic functions include addition, subtraction, multiplication, division, shifting, and scaling. Logic functions include masking, selective substitution, comparisons, complements, and word transfers between components of the arithmetic section or between the arithmetic section and memory. Control of the arithmetic section is maintained by command enable circuits of the control section. The arithmetic section is divided into seven basic subsections as shown in figure 2-9. These subsections are outlined as follows:

1. The Arithmetic Selector acts as a distribution point for data entering and leaving the arithmetic section (with the exception of data concerning P, S, and Z_1 registers). The selector receives data from the control section and arithmetic register. It sends data to X and D registers, control (parity), and memory (store selector) sections.

2. The adder is an 18-stage, subtractive, one's complement, end-around-borrow type adder used to perform arithmetic operations. It effectively subtracts the complemented contents of the D-register from the contents of the X-register.

3. The X, D, AU, AL, and W registers are 18-bit flip-flop registers. They are con-

trolled by command enables from the control section. The X-register is one of the input registers of the adder. It is also used with the upper arithmetic (AU) register for left and right shifting. The D-register is the other input register of the adder. The AU register is the upper half of the A-register and receives data from the adder. It is also used for shifting. The AL register is the lower half of the A-register and receives data from the adder. It is also used for left and right shifting with the W-register. The only function of the W-register is shifting. The parts of the A-register are connected so that left or right shifting can be accomplished on a 36-bit word.

2-38. MEMORY SECTION. The computer is designed for up to 65,536 18-bit words of addressable memory. Information concerning particular configurations is contained in Section 2-3, Detailed Functional Description. Computer storage is divided into main memory, control memory, and a nondestructive read-out bootstrap memory. Storage location for each word in all three memories is assigned a discrete address.

2-39. Main Memory. Main or core memory provides storage for programs, constants, and data. Main memory maximum capacity is 65,536 18-bit words. The number of usable addresses is 160 or 288 less than the number of actual storage locations. Control and Bootstrap memory addresses overlap with main memory addresses in stack 0. Main memory is a random-access coincident-current bit-oriented core memory. Cycle time is 2 microseconds and maximum access time is 750 nanoseconds. As illustrated in figure 2-9, main memory is divided into seven logically definable areas: main memory control, address translators, main memory (memory cores and read/write circuits), inhibit circuits, sense amplifiers, Z_m-register, and Z_1-register. They are listed as follows:

1. Main Memory Control provides timing and enable signals to regulate memory circuit operation. When a signal initiates a memory cycle, control circuits provide signals to perform a read cycle and a write cycle.

2. The Address Translators decode the memory address stored in S1-register and enable the appropriate X-line and Y-line to select the decoded address. Memory control signals pulse selected X and Y lines through the address translator to read and write at the selected address.

3. Main Memory (Memory Cores + R/W Circuits) provides actual storage for the program and data words. Cores are located on memory stacks. Each stack consists of 18 planes with 4096 cores per plane. Each plane stores a single bit of an 18-bit word. The 18 planes together form the stack and store 4096 18-bit words. One of 64 X-lines and one of 64 Y-lines are selected by address translators to select one of 4096 addresses on a stack. Address translators also select one of the stacks in the memory. R/W circuits provide regulated current for reading or writing into the main memory cores.

4. The Inhibit Circuits are used to inhibit writing a one at the bit positions that are to store a zero. At write time, X and Y lines are pulsed to write a one at all 18-bit positions of the selected address. If a zero is to be stored at any bit position in this word, an inhibit pulse must be generated for those positions.

5. The Sense Amplifiers monitor 18 output lines from each stack (one line for each bit of a word). When cores for a single word are pulsed at read time, the sense amplifiers detect the one (core switches) and zero (core does not switch) outputs and store

them in the Z_m-register.

6. The Z_m-Register provides necessary storage for the information read from main memory. Each of the four banks containing 8K-words in main memory has one Z_m-register; however, only one is used at a time. Thus, the Z_m-register is referenced singularly.

7. The Z_1-Register is used as a buffer for words being transferred between main memory and other sections of the computer.

2-40. Control Memory. Control memory (figure 2-9) is a word-oriented core memory consisting of 128 18-bit word storage locations. Access time of the control memory is a maximum of 300 nanoseconds; cycle time is 500 nanoseconds. Address assignments for the control memory are listed in table 2-9. The entire control memory circuit provides fast access to control words and to modifying index registers which allows this section to operate independently of the other two memory sections of the computer. Thus, because information is more readily available for input and output operations, operand and address modification, and real-time clock functions, the speed of the main computer operations is increased. Control memory employs the following circuitry:

1. S_0-Register. The S_0-register is the address register for control memory. Reference to an address in control memory is entered in the S_0-register for translation. Any location in control memory may be addressed through program control and the constants may be entered in the desired registers. S-register slaves are gates for the application of proper enables to succeeding circuits.

2. Word Translators. Word translators decode contents of the S-register slaves to locate selected memory address.

3. Digit Driver and R/W Diverters. Digit drivers perform the same function as inhibit circuits in the main memory. R/W diverters provide necessary drive currents for reading from or writing into the control memory cores.

4. Sense Amplifiers. Sense amplifiers detect bit-content of the addressed memory location and amplify detected signals to a level usable by the associated logic circuits.

5. Z_0-Register. The Z_0-register supplies required storage for the input and output information from control memory.

2-41. Bootstrap Memory. Bootstrap memory (figure 2-9) is a word-oriented non-destructive readout-type of memory with a maximum access time of 300 nanoseconds and a cycle time of 2 microseconds. This memory is a 32-word prewired program that loads a program of instructions via a peripheral device such as a paper tape reader or a magnetic tape transport. Some systems may automatically reload the program if the computer senses an invalid operation. Bootstrap memory utilizes logic circuitry assigned to control memory for the performance of its functions. Memory addresses are from 00500 through 00537.

2-42. BOOTSTRAP MODE switch (located on the front panel of drawer A4) provides a means of storing a second bootstrap program using main memory storage. In NDRO position, the switch allows the computer to reference nondestructive readout bootstrap memory where, for example, the paper tape load routine is stored. In MAIN MEMORY position, the computer references main memory addresses 00500 through 00537 where the magnetic tape load routine may be stored. The referenced main memory addresses are not the NDRO type, and information stored at these addresses may be changed or altered while the switch is in MAIN MEMORY position.

TABLE 2-9. MEMORY ADDRESS ASSIGNMENTS

Address	Memory	Assignment	
000000	Control	Fault Interrupt Entrance Register	
000001 thru 000010	Control	Index Registers (8)	
000011	Control	Intercomputer Time-Out Register	
000012	Control	Real-Time Clock Interrupt Register	
000013	Control	Clock Over Flow Interrupt Register	
000014	Control	Real-Time Clock Monitor Word Register	
000015	Control	Real-Time Clock Incrementing Register	
000016	Control	Synchronizing Interrupt Register	
000017	Control	Scale Factor Shift Count	
000020 thru 000037	Control	Continuous Data Mode (CDM) and External Function Buffer Control Registers	
000040 thru 000057	Control	Output Buffer Control Registers	
000060 thru 000077	Control	Input Buffer Control Registers	I/O Channels 0 through 7
000100 thru 000117	Main	External Interrupt Registers	
000120 thru 000137	Main	External Function Monitor Interrupt Registers	
000140 thru 000157	Main	Output Monitor Interrupt Registers	
000160 thru 000177	Main	Input Monitor Interrupt Registers	

TABLE 2-9. MEMORY ADDRESS ASSIGNMENTS (Cont)

Address	Memory	Assignment	
000200 thru 000217	Control	Unassigned	I/O Channels 0 through 17
000220 thru 000237	Control	Continuous Data Mode (CDM) and External Function Buffer Control Registers	
000240 thru 000257	Control	Output Buffer Control Registers	
000260 thru 000277	Control	Input Buffer Control Registers	
000300 thru 000317	Main	External Interrupt Registers	
000320 thru 000337	Main	External Function Monitor Interrupt Registers	
000340 thru 000357	Main	Output Monitor Interrupt Registers	
000360 thru 000377	Main	Input Monitor Interrupt Registers	
000400 thru 000477	* Main or Control	Unassigned	
000500 thru 000537	** Bootstrap or Main	Bootstrap: Nondestructive Readout Memory (Initial Input Routine) Main: Unassigned	

TABLE 2-9. MEMORY ADDRESS ASSIGNMENTS (Cont)

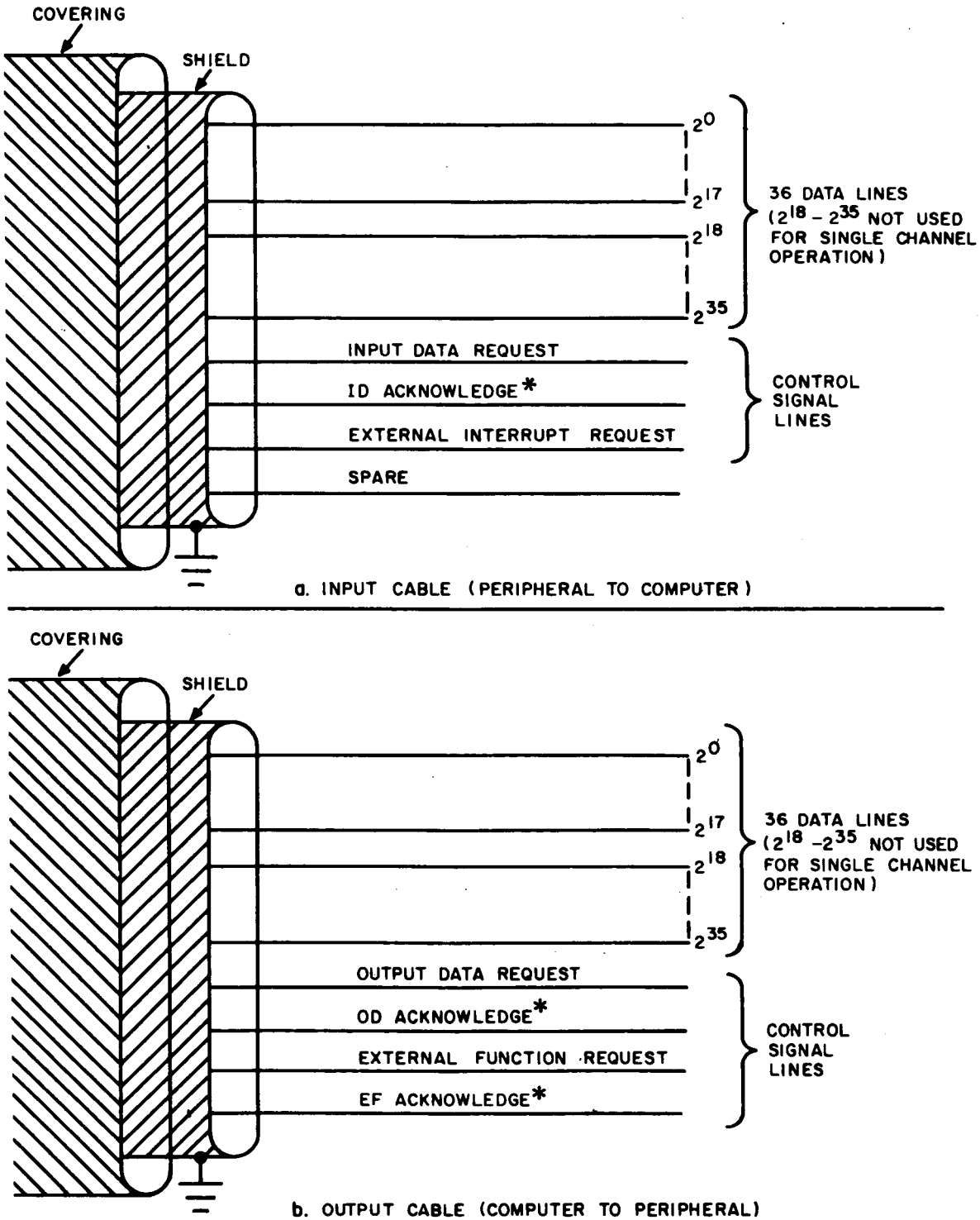
Address	Memory	Assignment
000540 thru 000577	Main	Unassigned
000600 thru 000677	* Main or Control	Unassigned
000700 thru 177777	Main	Unassigned

NOTES: I/O Terminal Address Control Words (TACW) are stored at even-numbered control memory address locations; Initial/Current Address Control Words (IACW, CACW) are stored at odd-numbered control memory address locations; Interrupt Entrance Registers are the even-numbered main memory interrupt registers; external interrupt status words are stored at the odd-numbered External Interrupt Register locations.

* These addresses are in control memory when 256 words of control memory are used for expanded ESI mode of operation.

** These addresses are in bootstrap memory when BOOTSTRAP MODE switch is positioned to NDRO and are in main memory when switch is positioned to MAIN MEMORY.

2-43. I/O SECTION. Digital Computer Mk 152 Series is able to communicate with a maximum of 16 peripheral devices. Each communication path is referred to as a channel. The channels are numbered $0\text{-}17_8$. Each data word transfer involves a maximum of 18 bits or 36 bits in parallel for single or dual channel. Each channel has an associated input and output cable. Refer to figure 2-11 and table 2-10 for cable configuration and I/O control signal data. The logic circuitry for the 16 channels is distributed among four I/O chassis with each chassis controlling four channels. The number of channels may be an multiple of four up to sixteen. Refer to tab 2-11 for the channel/chassis assignments. Each group of four channels is available in either -3 volts (fast) or -15 volt (slow) interface.



NOTES: EACH LINE IS TWISTED WITH A GROUND RETURN LINE (NOT SHOWN).
*ACKNOWLEDGE SIGNALS ARE SENT FROM COMPUTER TO PERIPHERAL. ALL OTHER SIGNALS ARE SENT TO COMPUTER FROM PERIPHERAL.

Figure 2-11. Typical I/O Cable Configurations

TABLE 2-10. DESCRIPTION OF INPUT/OUTPUT CONTROL SIGNALS

Channel	Signal Name	Origin	Meaning
Input Channel	Input Request (IR)	Peripheral equipment	I have a data word on the input lines ready for you to accept.
	Input Acknowledge (IA)	Computer	I have sampled the word on the input lines.
	External Interrupt (EI) (Request)	Peripheral Equipment	I have an interrupt code word on the input lines ready for you to accept
Output Channel	Output Request (OR)	Peripheral equipment	I am in a condition to accept a word of data from you.
	Output Acknowledge (OA)	Computer	I have put a data word for you on the output lines; sample them now.
	External Function (EF) Acknowledge	Computer	I have put an external function message for you on the output lines; sample them now.
	External Function Request (EFR)	Peripheral equipment	I am in a condition to accept an external function message from you.

TABLE 2-11. I/O CHASSIS CHANNEL ASSIGNMENTS

Chassis	Channels
2	0, 2, 4, 6
1	1, 3, 5, 7
10	10, 12, 14, 16
9	11, 13, 15, 17

2-44. Major I/O Operations. The major operations of the I/O section consist of three important communicative functions: data transfers, interrupts, and special operations. During data transfer, one I/O operation transfers a single word between the computer and a peripheral device. A complete data transfer operation may consist of a number of I/O operations, which transfer a number of data words. Data transfer between the computer and two or more devices is accomplished in accordance with established priorities. Individual I/O operations transferring words to the different devices are interlaced, or multiplexed. Interrupts provide a means to intervene in the main program thus giving the computer real-time, fault detection, and correction capabilities. Special operations allow programmed alteration of I/O logic for special program considerations such as checkout of interrupts. I/O operations are accomplished as follows:

2-45. Data Transfers. Two classes of operation are available for data transfers: normal or intercomputer. Any channel can be operated either as a normal channel (capable of communicating with peripheral equipment) or as an intercomputer channel (capable of communicating with another computer).

2-46. I/O Communication Modes . Four different communication modes are available: single channel, dual channel, Externally Specified Index (ESI), and Externally Specified Address (ESA). Each even-odd pair of I/O channels may operate in any one of the different modes. The mode of operation is selected by four four-position switches located on the I/O drawer front panel. The dual channel, Esi, and ESA modes require use of an odd channel and the next lower even channel.

1. Single Channel - In the single channel mode of operation, communication is by means of 18-bit words. The main memory locations to be used for data transfer are internally selected by buffer control words stored in control memory and selected automatically for the active channel.

2. Dual Channel - In dual channel mode of operation, an even-odd channel pair is combined to allow communication with 36-bit data words. Data transfer is by 36 parallel data lines contained in one cable. The 36-data-bit input and output cables must be connected to the odd channel of the channel pair. No external I/O cables can be connected to the even channel while in the dual channel mode. Control on dual channel is maintained by odd channel control lines. Separate parallel words of 18 bits or less are processed sequentially by the I/O section, and all data transfers are controlled in the same manner as for single mode.

3. ESI Mode - In ESI mode of operation an even-odd channel pair is combined with all 36 data lines in one cable. However, only 18-bit data words are transferred. The main memory locations to be used for data transfer are specified by an I/O buffer control word (index). The location of index words is limited to control memory addresses. For ESI input operations, the 18 bits on the even-numbered channel lines are stored at the address specifed by the index word. Address of the index is specified by 18 bits on the odd-numbered channel lines. For ESI output operation, the computer performs a normal output buffer with redundant 18-bit words being transferred.

4. ESA Mode - In ESA mode of operation, an even-odd channel pair is combined with

all 36 data lines in one cable. However, only 18-bit data words are transferred. For ESA input operations, 18 bits on the even-numbered channel lines are stored at the address specified by the 18 bits on the odd-number channel lines. For ESA output operations, the computer performs a normal output buffer with redundant 18-bit words being transferred.

2-47. Input/Output Signals. Input/Output signals consist of data-request signals, real-time clock-request signals, and interrupt signals.

2-48. Data Request Signals. These signals are sent from a peripheral device to the computer and cause the transfer of one data word. The request-honoring operation requires the use of memory to output or input a word. Therefore, it disables the instruction sequences to stop the program during its 2-microsecond use of memory.

1. Input Data Request (IDR). This signal is sent to the computer with a data word. The IDR requests the computer to accept the word.

2. Output Data Request (ODR). This signal requests the computer to output a data word to the peripheral device.

3. External Function Request (EFR). This signal requests the computer to output an external function word to the peripheral device. Usually, the external function word is used to control the peripheral device according to its bit configuration. The EFR operation could also be employed as a technique for outputting data to the peripheral device.

4. External Interrupt Request (EIR). This signal is sent to the computer possibly with a status word. Usually, the status word is used to inform the computer regarding the condition of the peripheral device as indicated by its bit configuration. For example, it may indicate timing errors, data word content errors, data ready condition, etc. The EIR requests the computer to accept the status word and inform the program of its occurrence by means of a program jump using the External Interrupt Monitor (EIM) signal.

2-49. Real Time Clock (RTC) Request Signals. The RTC Request is generated by an accurate internal oscillator operating at a frequency of 1024 pps. This oscillator is used to maintain a constantly updated 18-bit clock count in the control memory address 00015_8. The RTC request causes the computer to increment by +1 the content of this address. This clock count can be preset to any value by the program and periodically referenced so as to time certain program events.

2-50. Interrupt Signals. Interrupt signals cause an interruption to the program in the form of a jump. Each type of interrupt has a special jump address assigned to it. The jump address is also referred to as the interrupt entrance register. With the exception of the external sync interrupt, all interrupts are internally generated.

1. Input Data Monitor Interrupt (ID Mon). This signal can occur on any of the I/O channels. It indicates that the ID buffer set up by the program on its channel has been completed and that the program had desired the interrupt at the ID buffer termination. The ID buffer refers to the specified number of data words to be inputted on the particular channel.

2. Output Data Monitor Interrupt (OD Mon). This signal can occur on any of the I/O channels. It indicates that the OD buffer set up by the program on its channel has been completed and that the program had desired the interrupt at the buffer termination. The OD buffer refers to the specified number of output data transfers to occur on the part-

ıicular channel.

3. External Function Monitor Interrupt (EF Mon). This signal can occur on any of the I/O channels. It indicates that the EF buffer set up by the program on its channel has been ɔmpleted and that the program had desired the interrupt at the buffer termination. The EF buffer refers to the specified number of external function word transfers to occur on the particular channel.

4. External Interrupt Monitor Interrupt (EI Mon). This signal can occur on any of the I/O channels. It occurs as a result of honoring the external interrupt request on its channel. It simply indicates that the EIR was honored and its associated status word (if there is one) ıs been stored in memory.

5. RTC Overflow Interrupt. This signal is generated when the RTC request is honored and increments the clock count from 777777_8 to 000000_8. With the RTC oscillator operating t a frequency of 1024 pps, this interrupt ɔccurs every 256 seconds. It can be prevented by the RTC DISCONNECT switch.

6. RTC Monitor Interrupt. This signal is generated during the honoring of the RTC request if the clock count equals a predeter-ıined value. The program must also execute the RTC instruction to enable the interrupt when the clock reaches this count.

7. External Sync Interrupt. This is the only interrupt which is directly generated externally. It is a non-channel type and is carried by a separate I/O cable. It can be used as a timing input from an external device and to allow external control over the program.

8. Instruction Fault Interrupt. This is the only interrupt which is directly generated ɣ the program. It occurs when the program ɑttempts to execute a format 1 instruction with the function code of 00, 01, or 77_8. These are considered illegal or fault instructions.

9. Resume Fault Interrupt (Inter-Computer Time-Out Fault). This signal occurs only during output data transfers to another computer. It indicates that the receiving computer did not accept a data word within a specified period of time. It is timed by the RTC operations; therefore, the RTC request must not be disconnected.

2-51. Priorities. The input/output section of the computer handles two basic types of operation. These are the transfer of data or codes between peripheral equipment and the computer, and the processing of interrupts. The computer scans for transfer request or interrupts request during the execution of instructions or the processing of I/O operations. These scan functions can be divided into two types: data scan and interrupt scan. Each scan is responsible for gating request and selecting the order in which request will be processed. These are accomplished according to definite priority schemes. In accordance with the priority scheme, the higher-numbered channel activity is honored first. The scanning of functions for priority determination is based on the two-microsecond cycle time of the computer. During any major sequence one data scan cycle transpires during the first microsecond and one interrupt scan follows in the next microsecond unless some inhibiting condition exists.

2-52. I/O Circuitry. The following units are including within I/O section circuitry (see figure 2-9):

1. C_O- and C_E- Registers. These are two 18-bit output buffer registers used to transfer data or instruction words to external equipment. the C_O is the buffer register for four odd-numbered channels and the C_E is the buffer register for four even-numbered channels. In dual-channel (36-bit) operation, these registers are linked together. Another pair of C_O- and C_E- registers are required when 16 I/O channels

are used.

2. Input Selectors. This unit represents the gated input amplifiers. Each I/O channel has 18 selectors-one for each bit of data words. When a peripheral device sends a request to the computer, it also places a data word on the input lines. After priority has been granted to the requesting channels, the data word of that channel is gated into the computer through the input amplifiers.

3. Request Unit. This unit receives four types of request signals (external interrupt, input data, output data, and external function), from peripheral devices. The type of request is recorded and maintained until the request is honored by the computer.

4. Priority/Translation Block. Priority of the I/O requests is established according to the channel number and the type of function. Once priority is established, channel number and function are encoded into a binary value which is sent as an address to control memory. This address specifies which control words and which type of I/O operations are to be used for each channel.

5. Acknowledge. The acknowledge circuitry informs peripheral devices of the status of I/O operations by sending three types of acknowledge signals: input data, output data, and external function. The input acknowledge is sent for an external interrupt or input data transfer; output data acknowledge is sent for an output data transfer and the external function acknowledge is sent for an external function transfer.

6. Active Unit. In normal operation, the active circuitry functions as the on/off switch of the I/O section. Except for external interrupts, communication can occur between the computer and the peripheral devices only when the I/O channels are active.

2-53. INSTRUCTIONS ANALYSIS

2-54. INSTRUCTIONS REPERTOIRE. The repertoire of instructions consists of 102 single-address, flexible instructions. Each instruction contains at least one 6-bit function code that designates which operations are to be performed, and a 12-bit number which may be a memory address or a constant.

2-55. Symbols Used. The symbols used throughout the description of instructions are listed in table 2-12. The usage of each symbol is explicitly defined.

2-56. Instruction Word Formats. There are two basic word formats in use by the computer. An explanation of each is provided in table 2-13.

2-57. List of Instructions. The complete repertoire of instructions for the computer is presented in table 2-14. In the table, common usage and examples are included with instructions where the meaning may not be obvious. No attempt has been made to indicate more sophisticated use. The instructions are listed and defined in the following format:

1. (Octal code) (Instruction name) (TRIM code) (Symbolic Summary)
2. (Execution time)
3. (Definition of the y address or constant)
4. (Text defining the instruction in detail)
5. (Examples and/or notes if any)

The symbolic summary expression will use the symbol Y to include y or y+B, whichever is stated in the text for that instruction.

TABLE 2-12. SYMBOLS USED

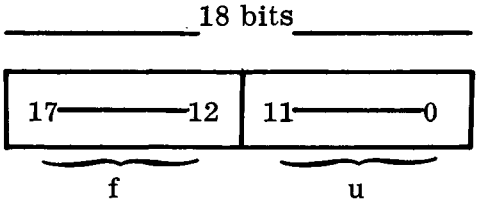
Symbol	Usage
AU	Upper accumulator, 18-bit arithmetic register.
AL	Lower accumulator, 18-bit arithmetic register.
A	AU and AL linked together to form one 36-bit arithmetic register.
B	Contents of the B active index register; 18-bit one's complement.
f	Function code, high-order 6 bits of all instruction words.
F	Function code register, 7 bits.
k	Designator contained in Format II instructions, 6 bits. May define shift count, I/O channel number, stop or skip condition, or the contents of the SR or ICR registers.
m	Minor function code contained in Format II instructions, 6 bits.
M	Memory word specified by (y), (y+B), L(y)(AU), or L(y+B)(AU) of compare instruction.
NI	Next instruction.
P	Program address register.
SR	Special register, 5-bit core memory bank designator.
u	Low-order 12 bits contained in Format I instruction words.
u_P	u prefaced with core memory bank designator bits of P.
u_{SR}	u prefaced with core memory bank designator bits of SR.
y	u extended, u_P or u_{SR}.
Y	Address of constant formed by y or y+B with or without sign extension.
( )	Contents of the address or register.
$( )_i$	Initial contents of the address or register.
$( )_f$	Final contents of the address or register.
$( )_n$	Designates any single nth bit of the contents of a register.
(Y+1, Y)	Designates the contents of two consecutive memory locations linked together to form a 36-bit word. Address Y+1 contains the most significant half of the word while address Y contains the least significant half.

TABLE 2-12. SYMBOLS USED (Cont)

Symbol	Usage
:	The colon in a logical expression indicates comparison.
L( ) ( ) or ( ) ⊙ ( )	The bit-by-bit or logical product (logical AND) defined by the table: 0 0 0 1 0 1
( ) v ( )	Logical sum, or inclusive OR defined by the table: 0 0 1 1 1 1
( ) ⊕ ( )	Half add, half subtract, or exclusive OR defined by the table: 0 0 1 1 1 0
( )' or $\overline{(\ )}$	The one's complement of the contents of the address or register.
( ) + ( )	Alegbraic sum of the contents of two locations.
( ) • ( )	Algebraic product of the contents of two locations.
(Y)	When the contents of Y are used as an address, only the lower portion of the word that can be contained in S is transferred.
⟶	Transfer the quantity stated at the left of the symbol to the address or register stated at the right of the symbol.
xY	x preceding some symbol indicates that the sign of the 12-bit constant has been extended to produce an 18-bit word, for example, xY = $u_{17} \ldots u_{12}$ ($u_{17} \ldots u_{12}$ = 6 bits) $u_{11} \ldots u_0$ ($u_{11} \ldots u_0$ = 12 bits) (all the same as u_{11})

TABLE 2-13. INSTRUCTION WORD FORMAT

(1) FORMAT I.



f : function code, six high-order bits

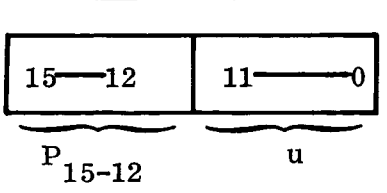
u : 12 low-order bits

The definition and usage of a u is determined by the function code utilizing u in two distinct manners:

u USED AS A CONSTANT. For this case, u itself is the operand and requires no further memory reference; however, u is extended to 18 bits. (See list of instructions.)

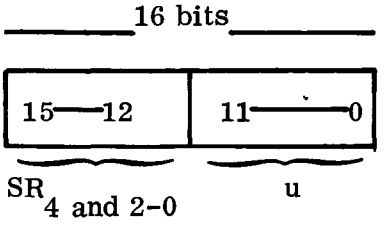
u USED AS AN ADDRESS. For this case, u is used as the lower-order 12 bits of the base address referring to a memory location. The base address is 16 bits, designated as u_P or u_{SR}, and is described below.

u_P:



u_P is defined as a 16-bit address; the four high-order bits consist of the four high-order bits of P and the 12 low-order bits are u.

u_{SR}:

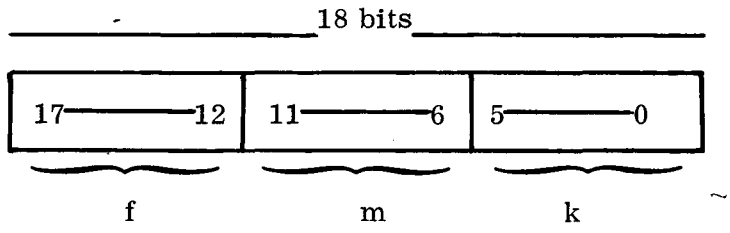


u_{SR} is defined as a 16-bit address; the four high-order bits consist of the three low-order bits of SR and the highest-order bit of SR, and the 12 lower-order bits are u. The peculiar arrangement of bits in the SR register, bit 2^3 being the active bit, was done to insure program compatibility with the 1218 computer.

TABLE 2-13. INSTRUCTION WORD FORMAT (Cont)

Certain Format I instructions allow the use of either u_P or u_{SR} as the operand address; for these instructions, u_{SR} is used if SR is active and u_P is used whenever SR is inactive. See list of instructions.

(2) FORMAT II.



f : six-bit function code (always equal to octal 50)

m : six-bit minor function code

k : six low-order bits (channel designator)

TABLE 2-14. LIST OF INSTRUCTIONS

(1) FORMAT I INSTRUCTIONS.

00
01 ILLEGAL CODE - Jump to fault entrance register, address 0 or address 500 (depending upon position of AUTO RECOVERY switch).

Execution time: 2 microseconds

02 COMPARE AL (CMAL) (AL) : (Y)

Execution time: 4 microseconds

$Y = u_P$ or u_{SR}

Compare algebraically (AL) with (y) and set the comparison designator as follows:

1) Set the compare stage
2) Set the greater stage if (AL) > (Y)
3) Set the equals stage if (AL) = (Y)

$(AL)_f = (AL)_i$

NOTE: The comparison designator is cleared by the execution of any subsequent instruction other than codes 60-67 or 50:60-67, and no interrupt is honored while the designator is set.

03 COMPARE AL (CMALB) (AL) : (Y)

Execution time: 4 microseconds

$Y = u_P$ or u_{SR}

Compare algebraically (AL) with (y+B) and set the comparison designator as follows:

1) Set the compare stage
2) Set the greater stage if (AL) > (y+B)
3) Set the equals stage if (AL) = (y+B)

$(AL)_f = (AL)_i$

NOTE: The comparison designator is cleared by the execution of any subsequent instruction other than codes 60-67 or 50:60-67, and no interrupt is honored while the designator is set.

04 SELECTIVE SUBSTITUTE (SLSU) $L(AU)'(AL)+L(AU)(Y) \rightarrow AL$ or $(Y)_n \rightarrow AL_n$ for $(AU)_n = 1$

Execution time: 4 microseconds

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

$y = u_P$ or u_{SR}

Replace the individual bits of (AL) with bits of (y) corresponding to ones in (AU), leaving the remaining bits of (AL) unaltered.

$(AU)_f = (AU)_i$

Example of selective substitute:

$(AU)_i$ = 007777 - Mask

(y) = 123451

$(AL)_i$ = 666666

$(AL)_f$ = 663451

05 SELECTIVE SUBSTITUTE (SLSUB) L(AU)'(AL). + L(AU)(Y) → AL

Execution time: 4 microseconds

$y = u_P$ or u_{SR}

Replace the individual bits of (AL) with bits of (y+B) corresponding to ones in (AU), leaving the remaining bits of (AL) unaltered.

$(AU)_f = (AU)_i$

06 COMPARE WITH MASK (CMSK) L(AU)(AL):L(AU)(Y)

Execution time: 4 microseconds

$y = u_P$ or u_{SR}

Algebraically compare selected bits of (AL) with corresponding bits of (y) and set comparison designator as follows:

1) Set the compare stage
2) Set the greater stage if L(AL)(AU) > L(y)(AU)
3) Set the equals stage if L(AL)(AU) = L(y)(AU)

$(AL)_f = (AL)_i$: $(AU)_f = (AU)_i$

NOTE: The comparison designator is cleared by the execution of any subsequent instruction other than codes 60-67 or 50:60-67, and no interrupt is honored while the designator is set.

Example of compare with mask:

$(AU)_i$ = 007777 - Mask

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

(y) = 123451

$(AL)_i$ = 222351

Compare 2351 with 3451

$(AU)_f$ = 007777 : $(AL)_f$ = 222351

07 COMPARE WITH MASK (CMSKB) L(AU)(AL):(AU)(Y)

Execution time: 4 microseconds

$y = u_P$ or u_{SR}

Algebraically compare selected bits of (AL) with corresponding bits of (y+B), and set the comparison designator as follows:

1) Set the compare stage
2) Set the greater stage if L(AL)(AU) > L(y+B)(AU)
3) Set the equals stage if L(AL)(AU) = L(y+B)(AU)

$(AL)_f = (AL)_i$: $(AU)_f = (AU)_i$

NOTE: The comparison designator is cleared by the execution of any subsequent instruction other than codes 60-67 or 50:60-67, and no interrupt is honored while the designator is set.

10 ENTER AU (ENTAU) (Y) → AU

Execution time: 4 microseconds

$y = u_P$ or u_{SR}

Clear AU. Then transmit (y) to AU.

11 ENTER AU (ENTAUB) (Y) → AU

Execution time: 4 microseconds

$y = u_P$ or u_{SR}

Clear AU. Then transmit (y+B) to AU.

12 ENTER AL (ENTAL) (Y) → AL

Execution time: 4 microseconds

$y = u_P$ or u_{SR}

Clear AL. Then transmit (y) to AL.

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

13	ENTER AL (ENTALB)	(Y) → AL
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR}	
	Clear AL. Then transmit (y+B) to AL.	
14	ADD AL (ADDAL)	(AL)+(Y) → AL
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR}	
	Add (y) to (AL) and leave the result in AL. Set overflow designator if overflow occurs.*	
	$(AL)_f$ are all ones if $(AL)_i$ and (y) are all ones.	
15	ADD AL (ADDALB)	(AL)+(Y) → AL
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR}	
	Add (y+B) to (AL) and leave the result in AL. Set overflow designator if overflow occurs.*	
	$(AL)_f$ are all ones if $(AL)_i$ and (y+B) are all ones.	
16	SUBTRACT AL (SUBAL)	(AL)-(Y) → AL
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR}	
	Subtract (y) from (AL) and leave the difference in AL. Set overflow designator if overflow occurs.*	
	$(AL)_f$ are all ones if $(AL)_i$ are all ones and (y) are all zeros.	
17	SUBTRACT AL (SUBALB)	(AL)-(Y) → AL
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR}	
	Subtract (y+B) from (AL) and leave the difference in AL. Set overflow designator if overflow occurs.*	

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

$(AL)_f$ are all ones if $(AL)_i$ are all ones and (y+B) are all zeros.

20 ADD A (ADDA) (A)+(Y+1, Y) → A

Execution time: 6 microseconds

$y = u_P$ or u_{SR}

Add to (A) the double length (36-bit) number contained in storage cells y+1 and y, and leave the result in A. Set overflow designator if overflow occurs.* The least-significant half is in cell y and the most significant half in y+1. The sign of the double length number is indicated by the most-significant bit of (y+1). Address y must be even; for example, the right-most octal digit must be 0, 2, 4, or 6.

NOTE: The instruction is executed in the following manner: The AU and AL registers are linked to form a continuous 36-bit A register. Any borrow required by AL comes from AU; any end-around-borrow required by AU is blocked and recorded in the borrow designator, leaving A uncorrected. The skip-on-no-borrow instruction (code 50:51) is used to test for required correction. Only add A or subtract A instructions set the designator.

Example of a double add with y = 07506:

$(A)_i$ = 201007430145

Address 07506 = 351123 (least significant half)

Address 07507 = 077430 (most significant half)

$(A)_f$ = 300440001271 - The result may be incorrect since the addition of some numbers results in an end-around-borrow. Since it is blocked, the result will be 1 larger than it should be (as in the example).

21 ADD A (ADDAB) (A)+(Y+1, Y) → A

Execution time: 6 microseconds

$y = u_P$ or u_{SR}

Add to (A) the double length (36-bit) number contained in storage cells y+B+1 and y+B, leaving the result in A. Set overflow designator if overflow occurs.* The least-significant half is in cell y+B and the most-significant half in cell y+B+1. The sign of the double length number is the sign of (y+B+1). Address y+B must be even. (See note, instruction 20.)

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

22 SUBTRACT A (SUBA) (A)-(Y+1, Y)→A

Execution time: 6 microseconds

$y = u_P$ or u_{SR}

Subtract from (A) the double length (36-bit) number contained in storage cells y+1 and y, and leave the difference in A. Set overflow designator if overflow occurs.* The least-significant half is in cell y and the most significant half in cell y+1. The sign of the double length number is the sign of (y+1). Address y must be even. The computer executes subtract A in a manner analogous to the add A instruction. (See note, instruction 20.)

23 SUBTRACT A (SUBAB) (A)-(Y+1, Y) → A

Execution time: 6 microseconds

$y = u_P$ or u_{SR}

Subtract from (A) the double length number contained in storage cells y+B+1 and y+B, and leave the difference in A. Set overflow designator if overflow occurs.* The least-significant half is in cell y+B and the most-significant half in cell y+B+1. The sign of the double length number is the sign of (y+B+1). Address y+B must be even. The computer executes subtract A in a manner analogous to the add A instruction. (See note, instruction 20.)

24 MULTIPLY AL (MULAL) (AL) · (Y) → A

Execution time: 14 microseconds

$y = u_P$ or u_{SR}

Multiply (AL) by (y) leaving the double length product in A. If the factors are considered integers, the product is an integer in A. The multiplication process is executed on the absolute value of the factors, then corrected for algebraic sign.

25 MULTIPLY AL (MULALB) (AL) · (Y) → A

Execution time: 14 microseconds

$y = u_P$ or u_{SR}

Multiply (AL) by (y+B) leaving the double length product in A. If the factors are considered integers, the product is an integer in A. The multiplication process is executed on the absolute value of the factors,

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

then correct for algebraic sign.

26 DIVIDE A (DIVA) (A) ÷ (Y); Quot → AL, Rem → AU

Execution time: 14 microseconds

$y = u_P$ or u_{SR}

Divide (A) by (y) leaving the quotient in AL and the remainder in AU. The remainder always bears the sign of the dividend A_i with the results satisfying the relationship:

dividend = quotient x divisor + remainder

Set overflow designator if overflow occurs.* Examples of the four possible sign combinations of the dividend/divisor and the results:

Dividend	Divisor	Quotient	Remainder
+5	+4	+1	+1
+5	-4	-1	+1
-5	+4	-1	-1
-5	-4	+1	-1

27 DIVIDE A (DIVAB) (A) ÷ (Y); Quot → AL, Rem → AU

Execution time: 14 microseconds

$y = u_P$ or u_{SR}

Divide (A) by (y+B), leaving the quotient in AL and the remainder in AU. The remainder bears the sign of the dividend A_i. (See instruction 26.)

30 INDIRECT RETURN JUMP (IRJP) (P)+1 → (Y); (Y)+1 → P

Execution time: 6 microseconds

Instruction executed from running program: $y = u_P$

Store (P)+1 at the address which is the low-order bits of (y), then increment that address by one (1) and enter it into the program address register.

Instruction executed from entrance register on interrupt: y = u
Store (P) at the address which is the low-order bits of (y), then increment that address by one (1) and enter it into the program address register.

*The overflow designator is set if $X_{17} = 0$ and $D_{17} = 0$ and no Borrow request to bit 17, or if $X_{17} = 1$ and $D_{17} = 1$ and a Borrow request to bit 17; it is cleared only by the execution of instruction skip-on-overflow (f m = 50:52) or instruction skip-on-no-overflow (f m = 50:53).

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

Example of an indirect return jump executed from address 22000:

Address	Initial Contents	Final Contents	Explanation
22000	30 6500	Same	Execute subroutine from main program.
26500	71 7420	Same	Constant defining location of desired subroutine.
117420	37 2164	02 2001	Subroutine exit address.
117421	. . . . .	Same	Subroutine entrance address (control is transferred here from indirect return jump).

The effect of the above sequence, upon execution of the indirect return jump at address 22000, is to transfer control to the subroutine starting at address 117421, but at the same time letting the subroutine know where to return control.

31 INDIRECT RETURN JUMP (IRJPB) (P)+1 → (Y); (Y)+1 → P

Execution time: 6 microseconds

Instruction executed from running program: $y = u_P$

Store (P)+1 at the address which is the low-order bits of (y+B), then increment that address by one (1), and enter it into the program address register.

Instruction executed from entrance register on interrupt: y = u

Store (P) at the address which is the low-order bits of (y+B), then increment that address by one (1) and enter it into the program register.

32 ENTER B (ENTB) (Y) → B Reg

Execution time: 4 microseconds

$y = u_P$ or u_{SR}

Transmit (y) to B_{ICR}

The full 18 bits of (y) are transmitted to the B-register (a normally addressable memory location).

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

33	ENTER B (ENTBB)	(Y) → B Reg
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR}	
	Transmit (y+B) to B_{ICR}	
	The full 18 bits of (y+B) are transmitted to the B-register (a normally addressable memory location).	
34	DIRECT JUMP (JP)	Y → P; NI = (Y)
	Execution time:	2 microseconds
	$y = u_P$	
	Unconditional jump to y. (Set P = y.)	
35	DIRECT JUMP (JPB)	Y → P; NI = (Y)
	Execution time:	2 microseconds
	$y = u_P$	
	Unconditional jump to y+B.	

NOTE: Since B is an 18-bit, one's-complement number, care must be taken when using this instruction. In addition, it is possible that the address y+B may not be relative to the same core bank from which the (35) DIRECT JUMP was executed: consider a direct jump with y = 03560 and b = 010000. In this case y+B = 03560 + 010000 = 13560.

36	ENTER B WITH CONSTANT (ENTBK)	xY → B Reg
	Execution time:	2 microseconds
	y = u (sign extended to 18 bits)	
	Clear B_{ICR}, then transmit y to B_{ICR}	

NOTE: u is a 12-bit, one's-complement number contained within the instruction; it does not refer to an address. Example of enter B with constant when u = 7701:

B_i = any value

B_f = 777701

37	MODIFY B WITH CONSTANT (ENTBKB)	B_i + xY → B Reg
	Execution time:	2 microseconds

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

	y = u (sign extended to 18 bits) Add y to B (add a constant to B). Note that u is a 12-bit, one's-complement number contained within the instruction and can be used to increment or decrement B.	
40	CLEAR Y (STORE ZERO) (CL)	0 → Y
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR} Store an 18-bit word of zeros at storage address y.	
41	CLEAR Y (STORE ZERO) (CLB)	0 → Y
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR} Store an 18-bit word of zeros at storage address y+B.	
42	STORE B (STRB)	B → Y
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR} Store B at storage address y.	
43	STORE B (STRBB)	B → Y
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR} Store B at storage address y+B.	
44	STORE AL (STRAL)	(AL) → Y
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR} Store (AL) at storage address y. $(AL)_f = (AL)_i$	
45	STORE AL (STRALB)	(AL) → Y

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

	Execution time:	4 microseconds
	$y = u_P$ or u_{SR}	
	Store (AL) at storage address y+B.	
	$(AL)_f = (AL)_i$	
46	STORE AU (STRAU)	(AU) → Y
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR}	
	Store (AU) at storage address y.	
	$(AU)_f = (AU)_i$	
47	STORE AU (STRAUB)	(AU) → Y
	Execution time:	4 microseconds
	$y = u_P$ or u_{SR}	
	Store (AU) at storage address y+B.	
	$(AU)_f = (AU)_i$	
50	(See format II instructions immediately following function code 77.)	
51	SELECTIVE SET (SLSET)	(AL) v (Y) → AL or SET $(AL)_n$ for $(Y)_n = 1$
	Execution time:	4 microseconds
	$y = u_P$	
	Set the individual bits of (AL) to one corresponding to ones in (y), leaving the remaining bits of (AL) unaltered. This is a bit-by-bit inclusive OR.	
	'Example of selective set:	
	$(AL)_i = 123456$	
	(y) = 000077	
	$(AL)_f = 123477$	
52	SELECTIVE CLEAR (SLCL)	L(AL)(Y) → AL or clear $(AL)_n$ for $(Y)_n = 0$

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

Execution time: 4 microseconds

$y = u_P$

Clear the individual bits of (AL) corresponding to zeros in (y), leaving the remaining bits of (AL) unaltered. The effect of this instruction is to compute the bit-by-bit (or logical) product of (AL) and (y) leaving the result in AL.

Example of selective clear:

$(AL)_i = 123456$

$(y) = 707070$

$(AL)_f = 103050$

53 SELECTIVE COMPLEMENT (SLCP) (AL) ⊕ (Y) ⟶ AL or complement $(AL)_n$ for $(Y)_n = 1$

Execution time: 4 microseconds

$y = u_P$

Complement the individual bits of (AL) corresponding to ones in (y), leaving the remaining bits of (AL) unaltered, for example, complement $(AL)_n$ for $(y)_n = 1$. This is a bit-by-bit exclusive OR.

Example of selective complement instruction:

$(AL)_i = 123456$

$(y) = 070007$

$(AL)_f = 153451$

54 INDIRECT JUMP AND ENABLE INTERRUPTS (IJPEI) (Y) ⟶ P; enable interrupts

Execution time: 4 microseconds

$y = u_P$ Address = $(y)_{15-0}$

Remove interrupt lockout (enable interrupts). Then jump to the address which is the low-order bits of (y). An application of this instruction is the termination of a subroutine activated by an interrupt.

55 INDIRECT JUMP (IJP) (Y) ⟶ P

Execution time: 4 microseconds

$y = u_P$ Address = $(y)_{15-0}$

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

Jump to the address which is the low-order bits of (y).

56	B SKIP (BSK)	If B = (Y), skip NI If B ≠ (Y), (B)+1 → B, read NI
	Execution time:	4 microseconds

$y = u_P$

Test B and (y) for equality. Skip next instruction if equal; otherwise increment B by one and read the next instruction.

57	INDEX SKIP (ISK)	If (Y) = 0, skip NI If (Y) ≠ 0, (Y)-1 → Y, read NI
	Execution time:	6 microseconds

$y = u_P$

If (y) ≠ 0, subtract one from (y) leaving the result in y, and take the next instruction; otherwise skip the next instruction leaving (y) unaltered.

60	JUMP AU ZERO (JPAUZ)	If [$\overline{\text{compare}}$ · (AU)=0]⊕ [compare · (AL)=M⊕ L(AL)(AU)=M] : Y → P
	Execution time:	2 microseconds

$y = u_P$

Jump to y; for example, set P = y, if:

1) Compare stage of the comparison designator is not set and (AU) = 0. (Negative zero acts as not zero.)
or
2) Compare stage of the comparison designator is set and the equals stage of the comparison designator is set.

Otherwise, execute next instruction.

61	JUMP AL ZERO (JPALZ) (JPEQ)	If [$\overline{\text{compare}}$ · (AL)=0]⊕ [compare · (AL)=M⊕ L(AL)(AU)=M] : Y → P
	Execution time:	2 microseconds

$y = u_P$

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

Jump to y; for example, set P = y, if:

1) Compare stage of the comparison designator is not set and (AL) = 0. (Negative zero acts as not zero.)

2) Compare stage of the comparison designator is set, and the equals stage of the comparison designator is set.

Otherwise, execute next instruction.

62 JUMP AU NOT ZERO (JPAUNZ)

If $[\overline{\text{compare}} \cdot (AU) \neq 0] \oplus [\text{compare} \cdot (AU) \neq M \oplus L(AL)(AU) \neq M] : Y \rightarrow P$

Execution time: 2 microseconds

$y = u_P$

Jump to y; for example, set P = y, if:

1) Compare stage of comparison designator is not set and (AU) $\neq$ 0.
or
2) Compare stage of comparison designator is set and the equals stage of the comparison designator is not set.

Otherwise, execute next instruction.

63 JUMP AL NOT ZERO (JPALNZ)

(JPNOT)

If $[\overline{\text{compare}} \cdot (AL) \neq 0] \oplus [\text{compare} \cdot (AL) \neq M \oplus L(AL)(AU) \neq M] : Y \rightarrow P$

Execution time: 2 microseconds

$y = u_P$

Jump to y; for example, set P = y, if:

1) Compare stage of comparison designator is not set and (AL) $\neq$ 0.
or
2) Compare stage of comparison designator is set and the equals stage of the comparison designator is not set.

Otherwise, execute next instruction.

64 JUMP AU POSITIVE (JPAUP)

If $[\overline{\text{compare}} \cdot (AU)\ \text{Pos}] \oplus [\text{compare} \cdot (AL) \geq M \oplus L(AL)(AU) \geq M] : Y \rightarrow P$

Execution time: 2 microseconds

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

	$y = u_P$ Jump to y; for example, set P = y, if: 1) Compare stage of comparison designator is not set and (AU) $\geq$ 0. or 2) Compare stage of comparison designator is set and the greater stage of comparison designator is set. Otherwise, execute next instruction.	
65	JUMP AL POSITIVE (JPALP) (JPMLEQ)	If [$\overline{\text{compare}}$ · (AL) Pos] ⊕ [compare · (AL) $\geq$ M ⊕ L(AL)(AU) $\geq$ M]: Y → P
	Execution time:	2 microseconds
	$y = u_P$ Jump to y; for example, set P = y, if: 1) Compare stage of comparison designator is not set and (AL) $\geq$ 0. or 2) Compare stage of comparison designator is set and the greater stage of comparison designator is set. Otherwise, execute next instruction.	
66	JUMP AU NEGATIVE (JPAUNG)	If [$\overline{\text{compare}}$ · (AU) Neg] ⊕ [compare · (AL) < M ⊕ L(AL)(AU) < M]: Y → P
	Execution time:	2 microseconds
	$y = u_P$ Jump to y; for example, set P = y, if: 1) Compare stage of comparison designator is not set and (AU) < 0. or 2) Compare stage of comparison designator is set and the greater stage of comparison designator is not set. Otherwise, execute next instruction.	
67	JUMP AL NEGATIVE (JPALNG) (JPMGR)	If [$\overline{\text{compare}}$ · (AL) Neg] ⊕ [compare · (AL) < M ⊕ L(AL)(AU) < M]: Y → P

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

Execution time: 2 microseconds

$y = u_P$

Jump to y; for example, set P = y, if:

1) Compare stage of comparison designator is not set and (AL) < 0.
or
2) Compare stage of comparison designator is set and the greater stage of comparison designator is not set.

Otherwise, execute next instruction.

70 ENTER AL WITH CONSTANT (ENTALK) xY → AL

Execution time: 2 microseconds

y = u (with sign extended to 18 bits)

Clear AL. Then transmit y to AL.

Example of enter AL with constant when u = 0001

$(AL)_i$ = any value

$(AL)_f$ = 000001 (+1)

Example of enter AL with constant when u = 7776

$(AL)_i$ = any value

$(AL)_f$ = 777776 (-1)

NOTE: u is a 12-bit, one's complement number contained within the instruction; it does not refer to an address.

71 ADD CONSTANT TO AL (ADDALK) (AL)+xY → AL

Execution time: 2 microseconds

y = u (sign extended to 18 bits)

Add y to (AL) and leave the result in AL. The effect of this instruction is to increment/decrement (AL) with a constant contained within the instruction.

Example of add constant to AL when u = 0002 (+2)

$(AL)_i$ = 057777

$(AL)_f$ = 060001 (incremented)

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

Example of add constant to AL when u = 7775 (-2)

$(AL)_i$ = 067055

$(AL)_f$ = 067053 (decremented)

72 STORE INDEX CONTROL REGISTER (STRICR) (ICR) $\rightarrow Y_{5-0}$

Execution time: 4 microseconds

$y = u_P$

Replace the low-order six bits of (y) with a six-bit value consisting of the four, lower-order bits equal to the contents of the index control register and the remaining two bits equal to zero. As this instruction affects a six-bit partial transfer, the upper 12 bits of (y) remain unchanged. $(ICR)_i = (ICR)_f$

NOTE: To clear the index control register see f = 50 72.

73 B JUMP (BJP) If B $\neq$ 0, B-1 $\rightarrow$ B Reg & Y $\rightarrow$ P
If B = 0, Execute NI

Execution time: 2 microseconds

$y = u_P$

If B $\neq$ 0, subtract one from B, then jump to y; otherwise, take the next instruction leaving B unaltered (neg zero $\neq$ 0).

NOTE: As B is a one's-complement number and can take values less than zero, the B jump will be effective only for program loops where B is initially positive.

74 STORE ADDRESS (STRADR) $(AL)_{11-0} \rightarrow Y_{11-0}$

Execution time: 4 microseconds

$y = u_P$

Replace the low-order 12 bits of (y) with the low-order 12 bits of (AL). As this instruction affects a partial transfer, the higher-order six bits of (y) remain undisturbed.

$(AL)_f = (AL)_i$

Examples of a store address instruction:

$(AL)_i$ = 762504

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

$(y)_i$ = 567777

$(y)_f$ = 562504

75 STORE SPECIAL REGISTER (STRSR) $(SR) \rightarrow Y_{5-0}$

Execution time: 4 microseconds

$y = u_P$

Replace the low-order six bits of (y) with a six-bit value of which the five low-order bits are equal to (special register) with the remaining bits equal to zero; store the result at y, then clear the active bit of the special register. As this instruction affects a six-bit partial transfer, the upper 12 bits of (y) remain undisturbed.

NOTE: This instruction deactivates the special register but does not clear the other 4 bits.

76 DIRECT RETURN JUMP (RJP) $(P)+1 \rightarrow Y$; $Y+1 \rightarrow P$

Execution time: 4 microseconds

$y = u_P$

Store (P)+1 at y, then jump to y+1. This instruction transfers to y a full 18-bit word, the lower bits being the address (P)+1 with the upper bits set to zero.

When this instruction is executed from an interrupt entrance register by an interrupt, store (P). Do not initiate the (P)+1 sequence.

77 ILLEGAL CODE - Jump to fault entrance register, address 0, or address 500 (depending on position of AUTO RECOVERY switch)

Execution time: 2 microseconds

(2) FORMAT II INSTRUCTIONS. The following are Format II, type 3 instructions and require a combination of a 50 function code and a subfunction code that determines the operation to be performed. The 50 code is detected when read from memory and causes $1 \rightarrow F_6$ and $Z_{11-6} \rightarrow F_{5-0}$ for execution. The computer maintains its regular timing sequence, plus a Format II sequence.

50:00 Not used

50:01 SET INPUT ACTIVE (SIN)

Execution time: 2 microseconds

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

Set input channel k to the active state. The buffer control words stored in memory locations 60 + 2k and 61 + 2k (channels 0-7) or 260 + 2(k-10) and 261 + 2(k-10) (channels 10-17) or as specified by the externally specified index or externally specified address will control the transfers.

50:02 SET OUTPUT ACTIVE (SOUT)

Execution time: 2 microseconds

Set output channel k to the active state. The buffer control words stored in memory locations 40 + 2k and 41 + 2k (channels 0-7) or 240 + 2(k-10) and 241 + 2(k-10) (channels 10-17) or as specified by the ESI or ESA will control the transfers.

50:03 SET EXTERNAL FUNCTION ACTIVE (SEXF)

Execution time: 2 microseconds

Set channel k external function mode active. The buffer control words stored in memory locations 20 + 2k and 21 + 2k (channels 0-7) or 220 + 2(k-10) and 221 + 2(k-10) (channels 10-17) will control the transfers.

50:04 Not used

50:05 Not used

50:06 Not used

50:07 Not used

50:10 Not used

50:11 INPUT TRANSFER (IN)

Channels 0-7	Channels 10-17
(P+1)→60+2K	(P+1)→260+2(K-10)
(P+2)→61+2k	(P+2)→261+2(K-10)

Set input active on channel k.

Execution time: 6 microseconds

Initiate input transfer on channel k.

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

Transfer buffer limit address words (for input buffer) from the following two addresses to the input buffer control registers for the designated channel. (Other I/O channel and processer activity proceeds normally.)

50:12 OUTPUT TRANSFER (OUT)

Channels 0-7	Channels 10-17
(P+1)→40+2k	(P+1)→240+2(K-10)
(P+2)→41+2K	(P+2)→241+2(K-10)

Set output active on channel k.

Execution time: 6 microseconds

Initiate output transfer on channel k.

Transfer buffer limit address words (for output buffer) from the following two instruction locations to the output buffer control registers for the designated channel. (Other I/O channel and processer activity proceeds normally.)

50:13 EXTERNAL FUNCTION (EXF)

Channels 0-7	Channels 10-17
(P+1)→20+2k	(P+1)→220+2(k-10)
(P+2)→21+2k	(P+2)→221+2(k-10)

Set external function active on channel k.

Execution time: 6 microseconds

Initiate external function transfer on channel k.

Transfer buffer limit addresses (for the function words to be used) from the following two instruction locations to the external function buffer control registers for the designated channel.

50:14 ENABLE REAL-TIME CLOCK (RTC) MONITOR

Execution time: 2 microseconds

Enable the real-time clock monitor interrupt; ignore k.

After execution of this instruction, equality between the RTC word register (location 15) and the RTC monitor word register (location 14) will interrupt the computer program. The next instruction is taken from the RTC monitor interrupt entrance register (location 12) and

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

	the RTC monitor is disabled.	
50:15	TERMINATE INPUT (INSTP)	Clear input active channel k.
	Execution time:	2 microseconds
	No monitor interrupt will occur as a result of the execution of this instruction.	
50:16	TERMINATE OUTPUT (OUTSTP)	Clear output active channel k.
	Execution time:	2 microseconds
	Terminate output on channel k.	
	No monitor interrupt will occur as a result of the execution of this instruction.	
50:17	TERMINATE EXTERNAL FUNCTION (EXFSTP)	Clear external function active channel k.
	Execution time:	2 microseconds
	Terminate external function on channel k.	
	No monitor interrupt will occur as a result of the execution of this instruction.	
50:20	SET RESUME (SRSM)	Set resume FF channel group K.
	Execution time:	2 microseconds
	Set the resume designator for the channel group specified by k to permit honoring the next requesting EF/OD function. Loss of any information currently held by the output register(s) for a peripheral device is allowed by this instruction.	
50:21	SKIP ON INPUT INACTIVE (SKPIIN)	
	Execution time:	2 microseconds skip or no skip
	Test for input buffer active on channel k. If inactive, skip the next instruction; otherwise, take next instruction.	
50:22	SKIP ON OUTPUT INACTIVE (SKPOIN)	
	Execution time:	2 microseconds skip or no skip
	Test for output buffer active on channel k. If inactive, skip the next	

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

	instruction; otherwise, take the next instruction.	
50:23	SKIP ON EXTERNAL FUNCTION INACTIVE (SKPFIN)	
	Execution time:	2 microseconds skip or no skip
	Test for external function activity on channel k. If inactive, skip the next instruction; otherwise, take the next instruction.	
50:24 or 50:25	WAIT FOR INTERRUPT (WTFI)	
	Execution time:	2 microseconds
	Stop the computer until any interrupt occurs; ignore k, then execute the instruction located in the interrupt entrance register designated by the interrupt.	
50:26	OUTPUT OVERRIDE (OUTOV)	
	Execution time:	2 microseconds
	Wait for the output device to accept the word in the C-register(s). Then simulate an output request on channel k and transfer the word designated by the address in the output buffer control register for that channel. Ignore the ESI or ESA mode if active. This instruction will transfer a word whether the buffer is active or not. Also, since the transfer takes place under control of the word in the buffer control register, the two buffer control words must not be equal. Equality terminates the action and no word will be transferred.	
50:27	EXTERNAL FUNCTION OVERRIDE (EXFOV)	
	Execution time:	2 microseconds
	Wait for the output device to accept the word in the C-register(s). Then simulate an external function request on channel k and transfer the word designated by the address in the external function buffer control register for that channel. Ignore the ESI or ESA mode if active. This instruction transfers a word whether the buffer is active or not. Also, since the transfer takes place under control of the word in the buffer control register the two buffer control words must not be equal. Equality terminates the action and no word is transferred.	
50:30 or 50:31	REMOVE INTERRUPT LOCKOUT (RIL)	
	Execution time:	2 microseconds

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

Remove the interrupt lockout - enable all interrupts, all channels; ignore k.

NOTE: A 50:30 or 50:31 instruction must be used in conjunction with a 50:34 or 50:35 instruction. It does not affect a 50:36 or 50:37 instruction.

50:32 or 50:33 — REMOVE EXTERNAL INTERRUPT LOCKOUT (RXL)

Execution time: 2 microseconds

Remove the external interrupt lockout - enable external interrupts, all channels; ignore k.

NOTE: A 50:32 or 50:33 instruction must be used in conjunction with a 50:36 or 50:37 instruction. It does not affect a 50:34 or 50:35 instruction.

50:34 or 50:35 — SET INTERRUPT LOCKOUT (SIL)

Execution time: 2 microseconds

Set the interrupt lockout - disable all interrupts, all channels; ignore k.

50:36 or 50:37 — SET EXTERNAL INTERRUPT LOCKOUT (SXL)

Execution time: 2 microseconds

Set the external interrupt lockout - disable external interrupts, all channels; ignore k.

50:40 — Not used

50:41 — RIGHT SHIFT AU (RSHAU)

Execution time:

4 usec (k = 0-4)	14 usec (k = 21-24)
6 usec (k = 5-8)	16 usec (k = 25-28)
8 usec (k = 9-12)	18 usec (k = 29-32)
10 usec (k = 13-16)	20 usec (k = 33-35)
12 usec (k = 17-20)	

Shift (AU) to the right, k bit positions. The higher-order bits are replaced with the original sign bit, AU_{17}, as the value is shifted. This is an end-off shift; for example, the low-order bits are lost upon completion of the shift.

Example of right shift AU with k = 2.

$(AU)_i$ (positive) = 370000

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

After first shift = 174000

After second shift = 076000

$(AU)_i$ (negative) = 400000

After first shift = 600000

After second shift = 700000

50:42 RIGHT SHIFT AL (RSHAL)

Execution time: Same as instruction 50:41

Shift (AL) to the right, k-bit positions. The higher-order bits are replaced with the original sign bit (AL_{17}) as the value is shifted. This is an end-off shift; for example, the low-order bits are lost upon completion of the shift.

50:43 RIGHT SHIFT A (RSHA)

Execution time: Same as instruction 50:41

Shift (A) to the right, k-bit positions. The higher-order bits are replaced with the original sign bit (A_{35}) as the value is shifted. This is an end-off shift; for example, the low-order bits are lost upon completion of the shift.

Example of right shift A with k = 2.

	AU	AL
$(A)_i$ (positive) =	370000	000000
After first shift =	174000	000000
After second shift =	076000	000000
$(A)_i$ (negative) =	400000	000000
After first shift =	600000	000000
After second shift =	700000	000000

50:44 SCALE FACTOR (SF)

Execution time:

6 usec (k = 0-4)	16 usec (k = 21-24)
8 usec (k = 5-8)	18 usec (k = 25-28)
10 usec (k = 9-12)	20 usec (k = 29-32)
12 usec (k = 13-16)	22 usec (k = 33-35)
14 usec (k = 17-20)	

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

Shift (A) circularly to the left until either $A_{35} \neq A_{34}$ or k-minus-shift-count = 0; then store the positive quantity k-minus-shift-count at memory address 00017. The effect of the instruction is to normalize (A) to the left subject to k. Scale factor is extremely useful when working with numerical values in floating point notation.

1) Example of scale factor with k = 7:

$(A)_i$ = 170000 000000 (positive, not normalized)

After first shift = 360000 000000 (positive, normalized)

The computer, sensing (A) now normalized, stores k - shift count (7-1) = the 18-bit quantity 000006 ⟶ 00017.

2) Example of scale factor with k = 3:

$(A)_i$ = 600000 000000 (negative, not normalized)

After first shift = 400000 000001 (negative, normalized)

The computer then stores the quantity 000002 ⟶ 00017.

3) Example of scale factor with k = 1:

$(A)_i$ = 070000 000000 (positive, not normalized)

After first shift = 160000 000000 (positive, not normalized)

The computer, having exhausted k, stores the quantity 000000 ⟶ 00017 leaving (A) only partially normalized.

50:45 LEFT SHIFT AU (LSHAU)

Execution time: Same as instruction 50:41

Shift (AU) circularly to the left, k-bit positions. The lower-order bits are replaced with the higher-order bits as the word is shifted. No bits are lost with the execution of left shift instructions.

Example of left shift AU with k = 2.

$(AU)_i$ = 300000

After first shift = 600000

After second shift = 400001

50:46 LEFT SHIFT AL (LSHAL)

Execution time: Same as instruction 50:41

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

Shift (AL) circularly to the left, k-bit positions. The lower-order bits are replaced with the higher-order bits as the word is shifted. No bits are lost with the execution of the left shift instructions.

50:47 LEFT SHIFT A (LSHA)

Execution time: Same as instruction 50:41

Shift (A) circularly to the left, k-bit positions. The lower-order bits are replaced with the higher-order bits as the word is shifted. No bits are lost with the execution of left shift instructions.

Example of left shift A with k = 2.

$(A)_i$	=	300000	000000
After first shift	=	600000	000000
After second shift	=	400000	000001

50:50 SKIP ON KEY SETTING (SKP)

Execution time: 2 microseconds skip or no skip

If bit 4, 3, 2, 1 or 0 of k is one and the corresponding skip key 4, 3, 2, 1, or 0 is set; or, if bit 5 of k is a one, skip the next instruction; otherwise, take the next instruction.

Examples of skip with:

k = 01 (bit 0)	Skip if skip key #0 is set.
k = 02 (bit 1)	Skip if skip key #1 is set.
k = 04 (bit 2)	Skip if skip key #2 is set.
k = 10 (bit 3)	Skip if skip key #3 is set.
k = 20 (bit 4)	Skip if skip key #4 is set.
k = 40 (bit 5)	Skip unconditionally.
k = 03 (bits 1, 0)	Skip if either key #1 or #0 is set.

50:51 SKIP ON NO BORROW (SKPNBO)

Execution time: 2 microseconds skip or no skip

If the last previous add A or subtract A required a borrow, take the next instruction; otherwise, skip the next instruction; ignore k. The skip occurs if no correction to (A) is needed. This allows a correcting

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

instruction to be inserted to save program steps. The correcting instruction will be subtract A where (Y+1, Y) = 000000000001.

50:52 SKIP ON OVERFLOW (SKPOV)

Execution time: 2 microseconds skip or no skip

If an overflow condition occurred on a previous arithmetic instruction, skip the next instruction; otherwise, take the next instruction. Ignore k and clear the overflow designator.

50:53 SKIP ON NO OVERFLOW (SKPNOV)

Execution time: 2 microseconds skip or no skip

If an overflow condition did not occur on any previous arithmetic instruction, skip the next instruction; otherwise, take the next instruction. Ignore k and clear the overflow designator.

50:54 SKIP ON ODD PARITY (SKODD)

Execution time: 2 microseconds skip or no skip

If the sum of the bits resulting from the bit-by-bit product of (AL) and (AU) is odd, skip the next instruction; otherwise, take the next instruction. Ignore k.

$(AU)_f = (AU)_i$; $(AL)_f = (AL)_i$

Example of skip odd parity:

(AU) = 000077 mask

(AL) = 127723

bit-by-bit product = 000023

bit sum = 3

Since the bit sum is odd, the next instruction is skipped.

50:55 SKIP ON EVEN PARITY (SKPEVN)

Execution time: 2 microseconds skip or no skip

If the sum of the bits resulting from the bit-by-bit product of (AL) and (AU) is even, skip the next instruction; otherwise, take the next instruction. Ignore k.

$(AL)_f = (AL)_i$; $(AU)_f = (AU)_i$

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

50:56 STOP ON KEY SETTING (STOP)

Execution time: 2 microseconds

If bit 4, 3, 2, 1, or 0 of k is one and the corresponding console stop key 4, 3, 2, 1, or 0 is set; or, if bit 5 of k is one, stop the computer; otherwise, take the next instruction.

Examples of stop with:

k = 01 (bit 0)	Stop if stop key #0 is set.
k = 02 (bit 1)	Stop if stop key #1 is set.
k = 04 (bit 2)	Stop if stop key #2 is set.
k = 10 (bit 3)	Stop if stop key #3 is set.
k = 20 (bit 4)	Stop if stop key #4 is set.
k = 40 (bit 5)	Stop unconditionally.
k = 03 (bits 1, 0)	Stop if either stop key #1 or #0 is set.

50:57 SKIP ON NO RESUME (SKPNR)

Execution time: 2 microseconds skip or no skip

If the resume designator specified by channel group k is not set (indicating unsuccessful transfer of a word to an output device), skip the next sequential instruction; otherwise, take the next instruction.

50:60 ROUND AU (RND)

If (AU) positive,
$(AU) + (AL)_{17} \longrightarrow AL$

If (AU) negative,
$(AU) - (AL)'_{17} \longrightarrow AL$

Execution time: 2 microseconds

If (AU) is positive, add bit position 17 of AL to (AU); if (AU) is negative, subtract the complement of bit position 17 of AL from AU and leave the resultant rounded (AU) in AL. Ignore k.

$(AU)_i = (AU)_f$

An application of this instruction would be: a double length value in A is normalized as far as possible to the left; however, only a rounded, single-length number is required for the accuracy desired.

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

50:61	COMPLEMENT AL (CPAL)	$(AL)' \rightarrow AL$
	Execution time:	2 microseconds
	Complement (AL), leaving the result in AL; ignore k.	

NOTE: This instruction affects a bit-by-bit complement with the following exception: all zeros (positive zero) will remain all zeros.

50:62	COMPLEMENT AU (CPAU)	$(AU)' \rightarrow AU$
	Execution time:	2 microseconds
	Complement (AU), leaving the result in AU; ignore k.	
	(See note: instruction 50:61.)	
50:63	COMPLEMENT A (CPA)	$(A)' \rightarrow A$
	Execution time:	2 microseconds
	Complement (A), leaving the result in A; ignore k.	
	(See note: instruction 50:61.)	
50:64	Not used	
50:65	Not used	
50:66	Not used	
50:67	Not used	
50:70	Not used	
50:71	Not used	
50:72	ENTER INDEX CONTROL REGISTER (ENTICR)	$k_{2-0} \rightarrow ICR$
	Execution time:	2 microseconds
	Clear the index control register. Then transmit the three, low-order bits of k to the ICR.	

TABLE 2-14. LIST OF INSTRUCTIONS (Cont)

50:73	ENTER SPECIAL REGISTER (ENTSR)	$k_{4-0} \rightarrow SR$
	Execution time:	2 microseconds
	Clear the special register. Then transmit the five low-order bits of k to the SR. ($SR_3 = 1$ activates the SR.)	
50:74	Not used	
50:75	Not used	
50:76	Not used	
50:77	Not used	

2-58. LOGIC

2-59. INTERNAL LOGIC. Internally (except in main memory), 0.0 volts represents a high (a logic 1), and -4.5 volts represents a low (a logic 0). Main memory uses +3.0 volts to represent a high and 0.0 volts to represent a low.

2-60. EXTERNAL LOGIC. Depending on configuration, either fast or slow interface logic provides communication with peripheral equipment. With fast interface, 0.0 volts represents a high (logic 1), and -3.0 volts represents a low (logic 0). With slow interface, 0.0 volts represents a high and -15.0 volts represents a low.

2-61. SYMBOLOGY

2-62. CIRCUIT SYMBOLOGY. The main symbols used on the functional schematics (Volume 2 Part 3) to represent electrical circuits and logic elements that constitute computer internal hardware are shown in figures 2-12 through 2-14. Volume 2 Part 1 contains physical and functional description of the printed-circuit cards. A brief description of the more common logic elements and the symbols used to represent them on the functional schematics are as follows. (Typical logic card numbers are given in parentheses.)

2-63. Inverter. The basic inverter (2070) is a single-input inverting amplifier (figure 2-12a). It operates with nominal static voltages of 0.0 volts and -4.5 volts. The circuit provides a low output for a high input, and a high output for a low input.

2-64. AND Inverters. The AND inverters (2030) are inverting amplifiers (figure 2-12b) with internal diode AND inputs. The number of inputs to each circuit is dependent upon the card type. The AND inverters operate with nominal static voltages of 0.0 volts and -4.5 volts. Each circuit provides a high output when all inputs are low and a low output when any one or all of its inputs are highs.

2-65. AND-OR Inverters. The AND-OR inverters (2040) are variations of the AND inverters and also operate with nominal static voltages of 0.0 volts and -4.5 volts. This inverter produces a high output when all inputs to any one of the AND circuits are lows. Conversely, the circuit produces a low output when a high is present on one or more inputs to each AND circuit (see figure 2-12c).

2-66. OR Inverter. The OR inverters (2220) are similar to the AND-OR inverters in that they are inverting amplifiers that perform OR functions and operate with nominal static voltages of 0.0 volts and -4.5 volts. The circuits produce low outputs only when all of their inputs are high, and a high output when any one of their inputs are low (see figure 2-12d).

2-67. Flip-Flops. The flip-flops (2000) are combinations of two internally interconnected inverters. The circuit may consist of either two AND-OR inverters or one inverter and one AND inverter. The flip-flops are represented on the functional schematics by a rectangle as shown in figure 2-13. The rectangle may be drawn on either a horizontal or vertical plane depending upon the function performed. A flip-flop has two stable states: the set state and the cleared state. The clear state exists when the output from the 1 side is high and the 0 side is low. The set state exists when the output from the 1 side is low and the 0 side is high. The flip-flop shown in figure

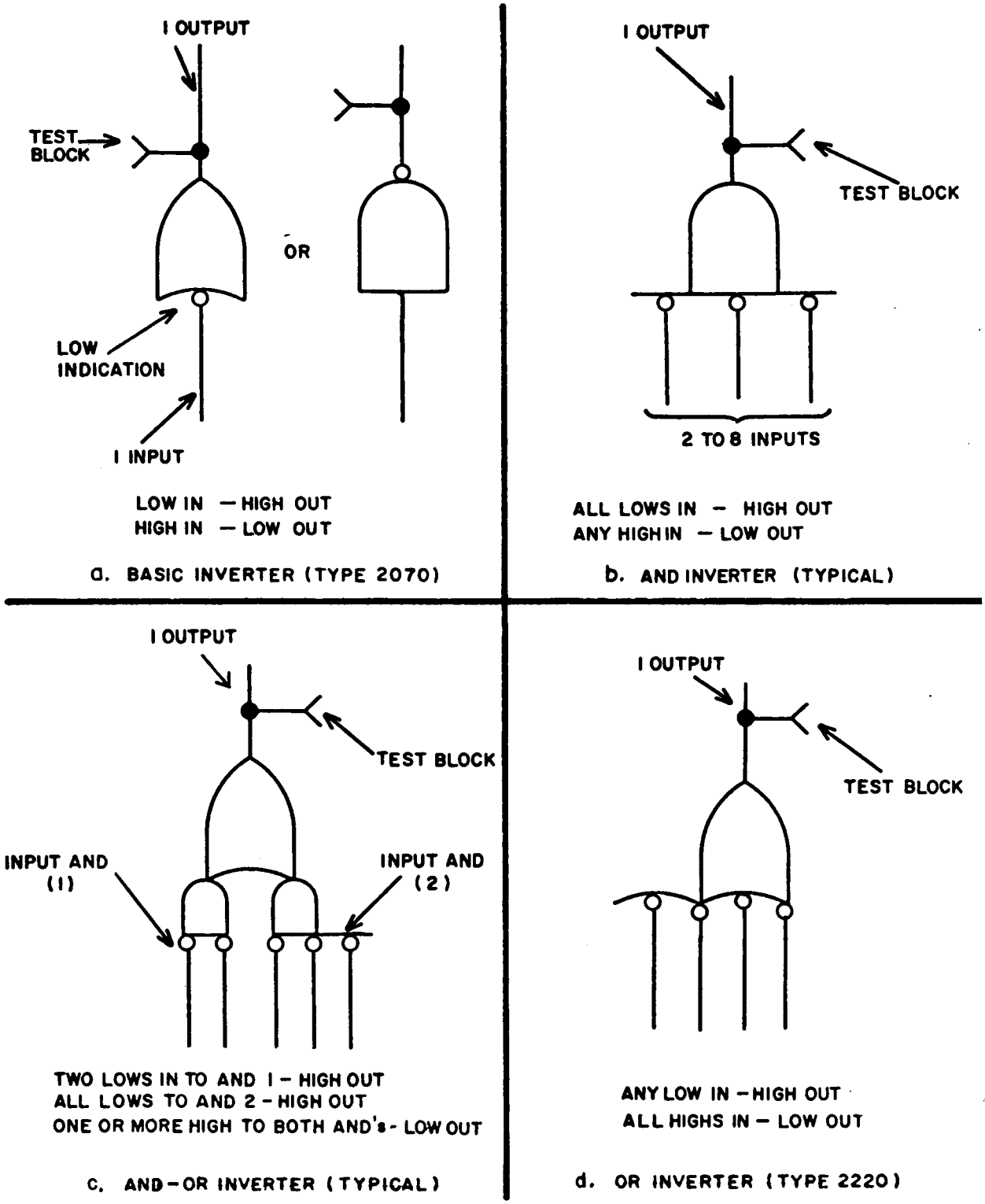
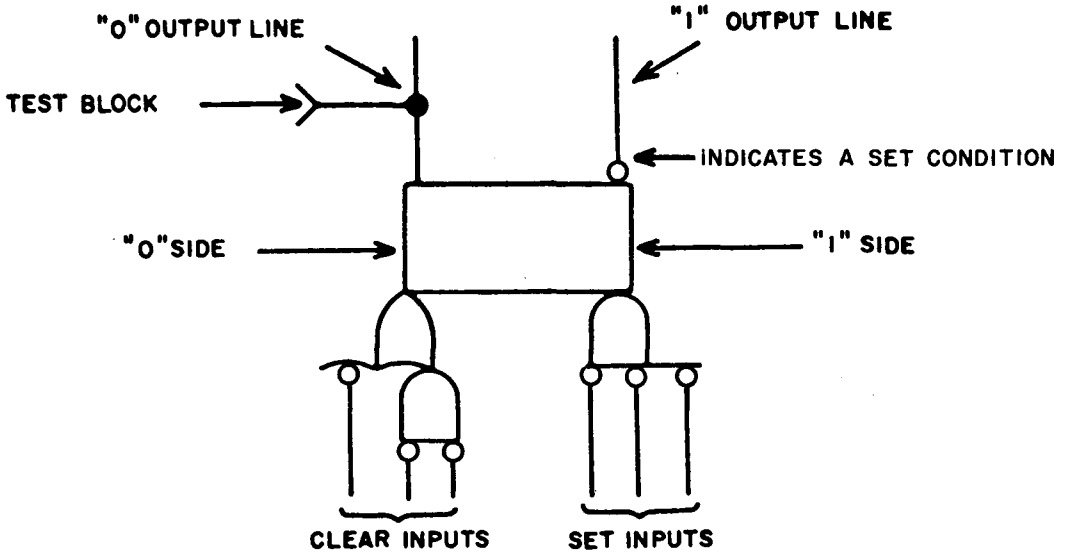
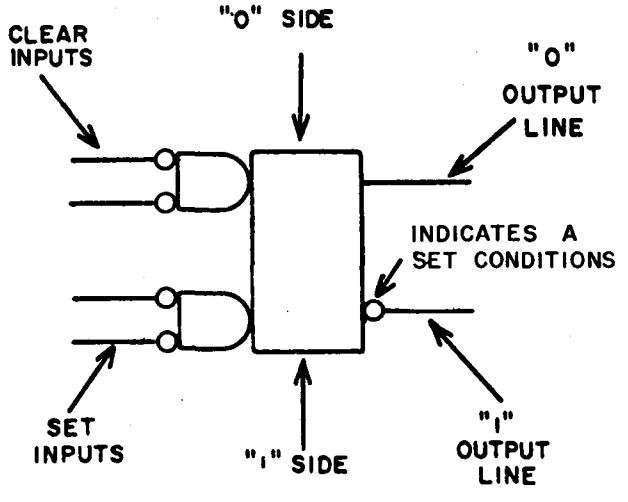


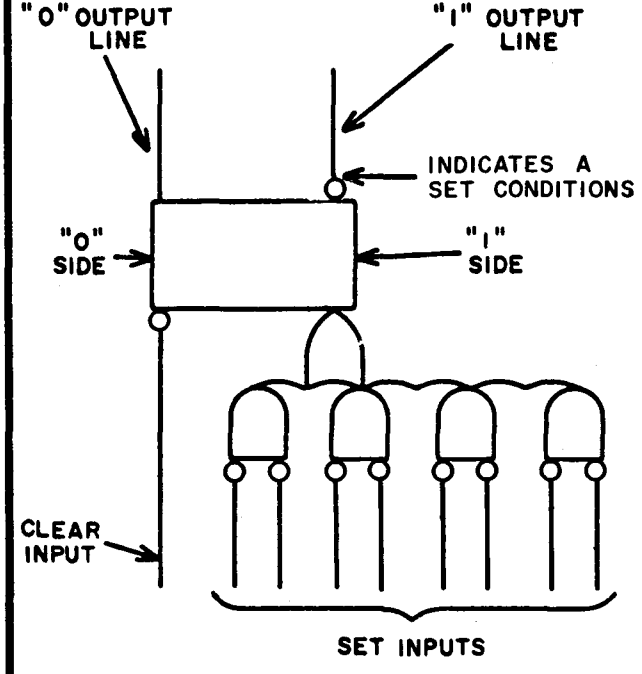
Figure 2-12. Typical Inverter Symbols



a. FLIP-FLOP (TYPE 2020)



b. TYPE 2020 FLIP-FLOP USED IN TIMING CHAIN



c. FLIP-FLOP (TYPE 2000)

Figure 2-13. Typical Flip-Flop Symbols

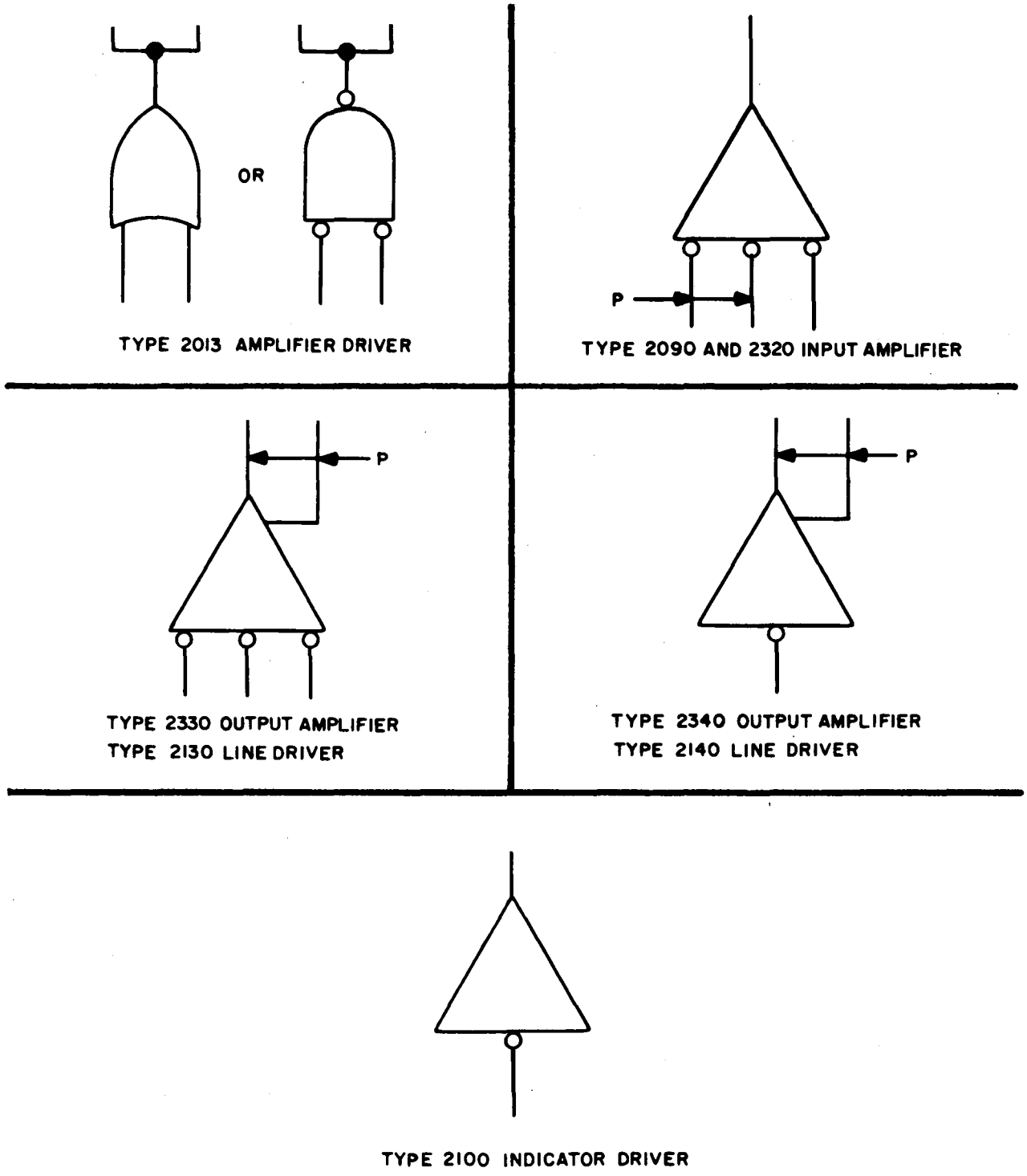


Figure 2-14. Typical Amplifier and Driver Symbols.

2-13a is set by simultaneously applying lows to all AND circuit inputs on the 1 side, and cleared by simultaneously applying lows to all AND circuit inputs on the 0 side. The outputs of all flip-flops are circled showing the set condition. In addition to setting and clearing a flip-flop by means of its input lines, a flip-flop can be set and cleared by applying the proper logic level to its set or clear output line. For example, a flip-flop can be forced to the set state by applying a high to the 0-output line. Conversely, a flip-flop can be forced to the cleared state by applying a low to the 0-output line.

2-68. Amplifier Driver. The amplifier driver (2013) is a double-inversion power amplifier used primarily for distributing clock and logic signals. The circuit operates with nominal static voltages of 0.0 volts and -4.5 volts and provides a low output only when all inputs to the circuit are lows. At all other times, the driver circuits are assigned a unique logic notation to identify the cycle and phase of the signal being transmitted. The amplifier driver is represented symbolically as shown in figure 2-14. The three common output pins each are capable of driving a number of logic circuits.

2-69. Input Amplifiers. The input amplifiers (2090) are used to amplify and invert the voltage levels of data and control signals received from externally located peripheral devices. Essentially, the circuits convert the externally generated signals to logic levels (0.0 and -4.5 volts) that are intelligible to the computer's internal circuits. The input amplifiers are represented symbolically on the functional schematics (refer to Part 3 in Volume 2 of this OP) by a triangle with three inputs and one output as shown in figure 2-14.

2-70. Output Amplifiers and Line Drivers. The output amplifiers and line drivers (2140) are inverting amplifiers used for the interface between the computer and input-output equipment. Essentially, the circuits convert computer-generated data and control signals to voltage levels (-3.0 and -15.0 volts) that are intelligible to the external peripheral devices. The output amplifiers and line drivers are represented symbolically by triangles as shown in figure 2-14.

2-71. Indicator Driver. The indicator driver (2100) is a dual-purpose inverting amplifier driver. It is used as a relay puller to control certain relays on the power control panel and as a driver for the PROGRAM STOP indicators on control panel number 2. The driver produces a zero-volt output for a low input, and presents an apparent open circuit to the relay or indicator for a high input. The circuit is represented symbolically on the functional schematics by a triangle with one input and one output as shown in figure 2-14.

2-72. REFERENCE DESIGNATORS. Reference designators used on the functional schematic diagrams include: unique designators, connector and jack location information, circuit card type identification, and special symbology. In addition, each schematic diagram contains such information as signal source and destination information, signal names, and page references.

2-73. Unique Designators. Unique designators are combinations of alphabetic characters and numerical superscripts used to identify uniquely each logic element and associated circuit in the computer system. The alphanumeric representation is in the form of abXcd as shown in figure 2-15. In

the notation, X (a letter) generally identifies a specific logic section, designator, or register. The abcd (numbers) indicates the level or rank of a circuit and/or its usage, as for example, the state of a register, the rank of a particular logic element, and 0 and 1 sides of a flip-flop, and is some cases, clock cycle and phase times.

2-74. Alpha Assignments. The alphabetic portion of the logic notation X is generally derived from the following alpha assignment list.

A - Arithmetic Register (Accumulator) and Adder

B - BU-Register

C - Input/Output (Communications) Register

D - Arithmetic Register

E - Control

F - Function Register and Translator

G - Control

I - Input Logic and Registers

J - Manual Control

K - Counter, K-Register

L - Secondary Timing Chains

M - Memory

N - Command Enables

P - Program Address Register

Q - Unassigned

R - Unassigned

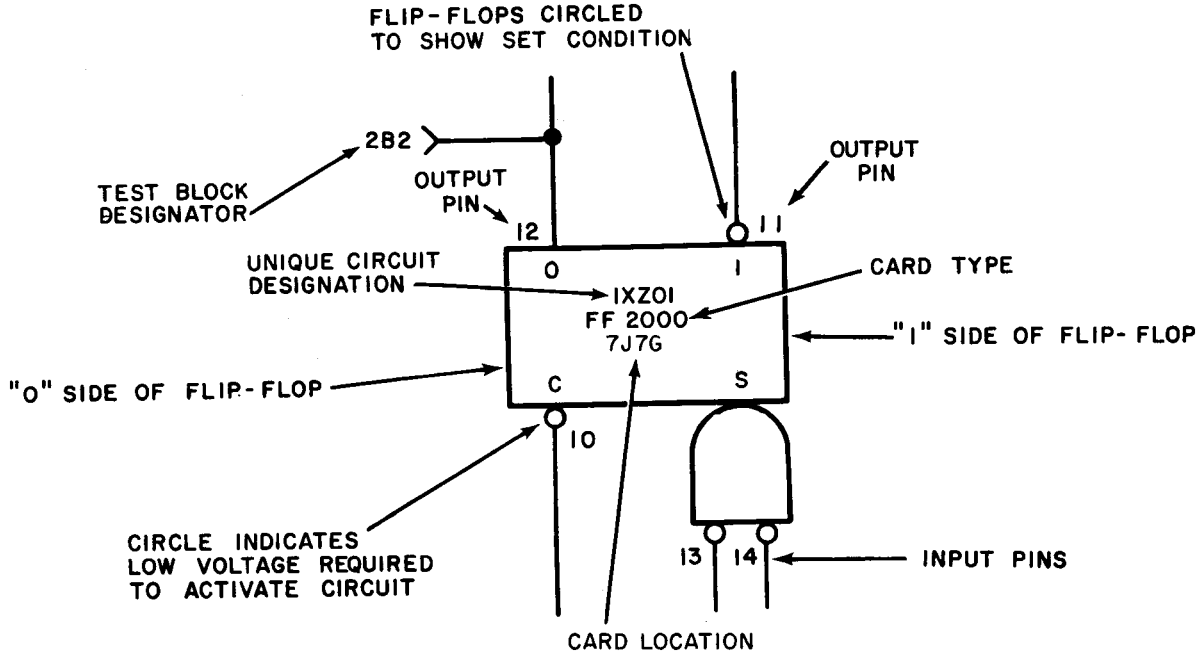


Figure 2-15. Examples of Logic Notations

S - Memory Address Register

T - Main Timing Chain

V - Input/Output Logic

W - Input/Output Control

X - Exchange Register

Y - Non-Standard Circuits

Z - Memory Data Register

2-75. Numeric Assignments. When the logic element is a flip-flop, the b portion the logic notation abXcd may be either a 0 or 1 (figure 2-15) depending upon the side of the flip-flop to be referenced and is written on the schematic as an X. The 01 designates the set (1) side of the flip-flop; the 00 designates the clear (0) side of the flip-flop. In the case of a register or counter, the cd portion of the logic notation identifies individual bit positions or stages of the register or counter. For example, a register or counter normally has its stages numbered from left to right in descending order (2^n. . . 2^{03}, 2^{02}, 2^{01}, 2^{00}) with individual bit position specified as a power of two. The cd portion, therefore, would be . . . 03, 02, 01, 00. In the case of a double-ranked register or counter, the c digit in the second rank is assigned arbitrarily (6, 7, or 8, for example), leaving a sufficient gap between the c numbers of the two ranks to allow for notating intermediate logic elements. If the numerics assigned the primary rank are as follows (considering only the 01 side of each flip-flop in the register):

01_X0n . . . 01_X03, 01_X02, 01_X01, 01_X00,

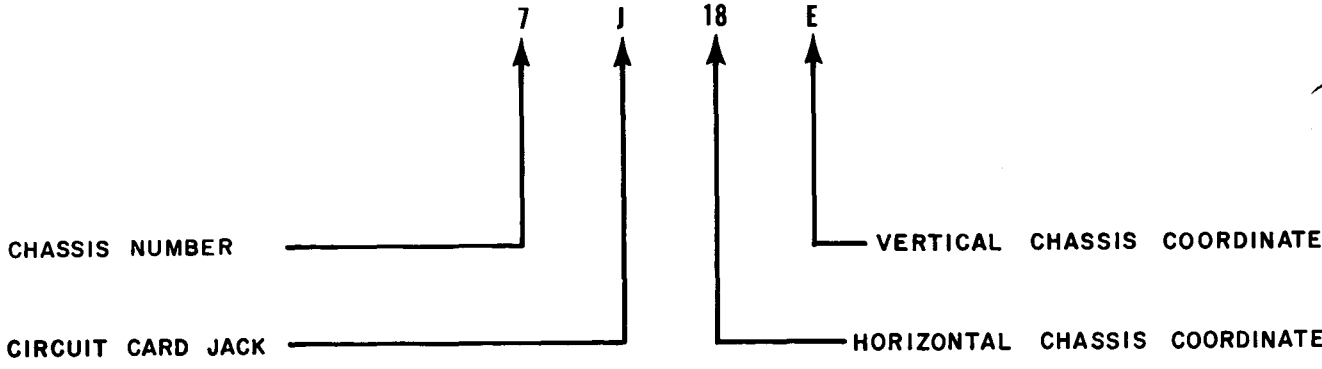
the numeric assignments for the second rank would be:

01_X4n . . . 01_X43, 01_X42, 01_X41, 01_X40.

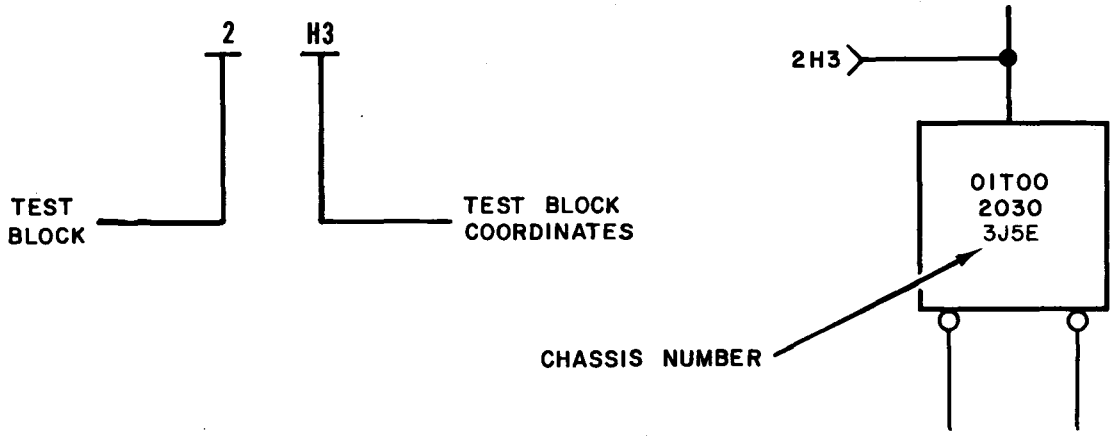
The arbitrary assignment of 4 as the value for c in the secondary rank allows a sufficient gap between the two ranks for designating associated logic elements. On functional schematics depicting more than one circuit, a g or h is used in place of a number in either the ab or cd position of the unique term. This g represents a different numerical value for each of the duplicated circuits and is usually defined in a note on the drawing.

2-76. Card Location Designators. A card location designator specifies the location of a card in the computer by chassis number and chassis coordinates. As shown in figure 2-16a, the first digit in the designator identifies the chassis number. For duplicate circuits, the chassis number may be shown as g or h. The J identifies the connector as a circuit card jack. The remainder of the designator identifies the coordinate location of the card connector on the specified chassis. Plate P-1 (figure 9-1 in Chapter 9 of this OP) identifies the various chassis by chassis number in the computer. Plates P-181 through P-187 are illustrations showing the arrangement of the circuit card connectors on the various chassis (refer to Volume 2 Part 3).

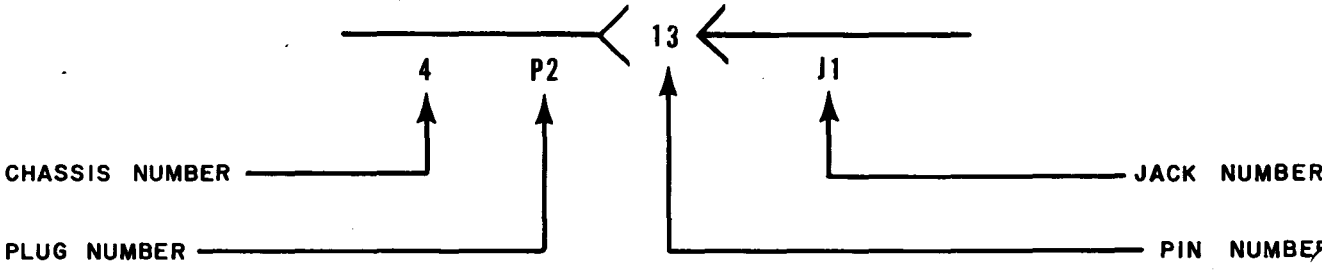
2-77. Test Block Designators. A test block designator identifies a test block terminal by test block number and pin number (in terms of its coordinates on the test block) as shown in figure 2-16b. However, the designator does not identify the panel or chassis number of the test block. To determine on which chassis the test block is located, refer to the first digit of the card location designator



a. Card Location Designator



b. Test Block Location Designator



c. Connector Designators

Figure 2-16. Card Location, Test Block, and Connector Designators

of the circuit associated with the test block in question.

2-78. Connector Designators. Each connector in the computer is assigned a unique reference designation consisting of a chassis identifier, plug and jack identifier, and pin number as shown in figure 2-16c. The permanent part of the connector is referred to as the jack and is identified by a J. The movable part of the connector is referred to as the plug and is identified by a P. Note in figure 2-16c that no chassis number is assigned to the jack; it is always the same as the plug. For duplicate circuits, the chassis may be shown as g or h.

2-79. Special Symbology. When practical, input and output lines to and from logic elements are identified by signal names or special symbology to name the signal transmitted and/or to define a logical event. In addition to the signal or special symbol that appears above the signal line, the following information is provided to aid in signal tracing and maintenance.

1. Logic designation of the circuit generating the signal (source of input signals) or the circuit receiving the signal (destination of output signals).

2. Plate number (Chapter 9) of the source or destination circuit.

Figures 2-17 and 2-18 show examples of these designators as well as other special symbols.

2-80. DUPLICATE CIRCUITS. The I/O drawers and memory drawer each contain identical chassis. Therefore, only 1 circuit is usually shown to depict 4 identical I/O chassis or 2 identical memory chassis. As stated earlier, the chassis numbers on these circuits are shown as g or h. A table is provided with each circuit containing these unique designators listing the appropriate unique designator, card jack and plug terms for each chassis. Most identical circuit drawings contain the g term only. However, some I/O circuits use both g and h to distinguish two chassis on any one drawer.

2-81. CHASSIS MAPS. The chassis maps show the physical layout of the card chassis, the location and type number of each circuit card used on the chassis, the number of circuits available on each card, logic notations for those circuits that are used, spare circuits on each card, and spare connectors on the chassis. Chassis maps for the computer are included with the functional schematic diagrams (refer to plates P-181 through P-187 in Volume 2). A simplified chassis map is shown on figure 2-19. As can be seen on the illustration, each chassis (except the memory chassis) contains 245 card connectors. Individual connectors are arranged on the chassis in seven rows. Columns are numbered left to right, 1 through 35. The following discussion pertains to individual cards on the sample chassis map shown in figure 2-19.

Location A1 — Contains type 2070 card. The card contains five circuits. Only the first four circuits are used. The first circuit is labled 02S10 on the functional schematics; the fourth circuit, 03S11. The dashes shown in the area reserved for the fifth logic notation indicate this circuit is a spare. The numbers 6 and 8 indicate that the circuits identified are found in plates P-6 and P-8. The 1-1-1-1-1 shows that each circuit has one input.

Location G35	Contains type 2930 card. The card contains two flip-flop circuits.
Location F35	A usable spare location.
Location F34	A nonusable spare location.

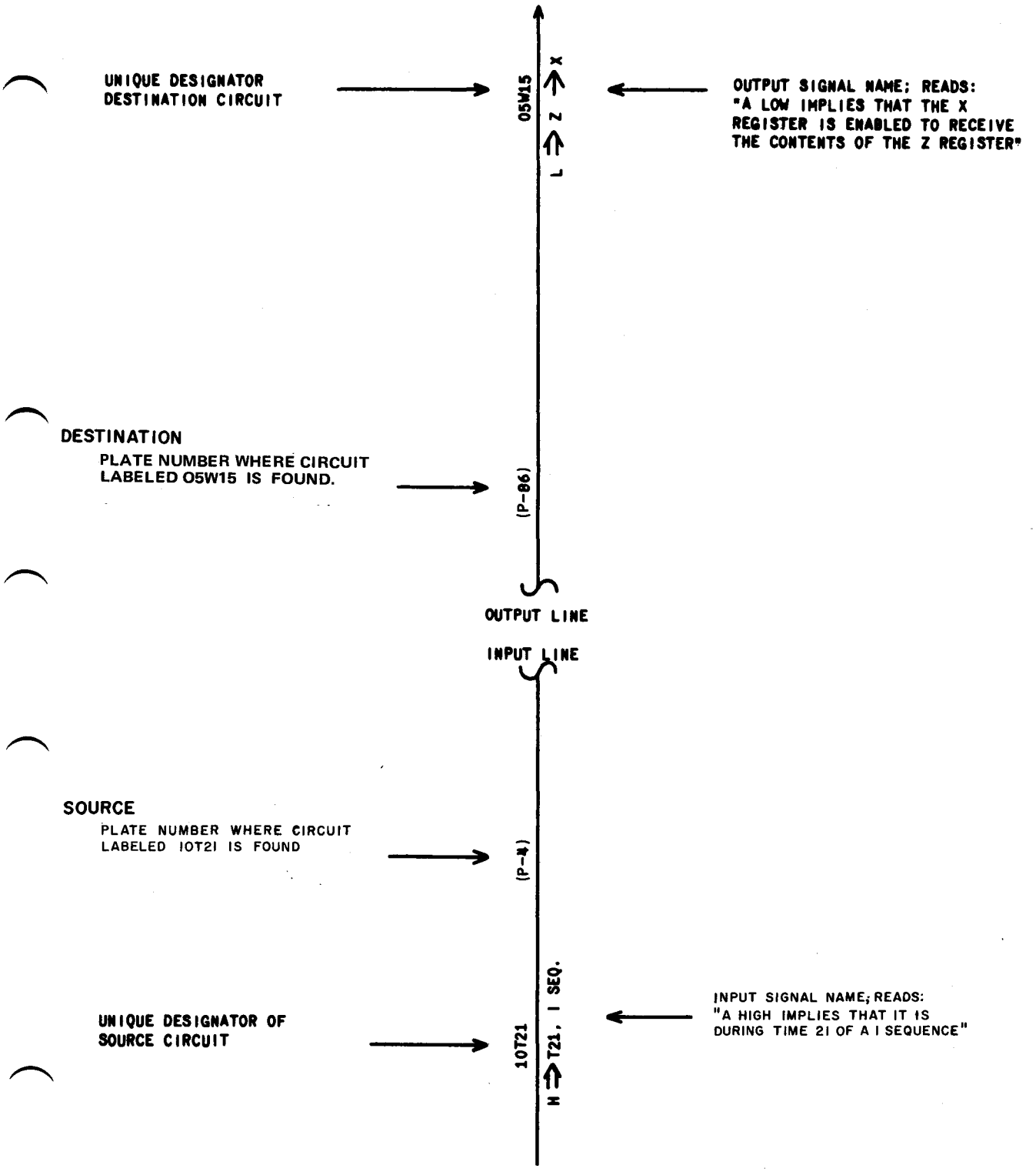


Figure 2-17. Signal Line Symbology

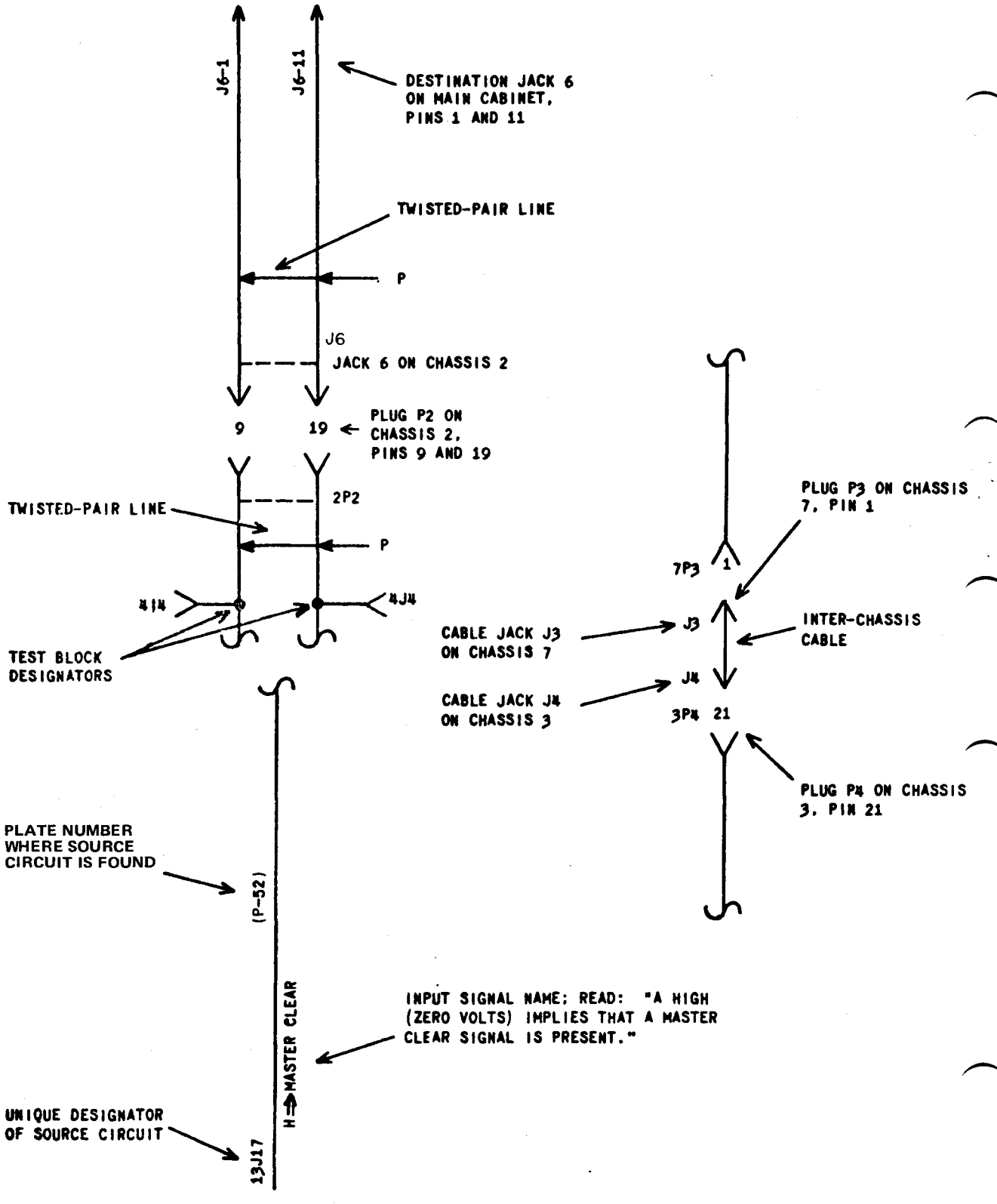


Figure 2-18. Special Symbology

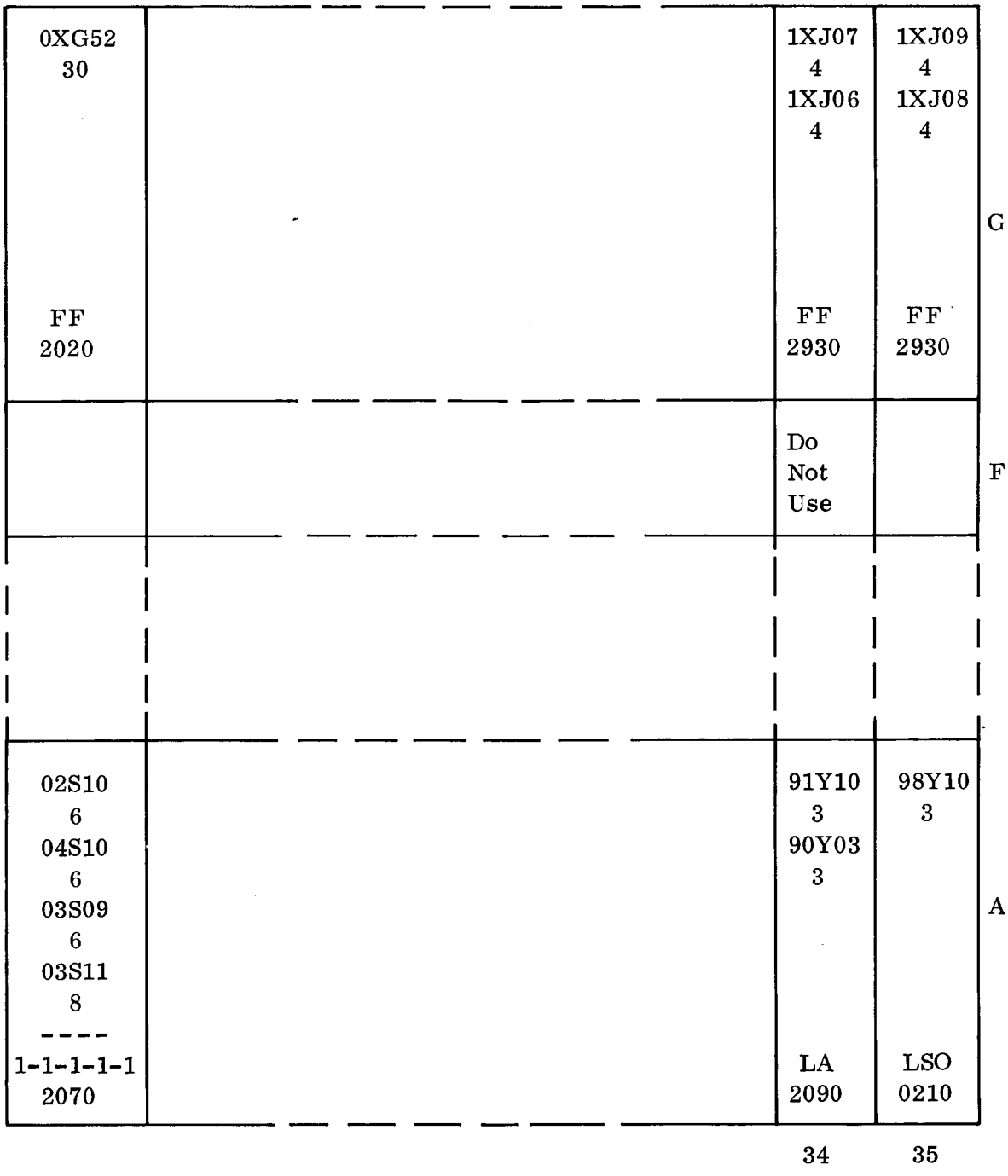


Figure 2-19. Simplified Chassis Map

Section 2-3. Detailed Functional Description (Control Section)

2-82. INTRODUCTION

This section provides a functional description of the Control Section (paragraph 2-83), Arithmetic Section (paragraph 2-233), Memory Section (paragraph 2-326), Input/Output Section (paragraph 2-413), and Power Distribution Circuitry (paragraph 2-603). Functional schematics are provided in Volume 3, Part 3.

2-83. CONTROL SECTION

2-84. The control section of the computer is divided functionally into five major groups. These groups are: console control, timing circuits, program translation and control, registers, and special circuits. Figure 2-20 depicts the structure of these groups and their interrelationships.

2-85. CONSOLE CONTROL. The console control group of the control section comprises three control circuits: start/stop logic, master clear, and mode select logic. Basic control of computer operation from initial application of power and the starting of a computer program to termination of program execution is provided by these circuits. (Refer to Table 2-15 for control settings and functions.)

2-86. Start/Stop Logic. Start/Stop logic circuitry enables the operator to start computer execution of a program, control issuance of phase and timing pulses for the several modes of operation, and stop computer operation. These operations are accomplished using the three-position spring-loaded RESTART/START STEP and SEQ STEP/STOP switches.

2-87. Start Circuit. Start logic circuitry consists primarily of the Start and Run flip-flops, RESTART SPEED CONT potentiometer, and associated logic circuitry. The Start flip-flop ensures that the RESTART/START STEP switch output is honored only once for each switch depression, or for each LO SPEED OSC pulse. This occurs because the flip-flop changes state (clears) when the computer goes into operation and prevents the honoring of any further pulses until the computer stops. At the initiation of computer operation, the Run flip-flop is set and supplies enabling signals to various control logic circuits. The RESTART SPEED CONT potentiometer, R1, controls the frequency of the LO SPEED OSC and thus governs the application of pulses applied to the Start flip-flop during a restart application.

2-88. Start circuitry is shown on Plate P-3 of the Chapter 9 Functional Schematics. RESTART/START STEP switch S9 is used to initiate computer operation. With S9 in the neutral position as shown, the high on contact 6 is applied to pin 8 of 31J10. This high combined with the disabling input at pin 6 ($\overline{\text{REMOTE}}$) fully disables 31J10. The low output at 31J10 pin 9 is applied to pin 14 of flip-flop 2XJ10 partially enabling this flip-flop. The low on pin 2 of switch S9 enables 36J10 whose high output disables 34J10. The low from pin 13 of 34J10 fully enables flip-flop 2XJ10 putting it in the set condition. The high off pin 11 of 2XJ10 partially enables AND gate 23J10. AND gate

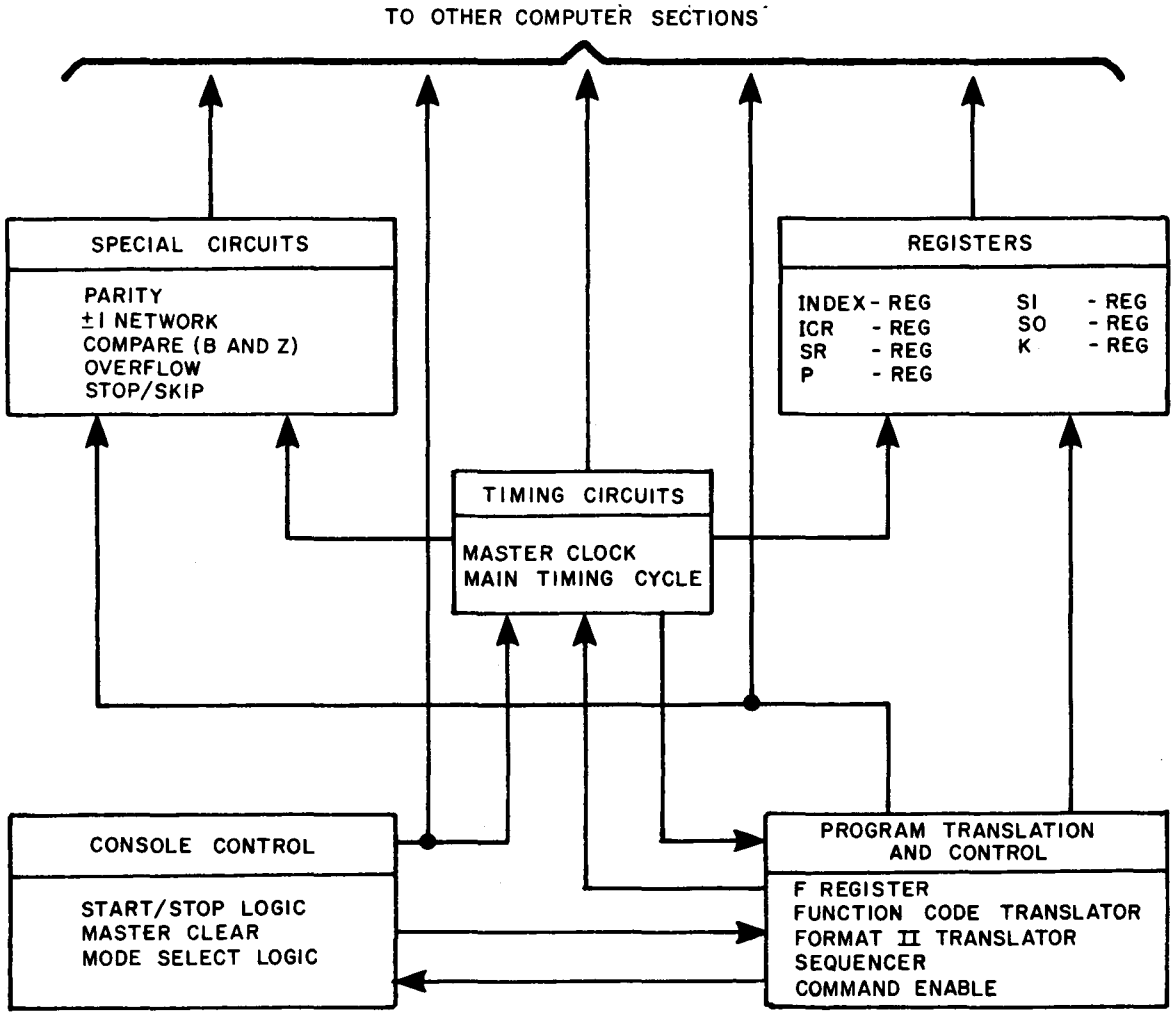


Figure 2-20. Control Section, Block Diagram

23J10 will not be enabled since lows exist on pin 13 (not phase RPT), pin 15 (low from pin 13 of 34J10), and pin 14 (33J10 is disabled due to high on pin 14 since 32J10 is disabled because of a low on pin 5 and REMOTE on pin 12). When the RESTART/START STEP switch is placed in the START STEP position, a high is developed as an input to inverter 32J10. The low output of 32J10 is ANDed with the local control pulse to cause 33J10 to output a high. The high is applied to pin 14 of 23J10. This gate is now completely enabled and the Start command enable is developed. The Start pulse is applied as an input to the Run 1 flip-flop causing this flip-flop to set at T4.2 due to 03T42 and ϕ2. An output from the set side of the Run 1 flip-flop is fed back to the Start flip-flop, causing it to clear at T4.4. When this flip-flop clears, a low occurs on pin 11 of 23J10 and the Start pulse is terminated.

2-89. Restart circuitry is shown in plate P-3. With the RESTART/START STEP switch in the RESTART position, a low is applied to 34J10 via gate 36J10. 34J10 is enabled when the LO SPEED OSC outputs a high which is inverted to a low by 95Y10. The high output from 34J10 enables 23J10 and the Start Command enable is developed.

TABLE 2-15. CONTROL PANEL SWITCH SETTINGS AND RELATED OPERATION

Control Panel Switches					Operation
RESTART/ START STEP	SEQ STEP/ STOP	AUTO RECOVERY	PHASE REPEAT	FUNCTION REPEAT	
Settings With RUN Mode Indicator Lighted					
START STEP	Neutral	Down	Down	Down	Initiates normal computer high-speed run.
START STEP	Neutral	Up	Down	Down	Initiates normal high-speed run with bootstrap memory referenced after program fault.
START STEP	Neutral	Down	Down	Up	Repeat instruction contained in F-register at computer high-speed run.
Neutral	STOP	Down	Down	Down	Stop computer high-speed operation after the execution of the I-sequence of the next instruction.
RESTART	Neutral	Either	Down	Either	Restart computer operation after a stop instruction or operation of the SEQ STEP/ STOP switch.
Settings With PHASE STEP Mode Indicator Lighted					
START STEP	Neutral	Either	Down	Either	Send indicated phase pulse to computer logic, advance phase generator, and stop.
RESTART	Neutral	Either	Down	Either	Continuously send sequential phase pulse as indicated, to computer logic and advance phase generator.
NEUTRAL	Neutral	Either	Up	Either	Send a continuous pulse of phase indicated to computer logic at normal high-speed rate.

TABLE 2-15. CONTROL PANEL SWITCH SETTINGS AND RELATED OPERATION (Cont)

Control Panel Switches					Operation
RESTART/ START STEP	SEQ STEP/ STOP	AUTO RECOVERY	PHASE REPEAT	FUNCTION REPEAT	
Settings With OP STEP Mode Indicator Lighted					
START STEP	Neutral	Either	Down	Either	Stop computer operation after execution of I-sequence of each instruction.
RESTART	Neutral	Either	Down	Either	Restart computer operation continuously after above listed stop.
START STEP	SEQ STEP	Either	Down	Either	Stop computer operation after each major sequence.
RESTART	SEQ STEP	Either	Down	Either	Restart computer operation continously after above listed stop.
Settings With LOAD Mode Indicator Lighted					
START STEP	Neutral	Down	Down	Down	Jump to address 00500, initiate bootstrap program, clear Load Mode, and continue operation in Run mode.

2-90. Stop. Stop logic is used to terminate computer operation by clearing the Run flip-flops. In Run mode, this is accomplished by programmed stops or by manually positioning the SEQ STEP/STOP switch to STOP. The stop function is dependent upon the switch settings listed in table 2-15 and any programmed computer stop instructions.

2-91. Positioning the SEQ STEP/STOP switch to STOP causes a program stop at the completion of the current sequence. Execution of the f = 50:56 instruction also causes a program stop if the stop condition is satisfied. With either of these two methods, the program stop is effected by disabling the control of the sequencer. No operation is performed without sequence control. The I/O sequences are not disabled and main timing continues unaffected, thus allowing I/O operations. The computer mode is not altered.

2-92. Once stopped, manipulation of the START STEP/RESTART switch will effect restart and the computer will continue operation in the same mode which had control prior to the stop.

2-93. Stop Circuit. Stop circuitry is shown in plate P-3. When SEQ STEP/STOP switch S10 is positioned to the STOP position (down) gate 25J10 is enabled by a high. A low is applied to Run 1 flip-flop.0XJ10 and at time T4.1 0XJ10 is cleared. The low output at 0XJ10 pin 12 is applied to Run 2 flip-flop 1XJ10 and at T1.2 1XJ10 is cleared. When both Run flip-flops are clear, all sequence control is disabled except for the I/O sequence.

2-94. Master Clear. Master clear circuitry allows the operator to clear either certain selected flip-flops, or all of the primary flip-flops in the computer. This option is provided by the three-position spring-loaded I/O CLEAR/MASTER CLEAR switch.

2-95. Placing the switch momentarily in the I/O CLEAR position clears the channel and Function Priority flip-flops, the I/O translator circuits, the EF/OD Active flip-flops, and the C-register. The Resume flip-flop and Request flip-flops are set.

2-96. Placing the I/O CLEAR/MASTER CLEAR switch momentarily in the MASTER CLEAR position when the computer is not in a run or phase step condition, clears all sections, registers, and control flip-flops in the control computer. If in a run condition, only the Voltage Fault and/or Program Fault flip-flops and the Memory Protect flip-flops (refer to P-134 and P-121) are cleared.

2-97. The Master Clear circuitry is shown in plate P-7. If the MASTER CLEAR switch S11 is activated, 00J11 outputs a low. This low output is ANDed with the output of 09J13, which is a low in local control. The high output of 01J11 enables 02J11 to develop a Master Clear.

2-98. If in a $\overline{\text{run}}$ condition, the MASTER CLEAR switch clears all sections, registers, and control flip-flops in the computer. The low output of 02J11 is ANDed with the low into 03J11, which outputs a high to enable 04J11 and 06J11, which enables 05J11, 09J11 and 10J11. Gate 05J12 is also enabled, which enables 06J12, 08J12 and 10J12. If in a run condition, the input into 03J11 is high, disabling 03J11 and only the Voltage Fault and/or Program Fault flip-flops 9XG14 and 9XG90 (refer to plate P-121) and the Memory Protect flip-flops 1XG8l and 0XG8l (refer to plate P-134) are cleared.

2-99. I/O Clear. Positioning I/O CLEAR/MASTER CLEAR switch to I/O CLEAR causes 00J12 to output a low. This low output is ANDed with the low from 09J13 in local control, and the low from not being

in RUN. The high out of 04J12 enables 05J12, which enables 06J12, 08J12 and 10J12, which develops an I/O Clear to clear the Channel and Function Priority flip-flops (refer to plates P-65 and P-66), the I/O translator circuits (refer to plate P-51), the EF/OD Active flip-flops (refer to plates P-61 through P-64), and the C-register (refer to plates P-71 and P-72). The Resume flip-flop (refer to plate P-69) and Request flip-flops (refer to plates P-61 through P-64) are set. (The actual procedure for clearing and setting these flip-flops is covered in detail under Command Instructions in the I/O portion of Section 2-3.) If in a run condition, 04J12 is disabled by the high input on pin 7 and gates 05J12, 06J12, 08J12, and 10J12 are disabled preventing clearing I/O. The computer can also be cleared remotely.

2-100. Mode Select Logic. The computer performs logic functions in one of four possible modes of operation. These four modes are Run, Phase Step, OP Step (operation step), and Load. The mode of operation must be manually selected on Control Panel 1 (A2) (refer to figure 2-21) by pressing the specific pushbutton indicator corresponding to that mode. Becuase of the interconnecting logic circuitry, only one mode at a time can be selected. Mode selection is made by mode selection circuitry. Pressing the desired pushbutton indicator supplies an enable for the selected mode and disables the three remaining modes. Enabling a given mode initiates sequencing of the computer for that mode through generation of command outputs.

2-101. Run Mode. Run mode is the normal high-speed operating mode of the computer. All operations not pertaining to malfunction isolation and program debugging are performed in Run Mode. Although initial program loading is performed in the Load mode, the computer reverts to Run mode after approximately 2 microseconds of operation.

2-102. The Run Mode (refer to plate P-3) is initiated by depressing the Run pushbutton DS14A and effectively enabling 00J0 This will disable all other modes by placing a high on 00J01, 00J02, and 00J03. The low output of 01J00 sends a partial enable to 05J10. When the Run 1 flip-flop 0XJ10 is set by operating the RESTART/START STEP switch, 05J10 will be enabled and develop a high output which is used to prevent the accidental clearing of the computer in case the MASTER CLEAR switch is activated. Gate 06J10 develops the Program Run command enable which is used to illuminate the RUN indicator on the control panel.

2-103. Phase Step Mode. Phase Step mode provides normal computer operation, including the memory circuitry, at a control rate of speed. Used primarily for logic malfunction isolation, Phase Step mode may be executed either by manually issuing each successive phase of the Master Clock through the START STEP position of the RESTART/START STEP switch, or by using the RESTART position of the switch and the LO SPEED OSC. The RESTART SPEED CONT potentiometer controls speed of computer operation in the Phase Step mode. By placing the PHASE REPEAT switch to the up position and selecting a desired phase, the computer repeatedly issues the chosen pulse at a high rate of speed. If all four phases are selected, a cycle-step operation results.

2-104. The Phase Step mode (refer to plate P-3) is initiated by depressing the PHASE STEP pushbutton DS14C. A high is placed on 00J02 effectively enabling it and disabling all other modes by applying a high at

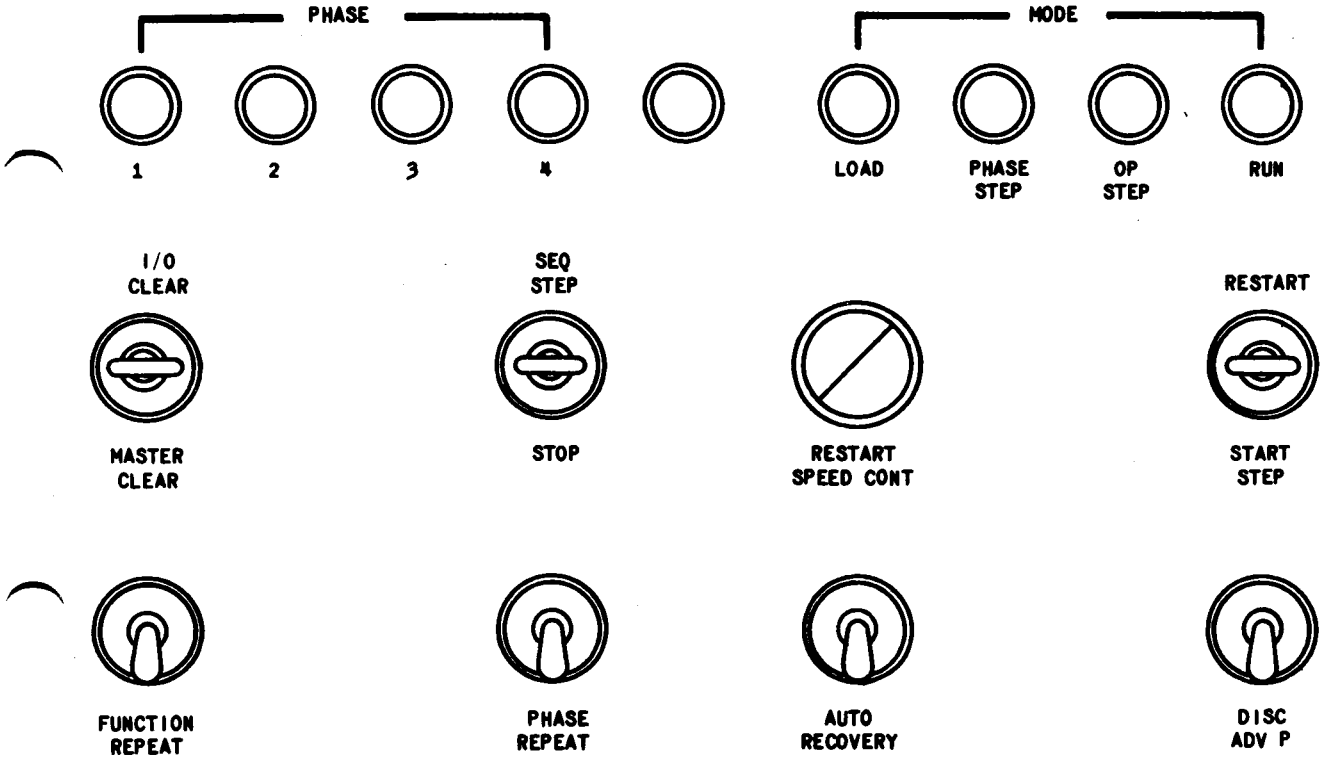


Figure 2-21. Operational Controls, Control Panel 1 (A2)

the input of 00J00, 00J01, and 00J03. The low output of 01J02 is applied to Start flip-flop 2XJ10, which is cleared every ∅4, and to 02J02, which outputs a high to 24J06 (refer to plate P-4) and a low to 30J06 and 21J06 through 21J09 (refer to plate P-4) via 03J02 (refer to plate P-3).

2-105. The desired phase is selected by depressing one of the PHASE pushbuttons, DS14G through DS14J (refer to plate P-4). For example, depressing DS14J will set the lower rank ∅1 flip-flop, 0XJ06. When the Start Step pulse is received, it sets the upper rank ∅1 flip-flop, 1XJ06 enables gate 1J06 and partially enables 71C01. The Start pulse, which is applied to pin 7 of 24J06 (B5), causes the Master Clock to develop two complete cycles to partially enable 71C01 through 71C04 in sequence. The only gate that would be completely enabled under these circumstances is 71C01. Therefore, clock phase one (∅1) is developed.

2-106. The low output of 1XJ06 is applied to 0XJ07. When 71C14 outputs a low, it is applied to the lower rank flip-flops 0XJ06, 0XJ07, 0XJ08, and 0XJ09, via 32J06 with no phase repeat. This allows the sequential setting and clearing of these flip-flops. Flip-flop 0XJ07 will set and when the Start Step pulse is again applied, ∅2 will be developed. Phases ∅3 and ∅4 are developed in the same manner. If the PHASE REPEAT switch S14 is set to ON, gate 30J06 is enabled and outputs a high to disable 32J06. This prevents the special ∅4 of 71C14 from being applied to the lower rank flip-flops and changing their state.

2-107. The high output from 30J06 is also applied to 23J10 (refer to plate P-3) to send

a constant Start to 24J06 in the Master Clock. The phase that was initially selected is continually repeated at a rate determined by the speed on the free running Master Clock. The lower rank flip-flops can be manually cleared by depressing MAN CLR PHASE switch, S8.

2-108. OP Step Mode. In the OP Step mode, the computer operates at a high rate of speed, but its operation is stopped at the end of an I-sequence (except for I/O instructions) throughout the performance of a program and must be manually restarted. During this mode, the operator examines the contents of the various registers for accuracy at specific times throughout the execution of a program. If a malfunction is detected, the operator can determine, from register content, the approximate point in the program at which the error occurred. Each operation of the RE-START/START STEP switch steps the computer through the major sequences, stopping after the I-sequence of each instruction except for I/O instructions. If the switch is on (SEQ STEP/STOP position), the computer stops after each major sequence of an instruction.

2-109. The OP Step mode (refer to plate P-3) is initiated by depressing the OP STEP pushbutton DS14B. A high is placed on the output of 00J01, effectively enabling it. All other modes are disabled by placing a high on 00J00, 00J02, and 00J03. The low output of 01J01 is applied to Run 1 flip-flop 0XJ10, gates 07J10 and 24J10. During an I-sequence, 24J10 is enabled and outputs a high to enable 25J10. At T4.1, Run 1 flip-flop is cleared enabling 07J10 which outputs a high to stop main timing by disabling 03T11 (refer to plate P-8) in main timing, thereby allowing one instruction to be performed. The computer remains stopped until the Run 1 flip-flop is again set.

2-110. With the SEQ STEP/STOP switch S10 (refer to plate P-3) in the SEQ STEP (up) position, a low is applied to 24J10 and it is constantly enabled. Therefore, the Run 1 flip-flop is cleared at T4.1 to effect a sequence stop.

2-111. Load Mode. The load mode forces the computer logic to address the first instruction of the wired memory program in bootstrap, or the routine loaded in 500_8 -537_8 of main memory. The computer performs at the high-speed rate. After the memory referencing has occurred, it performs in the normal Run mode.

2-112. Load mode (refer to plate P-3) is initiated by depressing LOAD pushbutton DS14D. A high effectively enables 00J03 and disables all other modes by placing a high to the inputs of 00J00, 00J01, and 00J02. The low out of 01J03 sends a Load mode to gates 10G70 (refer to plate P-28), 20N07 (refer to plate P-21) and 60N12 (refer to plate P-22).

2-113. TIMING CIRCUITS. Timing within the computer is initially provided by a master clock circuit. An additional clock circuit is used to govern functions of the main core memory and utilizes basic timing of the Master Clock. Master Clock, clock distribution, timing chains, and phase step logic are included in the timing circuitry.

2-114. Master Clock. The Master Clock (refer to plate P-4) generates and distributes four clock pulses every 500 nanoseconds. The time duration of each clock pulse is approximately 125 nanoseconds. These four pulses, identified as phases 1, 2, 3, and 4, comprise one clock cycle. Four clock cycles comprise one computer cycle.

2-115. The relationship between the phases, clock cycles, and computer cycle is illustrated in figure 2-22. This illustration also includes the relationship of timing chain outputs to the Master Clock output.

2-116. The clock-generating circuit consists of a delay line, two phase generators, two phase flip-flops, and gates and inverters, which produce the proper enables. A simplified logic diagram of the Master Clock is shown in figure 2-23. The time relationship between various outputs and clock pulses is shown in figure 2-24. When not in use, all outputs of the delay line are 0.0 volt.

2-117. Applying an external enable to the clock-input inverter initiates the propagation of a -4.5-volt signal through the delay line circuit. After approximately 15 nanoseconds, this signal reaches pin eight of the delay line and is sent to gate EF 02. Between 120-125 seconds after being applied to the delay-line chain, the signal reaches pin six and appears as a -4.5-volt signal at the input inverter. This signal disables the input inverter and a 0.0-volt signal is propagated through the delay-line. At 125 nanoseconds later, the input inverter is re-enabled and the process is repeated. The application of delay-line output signals to the gating circuits and phase generators produces outputs shown in figure 2-24.

NOTE: Output pin numbers from the Master Clock delay lines may vary between individual computers because of inherent delays in logic modules.

2-118. Phase 1 (Ø1). 24J06 is enabled by a Start or absence of Phase Step (refer to P-4). The high output of 24J06 along with the high from 03DY01 enables 01DD01 and a low level is applied to the delay lines. The low from 01DD01 is applied to 62C13 via 01EF01 and 60C01. At this time, pin 6 of 03DY01 is high and is inverted by 04EF01 and applied as low to 62C13. The low output from 63C13 is applied to Ø1 and Ø3 gates 71C01 and 71C03. Initially 7XC13 is cleared and the low output of the clear side enables Ø1 gate 71C01, and along with the low of 63C13, sets 7XC24. Phase 1 will be generated until the low moves down the delay line and outputs a low at pin 6 of 03DY01, which disables 62C13, ceasing generation of Ø1.

2-119. Phase 2 (Ø2). Initially the high output of pin 8 of 01DY01 provides a low to 62C24 via gate 02EF01. However, the high out of pin 5 of 03DY01 causes 62C24 to be disabled via gates 60C00 and 03EF01. As the low moves down the delay line, pin 8 of 01DY01 becomes a low putting a high on 62C24. When pin 5 of 03DY01 goes low it outputs a low to 62C24. When 01DD01 again outputs a high, pin 8 of 01DY01 becomes a high, putting a low to 62C24. The high out of 62C24 causes 63C24 to go low. The low is applied to Ø2 and Ø4 gates 71C02 and 71C04. The low out of 63C24 along with the low from the set side of 7XC24 sets 7XC13. When the output of 03DY01 pin 5 again becomes high, 62C24 is disabled via 60C00 and 03EF1 and Ø2 is terminated.

2-120. Phase 3 (Ø3). The low output from the set side of 7XC13 along with the low from 63C13(enabled as in Ø1) enables Ø3 gate 71C03. At this time the low from 63C13 and the set side of 7XC13 clears 7XC24. Phase Ø3 is terminated the same as Ø1.

2-121. Phase 4 (Ø4). The low output from the cleared side of 7XC24 along with the low from 63C24 (enabled as in Ø2) enables Ø4 gate 71C04. The low from 63C24 along with the low from the clear side of 7XC24 clears 7XC13. The generation of Ø4 is terminated

the same as Ø2. (Refer to table 2-16 for a description of the clock phase generator.) The output of 71C14 is used to clear 2XJ10 at Ø4.

2-122. In NARROW position, CLOCK NARROW-NORMAL switches allow the master clock circuit to generate pulses of a shorter time duration for marginal check procedures. Under ordinary operation, the switches will be in the NORMAL position.

2-123. Clock Distribution. Clock distribution circuits (refer to plate P-5 and P-6) supply the logic driving elements used to propagate clock phase pulses 1 through 4 throughout the computer. Each phase is applied through several non-inverting drivers to the associated computer logic. Plate P-5 depicts this distribution to chassis three and four, located physically in drawer A2. Plate P-6 depicts the distribution to the I/O chassis, and to chassis seven and eight. The I/O chassis are located in drawer A1, and, with the 12- or 16-I/O channel option, drawer A8. Chassis seven and eight are located in drawer A4.

2-124. Clock pulse distribution to the memory drawer A3 is shown in plate P-6 where phase 1 and 3 are routed to the main memory timing circuit on plate P-134. Further memory timing is supplied from the main memory timing logic.

2-125. Timing Chains. In order to effectively expand the clock phase outputs, a group of 16 flip-flops, referred to as the main timing chain, is used. These flip-flops change their set/clear configuration on every clock phase. There are 16 individual configurations that occur during four 4-phase clock cycles. These four clock cycles comprise one main timing cycle. The flip-flops are arranged sequentially, so each clock phase sets the next sequential flip-flop and clears a previously set flip-flop. Each flip-flop is set for the duration of one 4-phase clock cycle. The timing flip-flops are numbered according to the setting sequence. When the last flip-flop is set, it enables the setting of the first flip-flop on the next clock phase for recycling.

2-126. The necessary sequence of events to execute an operation is timed by the combination of the timing chain flip-flop outputs and the clock phases. As each sequential flip-flop is set, a new step (data transfer to a register, etc.) may be performed as a part of an instruction execution. As discussed in later sheets, more than one timing chain cycle may be necessary to complete the execution. For example, one cycle could be used to obtain an instruction from memory, and the next cycle could execute that instruction.

TABLE 2-16. NORMAL MASTER CLOCK OPERATION

63C13	63C24	7XC24 ff	7XC13 ff	Generated Ø
L	H	Set	Clear	Ø1
H	L	Set	Set	Ø2
L	H	Clear	Set	Ø3
H	L	Clear	Clear	Ø4
L	H	Set	Clear	Ø1

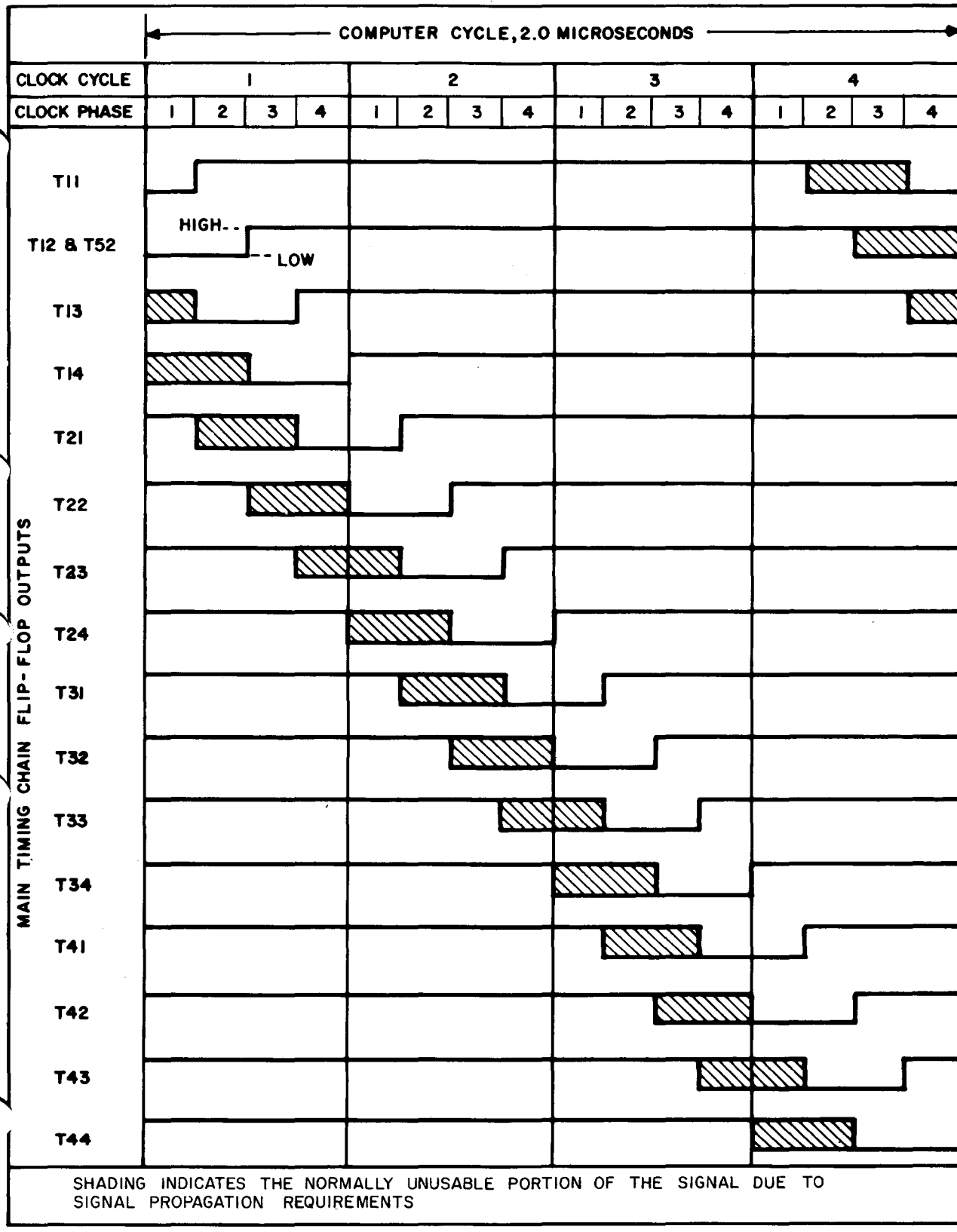


Figure 2-22. Main Timing Cycle Outputs

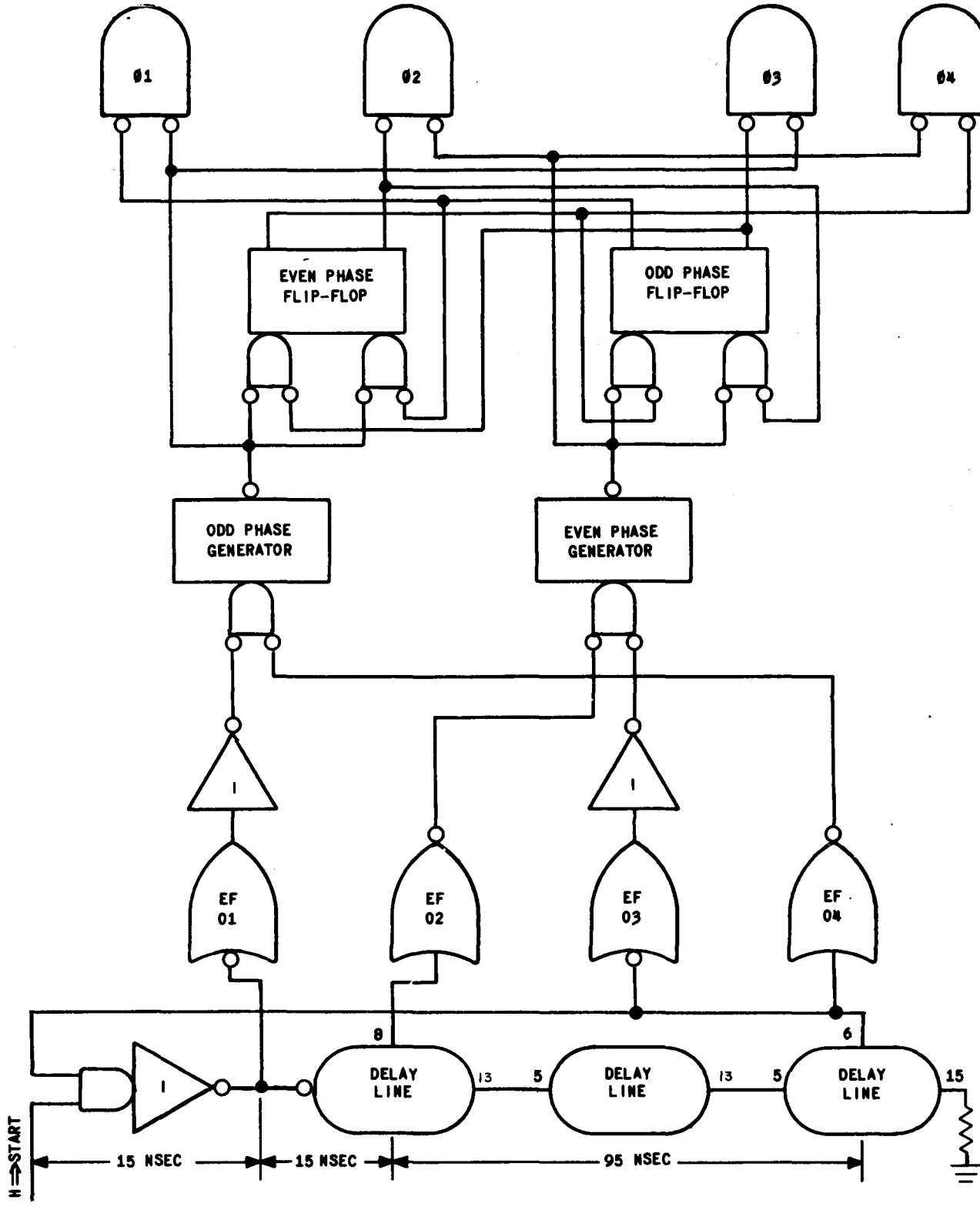


Figure 2-23. Master Clock Logic, Simplified Diagram

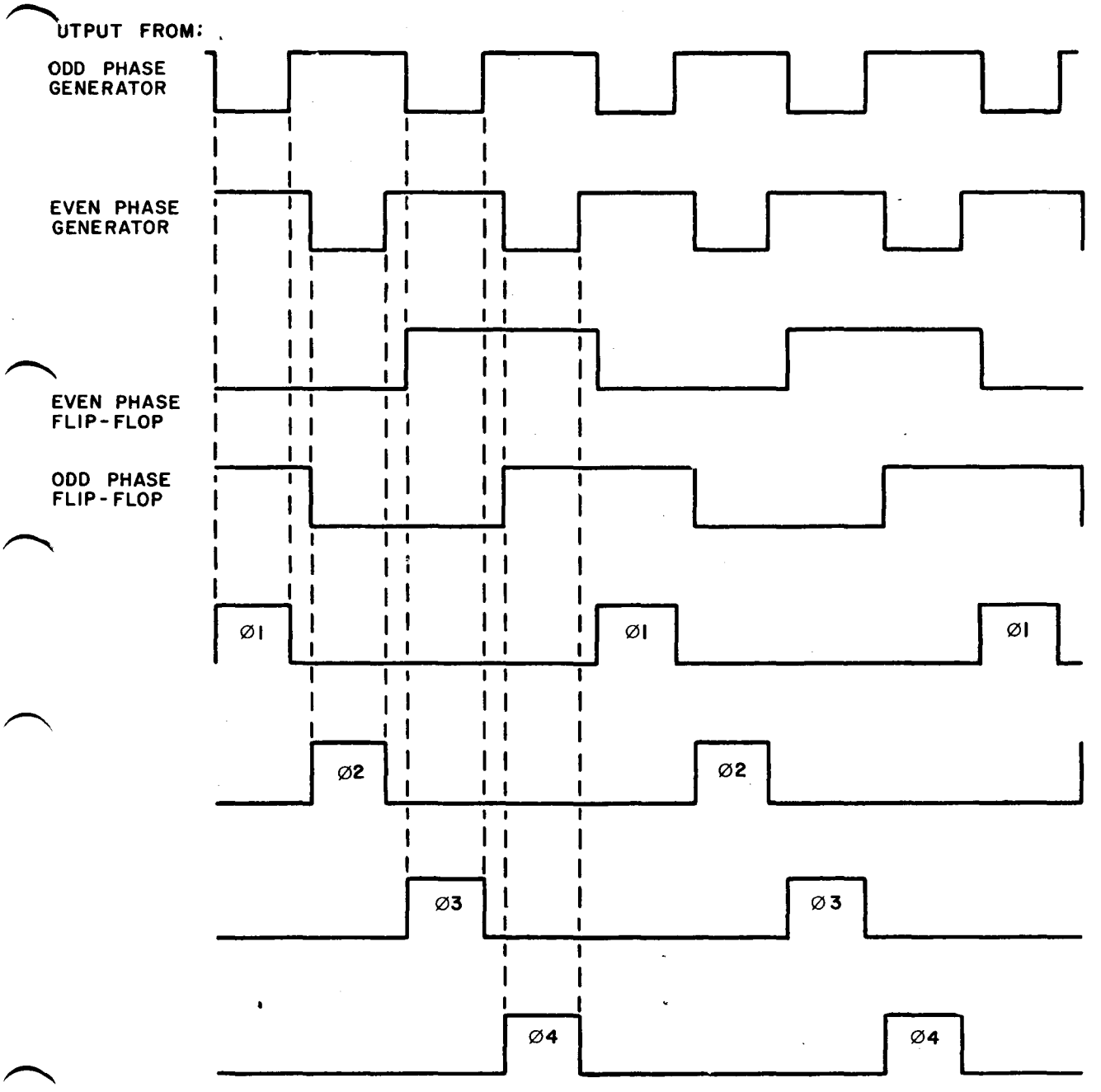


Figure 2-24. Master Clock Output Signals

2-127. Figure 2-22 illustrates the main timing chain. Notice the time relationships between the computer cycle, clock cycle, and clock phase. Four clock phases produce 1 clock cycle, and 4 clock cycles produce 1 computer cycle. The individual flip-flops T11 (0XT11) through T44 (0XT44), are shown on the extreme left of the figure.

2-128. When an individual flip-flop is set it produces a particular computer time; for example, T11 sets at clock cycle 4, clock phase 2, or computer time T4.2. It clears at clock cycle 1, clock phase 2, or computer time T1.2. The shaded portions of the computer times generated by individual flip-flops are not usually utilized.

2-129. The T11 flip-flop shaded area encompasses T4.2 and T4.3, indicating that the computer would not normally make use of the T11 flip-flop output at these times (although they are used in certain instances). The best usuable portion of the T11 flip-flop occurs at T4.4 and ends at T1.1. Refer to table 2-17 for the flip-flop setting/clearing sequence.

2-130. The main timing chain (refer to plates P-8 through P-11) consists of flip-flops 0XT11 through 0XT14, 0XT21 through 0XT24, 0XT31 through 0XT34, 0XT41 through 0XT52, and associated gates. The 0XT52 flip-flop is set for only two special operations and is discussed later in this section. The main timing chain runs continuously as long as the Master Clock is in operation and Op Step mode has not been selected. Gate 03T11 (refer to plate P-8) is enabled when not in Op Step mode and 0XT11 is set. Input gate 10T11 can only be enabled when OXT 33, OXT 23, and OXT 13 are clear which occurs from time T3.4 to T4.3. Since the first flip-flop, 0XT11, is using $\phi 2$ as an enable to both set and clear this flip-flop, it can only set at T4.2 and clear at the next $\phi 2$ (time T1.2). When 0XT11 sets at T4.2, it enables 0XT12, the next flip-flop in line, via 03T11, to set with the application of $\phi 3$, or at T4.3. Note that when 0XT12 sets, the low output will set 0XT13 at T4.4 and clear 0XT11 at T1.2. This action will continue so that each clock phase sets the next sequential flip-flop and clears some previously set flip-flop.

2-131. PROGRAM TRANSLATION AND CONTROL. The purpose of program translation and control circuitry is to decode the program instruction addressed and determine what operations must be performed to accomplish the instruction. This circuitry consists of the F-register function code subtranslators, function translators, and sequencers.

2-132. F-Register. The F-register receives the f portion of the instruction word from memory via the Z-selectors, stores this information, determines whether it is a Format I or Format II instruction, and then outputs to the function translation circuits for decoding. The F-register (refer to plate P-40) is cleared by the MASTER CLEAR switch, the CLEAR F-REGISTER pushbutton, or time T1.3 of an I (initial) sequence. The most common method of clearing the F-register, which is during an I-sequence, is described in the following paragraphs.

2-133. A H $\Rightarrow$ T14 I-Seq enables OR gate 05N16 causing it to output a low to pin 6 of AND gate 06N16. This low is ANDed with a $\phi 3$ low at pin 5 and a low due to the absence of a H $\Rightarrow$ Initiate Buffer at pin 7. The high produced by 06N16 is amplified by 08N16 and clears all of the flip-flops of the F-register.

TABLE 2-17. MAIN TIMING

Clock Phase	Flip-Flop Action	Time Notation
ø2	Set T11	
ø3	Set T12	
ø4	Set T13	
ø1	Set T14	T1.1
ø2	Set T21, clear T11	T1.2
ø3	Set T22, clear T12	T1.3
ø4	Set T23, clear T13	T1.4
ø1	Set T24, clear T14	T2.1
ø2	Set T31, clear T21	T2.2
ø3	Set T32, clear T22	T2.3
ø4	Set T33, clear T23	T2.4
ø1	Set T34, clear T24	T3.1
ø2	Set T41, clear T31	T3.2
ø3	Set T42, clear T32	T3.3
ø4	Set T43, clear T33	T3.4
ø1	Set T44, clear T34	T4.1
ø2	Set T11, clear T41	T4.2
ø3	Set T12, clear T42	T4.3
ø4	Set T13, clear T43	T4.4
ø1	Set T14, clear T44	T1.1

NOTES: Timing continues as long as recycle (setting T12) is enabled.
T52 flip-flop is not shown.

2-134. The instruction is classified as a Format I or II by AND gate 16F50. Its inputs from the Z-selector represent an octal-coded 50 in the six most significant bits of the instruction read from memory into the Z1 register. Some of the bits are inverted in the Z-selector so all inputs to 16F50 are lows if Z1 contains a 50 code in the upper six bits. Qualification of 16F50 disables pin 8 of AND gate 19N16 and enables that pin of 29N16 via inverter 17F50. The absence of an H ⇒ Initiate Buffer and the presence of ∅4 and L ⇒ T24 I-Seq qualify AND gate 10N16, which outputs a high. This high is inverted by 11N16 and applied simultaneously to pin 9 of AND gates 29N16 and 19N16 thereby enabling gate 29N16 which outputs a low to enable the transfer from Z-select to the F register. The low output from 29N16 sets the Format II flip-flop 0XF06 and effects the transfer of bits 6-11 of the Z-select. Conversely, if 19N16 is the active gate, 0XF06 remains cleared, and bits 12-17 are gated from Z select to the F-register.

2-135. If the FUNCTION REPEAT switch is placed in the up, or ON, position a high (ground-OV) disables AND gates 10N16 and 06N16 to prevent clearing or setting the function register except by the pushbuttons on the control panel. The instruction contained in the F-register will be continuously repeated until manual intervention stops the computer or a fault condition occurs.

2-136. Function Code Subtranslators. Function code subtranslators (refer to plates P-41, P-42, and P-43) receive input from the F-register and translate this data into the proper numeric value for sequencing the computer logic.

2-137. Subtranslator 1. Subtranslator 1 (refer to plate P-41) translates the more significant octal digits of the function code from F_{05-03}. The gates which perform this translation are 08F00 through 08F30, 08F60, 09F40, 09F50, and 09F70. The next to last number of the logic gate number indicates the octal digit for which the particular gate translates. For example, 08F00 outputs a high level if bits 05 through 03 of F contain A0's. Thus, this high level would indicate the presence of any function code whose more significant octal digit is 0_8, which is f = 00-07. Other gates (shown in plate P-41) translate for larger function code groups. For example, a low level output of 91F00 indicates f = 00-07, 10-17, 20-27, 30-37, 40-47 or 00-47.

2-138. Subtranslator 2. Subtranslator 2 (refer to plate P-42) translates the less significant octal digit of the function code. Only four of the eight possible conditions are indicated because only bits two and one (flip-flops 0XF02 and 0XF01, respectively, refer to plate P-40), are tested. For example, AND gate 09F01 outputs a low if both bits are zero. Bit zero (0XF00) is not tested, so this low output could signify either f = X0 or X1. The X means the more significant octal digit can be anything.

2-139. Subtranslator 3. Subtranslator 3 (refer to plate P-43) identifies the low-order digit as being either even (0, 2, 4, or 6) or odd (1, 3, 5, or 7). If 0XF00 (refer to plate P-40) contains a one, 02F00 is disabled by a high from the clear side of the flip-flop and 03F00 is qualified. The reverse is true if the flip-flop is not set. If combined with the logic outputs of subtranslator 2, the F_{00} indications can specify the exact least-significant octal digit of the function code. For example, if subtranslator 2 indicates f = X4, X5 and subtranslator 3 indicates f = even, then f = X4. This is exactly what is done in the function

translators. The flip-flop 0XF06 and the 03F06 gates indicate the format of the instruction. If 0XF06 = 1_2 (set), the instruction is format II. From this logic, format indication is combined with other function code translations to completely describe the instruction.

2-140. Function Translators. The function translators (refer to plates P-44 through P-48) comprise a network of AND, OR, and AND-OR gates and inverters which changes the function code contained in the F-register into enable signals to initiate the necessary computer operation.

2-141. Sequencer. During the execution of an instruction, the main timing cycle is under the control of a sequence. The major sequences used by the computer are I, R, W, B, I/O, and interrupt. These sequences determine the use of the timing signals taken from the main timing flip-flops. Different sequences cause different operations to be performed. Sometimes two sequences may run parallel with each other. The instruction itself determines what sequence is to be used, and the number of sequences required to complete the instruction determines the instruction execution time.

2-142. I-Sequence. The I-Sequence is the first sequence performed during the execution of every instruction. During this sequence the instruction word is read out of memory and initially acted upon. If required, B-modification is performed and the value of U_p, U_{SR}, or XU is formulated.

2-143. R-Sequence. During the R-sequence, an operand (value to be operated upon) is obtained from memory at an address specified by the instruction. If only one operand is required, an R1-sequence is generated; if two operands are required, such as with a double add or double subtract instruction (20-23), both R1 and R2-sequences are generated. The R1-sequence obtains the first 18-bit operand from memory and the R2-sequence obtains the second 18-bit operand.

2-144. W-Sequence. During the W-sequence the computer stores values into memory locations as designated by the instruction.

2-145. B-Sequence. B-sequences are used in preparation for I/O operations. During this sequence an I/O channel is activated and the buffer limits are established. The IB1-sequence transfers the TACW to control memory and the IB2-sequence transfers the IACW to control memory, for use by the I/O-sequences.

2-146. I/O Sequence. During the I/O-sequence the computer is communicating with some peripheral device. If in single channel operation (18-bit words), only the I/O1 sequence is required. If in dual channel operation (36-bit words), both the I/O1 and I/O2 sequences are generated.

2-147. Interrupt Sequence. During the Interrupt sequence the computer program is interrupted. The interrupt may be generated internally, by the computer itself, or externally by peripheral equipment.

2-148. Upper-Rank Sequence Designator. The upper-rank sequence designator contains a series of flip-flops, each set to initiate a unique sequence of events within computer logic. Except for the I/O sequence flip-flops, the upper rank contains the initial sequence flip-flops and the lower rank contains the final sequence flip-flops. The initial sequence flip-flops are cleared at time T4.1 and are enabled to be set at time T4.2. The following paragraphs describe an application of the upper rank sequence designator (refer to plate P-12).

2-149. The low at the input to 00E20 is inverted and enables 01E20, and at time T4.1, 08E20 is enabled. The high from 08E20 is amplified by 09E20 and used to clear the upper rank sequence register. To set the flip-flops (with the exception of I/O 1 and I/O 2) the low from 03T42, and not being in I/O, will enable 12E20. The low output of 12E20 will provide a partial enable for the flip-flops. Several conditions for the other enable may be required to set the upper rank flip-flops. For example, one of the conditions to inititate an I-sequence (set I flip-flops 0XG20) is to not be in a Wait and R1-sequence. The high from not being in Wait is applied to 21G20 for a partial enable and is a disable to 20G27. Not being in an I-sequence provides the other disable for 20G27. Gates 20G23 and 20G22 are disabled by not being in I or R1-sequence. The low outputs from 20G27, 20G22, and 20G23 enable 20G20. The high out of 20G20 provides the other enable for 21G20, and at time T4.2, I initial flip-flop 0XG20 is set and the low output of the set side is sent to the set side of lower rank flip-flops 1XG20 (refer to plate P-13) and to the sequence output gate 03G20 (refer to plate P-15). I/O and I/O2 flip-flops 0XG25 and 0XG26 are set by the low output from the I/O and I/O2 flip-flops in the lower-rank sequence designator (refer to plate P-13) and by the low output of 22E20.

2-150. The W-Sequence flip-flop (refer to plate P-12) is set from an I-sequence at time T4.2 in the following manner: L $\Rightarrow$ I-Seq$\cdot\overline{\text{Format II}}$ is ANDed with either a L $\Rightarrow$ f = 72, 74-76, or a L $\Rightarrow$ f = 40-47 by AND-OR gate 20G23 which produces a high. This gate can also be qualified by a L $\Rightarrow$ I-Seq and an L $\Rightarrow$ f = 50:44 or a L $\Rightarrow$ f 30, 31, 57, 76 ANDed with a L $\Rightarrow$ R1 Seq. The high from 20G23 is inverted through 21G23 and ANDed with a low from 12E20 (T42$\cdot\overline{\text{I/O}}\cdot\emptyset$2) to set 0XG23.

2-151. The R1-Seq flip-flop is set at time T4.2. A L $\Rightarrow$ I-Seq$\cdot\overline{\text{Format II}}$ is ANDed with a L $\Rightarrow$ f = 02-33, 50-57 by 20G22 (refer to plate P-12) and its output inverted by 21G22. Lows implying I/O, T42 and $\emptyset$2 are ANDed by 12E20. Its output and that of 21G22 are ANDed to set 0XG22.

2-152. For 20-23 instructions (double length Add and Subtract, normal and B-sensitive) and R1/R2-Sequence is initiated from an R1-sequence and is effected by setting both the R1 and R2 flip-flops. The absence of a H $\Rightarrow$ R2-Seq is ANDed with a L $\Rightarrow$ R1-Seq from the lower rank of the sequence designator and a L $\Rightarrow$ f = 20-23 by 20G22 and 20G24. The outputs from these gates are inverted by 21G22 and 21G24, respectively, then ANDed with the output from 12E20 (T42$\cdot\overline{\text{I/O}}\cdot\emptyset$2) to set the R1 and R2 flip-flops 0XG22 and 0XG24.

2-153. Lower-Rank Sequence Designator. The lower-rank sequence designator is composed of several sequencing flip-flops and associated set and clear circuitry. Each flip-flop, except the I/O1 flip-flop, has as one of its set conditions an input from an associated set and clear circuitry. Each flip-flop, except the I/O1 flip-flop, has as one of its set conditions an input from an associated flip-flop in the upper rank. Output signals from the lower rank are then applied as enables to set the flip-flop in the upper rank, which designates the next series of operations. The final flip-flops in the lower-rank sequence designator are cleared at time T2.1 and are enabled to be set at time T2.2.

2-154. The final flip-flop, set at time T2.2, corresponds to the initial flip-flop which has been set at time T4.2. The setting of the initial flip-flop is determined by the instruction and its use of the main timing cycle.

(Refer to table 2-18 for the conditions required to set the initial sequence flip-flops.)

2-155. Each main timing cycle is under control of a sequence. The initial sequence flip-flops change (advance) their configuration once every main timing cycle. The final flip-flops follow the initial flip-flops. Each sequence has a duration of one main timing cycle (two microseconds).

2-156. For a detailed order of events occurring during a given instruction, refer to the Timing Chart for that particular instruction. The following paragraphs describe an application of the lower-rank sequence designators (refer to plate P-13).

2-157. The low from 03T22, and being in Run, and not I/O and not Hold 1 at time T2.1, enables 30E20. The high output of 30E20 will clear, via 31E20, all lower rank flip-flops except I/O 1 and I/O 2. At clock Ø1, these flip-flops are cleared at the same time via gate 00E25. These same conditions at time T2.2 provide a partial enable to set all flip-flops except I/O1 and I/O2. If one of the upper rank flip-flops has been set, such as the I flip-flop, the low output of the upper rank flip-flop will set I flip-flop 1XG20. The output from 1XG20 is applied as enables to set the upper rank flip-flops, which designates the next series of operation. I/O 1 and I/O 2 flip-flops will be set at time T4.2 if an I/O REQ is received and other enables are provided to set the flip-flops.

2-158. MAJOR SEQUENCE ANALYSIS. The major sequences used in the control section of the computer are the I, R, and W sequences. An analysis of these sequences is presented in the following paragraphs. The Advance-P subsequence, although not a major sequence, is presented because it is an integral part of all three major sequences. (During the I, R and W-sequence analysis the registers used in the control section will be identified as they function within the computer.)

2-159. I (Initiate) Sequence. The I-sequence is used to obtain an instruction word from memory as specified by a running program. (Refer to table 2-19 for a detailed description

TABLE 2-18. INITIAL SEQUENCE FLIP-FLOP SETTING CONDITIONS (NON-I/O)

I_i ff	setting Int_i ff
or	(Int_i ff set + Wait Seq ff clear) · (not setting $Wait_i$, W_i, and $R1_i$ ff's)
$R1_i$ ff	I_f ff set · format I · f = 02-33, 50-57
or	$R1_f$ ff set · $R2_f$ ff clear · f = 20-23
$R2_i$ ff	$R1_f$ ff set · $R2_f$ ff clear · f = 20-23
W_i ff	I_f ff set · f = 50:44
or	I_f ff set · format I · f = 40-47, + 72, 74-76
or	$R1_f$ ff set · f = 30, 31, 57, 76

NOTE: Setting time is T4.2

of the I-sequence.) This sequence is illustrated in block form by figure 2-25. In this figure the S1-register receives the address of the desired instruction word at T1.1. The I-sequence presented in the following paragraphs is a typical I-sequence and events may be added or deleted according to the requirements of the individual instruction.

2-160. A memory reference, consisting of a read and write cycle, is initiated to read the desired instruction word from memory and write it back again after having placed it into the Z1-register. From the Z1-register the word is passed through the Z-select circuits to the three registers F, KO, and D via arithmetic selector. The six most significant bits of the word are examined to determine instruction word format. If a Format I word is detected, Z_{17-12} are considered to be the function code and are placed in the F-register. If a Format II word is detected, Z_{11-6} are considered to be the function code and are placed in the F-register. The function code, now contained within the F-register, is decoded by the Function Code Translator which controls the remaining operations in executing the instruction.

2-161. The KO-register, which is in effect a counter, always receives the six least significant bits of the instruction word. The current instruction word determines whether or not the KO-register is used (see figure 2-25). The entire instruction word is passed through the Arithmetic selectors from the Z selectors. However, bits 17-12 are masked out by 0's. The remaining 12 bits are placed into the D-register. This value is now referred to as U. Depending upon the instruction word, the upper bits of the D-register contents can receive bits from either the SR- or P-register, to formulate u_{SR} or u_P. For certain function codes (see instruction repertoire) and if the SR-register is active (ACT = 1_2), SR_{3-0} are set in D_{15-12}. With possible exceptions, f (see repertoire) P_{15-12} are set in D_{15-12} if the SR-register is inactive (ACT = 0_2).

2-162. If the function code is Format I, odd, and below an octal 50, the value in D can be modified by the index register. The index register, commonly referred to as the B-register, is actually the content of eight addresses ($000001-000010_8$) in control memory. The address of a specific B-register is determined by the content of the ICR-register. Except for the value of 000_2, the value of ICR is the exact address of the B-register. The value of 000_2 specifies the B-register at address 000010_8.

2-163. When the content of the B-register is used, it is added to the D-register content and outputted as a modified operand, or the modified address of an operand, depending upon the instruction.

2-164. Advance-P Subsequence. The Advance-P subsequence (refer to table 2-20 for a detailed description of the Advance-P sequence) is used to increment the P-register. This operation sets the P-register to the next sequential address in preparation for the next sequential instruction of the program. If a Jump or Skip instruction is being executed by the program, the new address will be inserted into the P-register as specified by the Jump or Skip instruction.

2-165. R (Read) Sequence. The R-sequence (refer to table 2-21 for a detailed description) serves to take an operand from a designated memory location and place it in a specified register. (Remember that an I-sequence would have preceded the R-sequence, so that the instruction word could have been called

TABLE 2-19. I-SEQUENCE

Time Notation	Event/Description
T4.4	Clear S1. H⇒I-Seq is inverted by OR gate 05G20 (refer to plate P-22) to enable AND gate 20N12 with the absence of the H ⇒ Int Seq at T11. Gate 20N12 enables 09N12 via OR gate 08N12 at Ø4. Gate 09N12 produces a L ⇒ Clear S1 at the clear side of the Sl-register flip-flops 1XS00 through 1XS15 (refer to plates P-104 and P-105). Inhibit Cont Mem⇒Z0. AND-OR gate 10N13 (refer to plate P-23) is qualified by lows at pins 11, 12, 13 signifying T13, not Cont Data Seq, and I or I/O Seq, respectively. It inputs a high to AND gate 11N11 (refer to plate P-25), causing it to output a high to AND gates 17N11, 19N11, and 18N11, disabling them. This inhibits transfer of Control Memory to Z0.
T1.1	P ⇒ S1. The high at 20N12 (refer to plate P-22) cause OR gate 28N12 to produce a low to partially enable AND gate 29N12. The absence of L ⇒ Load Mode disables AND gate 60N12 to enable 29N12. At Ø1, 29N12 produces a L ⇒ P ⇒ S1 to load the S1-register with the contents of the P-register. Initiate Main Memory. At the same time, flip-flop 0XG80 (refer to plate P-134) is set by L ⇒ I, R1, W, or I/O, T11 and Ø1, producing L ⇒ Initiate Memory which causes inverter 74Yg2 (refer to plate P-136) to produce a high at inverter-amplifier 54MT01. From 54MT01 a L ⇒ Initiate Memory is applied to 50MT20 (refer to plate P-135) and the memory cycling is initiated. Set Increment P Flip-Flop. Simultaneously, the L ⇒ I-Seq, L ⇒ $\overline{\text{Int Seq}}$ and T11 qualifies gate 10L10 which outputs a high. The output is inverted and ANDed with a low from the DISC ADV P switch and a Ø1 to set 0XL10 and light DS7C (refer to plate P-35). Clear Z0. A L ⇒ Cont Data Seq is not present at AND gate 00N11 (refer to plate P-25) causing it to output a low which is ANDed with a Ø1 by 09N11 to produce a L ⇒ Clear Z0. Z0 is cleared every Ø1 except during Cont Data Seq.
T1.2	Set Run 2 Flip-Flop. At time T1.2, a low from the one side of the Run 1 flip-flop 0XJ10 (refer to plate P-3) is ANDed with a T13 and Ø2 to set 1XJ10, the Run 2 flip-flop.

TABLE 2-19. I-SEQUENCE (Cont)

Time Notation	Event/Description
T1.3	Clear D. At time T1.3, the 0XL10 causes OR gate 03L10 (refer to plate P-35) to output a L ⇒ Advance-P to OR gate 30N02 (refer to plate P-18). The high output of 30N02 causes OR gate 08N02 to provide a low to AND gate 09N02. Since AND gate 01N02 is disabled by the absence of L ⇒ T24 R1-Seq, 09N02 is enabled with a ø3 produces a L ⇒ Clear D to the 00D00 through 00D17 flip-flops (refer to plates P-85 and P-86). Clear W. A H ⇒ Advance-P qualifies OR gate 08N03 (refer to plate P-19) which outputs a low. This low is ANDed with a ø3 by 08N06, which outputs a H⇒Clear W via amplifier 09N06. Clear X. The low output from 08N03 is also ANDed with ø3 by 09N03 which outputs a L ⇒ Clear X to flip-flops 0XX00 to 0XX17 (refer to plates P-87 and P-88). Clear Z1. The low input from 03T14 is inverted by 00N13 (refer to plate P-23) and applied to 09N13 via 08N13. At ø3, 09N13 produces a L ⇒ Clear Z1 to flip-flops 10Z00 to 10Z17 (refer to plates P-111 through P-113). Clear F-Register. (Refer to plate P-40.) Time T14 of an I-sequence causes OR gate 05N16 to produce a low at pin 6 of 06N16. At ø3, 06N16 produces a high due to the absence of a H ⇒ Initiate Buffer. The high is amplified by OR gate 08N16 and clears the F-register flip-flops (0XF00 through 0XF06). Clear Hold 2 Flip-Flop. If the Hold 1 flip-flop 1XG38 (refer to plate P-28) is clear, its zero side outputs a low to 2XG38 which is ANDed with a ø3 and the low output of inverter 13T14. This combination of signals will clear the Hold 2 flip-flop (2XG38). Clear Parity Flip-Flop. The Parity flip-flop OXG54 (refer to plate P-30) is cleared by ANDing a L ⇒ T14 I Seq with a ø3. Clear Select Stop Flip-Flops. A H ⇒ T14 I-Seq is inverted by OR gate 05N16 (refer to plate P-40) and the L ⇒ Master Clear which it produces is ANDed with a ø3 by 00E60 (refer to plate P-31) to clear flip-flops OXG60 through OXG65. Set OXL11 Flip-Flop. The final operation at time T1.3 is the setting of flip-flop OXL11 (refer to plate P-35) by the low output of flip-flop O1L10 and a ø3.

TABLE 2-19. I-SEQUENCE (Cont)

Time Notation	Event/Description
T1.4	Set INHIBIT EAB Flip-Flop. A L ⇒ Advance-P causes OR gate 30N02 (refer to plate P-18) to output a high. This is inverted by 36N02 to produce a L ⇒ Set Inhibit EAB which is ANDed with a ∅4 to set flip-flop OXG33 (refer to plate P-27). PL → DL, PU → DU. (Refer to plate P-18.) The low from 36N02 is ANDed with a ∅4 by 38N02 to produce a L ⇒ PL → DL. Similarly, the high from 30N02 causes OR gate 37N02 to output a low which is ANDed with a ∅4 to produce a L ⇒ PU → DU at 39N02. Clear KO. Also at this time a H ⇒ T14 I-Seq qualifies OR gate 08N14 (refer to plate P-37). Its low output is ANDed at gate 09N14 with a low due to the absence of H⇒B SEQ and at ∅4 produces at AND gate 09N14 a low to clear the KO-register flip-flops (OXKOO through OXKO5). Set Parity Flip-Flop. Even or odd parity is determined by the network (refer to plate P-101) which will be specifically described during examination of the Parity instructions. If the number of ones in the arithmetic selectors is odd, a L ⇒ Parity Odd is ANDed with a L ⇒ T14 I-Seq and a ∅4 to set OXG54 (refer to plate P-30). If parity is even, the flip-flop will remain clear.
T2.1	Clear Sequence Designator (Lower Rank). The low from 03T22, being in Run and not I/O, and not Hold 1 at time T2.1, enables 30E20 (refer to plate P-13). The high output from 30E20 will clear, via 31E20, all lower rank flip-flops except I/O 1 and I/O 2. These flip-flops are cleared at the same time via gate OOE25. Clear Z0. (Refer to T1.1 and plate P-25). Z Select → Arithmetic Select. Also at time T2.1, AND-OR gate 40N01 (refer to plate P-17) is qualified by L ⇒ T24 and L ⇒ I, R1 or W-Seq. The high output from this gate is inverted and amplified through gates 48N01 and 49N01 and emerges as a L ⇒ Z → Select. Z1 → Z Select. Since the output of 91S00 (refer to plate P-107) does not imply a control or bootstrap memory address in S1 (P was gated to S1 at time T1.1), 92S00 outputs a low to AND gate 10N00. This signal is ANDed with a T24 (which begins at time T2.1) and the high output is inverted through 11N00 and amplifier 30N00 to produce a L ⇒ Z1 → Z Select.

TABLE 2-19. I-SEQUENCE (Cont)

Time Notation	Event/Description
T2.1 (cont)	Enable Main Memory → Z1. AND-OR gate 10N13 (refer to plate P-23) is disabled by the absence of L ⇒ I/O Seq, L ⇒ W-Seq · Run · NDRO Bootstrap Address, and T13 at pins 9, 5 and 11, respectively. It outputs a low which is ANDed with T22 and T24 at 11N13, which in turn supplies a low to AND gates 17N13, 18N13, and 19N13. Gates 12N13 and 13N13 are both disqualified by the absence of a L ⇒ W-Seq etc., thereby producing lows to enable 18N13 and 17N13. The three gates output L ⇒ Memory → Z1. O's → Select. At AND-OR gate 70N01 (refer to plate P-17) L ⇒ T24 I-Seq and the absence of a H ⇒ Initiate Buffer provides a high to OR gate 78N01 which via 79N01 produces a L ⇒ 0's → Select. This signal is applied only to the upper six bits of the selector. Clear Increment P Flip-Flops. Also at time T2.1, the Increment P flip-flop OXL10 (refer to plate P-35) is cleared. Pin six of 10L10 is disabled after time T1.1 causing 10L10 to output a low which at ∅1 clears OXL10.
T2.2	Clear P. Flip-flops OXP00 through OXP15 (refer to plate P-102 and P-103) have not yet been cleared and a H ⇒ Advance-P enables 08N07 via OR gate 07N07 (refer to plate P-21). At ∅2, 08N07 outputs a high via OR gate 09N07 to clear P. Adder → P. The Clear P signal is delayed a short time (inherent delay) by inverter 11N07 before being ANDed at 19N07 with a low output from 18N07 which was qualified by the same Advance-P signal as 07N07. The output of AND gate 19N07 is a L ⇒ Adder → P. Sequence Designator UR → LR. Lows implying Run · $\overline{\text{I/O}}$ · $\overline{\text{Hold 1}}$, T22 and ∅2 qualify AND gate 42E20 (refer to plate P-13) which outputs a low enabling the transfer of the sequence designation from the upper to the lower rank of the designator.
T2.3	Clear Inhibit and Insert EAB Flip-Flops. OR Gate 01E32 (refer to plate P-27) outputs a low which enables pin 6 of AND gate 02E32. This input is ANDed with ∅3 to clear the EAB flip-flops OXG33 and OXG32. Clear OXL11 Flip-Flop. Lows from the clear sides of flip-flops OXL10 and 4XL10 (refer to plate P-35) are ANDed with a ∅3 causing flip-flop OXL11 to be cleared.

TABLE 2-19. I-SEQUENCE (Cont)

Time Notation	Event/Description
T2.3 (cont)	Clear W, Clear X. A H ⇒ T24 I-Seq causes 08N03 (refer to plate P-19) to output a low which is ANDed with a ∅3 by gates 08N03 and 09N03 to produce a H ⇒ Clear W via 09N06 and a L ⇒ Clear X via 09N03. Clear Main Memory Flip-Flop. Time T24 is ANDed with a ∅3 to clear OXG80 (refer to plate P-134). Clear D. A H ⇒ T24 I-Seq causes OR gate 08N02 (refer to plate P-18) to output a low. The absence of a L ⇒ T24 R1-Seq causes 01N02 to output a low. These outputs are ANDed with a ∅3 to produce a L ⇒ Clear D from AND gate 09N02.
T2.4	Select → D. A H ⇒ T24 I-Seq qualified pin 8 of 19N02 (refer to plate P-18) via OR inverter 18N02. Pin 9 is enabled since a L ⇒ T24 R1-Seq is absent from pin 9 of AND gate 17N02, disqualifying it and causing it to output a low. A L ⇒ ∅4 then causes 19N02 to output a L ⇒ Select → D. Z SELECT 0-5 → KO. A L ⇒ $\overline{\text{I/O}}$ $\overline{\text{Seq}}$, L ⇒ T24 I-Seq and ∅4 are ANDed by 29N14 (refer to plate P-37) to gate bits 0 through 5 of the Z selectors to the KO registers, flip-flops 0XK00 through OXK05, via pins 5 and 6 of the flip-flops. Z Select → F-Register. The loading of the F-register is described in paragraph 2-160.
T3.1	ICR → SO, Initiate CM. A H ⇒ T31 I-Seq unconditionally qualifies AND gate 48N10 (refer to plate P-24). Its low output is ANDed with a ∅1 at gate 49N10 to produce a L ⇒ ICR → SO, Initiate CM (control memory). This signal sets flip-flop OXDT10 (refer to plate P-122) which initiates the control memory cycle. Clear Skip Flip-Flop. A T32 is ANDed with a ∅1 to clear the Skip flip-flop OXG50 (refer to plate P-31). Clear ZO. This was described at time T1.1. ZO → Z Select. Also at time T3.1, AND gate 10N00 (refer to plate P-107) is disabled since it no longer receives a L ⇒ T24; its low output is amplified by 20N00 to gate ZO → Z Select.

TABLE 2-19. I-SEQUENCE (Cont)

Time Notation	Event/Description
T3.1 (cont)	Z Select → Arith Select or Drop Z Select → Arith Select. Simultaneously, if the instruction in the F-register is between 00 and 47, and is odd, a L ⇒ f = 00-47. Odd is developed by gate 93F00 (refer to plate P-41) and is ANDed with a L ⇒ T34 I-Seq Format I developed by gate 11T34 (refer to plate P-15) at pins 7 and 8 of 40N01 (refer to plate P-17), causing it to output a high which is inverted and amplified by 48N01 and 49N01 respectively, producing AL ⇒ Z → Select. If the instruction is not 00-47 (FI) Odd, the signal is dropped. Drop O's → Select. Since time T24 ends as time T3.1 begins the L ⇒ T24 I-Seq is dropped from pin 7 of AND-OR gate 70N01 (refer to plate P-17) causing it to output a low, since it is also disqualified by absence of a L ⇒ T14 R1, W-Seq. This signal absence also disqualifies 90N01 and 50N01 causing them to output lows to 78N01. This OR gate is therefore completely disabled and the L ⇒ O's → Select via 79N01 is dropped. Drop Z1 → Z Select. Similarly, the running out of time T24 at time T3.1 disqualifies 10N00 (refer to plate P-107) which then outputs a low to 11N00. The L ⇒ Z1 → Z Select via 30N00 is dropped at time T3.1.
T3.2	Clear B. Time T34 is ANDed with a ∅2 by 08N08 (refer to plate P-21) to produce a H ⇒ Clear B via amplifier 09N08. f • KO → Translator. Gate 21E00 (refer to Plate P-49) is qualified by a L ⇒ I-Seq and a low from 03T32. The high output is inverted by 22E00 and at ∅2 23E00 outputs a L ⇒ f · KO → XLATOR.
T3.3	Clear D. (Refer to plate P-18). The signal L ⇒ Clear D will be generated by ANDing a L ⇒ T34 I-Seq Format II and L ⇒ f = 50:21-23, 50-55, 57 at 30N02. A high from this gate qualifies 08N02 and its output is ANDed with a low from 01N02 (disabled by lack of a L ⇒ T24 R1 Seq) and ∅3 by 09N02 to produce a L ⇒ Clear D. Clear X. (Refer to plates P-18 and P-19). The H ⇒ Advance P from 30N02 is applied to 08N03 (refer to plate P-19) causing it to output a low which is ANDed with a ∅3 by 09N03 to produce a L ⇒ Clear X.
T3.4	PL → DL, PU → DU. (Refer to plate P-18). The high output of 30N02 is inverted by 36N02 and ANDed with a ∅4 to produce a L ⇒ PL → DL. Similarly, OR inverter 37N02 is enabled by 30N02 and outputs a low which is ANDed with a ∅4 to produce a L ⇒ PU DU.

TABLE 2-19. I-SEQUENCE (Cont)

Time Notation	Event/Description
T3.4 (cont)	Set Inhibit EAB. (Refer to plates P-18 and P-27.) The (L ⇒ Set Inhibit EAB) low from 36N02 (refer to plate P-18) is ANDed with a ∅4 to set the Inhibit EAB flip-flop OXG33 (refer to plate P-27). ZO → B. AND gate 10N08 (refer to plate P-21) is disabled by the lack of a L ⇒ I/O Seq or ESA Mode and outputs a low which is ANDed with T34 and ∅4 by 29N08 to produce a L⇒ ZO → B. SR or PU → DU. A L ⇒ T34 I-Seq Format I, plus L ⇒ SR Active, plus L ⇒ f = 00-47, plus L ⇒ f = 30, 31, 34-37, plus L⇒ Int Seq (or B-Seq, or I/O, or Hold 2) qualify AND gate 40N02 (refer to plate P-18). It outputs a high which is inverted through OR gate 48N02 and applied to pin 5 of 49N02. This is ANDed with a ∅4 (time T3.4) to output a L ⇒ SR + Set → D. This signal enables the transfer of a SR modification to the address being formed in the D-register. If SR is not active, the absence of a L ⇒ SR Active disables 40N02, causing it to output a low. This low is ANDed with some inputs of 40N02, plus a L ⇒ f = 36, 37, 70, 71, to qualify 31N02. The high output of 31N02 is inverted through OR gate 37N02 and ANDed with a ∅4 by 39N02 to provide a L ⇒ PU → DU. This signal is used in lieu of an SR modification, loading the upper six bits of the P-register into the address being formed in the D-register. Select → X. A L ⇒ T34 I-Seq Format I qualifies OR gate 10N03 (refer to plate P-19), causing it to output a high which enables OR gate 18N03. A low output from 18N03 is ANDed with a L ⇒ ∅4 by 19N03 to produce a L ⇒ Select → X.
T4.1	Clear Sequence Designator (UR). Time T42 causes 00E20 (refer to plate P-12) to output a high which is inverted through OR gate 01E20 and applied to 08E20. It is ANDed with a ∅1 and clears via 09E20 the upper rank of the Sequence Designator (flip-flops 0XG20 through 0XG27, and 6XG20). Clear ZO Register. (Refer to T1.1 and plate P-25.) Clear B ± 1 Flip-Flop. Time T42 is ANDed with a ∅1 to clear OXG37 (refer to plate P-27). Clear Comparison Designators. If the contents of the F-register are not 60-67 (format I or II), AND gate 00E34 (refer to plate P-29) will be qualified by lows signifying T42 I-Seq, ∅1, and f = $\overline{60\text{-}67}$ (or $\overline{50:60\text{-}50:67}$). The gate will output a high, clearing OXG34 through OXG36, the Equal, Greater, and Compare flip-flops.

TABLE 2-19. I-Sequence (Cont)

Time Notation	Event/Description
T4.1 (cont)	Drop Z → Select. The signal L ⇒ Z → Select developed by 49N01 (refer to plate P-17) at time T3.1 is dropped. The signal L ⇒ T34 I-Seq Format I becomes a high with the end of time T3.4, thereby disabling AND-OR gate 40N01.
T4.2	Clear A Neg and Y Neg Flip-Flops. A L ⇒ Run is ANDed with a L ⇒ I-Seq by 13G20 (refer to plate P-15) and its output is ANDed at time T4.2 by 11T42 to produce a L ⇒ Clear A Neg and a L ⇒ Clear Y Neg. These signals are ANDed with ∅2 to clear flip-flops OXG30 and 1XG30 (refer to plate P-33). Clear HOLD 1 Flip-Flop. If the Clear Hold flip-flop OXG38 (refer to plate P-28) is set, its output is ANDed with a T42 and ∅2 to clear the Hold 1 flip-flop 1XG38. Set RUN Flip-Flop. Refer to Start Sequence. Set Sequencer (UR). The operational sequence (refer to plate P-12) for the sequence designator is set. With a L ⇒ $\overline{I/O}$ on pin 9 of gate 12E20, and gate 03T42; 12E20 is enabled. This low is applied to flip-flops OXG20 through OXG24, and 6XG20 and 4XG20. The other low necessary to enable the set side of the flip-flops is from the various enabling gates, depending on the desired sequence.
T4.3	Clear EAB Flip-Flops. OR GATE 01E32 (refer to plate P-27) outputs a low due to H ⇒ T43 on pin 7. This low is ANDed with a ∅3 to clear flip-flops OXG33 and OXG32, the Inhibit and Insert EAB flip-flops.

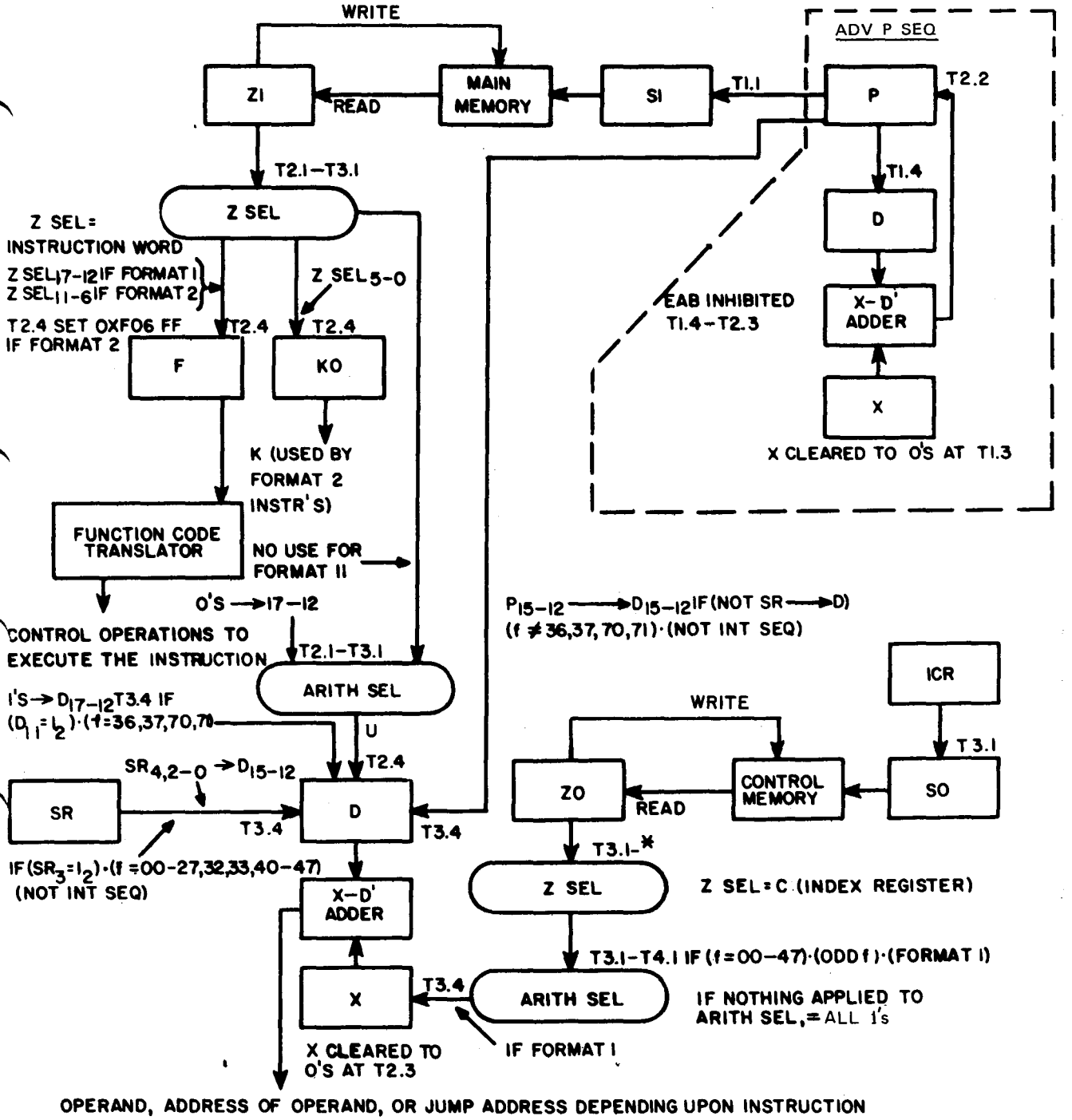


Figure 2-25. I-Sequence Data Flow

TABLE 2-20. ADVANCE-P SUBSEQUENCE

Time Notation	Event/Description
T1.1	Set Increment P Flip-Flop. (Refer to plate P-35) A L ⇒ I-Seq, L ⇒ $\overline{\text{INT Seq}}$ and T11 qualify AND/OR gate 10L10 which outputs a high. The output is inverted and ANDed with a low from the DISC ADV P switch and a Ø1 to set 0XL10.
T1.3	Clear D. A high from the clear side of 0XL10 (refer to plate P-35) causes OR gate 03L10 to output a L ⇒ Advance P to OR gate 30N02 (refer to plate P-18). The high output of 30N02 causes OR gate 08N02 to provide a low to AND gate 09N02. Since AND gate 01N02 is disabled by the absence of L ⇒ T24 R1-Seq, 09N02 is enabled and with Ø3 produces a L ⇒ Clear D to the 00D00 to 00D17 flip-flops (refer to plates P-85 and P-86). Clear X. (Refer to plate P-18.) A H ⇒ Advance P qualifies OR gate 08N03 (refer to plate P-19) which outputs a low. This low is ANDed with a Ø3 by 09N03 which outputs a L ⇒ Clear X to 0XX00 - 0XX17 flip-flops (refer to plates P-87 and P-88). Set 0XL11 Flip-Flop. (Refer to plate P-35.) A Ø3 is ANDed with the low from the set side of 0XL10 to set OXL11.
T1.4	Set Inhibit EAB Flip-Flop. A L ⇒ Advance P causes OR gate 30N02 (refer to plate P-18) to output a high. This is inverted by 36N02 to produce a L ⇒ Set Inhibit EAB which is ANDed with a Ø4 to set flip-flop OXG33 (refer to plate P-27). PL → DL, PU → DU. (Refer to plate P-18.) The low from 36N02 is ANDed with a Ø4 by 38N02 to produce a L ⇒ PL → DL. Similarly, the high from 30N02 causes OR gate 37N02 to output a low which is ANDed with a Ø4 to produce L ⇒ PU → DU at 39N02.
T2.1	Clear INCREMENT P Flip-Flop. (Refer to plate P-35.) Pin six of 10L10 is disabled after time T1.1 runs out at the beginning of time T2.1 causing 10L10 to output a low which, when ANDed with Ø1, clears 0XL10.
T2.2	Clear P. (Refer to plates P-35, P-21.) At this time T2.2 flip-flop OXL11 has not yet been cleared and a H ⇒ Advance P enables 08N07 via OR gate 07N07 (refer to plate P-21). At Ø2, 08N07 outputs a high via OR gate 09N07 to clear P.

TABLE 2-20. ADVANCE-P SUBSEQUENCE (Cont)

Time Notation	Event/Description
T2.2 (cont)	Adder → P. (Refer to plate P-21.) The Clear P signal is delayed a short time by inverter 11N07 before being ANDed at 19N07 with a low output from 18N07 which was qualified by the same Advance P signal as 07N07. The output of AND gate 19N07 is a L ⇒ Adder → P.
T2.3	Clear EAB Flip-Flops. (Refer to plate P-27.) At time T2.3 OR gate 01E32 outputs a low which enables pin 6 of AND gate 02E32. This input is ANDed with ∅3 to clear the EAB flip-flops 0XG33 and 0XG32. Clear 0XL11 Flip-Flop. (Refer to plate P-35.) At this time, lows from the clear sides of flip-flops 0XL10 and 4XL10 are ANDed with a ∅3 and flip-flop 0XL11 is cleared.
T3.1	Set Increment P Flip-Flop. (Refer to plate P-35.) OR GATE 10L10 can be enabled at T31 by one of two methods. During an R1 SEQ a L ⇒ R1 SEQ, L ⇒ f = 56, L ⇒ T31 will enable 10L10. The high from 10L10 is inverted by 11L10 and together with a low from DISC ADV P switch and a ∅1 sets 0XL10. 10L10 can also be enabled during a W SEQ. A L ⇒ W SEQ, L ⇒ f = 30, 31, 57, 76, L ⇒ T31 and a low from 12L10 which is disabled if f ≠ 57 or D = X' will enable 10L10 and set 0XL10 as before.
T3.3	CLEAR D. See T1.3. CLEAR X. See T1.3. SET 0XL11 FLIP-FLOP. See T1.3.
T3.4	SET Inhibit EAB FLIP-FLOP. See T1.4. PL → DL, PU → DU. See T1.4.
T4.1	Clear Increment P Flip-Flop. (Refer to plate P-35.) Pins 9 and 12 of 10L10 are disabled after time T31 runs out. The low from 10L10 is ANDed with the next ∅1 to clear 0XL10.
T4.2	Clear P. See T2.2. ADDER → P. See T2.2.
T4.3	Clear EAB Flip-Flops. (Refer to plate P-27.) At time T43 or gate 01E32 will output a low which partially enables 02E32. This low is ANDed with a ∅3 to clear the EAB Flip-Flops 0XG33 and 0XG32. Clear 0XL11 Flip-Flop. See T2.3.

TABLE 2-21. R-SEQUENCE

Time Notation	Event/Description
T4.4	Clear S1. The low input to 05G20 (refer to plate P-22), due to the absence of an I-sequence and not Function Repeat, is inverted and disables 20N12, providing a low to 10N12. Other inputs to 10N12 are a L ⇒ R1 W Seq, L ⇒ $\overline{\text{R2-Seq}}$, a low from 03T11, and a low from 12F57, which is disabled by the absence of a W-sequence or f = 57. The high from 10N12 is inverted by 08N12 and at ϕ4, 09N12 outputs a L ⇒ Clear S1.
T1.1	Adder → S1. The high from 10N12 (refer to plate P-22) is inverted by 18N12 and applied to 19N12. Gate 70N12 outputs a low because of the absence of a proper instruction or sequence. This low is applied to 19N12 and at ϕ1, 19N12 outputs a L ⇒ Adder → S1. Initiate Memory. Same as I-sequence. Clear Z0. Same as I-sequence. AL → Arith Sel. Gate 20N01 (refer to plate P-17) is enabled by a L ⇒ T14 → R1, W-Seq and a L ⇒ f = 02, 03, 06, 07, 14 - 17, 20 - 23·$\overline{\text{R2}}$, 24, 25, 44, 45, 53, 74, and outputs a high which is inverted by 28N01. Gate 90N01 is disabled by the absence of an f = 50:44, and the low output into 29N01 produces a L ⇒ AL → Select. For the instructions described except 06, 07, the AU → Sel is dropped at T2.1. AU → Arith Sel. Gate 10N01 (refer to plate P-17) is enabled by a L ⇒ T14 R1, W-Seq and a L ⇒ f = 06, 07, 20 - 23 · R2-Seq; 46, 47 or a L ⇒ f = 04, 05 or a L ⇒ f = 26, 27. The high is inverted by 18N01 and gate 19N07 produces a L ⇒ AU → Select. For a 74 instruction gate 70N01 is enabled and only gate 79N01 produces a L ⇒ 0's → Select. If f = 06, 07, 10N01 is kept enabled by L ⇒ T24 R1 Seq and L ⇒ f = 06, 07. O's → Arith Sel. Gate 50N01 (refer to plate P-17) is enabled by a L ⇒ T14 R1, W-Seq, L ⇒ f = 40, 41, 72, 75, and the high is inverted by 78N01 and 58N01, and gates 79N01 and 59N01 produce a L ⇒ O's → Select.

TABLE 2-21. R-SEQUENCE (Cont)

Time Notation	Event/Description
T1.3	Clear X, W. (Refer to plate P-19.) Gate 00N03 is enabled by a L ⇒ f ≠ 57 and a L ⇒ T14 R1-Seq. The high from 00N03 enables 08N03; the low from 08N03 enables 09N03 and 08N06 during a ∅3; 09N03 outputs a L ⇒ clear X. The output of 08N06 is amplified by 09N06 which outputs a H ⇒ clear W. Clear D. Gate 20N02 (refer to plate P-18) is enabled by a L ⇒ 04, 05, 24-27, 53, T14 R1, W-Seq to produce a high at 08N02 which is inverted and applied to 09N02. Gate 01N02 is disabled by a L ⇒ T24 R1-Seq to provide a low to 09N02 and at ∅3, 09N02 produces a L ⇒ Clear D. Clear Z1. Same as I-sequence.
T1.4	Arith Sel → X. The low inputs which enable gate 10N03 (refer to plate P-19) are L ⇒ T14, R1, W-Seq and L ⇒ f ≠ 24-27. The high from 10N03 partially enables 19N03 by OR inverter 18N03 and with a ∅4, 19N03 outputs a L ⇒ Select → X. Arith Sel' → D. Gate 20N02 (refer to plate P-18) is enabled by a L ⇒ T14 R1, W-Seq and a L ⇒ f = ∅4, ∅5, 24-27, 53. The high is inverted by 28N02 and applied to 29N02. Gate 27N02 is disabled by the absence of a L ⇒ T24R1 Seq and at ∅4, 29N02 produces a L ⇒ $\overline{\text{Select}}$ → D. Clear KO. Same as I-sequence except 08N14 (refer to plate P-37) has as its input H ⇒ T14 R1 Seq.
T2.1	Z1 → Z Sel and Z Sel → Arith Sel. Same as I-sequence. Set KO to 22 or 23. Gate 39N14 (refer to plate P-37) is enabled by a L ⇒ T22 R1-Seq and a ∅1, outputs a low to OXK00, OXK01, and OXK04. If f ≠ X4 or X4 or X5, OXK00 is not set and KO = 22. If f = X4 or X5, OXK00 is set and KO = 23. Drop AU → Arith Select. If f ≠ 06, 07 see T1.1. Drop AL → Arith Select. If f ≠ 06, 07 see T1.1.

TABLE 2-21. R-SEQUENCE (Cont)

Time Notation	Event/Description
T2.3	Clear D. Gate 01N02 (refer to plate P-18) is disabled by the absence of an f = ∅4, ∅5, and 53 to produce a low at ∅9N02. Gate 10N02 produces a high by the L ⇒ T24 R1-Seq and a L ⇒ f = ∅4, ∅5, 1∅-15, 20, 21, 24-27, 30-47, 51, 52, 54, 55, 57. The high is inverted by 08N02 and at ∅3, 09N02 is enabled to produce a L ⇒ Clear D. Clear Inhibit and Insert EAB ff. Same as I-sequence.
T2.4	Arith Sel → D. The high out of 10N02 (refer to plate P-18) is inverted by 18N02 to produce a low at 19N02. Gate 17N02 is disabled by the absence of an f = 24-27 to produce a low at 19N02, and at ∅4, 19N02 produces a L ⇒ Select → D. Arith Sel' → D. Gate 20N02 (refer to plate P-18) is enabled by a L ⇒ f = 02, 03, 06, 07, 16, 17, 22-27, 53, 56 and a L ⇒ T24 R1-Seq. The high output is inverted by 28N02 and applied to 29N02. Gate 27N02 is disabled by the absence of an f = 24, 25, 26, or 27 instruction and Y Pos or Y Neg and gate 29N02 produces L ⇒ $\overline{\text{Select}}$ → D.
T3.2	Enable Compare, Greater and Equal Flip-Flops. Gate 10E34 (refer to plate P-29) produces a low output because of a L ⇒ R1-Seq, ∅2 and a low from 01T32. The low output is applied to flip-flops OXG35, OXG36, and OXG34, which will be set when a low is applied to the other inputs. Clear B. Same as I-sequence.
T3.4	ZO → B. Same as I-sequence.
T4.1	Clear AL. Gate 10N05 (refer to plate P-20) is enabled by a L ⇒ 04, 05, 12-17, 20-23 · $\overline{\text{R2-Seq}}$, 51-53, and a L ⇒ T42 R1 W-Seq. The high output is inverted by 07N05 and applied to 08N05. Gate 00N05 outputs a low because of the absence of a f = 51 instruction and at ∅1, 08N05 outputs a H ⇒ Clear AL via 09N05.
T4.2	Adder → AL. The high out of 10N05 is inverted by 18N05 and at ∅2, 19N05 produces a L ⇒ Adder → AL.

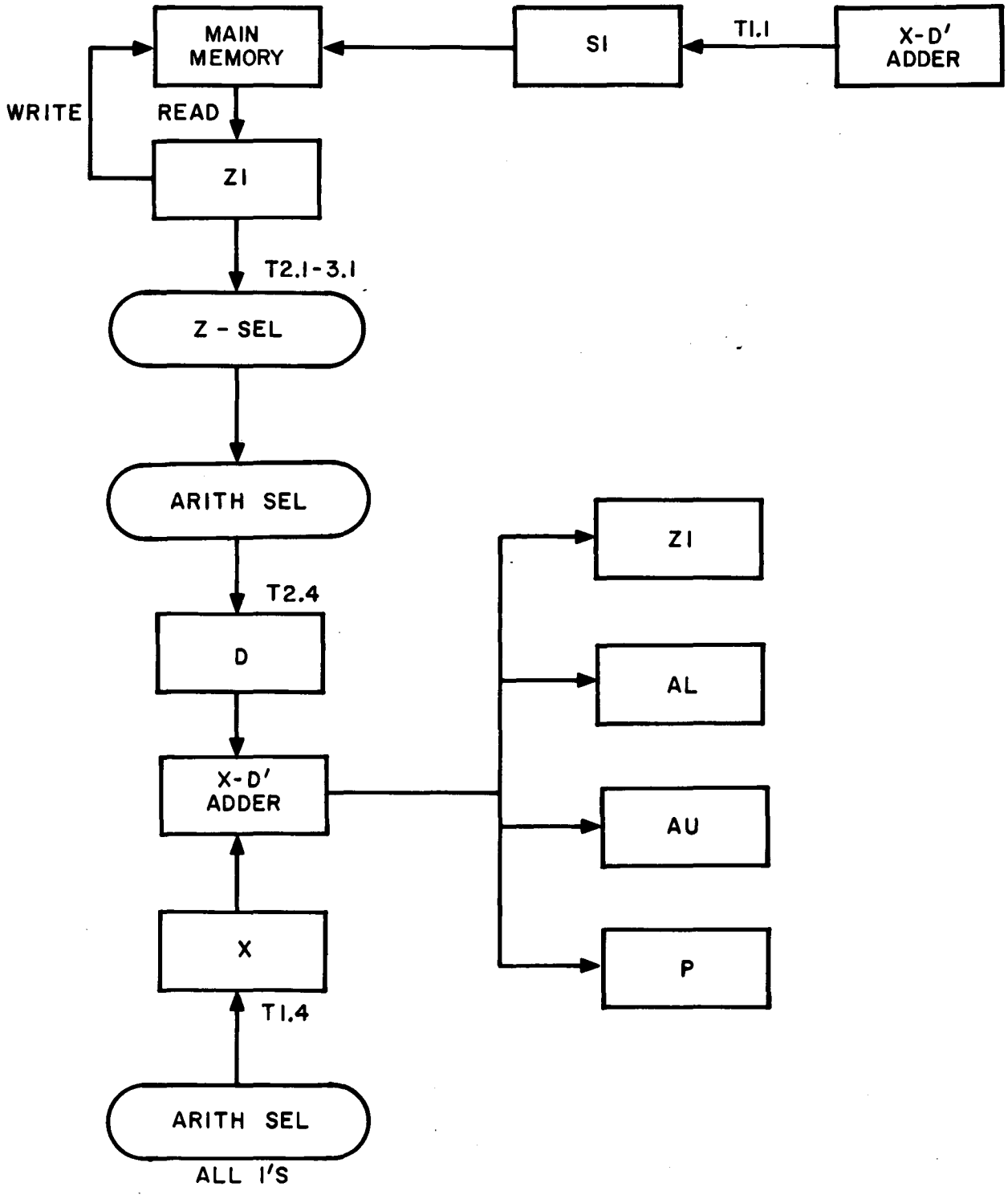


Figure 2-26. Basic R-Sequence

out of memory and initially acted upon.) A basic R-sequence is illustrated by figure 2-26.

2-166. The operand address, formulated during the preceding I-sequence, is placed into the S1-register. A memory cycle is initiated and the addressed operand is called out of memory and placed in the Z1-register. After being fed through the Z and arithmetic selectors the operand is placed in the D-register. Notice that the D-register feeds one side of the adder, the other side being fed by the X-register, previously set to all 1's. The adder effectively performs a X-D' subtraction and the results of the subtraction are transferred to the register designated by the instruction being executed. For example, if address 00600 contained the operand 351725, it would be placed in the S1-register and 351725 would be called out of memory and transferred to the D-register. Since the X-register has previously been set to all 1's, when the X-D' subtraction occurs, the operand is developed as follows:

X = 777777 (all 1's)

D' = 426052 (complemented operand)

351725 (operand)

Thus the operand is made available for transfer to the register as designated by the instruction: AU, AL, Z1, or P.

2-167. W (Write) Sequence. The W-sequence (refer to table 2-22 for a detailed description) is used to write new data into a designated memory location. The data previously stored at that location is destroyed during this sequence. An I-sequence would have preceded the W-sequence to allow the instruction word to be called out of memory. A basic W-sequence is shown in figure 2-27.

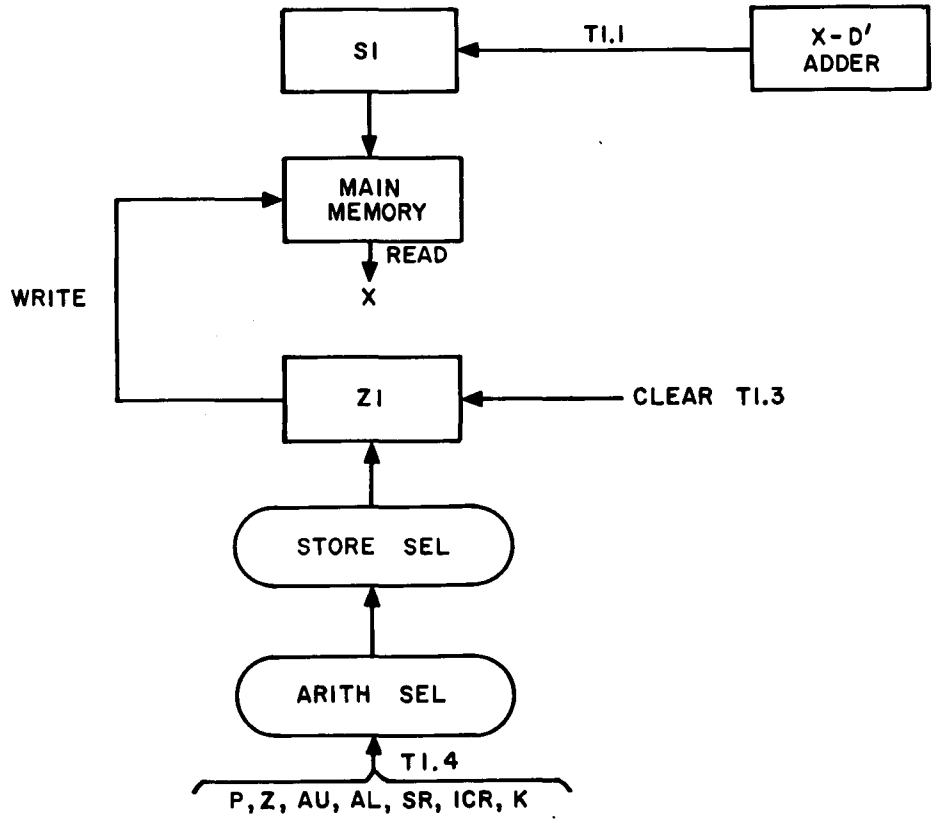


Figure 2-27. Basic W-Sequence

TABLE 2-22. W-SEQUENCE

Time Notation	Event/Description
T4.4	Clear S1. Same as R1-sequence.
T1.1	Adder → S1. Same as R1-sequence.
	Initiate Memory. Same as I-sequence.
	Clear Z0. Same as I-sequence.
	O's → Arith Sel. (Refer to plates P-17, P-82 and P-84.) A L⇒ T14 R1, W-Seq is applied to AND gates 90N01, 50N01 and to AND-OR gate 70N01. If f = 50:44, gate 90N01 is qualified and outputs highs to OR inverters 78N01 and 58N01. They output lows via 79N01 and 59N01, which gate zeros to bits 06 through 17 of the arithemtic selectors (refer to plates P-82 through P-84). If f = 40, 41, 72, 75, the AND gate 50N01 outputs the highs to 78N01 and 58N01. If f = 74, only 78N01 is enabled by AND-OR gate 70N01 and zeros are gated only to bits 12 through 17 of the selectors.
	P → Arith Sel and O's → Arith Sel 17-16. (Refer to plates P-17, P-82 and P-84.) If f = 30, 31, 76, the L⇒ T14 R1, W-Seq is ANDed with a L ⇒ W-Seq, and a L⇒ f = 30, 31, 76 by 30N01 which produces a P → Select signal via 38N01 and 39N01. Note that this enable is ANDed with the contents of P only in bits 00 through 15 and that it applies a zero directly to 13X17 and 13X16 (refer to plate 9-84).
T1.3	Clear Z1. Same as I-sequence.
T1.4	Store Sel → Z1. (Refer to plate P-23.) AND-OR gate 20N13 produces a high because of a L ⇒ W-Seq·Run · $\overline{\text{NDRO Boot-strap address}}$ and a low from 03T14. The high is inverted by 28N13 and applied to 29N13. Gate 30N013 outputs a low due to the absences of T44 and f = 57 and at Ø4, 29N13 produces a L ⇒ Store Select → Z1.
	Arith Sel → Store Sel. (Refer to plates P-108 through P-110.) This signal is developed throughout the W-sequence by a L⇒

TABLE 2-22. W-SEQUENCE (Cont)

Time Notation	Event/Description
T1.4 (cont)	$\overline{I/O}$ which is ANDed with the output of the arithmetic selectors. $\overline{I/O}$ effectively gates Arith Sel → Z1 when the Store Sel → Z1 signal is produced. Enable, Disable (Parts of) Memory → Z1. (Refer to plate P-23.) AND-OR gate 10N13 is qualified by a L ⇒ f = 30, 31, 40-47, 57, 76, a L ⇒ W-Seq•Run • $\overline{\text{NDRO Bootstrap address}}$ and a L ⇒ T23, to output a high which disables 11N13, causing it to output a high to disqualify 17N13, 18N13, and 19N13. If f = 72, 74, 75, 76, AND gate 13N13 disables only 17N13 (lower six bits). If the instruction is a f = 74, AND gate 12N13 outputs a high which disables 18N13 (middle six bits). In these cases, 10N13 outputs a low to 11N13 where it is ANDed with a T22 and T24, outputting a low enable to 19N13, 18N13, and 17N13 beginning at time T2.1.
T2.1	Drop O's → Arith Sel. (Refer to plate P-17.) Time T1.4 ends with the beginning of time T2.1 so the L ⇒ T14 R1, W-Seq no longer enables gates 90N01, 50N01, 70N01. Drop P → Arith Sel and O's → Arith Sel 17-16. (Refer to plate P-17.) Without the L ⇒ T14 R1, W-Seq gate 30N01 no longer outputs the necessary enable. Z → Select. Same as I-Sequence.
T2.2	Clear P, S1 → P. (Refer to plate P-21.) A combination of T22, W-Seq and f = 30, 31, 76 qualifies AND-OR gate 20N07. The high it outputs is inverted through 07N07 and 28N07. From 07N07, the signal is ANDed with a $\emptyset 2$ by 08N07 to produce a H ⇒ Clear P via 09N07. The high is inverted and delayed slightly through 11N07 and applied to 29N07 where it is ANDed with the output of 28N07. This gate then outputs a L ⇒ S1 → P.
T2.3	Drop Enables (Parts of) Mem → Z1. (Refer to plate P-23.) When T22 ends, AND gate 11N13 is disabled and no longer outputs a low enable to 17N13, 18N13, and 19N13. Clear EAB Flip-Flops. OR gate 01E32 (refer to plate P-27) outputs a low due to the high from 00T23. This low, along with a $\emptyset 3$, enables 02E32 which outputs a high to clear OXG32 and OXG33, the Inhibit and Insert EAB flip-flops.

2-168. The memory location address where the new data will be stored was developed during the I-sequence preceding this W-sequence. At T1.1, this address is transferred to the S1-register and the old data contained in the address is read out of memory. Notice that the old data is not allowed to enter the Z1-register. It is effectively destroyed. The new data, contained in the P, Z, AU, AL, SR, ICR and K-registers, is passed through the arithmetic and store selectors into the Z1-register. During the write portion of the memory cycle, the new data is written into memory at the address specified by the contents of the S1-register.

2-169. COMMAND INSTRUCTIONS. The command instructions generated by the control section are used to order the computer to stop, add data, store data, jump addresses, or skip addresses. (Table 2-23 lists the command instructions generated by the control section.) Most of these instructions use some part of the major sequences described previously. Therefore, only those portions of the instructions not previously described will be described in detail in the following paragraphs.

2-170. Execution of Stop Instruction (f = 50:56). The STOP instruction (f = 50:56) allows the program to stop the computer either conditionally or unconditionally. The k portion of the instruction word determines which type of stop will be executed. If bit 4, 3, 2, 1 or 0 of k is a one, and the corresponding console STOP key is set, a conditional stop will be executed. If bit 5 of k is a one, the computer will stop unconditionally, regardless of the position of the console STOP keys. The computer is stopped by disabling control of all sequences except I/O. Since Main Timing will not be inhibited, I/O operations can continue without interruption.

2-171. I-Sequence Data Flow for f = 50:56. Refer to figure 2-28 for a block diagram description of the execution of STOP (f = 50:56), and table 2-24 for a list of I-sequence essential commands during the execution of STOP. A STOP instruction is completed during the I-sequence. Most of the I-sequence operations are previously described for a typical I-sequence. As shown in figure 2-28, at time T2.4, the KO-register receives the k portion of the instruction word. Whether or not the computer will stop depends upon which bits of k are ones, and which console STOP keys are set. (Refer to plate P-32 for the logic governing the console STOP keys.) For example, if STOP key 4 were set, 20G64 would output a low, which would be felt on pin 6 of OXG64 (refer to plate P-31). If bit 4 of k were a one, pin 7 of OXG64 would be low. When 10E60 is enabled by L ⇒ f = 50:56, L ⇒ T34 I-Seq Format II, and Ø4, it will produce a low to set OXG64, thereby lighting STOP 4 lamp on the console. A high from the clear side of OXG64 is produced by 11G60 via OR gate 10G60. This high is felt on pin 15 of 25J10 (refer to plate P-3) which outputs a low, clearing the Run 1 flip-flop at time T4.1 (Ø1 of T42).

2-172. Execution of ENTALK, ADDALK Instructions (f = 70, 71). The ENTALK (f = 70) and ADDALK (f = 71) instructions are used to enter or add a constant to the AL-register. If f = 70, the lower 12-bits of the instruction word (U) is extended to an 18-bit word (XU) and placed in the AL-register. The original contents of the AL-register are destroyed in the process. If f = 71, an 18-bit word is formulated as before, but this value is now added to the value already contained in the AL-register.

TABLE 2-23. CONTROL SECTION COMMAND INSTRUCTIONS

Format Code	Instruction
ENTER (MEMORY → REGISTER)	
10	ENTAU
11	ENTAUB
12	ENTAL
13	ENTALB
32	ENTB
33	ENTBB
ENTER (NON-MEMORY)	
36	ENTBK
37	ENTBKB
70	ENTALK
50:72	ENTICR
50:73	ENTSR
CLEAR	
40	CL
41	CLB
STORE (REGISTER → MEMORY)	
42	STRB
43	STRBB
44	STRAL
45	STRALB
46	STRAU
47	STRAUB
72	STRICR
74	STRADR
75	STRSR
ADD (AL +XU → AL)	
71	ADDALK
STOP	
50:56	STOP

Format Code	Instruction
SKIP	
56	BSK
57	ISK
50:50	SKP
50:51	SKPNBO
50:52	SKPOV
50:53	SKPNOV
50:54	SKPODD
50:55	SKPEVN
JUMP (DIRECT)	
34	JP
35	JPB
60	JPAUZ
61	JPALZ
62	JPAUNZ
63	JPALNZ
64	JPAUP
65	JPALP
66	JPAUNG
67	JPALNG
73	BJP
JUMP (INDIRECT)	
54	IJPEI
55	IJP
JUMP (DIRECT RETURN)	
76	RJP
JUMP (INDIRECT RETURN)	
30	IRJP
31	IRJPB

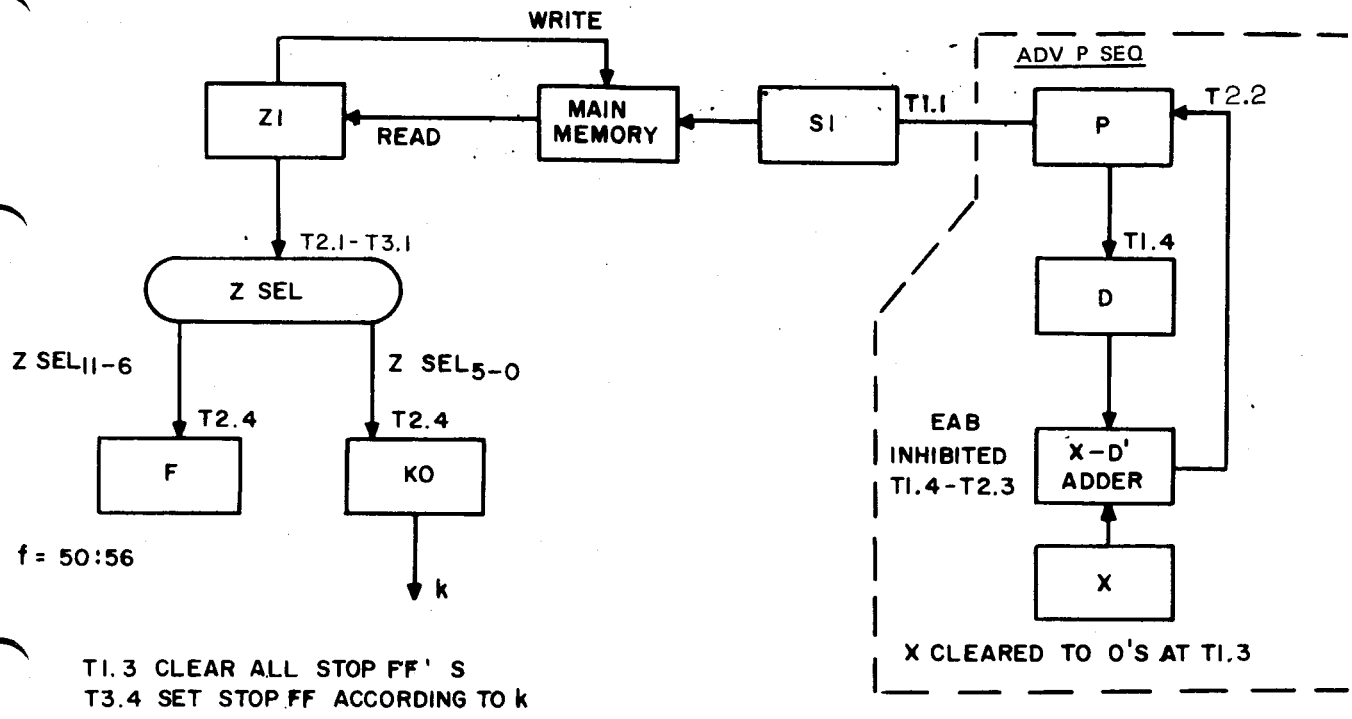


Figure 2-28. I-Sequence Data Flow for f = 50:56

TABLE 2-24. I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:56

Time Notation	Commands
T4.4	Clear S1
T1.1	P → S1, Init Memory, *set Incr P ff
T1.3	*Clear D, *clear X, clear F, clear Z1, *set OXL11 ff, clear all Stop ff's
T1.4	*P_L → D_L, *P_U → D_U, clear KO, *set Inhib EAB ff
T2.1	Z1 → Z Sel, *clear Incr P ff
T2.2	Clear P, Adder → P
T2.3	*Clear OXL11 ff, *clear Inhib EAB ff
T2.4	Z Sel_{11-6} → F, set 0XF06 ff, Z Sel_{5-0} → KO
T3.4	Set stop ff corresponding to KO bit - 1_2 and selected STOP switch
T4.1	Clear run 1 ff if any stop ff set

*These events are concerned with or are controlled by the Advance-P subsequence. Refer to Table 2-19 for a detailed description of the I-Sequence essential commands. For a description setting the Stop ff and clearing run 1 ff, refer to paragraph 2-171.

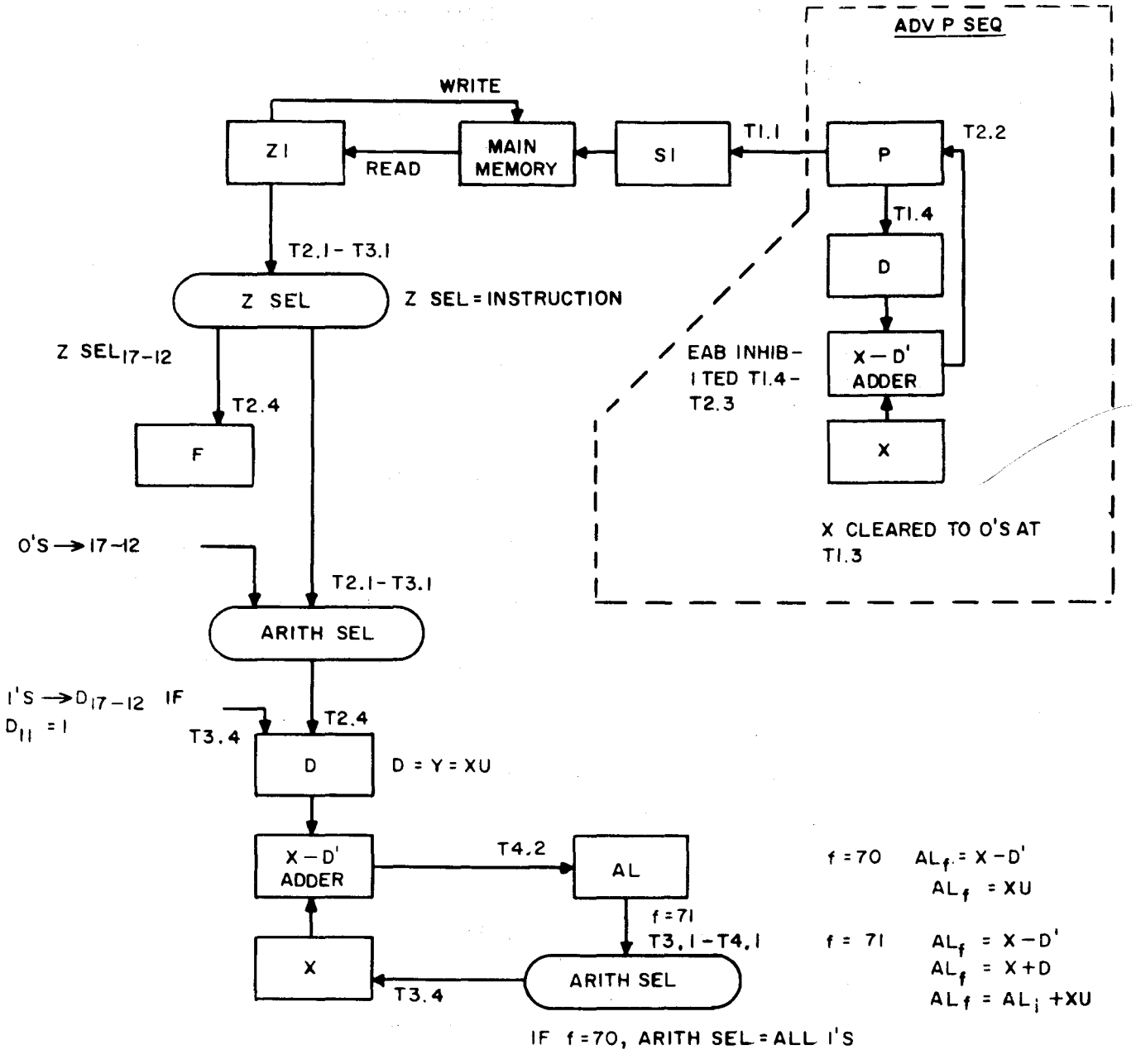
2-173. I-Sequence Data Flow for f = 70, 71. Refer to figure 2-29 for a block diagram description of the execution of ENTALK and ADDALK instructions (f = 70, 71). Most of the I-sequence operations are as previously described in a typical I-sequence. As shown in figure 2-29 at T2.4, the D-register receives the lower 12 bits (U) of the instruction word from the arithmetic selectors. The sign bit of U is extended (XU) from D_{17-12}. The operand (XU) is applied to one side of the X-D adder. The operand from the X-register is applied to the other side of the adder. If f = 70, the X-register receives a -0 from the arithmetic selectors and the adder effectively outputs -0+XU to the AL-register. If f = 71, the X-register receives the initial contents of the AL-register and the adder effectively outputs AL_i+XU to the AL-register. The initial instruction word may be obtained from main, control, or bootstrap memory. Refer to table 2-25 for a list of the essential commands used if f = 70, 71. Those commands that have not been previously described in an I-sequence are described in table 2-26 in detail.

2-174. Execution of ENTICR, ENTSR Instructions (f = 50:72, 50:73). The ENTICR instructions are used to enter the ICR or the SR-register with the lower bits of the instruction word. If f = 50:72, the three least significant bits of the instruction word are placed in the ICR-register, designating the B-register to be used in address modification. If f = 50:73, the five least significant bits of the instruction word are placed in the SR-register making this register available for address selection. Bit 3 of the instruction word is gated to the SR-active bit while bit 4 of the instruction word is gated to SR bit 3.

2-175. I-Sequence Data Flow for f = 50:72, 50:73. Refer to figure 2-30 for a block diagram description of the execution of ENTIC ENTSR (f = 50:72, 50:73). Although this figure shows the instruction word being obtained from main memory, it can also be obtained from control or bootstrap memories. As shown in figure 2-30 at T2.4, the KO-register receives the six least significant bits of the instruction word via the Z selectors. If f = 50:72 at time T3.3, the ICR-register is cleared and receives bits 2-0 of the KO-register. If f = 50:73 at time T3.3, the SR-register is cleared and receives bits 4-0 of the KO-register. If bit 3 of the instruction word is set, it will make the SR-register active. Refer to table 2-27 for a list of the essential commands used if f = 50:72 or 50:73. Those commands that have not been previously described in an I-sequence are described in table 2-28 in detail.

2-176. Execution of ENTBK, ENTBKB Instructions (f = 36, 37). The ENTBK (f = 36) and ENTBKB (f = 37) instructions are used to enter the B-register with a constant. If f = 36, the lower 12-bits of the instruction word (U) are extended to an 18-bit word (XU) and placed into the B-register as determined by the contents of the ICR-register. The B-register is cleared before being extended by the constant (XU). If f = 37, an 18-bit word (XU) is formulated as before, but this value is then added to the original contents of the designated B-register. Therefore, $B_f = B_i + XU$.

2-177. I and Next I-Sequence Data Flow for f = 36, 37. Refer to figure 2-31 for a block diagram description of the execution of ENTBK, ENTBKB (f = 36, 37). As shown in this figure, the value of XU is formulate in the D-register which applies it to one side of the X-D' adder. The ICR-register selects the B-register by setting the SO-register to the proper address (00001_8 - 00010_8). A control memory reference then calls out the



T4.2 SET OVERFLOW ff IF (f = 71)·(SIGN OF $AL_f \neq$ SIGN OF AL_i)*

NOTE: * THE OVERFLOW CONDITION IS INDICATED FROM THE X – D' ADDER BY:
($X_{17} = 0_2 \cdot D_{17}' = 1_2$ · NO BORROW REQUEST → BIT 17) +
($X_{17} = 1_2 \cdot D_{17}' = 0_2$ · BORROW REQUEST → BIT 17)

Figure 2-29. I-Sequence Data Flow for f = 70, 71

contents of the designated B-register. If f = 36, the arithmetic selector outputs a -0 to the X-register. Under these circumstances, the X-D' adder effectively outputs XU to the Z1-register. This value is written into the designated B-register, via the ZO-register, during the next I-sequence. The instruction word can be obtained from main, control, or bootstrap memory. Refer to table 2-29 for a list of the essential commands used if f = 36 or 37. Those commands that have not been previously described in an I and Next I-sequence are described in table 2-30 in detail.

TABLE 2-25. I-SEQUENCE ESSENTIAL COMMANDS FOR f = 70, 71

Time Notation	Commands
T4.4	Clear S1
T1.1	P → S1, Init Memory, *set Incr P ff
T1.3	Clear Z1, *set OXL11 ff, *clear D, *clear X, clear F
T1.4	$*P_L \rightarrow D_L$, $*P_U \rightarrow D_U$, *set Inhib EAB ff
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel, 0's → Arith Sel 17-12, *Clear Incr P ff
T2.2	*Clear P, *Adder → P
T2.3	Clear X, *clear Inhib EAB ff, *clear OXL11 ff
T2.4	$Z\text{-}Sel_{17-12} \rightarrow F$, Arith Sel → D
T3.1	AL → Arith Sel if f = 71, drop Z1 → Z-Sel, drop Z-Sel → Arith Sel, drop 0's → $Arith\ Sel_{17-12}$
T3.4	1's → D_{17-12} if D_{11} = 1, Arith Sel → X
T4.1	Clear AL, drop AL → Arith Sel
T4.2	Adder → AL, set Overflow ff if (f = 71) • (sign of $AL_f \neq$ sign of AL_i)

*These events are concerned with or are controlled by the Advance-P subsequence.

TABLE 2-26. COMMANDS/DESCRIPTIONS FOR EXECUTION OF ENTALK, ADDALK INSTRUCTIONS (f = 70, 71) (NOT PREVIOUSLY DESCRIBED IN AN I-SEQUENCE) (ALL COMMANDS ARE DESCRIBED IN TABLE 2-19)

Time Notation	Command/Description
T3.1	AL → Arith Sel if f = 71. Gate 20N01 (refer to plate P-17) is enabled by f = 71 and L ⇒ T34 I-Seq Format I to produce a high output which is inverted by 28N01 and applied to 29N01. Gate 90N01 is disabled by not being R1 or W-Sequence, and the low output is also applied to 29N01 to produce a L ⇒ AL → Select.
T3.4	1's → D_{17-12} if D_{11} = 1. The L ⇒ D Lower 12 Neg from OXD11 (refer to plate P-86) is applied to gate 50N02 (refer to plate P-18) and, along with L ⇒ f = 36, 37, 70, 71 and L ⇒ T34 I-Seq Format I, enables 50N02 which produces a high output and is inverted by 58N02 to produce a L ⇒ Set DU to 01D16 and 01D17. The high output from 50N02 is also applied to gates 03G40 through 03G44 (refer to plate P-39) and converted and applied to 01D12 through 01D15 (refer to plate P-86).
T4.1	Clear AL. Gate 00N05 (refer to plate P-20) is disabled by not being in a R1 or W-sequence. The low output is applied to 08N05. Gate 10N05 is enabled by a L⇒ f = 70, 71, 50; 60 and L ⇒ T42 I-Seq and outputs a high which is inverted by 07N05 and applied to 08N05. Gate 08N05 outputs a high at ∅1 and produces a H ⇒ Clear AL via 09N05 to the AL-register. Drop AL → Arith Sel. Gate 20N01 (refer to plate P-17) is disabled by not being at T34. This low output is inverted by 28N01 to disabled 29N01 which produces a high to drop AL → Arith Sel.
T4.2	Adder → AL. The high output of 10N05 is inverted by 18N05 and ∅2 enables 19N05 to produce a L ⇒ Adder → AL (refer to plate P-20). Set Overflow ff if (f = 71). (Sign of $AL_f \neq$ sign of AL_i). Gate 22G52 (refer to plate P-30) is enabled by sign of $AL_f \neq$ sign of AL_i and produces a low output. Gate 11E52 is enabled by high output of 10E52 which is enabled by L ⇒ f = 71, L ⇒ Format I and L ⇒ T42 I-Seq. The low outputs of 22G52 and 11E52 at ∅2, set Overflow flip-flop OXG52.

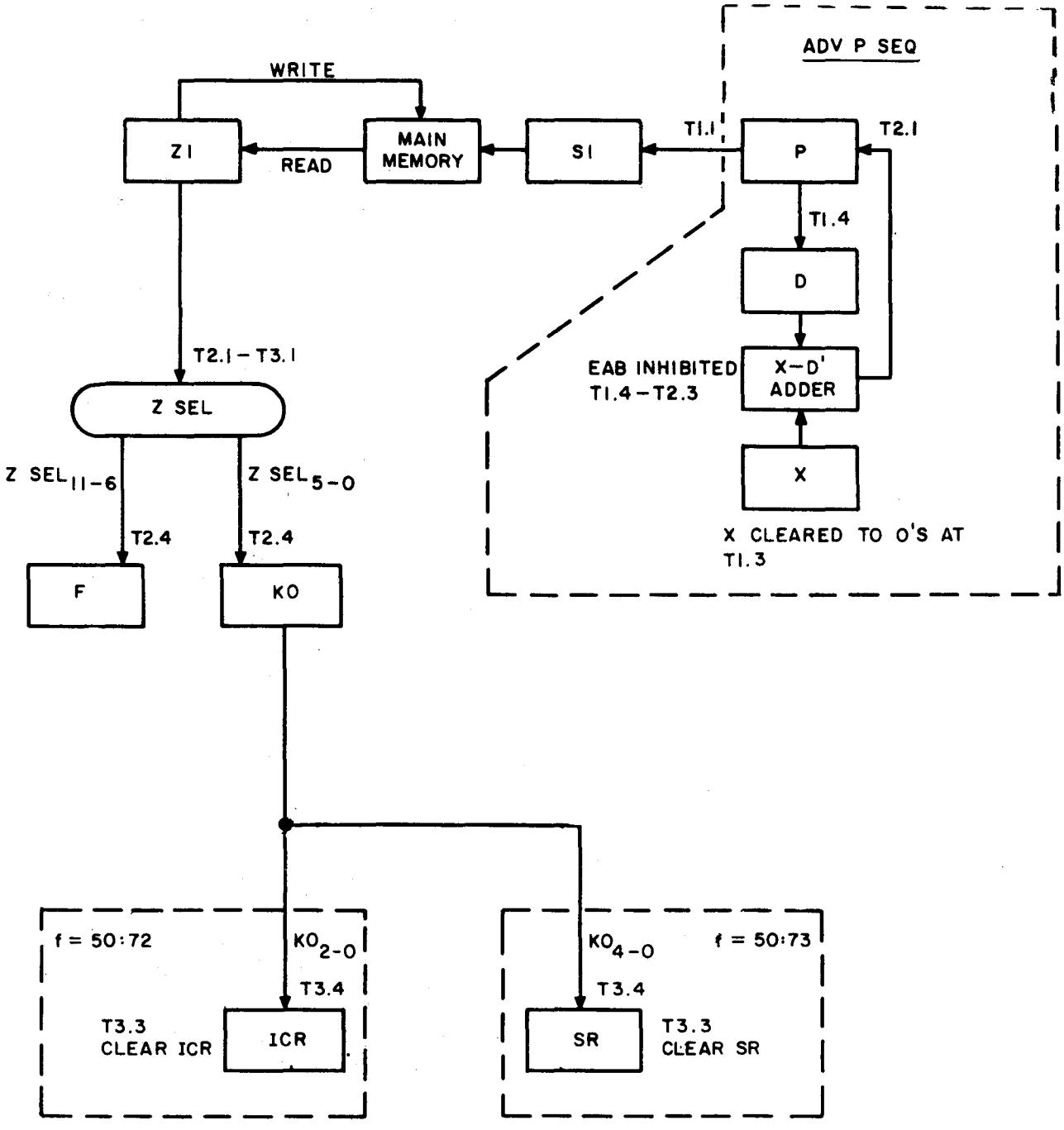


Figure 2-30. I-Sequence Data Flow for f = 50:72, 50:73

TABLE 2-27. I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:72, 50:73

Time Notation	Commands
T4.4	Clear S1
T1.1	P → S1, Init Memory, *set Incr P ff
T1.3	*Clear D, *clear X, clear Z1, clear F, *set 0XL11 ff
T1.4	*P_L → D_L, *P_U → D_U, clear KO, *set Inhib EAB ff
T2.1	Z1 → Z-Sel, *clear Incr P ff
T2.2	*Clear P, *Adder → P
T2.3	*Clear 0XL11 ff, clear Inhib EAB ff
T2.4	Z-Sel_{11-6} → F, set 0XF06 ff, Z-Sel_{5-0} → KO
T3.1	Drop Z1 → Z-Sel
T3.3	Clear ICR if f = 50:72, clear SR if f = 50:73
T3.4	KO_{2-0} → ICR if f = 50:72, KO_{4-0} → SR if f = 50:73

*These events are concerned with or are controlled by the Advance-P subsequence.

TABLE 2-28. COMMANDS/DESCRIPTIONS FOR EXECUTION OF ENTICR, ENTSR INSTRUCTIONS (f = 50:72, 50:73)
(NOT PREVIOUSLY DESCRIBED IN AN I-SEQUENCE)
(REFER TO TABLE 2-19 FOR A DESCRIPTION OF ALL COMMANDS)

Time Notation	Command/Description
T3.3	Clear ICR if f = 50:72. Gate 10E44 (refer to plate P-38) is enabled by a L ⇒ f = 50:72 and a L ⇒ T34 I-Seq Format II. The high output is inverted by 08E44 and applied to 09E44 and, at ∅3, 09E44 outputs a H ⇒ Clear ICR to clear ICR-register OXG45 through OXG47 (refer to plate P-39). Clear SR if f = 50:73. Gate 10E40 (refer to plate P-38) is enabled by a L ⇒ 50:73 and a L ⇒ T34 I-Seq Format II. The high output is inverted by 08E40 and applied to 09E40 and, at ∅3, outputs a H ⇒ Clear SR to clear SR-register OXG40 through OXG44 (refer to plate P-39).
T3.4	KO_{2-0} → ICR if f = 50:72. The high output of 10E44 is inverted by 18E44 and, at ∅4, 19E44 outputs a L ⇒ K00-K02 → ICR enable to ICR-register. KO_{4-0} → SR if f = 50:73. The high output from 10E40 is inverted and applied to 19E40 and, at ∅4, 19E40 produces a L ⇒ K00-K04 → SR enable to the SR-register.

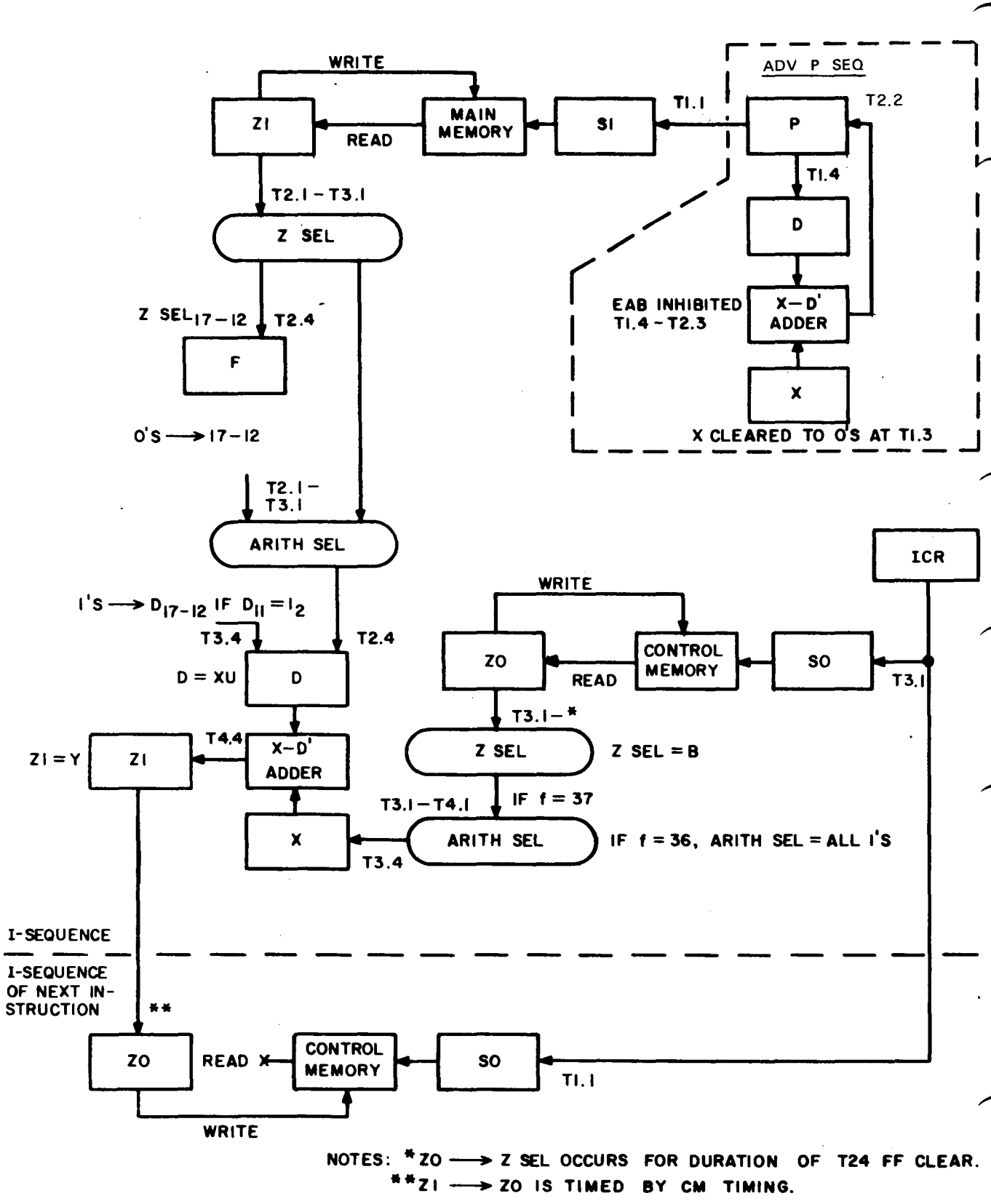


Figure 2-31. I and Next I-Sequence Data Flow for f = 36, 37

TABLE 2-29. I AND NEXT I-SEQUENCE ESSENTIAL COMMANDS FOR f = 36, 37

Time Notation	Commands
	I-SEQUENCE
T4.4	Clear S1
T1.1	P → S1, Init Memory, *set Incr P ff
T1.3	Clear Z1, *set OXL11 ff, *clear D, *clear X, clear F
T1.4	$*P_L \rightarrow D_L$, $*P_U \rightarrow D_U$, *set Inhibit EAB ff
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel, 0's → Arith Sel_{17-12}, *clear Incr P ff
T2.2	*Clear P, *Adder → P
T2.3	Clear X, *clear Inhib EAB ff, *clear OXL11 ff
T2.4	$Z\text{-}Sel_{17-12} \rightarrow F$, Arith Sel → D
T3.1	Drop Z1 → Z-Sel, drop Z-Sel → Arith Sel, drop 0's → Arith Sel_{17-12}, ICR → SO, Init CM, ZO → Z Sel **, Z Sel → Arith Sel if f = 37
T3.4	1's → D_{17-12} if $D_{11} = 1$, Arith Sel → X
T4.1	Drop Z-Sel → Arith Sel
T4.3	Clear Z1
T4.4	Adder → Z1***, disable CM → ZO
	I-SEQUENCE OF NEXT INSTRUCTION
T1.1	ICR → SO, Init CM
T1.4	Drop disable CM → ZO

* These events are concerned with or are controlled by the Advance-P subsequence.
** ZO → Z-Sel occurs for duration of T24 ff clear.
***Z1 → ZO is timed by control memory timing.

TABLE 2-30. COMMANDS/DESCRIPTIONS FOR EXECUTION OF ENTBK, ENTBKB INSTRUCTIONS (f = 36, 37)
(NOT PREVIOUSLY DESCRIBED IN AN I-SEQUENCE)
(REFER TO TABLE 2-19 FOR A DESCRIPTION OF ALL COMMANDS)

Time Notation	Command/Description
T3.4	1's → D 17-12 if D11 = 1. Same as f = 70, 71.
T4.3	Clear Z1. Gate 30N13 (refer to plate P-23) is enabled by a L ⇒ f = 36, 37, L ⇒ I-Seq and by a low from 03T44. This high output is inverted by 08N13 and at ∅3, 09N13 outputs a L⇒ Clear Z1.
T4.4	Adder → Z1. The high from 30N13 is inverted by 38N13 and at ∅4, 39N13 outputs a L ⇒ Adder → Z1.
	Disable CM → ZO. Gate 10N13 (refer to plate P-23) is enabled by L ⇒ I- or I/O Seq, a low from 03T13 and not in Cont Data Seq. The high output is applied to 11N11 (refer to plate P-25) which outputs a high disabling gates 17N11, 18N11 and 19N11.
	I-Sequence of Next Instruction
T1.1	ICR → SO, Init CM. The output of gate 15G34 (refer to plate P-29) is low because of the absence of an f = 56 and f = 73. The L⇒Jump is applied to 40N10 (refer to plate P-24) which, along with L ⇒ f = 36, 37, L ⇒ I-Seq and a low from 03T11, provides a high to 48N10. Also, Gate 48N10, along with the absence of a Format II, outputs a low to 49N10 which at ∅1, produces a L ⇒ ICR → SO. One of the low outputs goes to the SO-register (refer to plate P-114) and the other to OXDT10 (refer to plate P-122).
T1.4	Drop Disable CM → ZO. Gate 10N13 (refer to plate P-23) outputs a low because the input from 03T13 is high. The low is applied to gate 11N11 (refer to plate P-25). Other inputs to 11N11 are L ⇒ Strobe C or B memory, and the low from 00N11, which is disabled by the high from 31T42. The low output of 11N11 is applied to gates 17N11, 18N11 and 19N11. The absence of Bootstrap and a L ⇒ Lower 6 Bits → ZO at 17N11 and a L ⇒ Middle 6 Bits → ZO provide a L ⇒ Cont Mem → ZO.

2-178. Execution of ENTB, ENTBB Instructions (f = 32, 33). The ENTB (f = 32) and ENTBB (f = 33) are used to enter the B-register with contents of memory. If f = 32, an operand that was obtained from memory is entered into the designated B-register. The memory address of the operand is Up if the SR-register is inactive, and U_{SR} if the SR-register is active. If f = 33, an operand that was obtained from memory is entered into a designated B-register as before. In this case, however, the memory address of the operand is Up + B, provided the SR-register is inactive, or U_{SR} + B if the SR-register is active.

2-179. R1 and Next I-Sequence Data Flow for f = 32, 33. Refer to figure 2-32 for a block diagram description of the execution of ENTB, ENTBB (f = 32, 33). The figure does not show the initial I-sequence, during which the instruction word was called out of memory, and during which the operand address was developed, since this address is transferred to the S1-register at time T1.1 of the R1-sequence. As shown in this figure, the R1-sequence uses a memory reference to obtain the operand, which is placed in the D-register and then applied to one side of the X-D' adder. Since the X-register is set to a -0, the X-D' adder will output the operand to the Z1-register. During the next I-sequence, the B-register address is placed in the SO-register from the ICR-register. The operand in the Z1-register is transferred to the ZO-register. A control memory reference is initiated and the operand is written into the designated B-register. The operand could have been obtained from main, control, or bootstrap memory. Refer to table 2-31 for a list of the essential R1-sequence commands along with next I-sequence commands for f = 32, 33. Those commands that have not been previously described in an I and R1-sequence are described in table 2-32 in detail.

2-180. Execution of ENTAU, ENTAUB, ENTAL, ENTALB Instructions (f = 10, 11, 12, 13). Instructions f = 10 through 13 are used to enter either the AL or AU-registers with the contents of memory.

2-181. ENTAU (f = 10). The ENTAU instruction extracts an operand (Y) from memory, placing it in the AU-register. The address of the operand is Up if the SR-register is inactive, and U_{SR} if the SR-register is active.

2-182. ENTAUB (f = 11). The ENTAUB instruction provides the same function as ENTAU, except that the address of the operand is either Up + B or U_{SR} + B, depending upon the active-inactive status of the SR-register. The B-register is specified by the ICR-register.

2-183. ENTAL (f = 12). The ENTAL instruction provides the same function as ENTAU except that the contents of memory are placed in the AL-register.

2-184. ENTALB (f = 13). The ENTALB instruction provides the same function as ENTAUB except that the contents of memory are placed into the AL-register.

2-185. R1-Sequence Data Flow for f = 10-13. Refer to figure 2-33 for a block diagram description of the execution of ENTAU, ENTAUB, ENTAL, ENTALB (f = 10, 11, 12, 13). This figure does not show the I-sequence which called the instruction word out of memory and formulated the operand memory address. As shown in figure 2-33, at time T1.1 of the R1-sequence, the address of the operand is transferred to the S1-register, a memory reference is initiated placing the operand (Y) into the D-register; and, since

the X-register is set to a = 0, the adder output is the operand. This operand is now placed in either the AL or AU-register, depending upon the instruction being executed. Refer to table 2-33 for a list of R1-sequence commands for f = 10 through 13. Those commands not previously described in an R1-sequence are described in table 2-34 in detail.

2-186. Execution of CL, CLB, STRB, STRBB, STRAL, STRALB, STRAU, STRAUB Instructions (f = 40, 41, 42, 43, 44, 45, 46, 47). Instructions f = 40 through 47 are used to store 0's, or the contents of the B, AL or AU-register in memory.

2-187. CL (f = 40). The CL instruction effectively clears (stores 0's) the content of a designated memory location. The address of the memory location is Up if the SR-register is inactive, or U_{SR} if the SR-register is active.

2-188. CLB (f = 41). The CLB instruction provides the same function as CL, except that the designated memory address is modified by the B-register. The memory address is now Up+B if the SR-register is inactive, and U_{SR}+B if the SR-register is active.

2-189. STRB (f = 42). The STRB instruction stores the contents of a designated B-register in memory, the address being Up if the SR-register is inactive, or U_{SR} if the SR-register is active. The contents of the B-register are not altered by execution of this instruction.

2-190. STRBB (f = 43). The STRBB instruction provides the same function as STRB except that the designated memory address is modified by the B-register. The memory address is Up+B if the SR-register is inactive, or U_{SR}+B if it is active.

2-191. STRAL (f = 44). The STRAL instruction stores the contents of the AL-register in a designated memory location, where the address is Up if the SR-register is inactive, and U_{SR} if it is active.

2-192. STRALB (f = 45). The STRALB instruction provides the same function as STRAL except that the designated memory address is modified by the B-register. The memory address is Up+B if the SR-register is inactive or U_{SR}+B if it is active.

2-193. STRAU (f = 46). The STRAU instruction provides the same function as STRAL except that the contents of the AU-register are stored, rather than the contents of the AL-register.

2-194. STRAUB (f = 47). The STRAUB instruction provides the same function as STRALB except that the contents of the AU-register are stored, rather than the contents of the AL-register.

2-195. W-Sequence Data Flow for f = 40-47. Refer to figure 2-34 for a block diagram description of the execution of CL, CLB, STRB, STRBB, STRAL, STRALB, STRAU, STRAUB (f = 40, 41, 42, 43, 44, 45, 46, 47). The I-sequence that called the instruction word out of memory and formulated the storage address of the operand is not shown in this figure. As shown in figure 2-34, at time T1.1, the adder output is applied to the S1-register and a memory reference is initiated. The original content of the address memory location is destroyed because the gating of memory to the Z1-register is disabled during the read portion of the memory cycle. The operand, to be stored in memory, is placed in the Z1-register from one of the following sources: from the B-register if f = 42, 43 (the ICR-register specifies the B-register); from the AU-register if f = 46

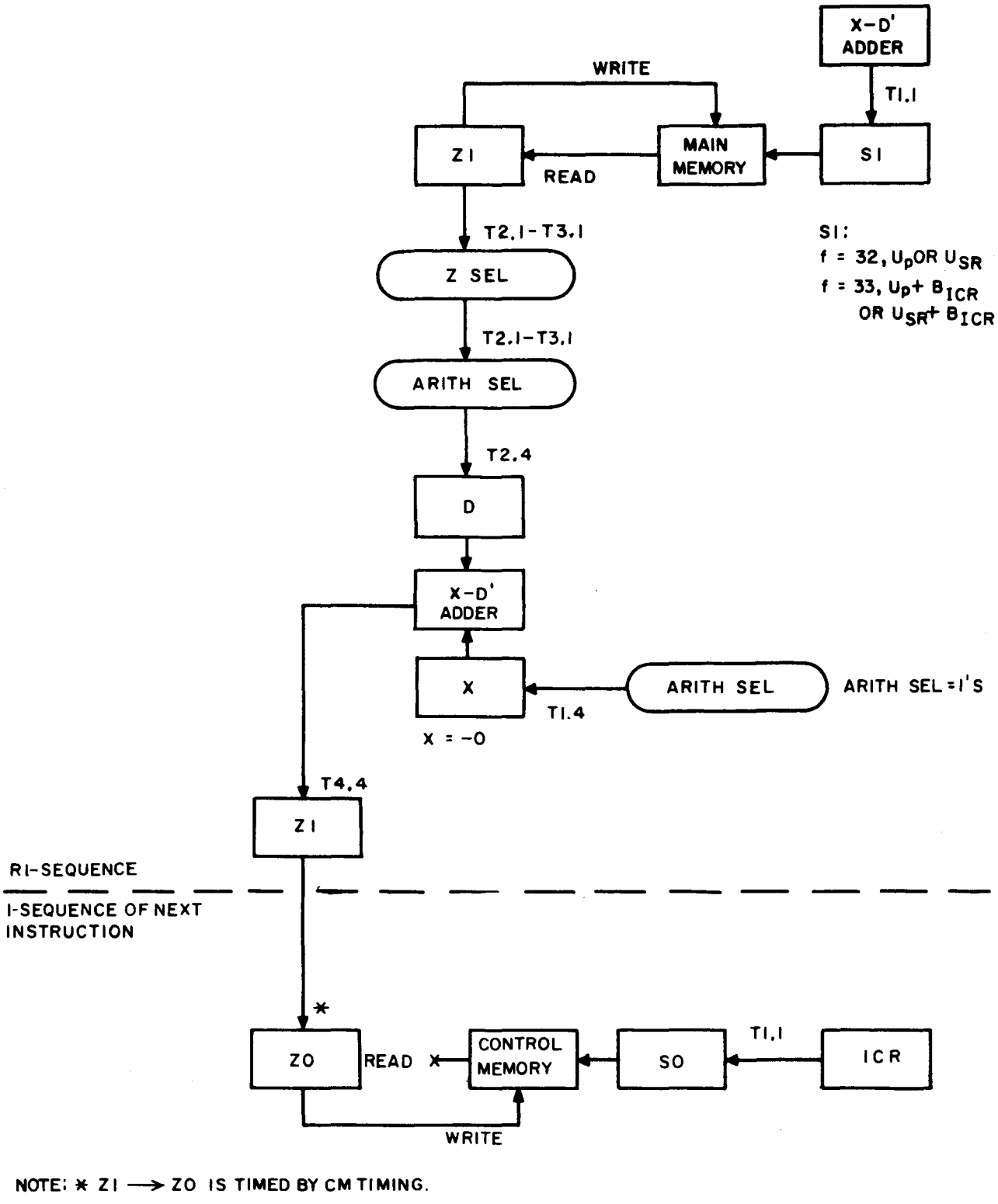


Figure 2-32. R1 and Next I-Sequence for f = 32, 33

TABLE 2-31. R1 AND NEXT I-SEQUENCE ESSENTIAL COMMANDS FOR f = 32, 33

Time Notation	Commands
	R1-SEQUENCE
T4.4	Clear S1
T1.1	Adder → S1, Init Memory
T1.3	Clear X, clear Z1
T1.4	Arith Sel → X
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel
T2.3	Clear D
T2.4	Arith Sel → D
T3.1	Drop Z1 → Z-Sel, drop Z-Sel → Arith Sel
T4.3	Clear Z1
T4.4	Adder → Z1*, disable CM → ZO
	I-SEQUENCE OF NEXT INSTRUCTION
T1.1	ICR → SO, Init CM
T1.4	Drop Disable CM → ZO

*Z1 → ZO is timed by control memory timing.

TABLE 2-32. COMMANDS/DESCRIPTIONS FOR EXECUTION OF ENTB, ENTBB INSTRUCTIONS (f = 32, 33)
(NOT PREVIOUSLY DESCRIBED IN AN I AND R1-SEQUENCE)
(REFER TO TABLE 2-21 FOR A DESCRIPTION OF ALL COMMANDS NOT DESCRIBED)

Time Notation	Command/Description
T4.3	Clear Z1. Gate 30N13 (refer to plate P-23) is enabled by a L ⇒ f = 32, 33, L ⇒ R1-Seq, and a low from 03T44. The high output is inverted by 08N13 and at ∅3, 09N13 produces a L ⇒ Clear Z1.
T4.4	Adder → Z1. The high out of 30N13 is inverted by 38N13 and at ∅4, 39N13 produces a L ⇒ Adder → Z1.
	Disable CM → ZO. At T4.2, the I_i flip-flop in the upper-rank sequence was set, providing gate 10N13 (refer to plate P-23) with a L ⇒ I-Seq. This gate is enabled the same as the f = 36, 37 instruction previously described.
	I - SEQUENCE OF NEXT INSTRUCTION
T1.1	ICR → SO, Init CM.
T1.4	Drop disable CM → ZO. Same as described for a f = 36, 37 instruction.

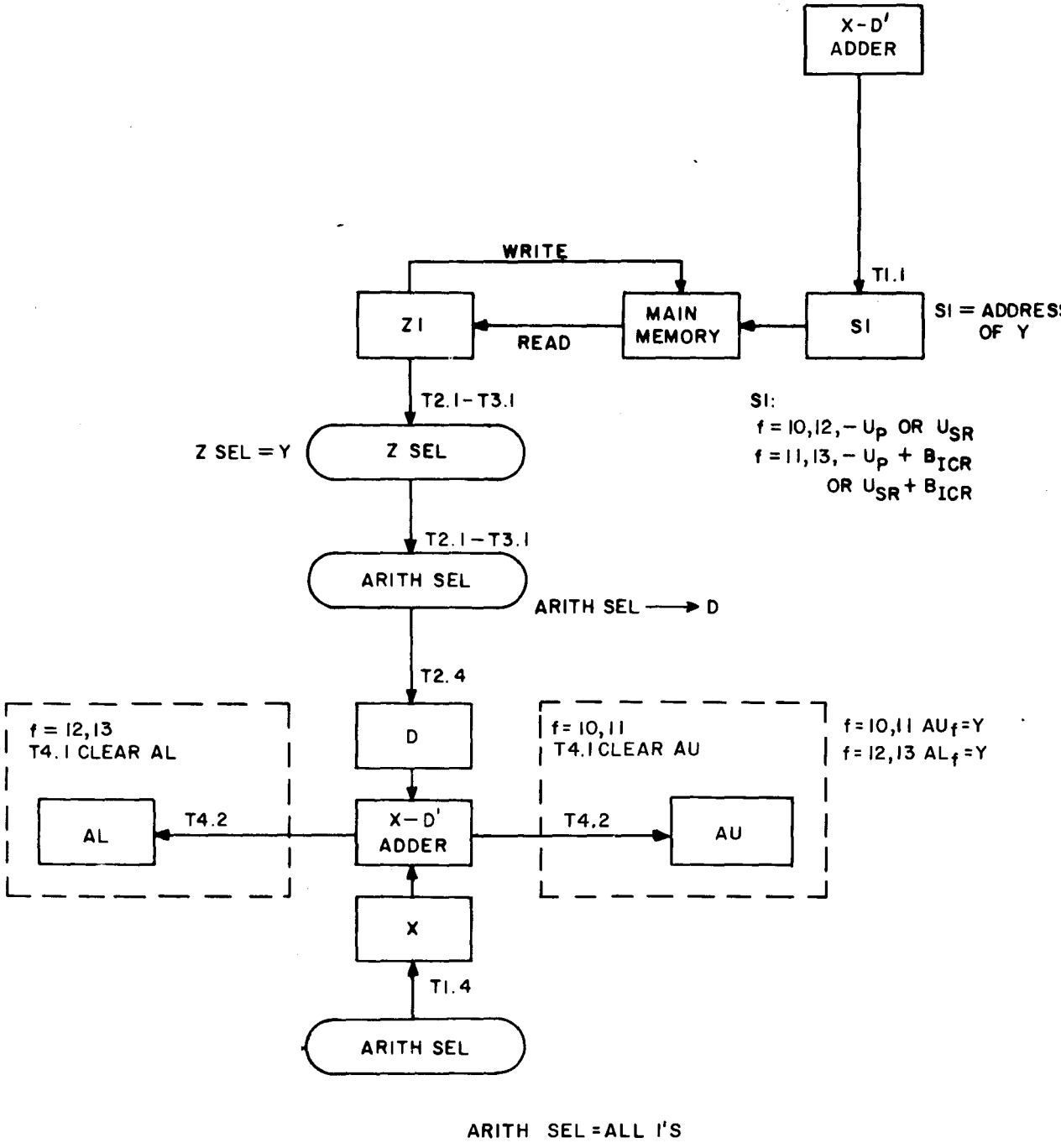


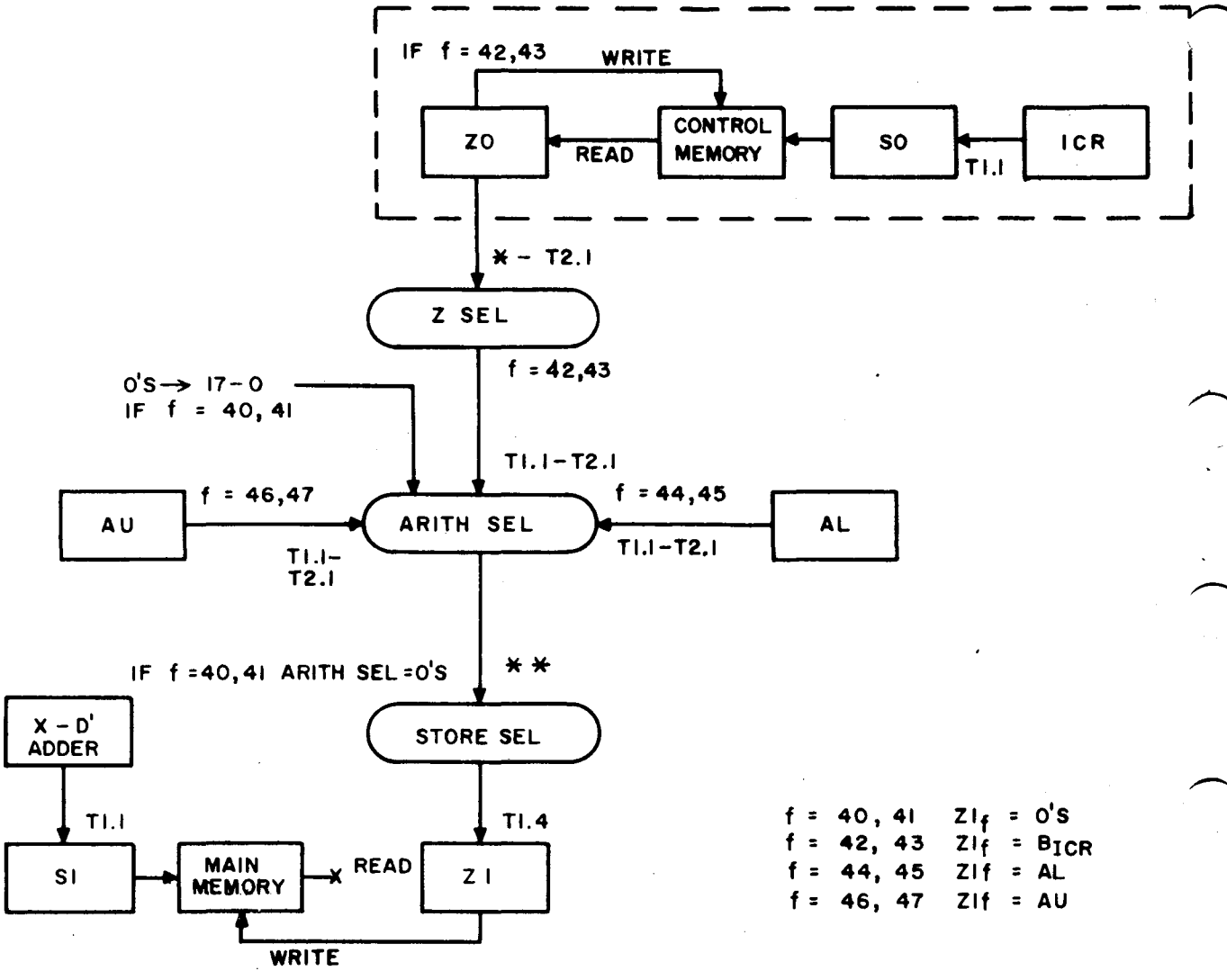
Figure 2-33. R-Sequence Data Flow for f = 10-13

TABLE 2-33. R1-SEQUENCE ESSENTIAL COMMANDS FOR f = 10 through 13

Time Notation	Commands f =	10, 11	12, 13
T4.4	Clear S1	X	X
T1.1	Adder → S1, Init memory	X	X
T1.3	Clear X, clear Z1	X	X
T1.4	Arith Sel → X	X	X
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel	X	X
T2.3	Clear D	X	X
T2.4	Arith Sel → D	X	X
T3.1	Drop Z1 → Z-Sel, drop Z-Sel → Arith Sel	X	X
T4.1	Clear AU	X	
	Clear AL		X
T4.2	Adder → AU	X	
	Adder → AL		X

TABLE 2-34. COMMANDS/DESCRIPTIONS FOR EXECUTION OF ENTAU, ENTAUB, ENTAL, ENTALB INSTRUCTIONS (f = 10, 11, 12, 13)
(NOT PREVIOUSLY DESCRIBED IN AN R1-SEQUENCE)
(REFER TO TABLE 2-21 FOR ALL COMMANDS NOT DESCRIBED)

Time Notation	Command/Description
T4.1	Clear AU. Gate 10N05 (refer to plate P-20) is disabled by the absence of a proper instruction sequence and outputs a low to 10N04 and along with a L ⇒ f = 10, 11, 20-23 and L ⇒ T42 R1-Seq enables 10N04 to produce a high. This high is inverted by 07N04 and, at ∅1, 08N04 is enabled and produces a H ⇒ Clear AU via 09N04.
T4.2	Adder → AU. The high output of 10N04 is inverted and applied to 19N04. Gate 00N04 is disabled by the absence of a f = 24, 25 instruction and the low output is applied to 19N04 and, at ∅2, 19N04 produces a L ⇒ Adder → AU.



SI:

f=40,42,44,46-U_P OR U_{SR}

f=41,43,45,47-U_P + B_{ICR} OR U_{SR} + B_{ICR}

NOTES: * ZO ⟶ Z SEL OCCURS FOR DURATION OF T24 CLEAR.

** ARITH SEL ⟶ STORE SEL OCCURS FOR ENTIRE W-SEQUENCE.

Figure 2-34. W-Sequence Data Flow for f = 40-47

from the AL-register if f = 44, 45; and from the arithmetic selector in the form of 0's if f = 40, 41. Refer to table 2-35 for a list of W-sequence essential commands for f = 40-47. Those commands that have not been previously described in a W-sequence are described in table 2-36 in detail.

2-196. Execution of STRICR, STRADR, STRSR Instructions (f = 72, 74, 75). Instructions f = 72, 74, and 75 are used to store the contents of the SR, ICR, or the lower 12-bits of the AL-register in memory.

2-197. STRICR (f = 72). The STRICR instruction stores the contents of the ICR-register in memory at an address specified by Up. The 12 most significant bits are retained from the original memory contents, but the six least significant bits are changed as follows: bits 2^5 and 2^4 are set to 0's, while bits 2^3 through 2^0 store the contents of the ICR-register. If ICR = 000_2, then bit 2^3 is set; otherwise it is cleared.

2-198. STRADR (f = 74). The STRADR instruction stores the 12 least significant bits of the AL-register in memory at an address designated by Up. Of the original memory contents, the six most significant bits are retained, and the 12 least significant bits are replaced by the 12 least significant bits of the AL-register.

2-199. STRSR (f = 75). The STRSR instruction stores the contents of the SR-register in memory at the address specified by Up. The 12 most significant bits of the original memory contents are retained, and the six least significant bits are changed as follows: bit 2^5 is set to a 0, while bits 2^4 through 2^0 store the contents of the SR-register. After storage, SR is cleared and deactivated. Bit 2^3 stores the active bit while bit 2^4 stores SR bit 2^3.

2-200. W-Sequence Data Flow for f = 72, 74, 75. Refer to figure 1-35 for a block diagram description of the execution of STRICR, STRADR, STRSR instructions. The I-sequence that called the instruction word out of memory and formulated the memory storage address (Up) is not shown in the figure. As shown in figure 1-38, the adder outputs the memory storage address to S1 at time T1.1 of the W-sequence, and a memory cycle is initiated. The value to be stored (ICR, SR or AL) is applied to the arithmetic selectors. In the selectors, the bit positions not to be stored are masked out with 0's. The original bits of memory to be replaced, according to instruction execution, are destroyed by not being gated into Z1 during the read portion of the memory cycle. Refer to table 2-37 for a list of W-sequence commands for f = 72, 74, 75. Those commands that have not been previously described in a W-sequence are described in table 2-38 in detail.

2-201. Execution of Jump (JP, JPB, JPAUZ, JPALZ, JPAUNZ, JPALNZ, JPAUP, JPALP, JPAUNG, JPALNG, and BJP) Instructions (f = 34, 35, 60-67, and 73). Jump instructions allow great flexibility in the execution of a program. The use of nonconditional jump instructions allows a jump to any address in a memory stack each time the instruction is executed, whereas a conditional jump instruction can take advantage of some logical decisions the computer must make before the jump is executed. Instructions f = 34, 35, 60 through 67 and 73 cause a direct program jump to an address designated by Up or Up+B, either conditionally or unconditionally.

2-202. JP (f = 34). The JP instruction unconditionally performs a direct program jump to an address specified by Up.

TABLE 2-35. W-SEQUENCE ESSENTIAL COMMANDS FOR f = 40-47

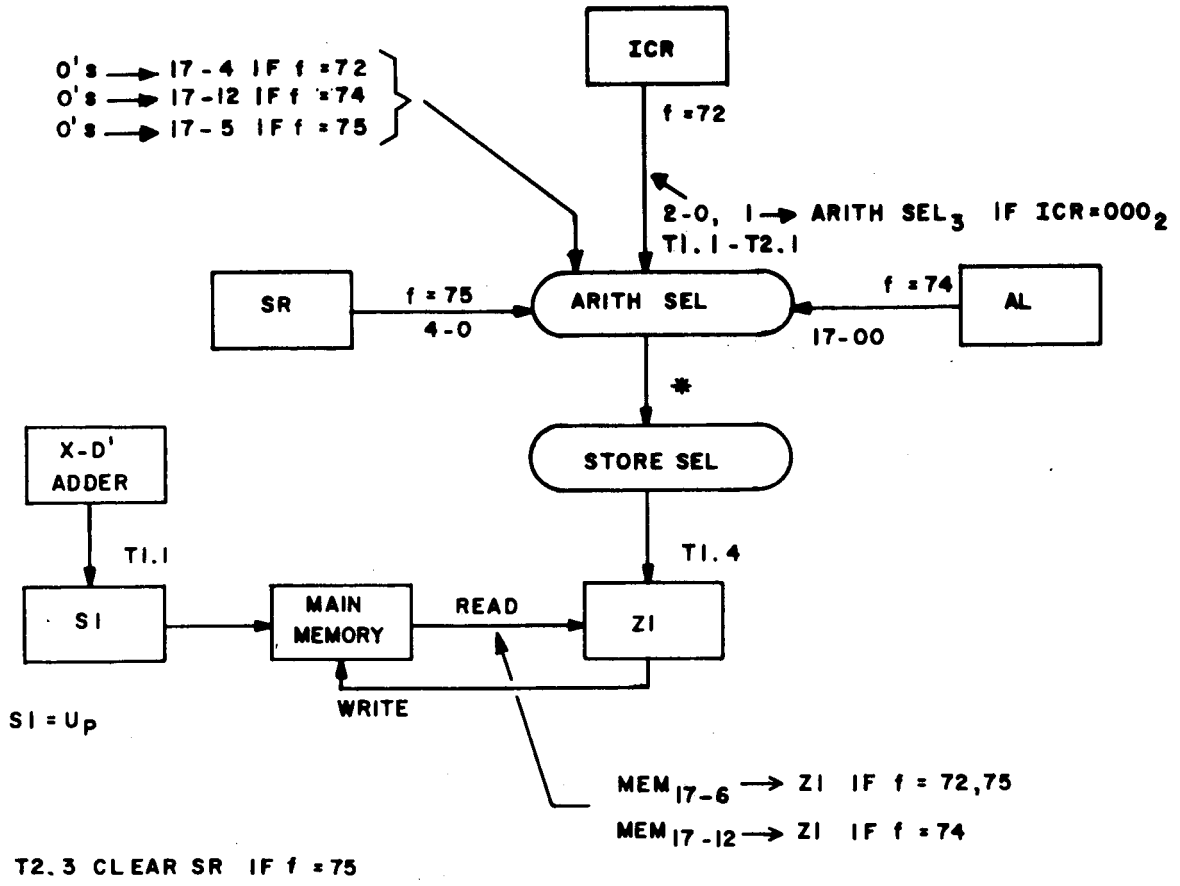
Time Notation	Commands f =	40,41	42,43	44,45	46,47
T4.4	Clear S1	X	X	X	X
T1.1	Adder → S1, Init Memory	X	X	X	X
	ICR → SO, Init CM, *Z-Sel → Arith Sel		X		
	O's → Arith Sel	X			
	AL → Arith Sel			X	
	AU → Arith Sel				X
T1.3	Clear Z1	X	X	X	X
T1.4	**Store Sel → Z1, disable Mem → Z1	X	X	X	X
T2.1	Drop O's → Arith Sel	X			
	Drop AL → Arith Sel			X	
	Drop AU → Arith Sel				X

NOTES: * ZO → Z Sel occurs for duration of T24 ff clear.
** Arith Sel → Store Sel occurs for entire W-sequence.

TABLE 2-36. COMMANDS/DESCRIPTIONS FOR EXECUTION OF CL, CLB, STRB, STRBB, STRAL, STRALB, STRAU, STRAUB (f = 40, 41, 42, 43, 44, 45, 46, 47)

(NOT DESCRIBED PREVIOUSLY IN A W-SEQUENCE)
(REFER TO TABLE 2-22 FOR COMMANDS DESCRIBED PREVIOUSLY)

Time Notation	Command/Description
T1.1	ICR → SO, Init CM. Gate 40N10 (refer to Plate P-24) is qualified by a L ⇒ f = 42, 43, 56, L ⇒ R1, W Seq & a low from 03T11. The high output is applied to 48N10 which outputs a low to 49N10 due to the absence of a Format II. At ∅1 49N10 produces a L ⇒ ICR → SO which also initiates CM.
	Z Sel → Arith Sel. Gate 40N01 (refer to Plate P-17) is qualified by a L ⇒ f = 42, 43, 56, & a L ⇒ T14 R1, W-Seq. The high output is inverted by 48N01 & 49N01 to provide a L ⇒ Z → SELECT. At T2.1 the T14 input goes high, however, 40N01 is kept qualified due to the low inputs on pin 9 & 10 and the L ⇒ Z → SELECT is not dropped until T3.1.
	AL → Arith Sel. AU → Arith Sel. } Described in the R Sequence (Table 2-21)
T2.1	Drop O's → Arith Sel. Drop AL → Arith Sel. Drop AU → Arith Sel. } Gates 10N01, 20N01, 40N01 & 50N01 (refer to Plate P-17) are now disabled due to the loss of T14.



NOTE: * ARITH SEL → STORE SEL OCCURS FOR ENTIRE W-SEQUENCE.

$f = 72 \quad MEMORY_f = \underbrace{17-6}_{MEMORY_i} \quad \underbrace{5-4}_{0's} \quad 3 \quad \underbrace{2 \quad 0}_{ICR}$

0 IF ICR ≠ 000₂

1 IF ICR = 000₂

$f = 74 \quad MEMORY_f = \underbrace{17-12}_{MEMORY_i} \quad \underbrace{11-0}_{AL_{11-0}}$

$f = 75 \quad MEMORY_f = \underbrace{17-6}_{MEMORY_i} \quad 5 \quad \underbrace{4-0}_{SR}$

0

Figure 2-35. W-Sequence Data Flow for f = 72, 74, 75

TABLE 2-37. W-SEQUENCE ESSENTIAL COMMANDS FOR f = 72, 74, 75

Time Notation	Commands f =	72	74	75
T4.4	Clear S1	X	X	X
T1.1	Adder → S1, Init memory	X	X	X
	ICR → Arith Sel_{2-0}, 0's → Arith Sel_{17-4}, 1 → Arith Sel_3 if ICR = 000_2	X		
	AL → Arith Sel, 0's → Arith Sel_{17-12}		X	
	SR → Arith Sel_{4-0}, 0's → Arith Sel_{17-5}			X
T1.3	Clear Z1	X	X	X
T1.4	*Store Sel → Z1, disable Mem_{5-0} → Z1	X	X	X
	Disable Mem_{11-6} → Z1		X	
T2.1	Drop ICR → Arith Sel_{2-0} drop 0's → Arith Sel_{17-4}, drop 1 → Arith Sel_3	X		
	Drop AL → Arith Sel, drop 0's → Arith Sel_{17-12}		X	
	Drop SR → Arith Sel_{4-0}, drop 0's → Arith Sel_{17-5}			X
T2.3	Clear SR			X

*Arith Sel → Store Sel occurs for entire W-sequence.

TABLE 2-38. COMMANDS/DESCRIPTIONS FOR EXECUTION OF STRICR, STRADR, STRSR (f = 72, 74, 75)

(NOT DESCRIBED PREVIOUSLY IN A W-SEQUENCE)
(REFER TO TABLE 2-22 FOR COMMANDS DESCRIBED PREVIOUSLY)

Time Notation	Command/Description
T1.1	O's → Arith Sel 17-14 if f = 72. O's → Arith Sel 17-12 if f = 74. O's → Arith Sel 17-5 if f = 75. f = 72. Gate 50N01 (refer to Plate P-17) is qualified by a L ⇒ f = 40, 41, 72, 75, and a L ⇒ T14 R1, W-Seq. The high output is inverted by 78N01 and 58N01 to provide a L⇒ O's → Select from 79N01 and a L ⇒ O's → Select + SR + 1CR + K → Select from 59N01. The L ⇒ O's select is applied to Arithmetic Selector gates 13X06 through 13X17 (Plate P-83 & P-84) to output a high. The L ⇒ SR + ICR + K is applied to Arithmetic Selector gates 13X00 through 13X05, however, only gates 13X04 & 13X05 will be enabled due to the low inputs from 12X04 & 12X05. (Plate P-39). f = 74. Gate 70N01 (refer to Plate P-17) is qualified by a L ⇒ f = 74 and a L⇒ T14, R1-Seq. and the high output is inverted by 78N01 to provide a L⇒ O's. Select into Arithmetic Selector gates 13X12 through 13X17. f = 75. Same as described for a f = 72 except that gate 13X04 (Plate P-82) is not qualified due to the high input from 12X04 (Plate P-39). 1 → Arith Sel$_3$ if ICR = 000_2. If flip-flops 0XG45 through OXG47 are all cleared (refer to plate P-39), the low outputs will produce a high out of 01G45 inverted by 11G45, and with a f = 72, 12X03 will produce a high into 13X03 (refer to plate P-82) to output a low. SR → Arith Sel $_{4-0}$. Gates 12X00 through 12X04 (refer to plate P-39) will produce a high due to the input of a L ⇒ f = 75, if their respective flip-flops, OXG40 through OXG44, are set. This high will cause gates 13X00 through 13X04 (refer to plate P-82) to produce a low output. If any of the inputs are low, the respective gate will produce a high

TABLE 2-38. COMMANDS/DESCRIPTIONS FOR EXECUTION OF STRICR, STRADR, STRSR (f = 72, 74, 75) (Cont)

Time Notation	Command/Description
T1.1 (cont)	output due to the L ⇒ SR + ICR + K → Select. AL → Arith Sel. Same as described in the R-Sequence Table 2-21. ICR → Arith Sel_{2-0}. Gate 59N01 (refer to plate P-17) produces a L ⇒ SR + ICR + K → Select into Arith Sel 13X00 through 13X05 (refer to plate P-82), however, only gates 13X00 through 13X02 will be effected due to inputs from gates 12X00 through 12X02 (refer to plate P-39). These gates will output a high due to the input of a L ⇒ f = 72, if the respective flip-flop is set (OXG45 through OXG47).
T2.1	Drop ICR → Arith Sel_{2-0}, drop O's → Arith Sel 17-4 drop 1 → Arith Sel_3. Drop AL → Arith Sel, drop O's → Arith Sel_{17-12}. Drop SR → Arith Sel_{4-0}, drop O's → Arith Sel_{17-5}. Gates 20N01, 50N01 and 70N01 (refer to plate P-17) are all now disabled by T14 going high dropping the commands.
T2.3	Clear SR. Gate 00E40 (refer to plate P-38) is enabled by a L ⇒ f = 75, L ⇒ W-Seq, and a low from 01T23. The high output is inverted by 08E40 and at T3, 09E40 produces a H ⇒ Clear SR.

2-203. JPB (f = 35). The JPB instruction performs the same function as JP, except that the direct program jump address becomes Up + B.

2-204. JPAUZ (f = 60). The JPAUZ instruction performs a direct program jump to an address specified by Up if the Compare flip-flop (refer to plate P-29) is clear, and the AU-register contains a positive zero, or if both the Compare and Equal flip-flops are set.

2-205. JPALZ (f = 61). The JPALZ instruction performs a direct program jump to an address specified by Up if the Compare flip-flop is clear and the AL-register contains a positive zero, or if both the Compare and Equal flip-flops are set.

2-206. JPAUNZ (f = 62). The JPAUNZ instruction performs a direct program jump to the address specified by Up if the Compare flip-flop is clear and the contents of the AU-register are not a positive zero, of if the Compare flip-flop is set and the Equal flip-flop is clear.

2-207. JPALNZ (f = 63). The JPALNZ instruction performs a direct program jump to the address specified by Up if the Compare flip-flop is clear and the contents of the AL-register are not a positive zero, or the Compare flip-flop is set and the Equal flip-flop is clear.

2-208. JPAUP (f = 64). The JPAUP instruction performs a direct program jump to the address specified by Up if the Compare flip-flop is clear and the contents of the AU-register are positive, or if both the Compare and Greater-Than flip-flops are set (refer to plate P-29).

2-209. JPALP (f = 65). The JPALP instruction performs a direct program jump to the address specified by Up if the Compare flip-flop is clear and the contents of the AL-register are positive, or if both the Compare and Greater-Than flip-flops are set (refer to plate P-29).

2-210. JPAUNG (f = 66). The JPAUNG instruction performs a direct program jump to the address specified by Up if the Compare flip-flop is clear and the contents of the AU-register are negative, or if the Compare flip-flop is set and the Greater-Than flip-flop is clear.

2-211. JPALNG (f = 67). The JPALNG instruction performs a direct program jump to the address specified by Up if the Compare flip-flop is clear and the contents of the AL-register are negative, or if the Compare flip-flop is set and the Greater-Than flip-flop is clear.

2-212. BJP (f = 73). The BJP instruction checks the contents of the B-register. If B equals a positive zero, nothing is affected, if B does not equal a positive zero, B is decremented by one and a direct program jump is initiated to the address designated by Up.

2-213. I-Sequence Data Flow for f = 34, 35, and 60 through 67, and I and Next I-Sequence Data Flow for f = 73. Refer to figure 2-36 for a block flow diagram description of the execution of jump instructions f = 34, 35, and 60 through 67, and to figure 2-37 for execution of jump instruction f = 73. Refer to table 2-39 for a list of I-Sequence commands for f = 34, 35, and 60 through 67, and table 2-40 for a list of I and next I-sequence commands for f = 73. Those commands that have not been previously described in an I-sequence are described in table 2-41 in detail.

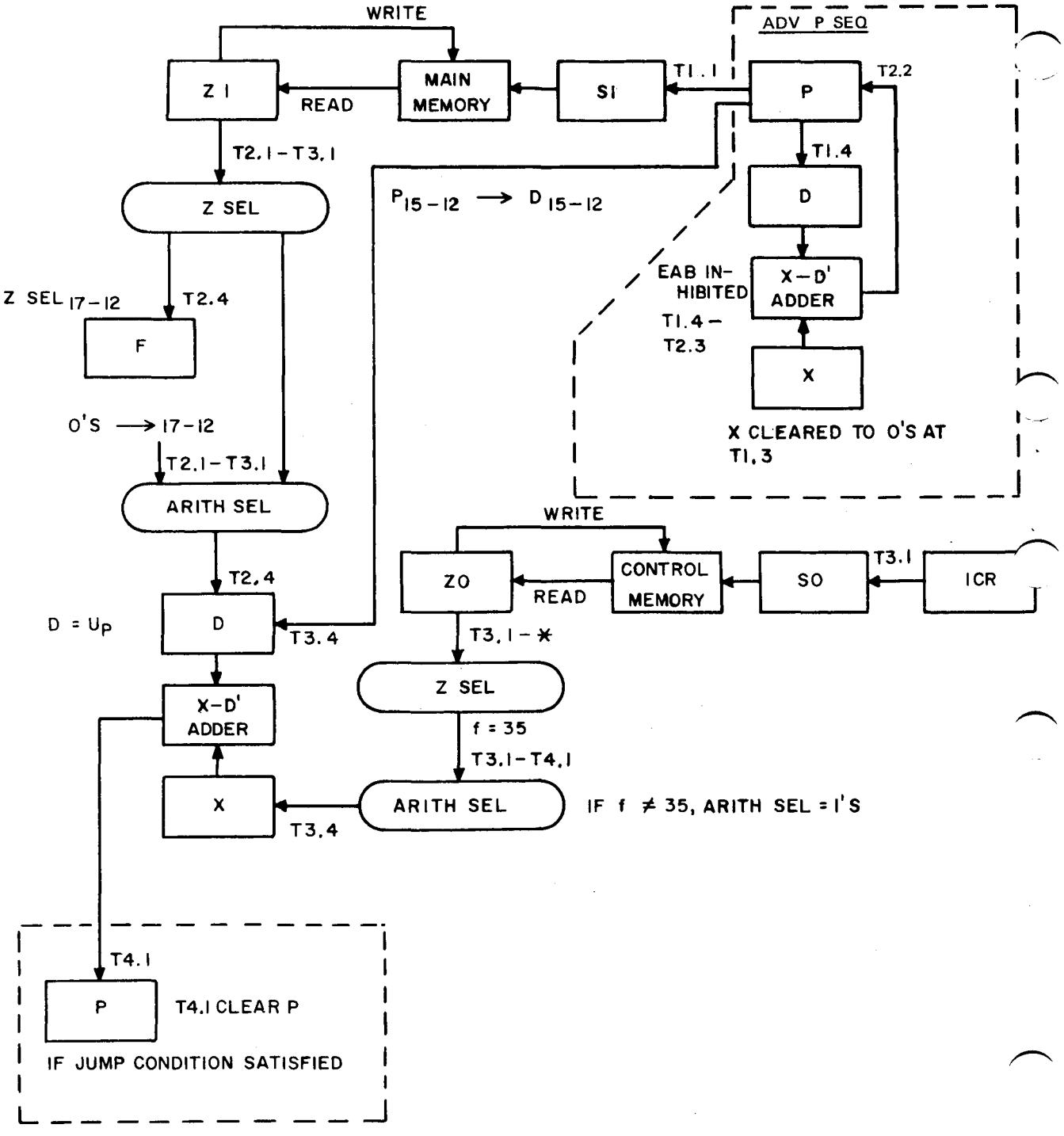


Figure 2-36. I-Sequence Data Flow for f = 34, 35, 60-67

TABLE 2-39. I-SEQUENCE ESSENTIAL COMMANDS FOR f = 34, 35, 60 - 67

Time Notation	Commands
T4.4	Clear S1
T1.1	P → S1, Init Memory, *Set Incr P ff
T1.3	*Clear D, *clear X, clear Z1, clear F, *set OXL11 ff
T1.4	*$P_L \rightarrow D_L$, *$P_U \rightarrow D_U$, *set Inhib EAB ff
T2.1	*Clear Incr P ff, Z1 → Z-Sel, Z-Sel → Arith Sel, 0's → Arith Sel_{17-12}
T2.2	*Clear P, *Adder → P
T2.3	Clear X, clear D, *clear OXL11 ff, *clear Inhib EAB ff
T2.4	$Z\text{-}Sel_{17-12} \rightarrow F$, Arith Sel → D
T3.1	Drop Z1 → Z-Sel, drop Z-Sel → Arith Sel, drop 0's → Arith Sel_{17-12}
	ICR → SO, Init CM, ZO → Z-Sel**, Z-Sel → Arith Sel if f = 35
T3.4	$P_{15-12} \rightarrow D_{15-12}$, Arith Sel → X
T4.1	Drop Z-Sel → Arith Sel
T4.2	***Clear P if jump satisfied
	***Adder → P if jump satisfied

NOTES: * These events are concerned with or are controlled by the Advance-P subsequence.
** ZO → Z-Sel occurs for duration of T24 ff clear.
***Jump condition is satisfied if:
f = 34, 35 unconditionally
f = 60 Compare ff clear • AU = +0 or Compare ff set • Equal ff set
f = 61 Compare ff clear • AL = +0 or Compare ff set • Equal ff set
f = 62 Compare ff clear • AU ≠ +0 or Compare ff set • Equal ff clear
f = 63 Compare ff clear • AL ≠ +0 or Compare ff set • Equal ff clear
f = 64 Compare ff clear • AU pos or Compare ff set • Greater ff set
f = 65 Compare ff clear • AL pos or Compare ff set • Greater ff clear
f = 66 Compare ff clear • AU neg or Compare ff set • Greater ff clear
f = 67 Compare ff clear • AL neg or Compare ff set • Greater ff clear

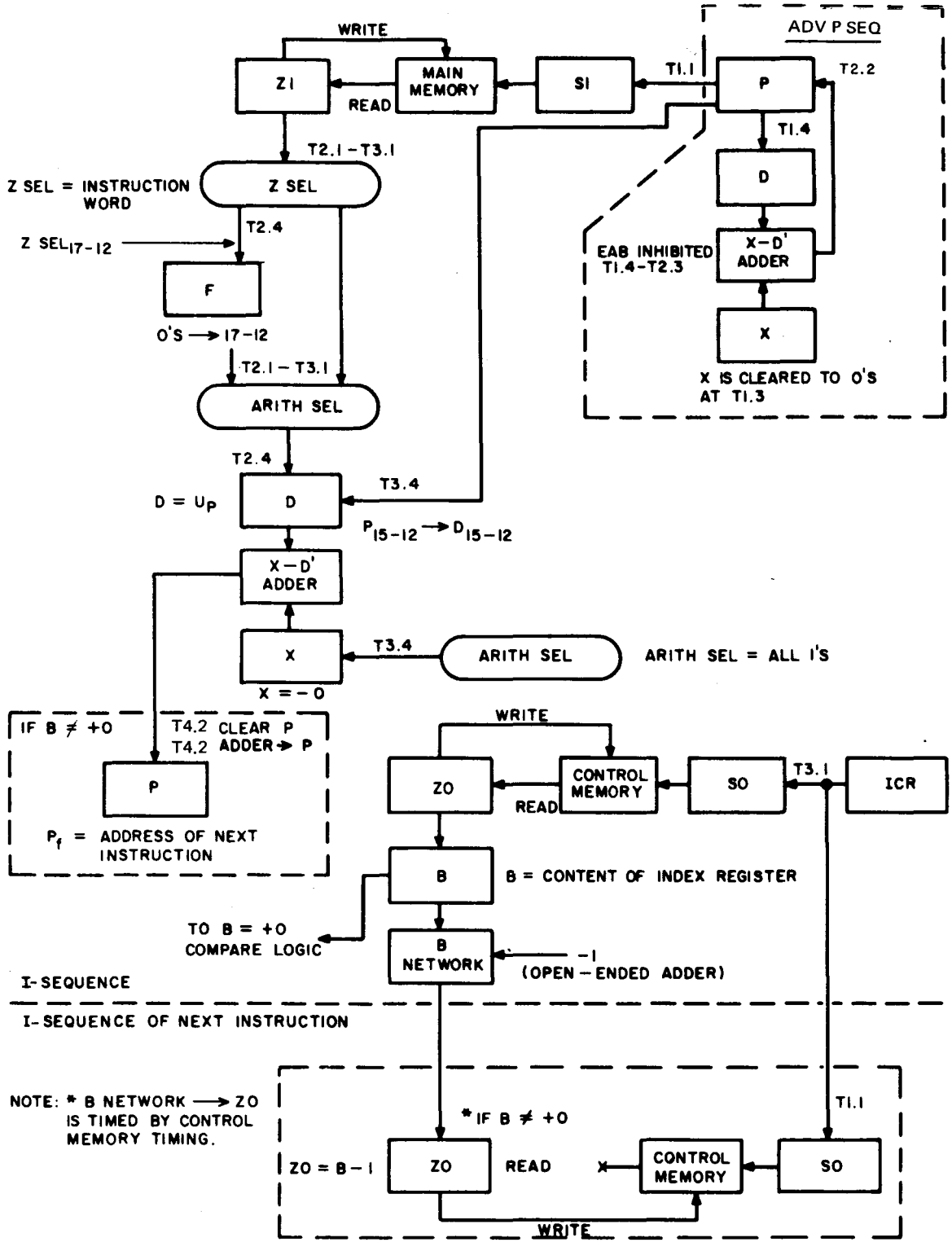


Figure 2-37. I and Next I-Sequence Data Flow For f = 73

TABLE 2-40. I AND NEXT I-SEQUENCE ESSENTIAL COMMANDS FOR f = 73

Time Notation	Commands
	I-SEQUENCE
T4.4	Clear S1
T1.1	P → S1, Init Memory, *set Incr P ff
T1.3	*Clear D, *clear F, clear Z1, *set OXL11 ff
T1.4	$*P_L \rightarrow D_L$, $*P_U \rightarrow D_U$, *set Inhib EAB ff
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel, 0's → Arith Sel_{17-12} *Clear Incr P ff
T2.2	*Clear P, *Adder → P
T2.3	Clear D, clear X, *clear OXL11 ff, *clear Inhib EAB ff
T2.4	$\text{Z-Sel}_{17-12} \rightarrow$ F, Arith Sel → D
T3.1	ICR → SO, Init CM, drop Z1 → Z-Sel, drop Z-Sel → Arith Sel drop 0's → Arith Sel_{17-12}
T3.2	Clear B
T3.4	$P_{15-12} \rightarrow D_{15-12}$, Arith Sel → X, ZO → B
T4.1	Clear B ± 1 ff
T4.2	Clear P if B ≠ +0
T4.2	Set B ± 1 ff (-1 → B Network) Adder → P if B ≠ +0
T4.4	Disable CM → ZO**
	I-SEQUENCE OF NEXT INSTRUCTION
T1.1	ICR → SO Init CM if B ≠ +0
T1.4	Drop disable CM → ZO

* These events are concerned with or are controlled by the Advance-P subsequence.
** B network → ZO is timed by control memory timing.

TABLE 2-41. INSTRUCTIONS/DESCRIPTIONS FOR f = 34, 35, 60-67 AND 73
(NOT PREVIOUSLY DESCRIBED IN AN I-SEQUENCE)
(REFER TO TABLE 2-19 FOR ALL COMMANDS NOT DESCRIBED)

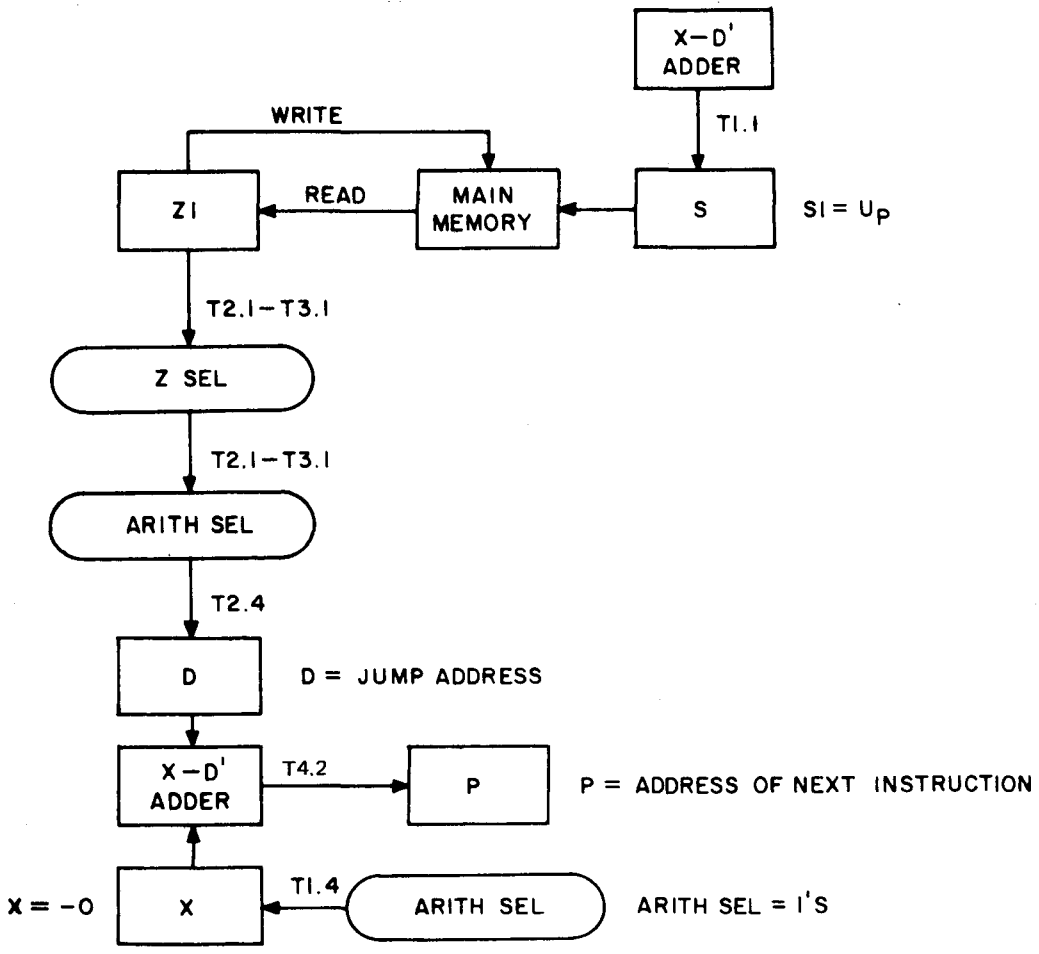
Instruction	Description
JP, JPB (f = 34, 35)	AND gate 13G34 (refer to plate P-29) is qualified by highs at pins 11, 10, and 6, signifying that the instruction is not f = 60 through 67 or f = 73. This gate outputs a L ⇒ Jump to AND-OR gate 10N07 (refer to plate P-21). Other enables (L ⇒ 34, 35, 60 - 67, 73 and L ⇒ Format I) are received from the function translators and at time T4.2 of an I-sequence, 10N07 outputs a high which is inverted through OR gates 07N07 and 18N07 and applied to AND gates 08N07 and 19N07 respectively. A Ø2 qualifies 08N07 which outputs a high via OR gate 09N07 to clear the P-register. This high is produced with a slight delay through 11N07 to qualify 19N07, which then outputs a L ⇒ Adder → P, the contents of the adder being the address to which the jump is to be made. NOTE: Jumps for f = 60 through 67 are quite similar in operation. AND gate 13G34 is disabled at pins 11 and 10 by a L ⇒ f = 60 through 67 to make possible the evaluation of the jump.
JPAUZ (f = 60)	The Compare flip-flop is not set. If AU is clear, AND gates 90A31, 90A27, 90A22, and 90A18 (refer to plates P-97, and P-98) outputs highs to AND gate 91A18 (refer to plate P-29). A L ⇒ AU = O from this gate and a L ⇒ f = Even from the function translators qualify AND-OR gate 20G34. AND-OR gate 22G34 is disqualified by the high on pin 8 and the lack of a L ⇒ f = X4 through X7 on pin 5. It outputs a low which is ANDed with a L ⇒ f = X0, X1 X4, X5 at 24G34 causing a high to be placed on pin 8 of 13G34. Simultaneously, a high from the set side of the Compare flip-flop (signifying that it is not set) furnishes the final enable to cause 13G34 to output a L ⇒ Jump. The operation then proceeds the same as an f = 34 instruction. If the Compare and Equal flip-flops have been set by the previous instructions, a high from the zero side of the Compare flip-flop qualifies pin 9 of 13G34. A low from the Equal flip-flop is ANDed with a L ⇒ f = X0 through X3 at 10G34, as the high output is inverted by 11G34 and applied to 12G34. A L ⇒ f = X0, X1, X4, X5 then causes 12G34 to output a high, qualifying 13G34. The jump then proceeds as outlined.
JPALZ (f = 61)	The f = 61 instruction is executed in the same manner as f = 60, but utilizes AND gate 91A00 instead of 91A18.

TABLE 2-41. INSTRUCTIONS/DESCRIPTIONS FOR f = 34, 35, 60 - 67 and 73 (Cont)
(NOT PREVIOUSLY DESCRIBED IN AN I-SEQUENCE)

Instruction	Description
JPAUNZ, JPALNZ (f = 62, 63)	Instructions f = 62, 63 are similar to preceding instructions, but qualify gates which were disqualified and disable gates which were enabled during execution of f = 60, 61 instructions. However, gates 12G34 and 24G34 must output highs to enable the respective jumps, just as before, and are qualified via the second set of inputs, pins 11 and 12.
JPAUP, JPALP, JPAUNG, JPALNG (f = 64 - 67)	With the Compare flip-flop clear, decisions for instructions f = 64 through 67 are handled by AND-OR gate 21G34. The zero sides of flip-flops OXA35 and OXA17 are sampled and ANDed with L ⇒ f = Even or L ⇒ f = Odd depending on the instruction. A combination such as f = 64 and OXA35 clear will qualify 21G34 which will output a high that disables 22G34, since pin 7 is also high. The jump operation then proceeds as previously described. Minor variations exist for the other instructions which are similar to those previously described for the f = 60 through 63 instructions. If the Compare flip-flop is set, the output of the Greater flip-flop is monitored for f = X4 through X7 by 10G34 in the same fashion as it monitors and gates the output of the Equal flip-flop for instructions f = X0 through X3. All jumps in the f = 60 through 67 group proceed in essentially the same manner with only minor individual variations.
BJP (f = 73)	During the I-sequence, the instruction word is obtained from memory and the jump address (Up) is formulated in the D-register. A control memory reference is initiated to obtain the contents of the B-register (control memory). If the B-register value is a positive zero, no further action is taken. If the value is not a positive zero, the P-register is cleared and receives the contents of the adder (Jump address). During the next I-sequence, if B = 0, a jump is initiated. If B ≠ 0, a one is subtracted from the contents of B by the B-1 network, and this new value is restored to its original location in control memory. The clear sides of the B-register flip-flops (refer to plates P-118 through 120) cause one of the AND gates 90B04, 90B00, 90B09, 90B13 (refer to plate P-107) to output a low if any flip-flop is set (B ≠ 0). The low will disqualify 91B00 and it will output a high. If B = 0, 91B00 will produce a L ⇒ B = 0. If B ≠ 0 and f = 73, AND-OR gate 15G34 (refer to plate P-29) outputs a L ⇒ Jump P and 13G34 will be qualified due to the lack of lows implying f = 60 through 67 and B = 0 on pins 11, 10 and 5, respectively, causing it to produce a L ⇒ Jump. The L ⇒ Jump is ANDed with a L ⇒ T42 I-Seq, L ⇒ f = 34, 35, 60, 67, 73, and L ⇒ Format I to qualify 10N07 (refer to plate P-21).

TABLE 2-41. INSTRUCTIONS/DESCRIPTIONS FOR f = 34, 35, 60 - 67 and 73 (Cont)
(NOT PREVIOUSLY DESCRIBED IN AN I-SEQUENCE)

Instruction	Description
BJP (f = 73) (cont)	A high output from 10N07 causes 09N07 and 19N07 to output Clear P and Adder → P signals in the same manner as during an Advance-P subsequence. A L ⇒ T42 I-Seq and a L ⇒ f = 73 and ANDed with a ∅2 to set the B±1 flip-flop OXG37 (refer to plate P-27). Setting this flip-flop causes 03G37 to output a L ⇒ -1 → B to the B-adder network. This decrements the contents of B by 1. If B remains B ≠ 0, the L ⇒ Jump P from 15G34 (refer to plate P-29) is ANDed with a L ⇒ I Seq, L ⇒ f = 32, 33, 36, 37, 56, 73 and with a low from 03T11 by 40N10 (refer to plate P-24). The high output from this is ANDed with a high from 48N10 due to the absence of a L ⇒ Format II. Its output is then ANDed with a ∅1 at 49N10 to produce an ICR → SO. At T1.4 the disable CM → ZO is dropped as described in the f = 36, 37 instruction.



NOTE: FOR f = 54, CLEAR ALL INTERRUPT LOCKOUT FF AT T3.4 OF I-SEQUENCE.

Figure 2-38. R1-Sequence Data Flow For f = 54, 55

2-214. Execution of Jump (IJPEI, IJP, RJP, IRJP, and IRJPB) Instructions for (f = 54, 55, 76, 30, and 31). Instructions IJPEI and IJP unconditionally cause an indirect program jump to an address contained in memory. In addition, the execution of the IJPEI instruction enables all interrupts. Instructions IRJP and IRJPB will store the contents of the P-register in memory and perform an indirect program jump.

2-215. IJPEI (f = 54). The IJPEI instruction causes an indirect jump address to be extracted from memory at an address determined by Up. The actual jump address is contained in the 15 least significant bits of the word extracted from memory. The All Interrupt Lockout flip-flop (refer to plate P-28) is cleared during the execution of this instruction to allow the honoring of a future interrupt.

2-216. IJP (f = 55). The IJP instruction provides the same function as f = 54 except that the All Interrupt Lockout flip-flop is not affected.

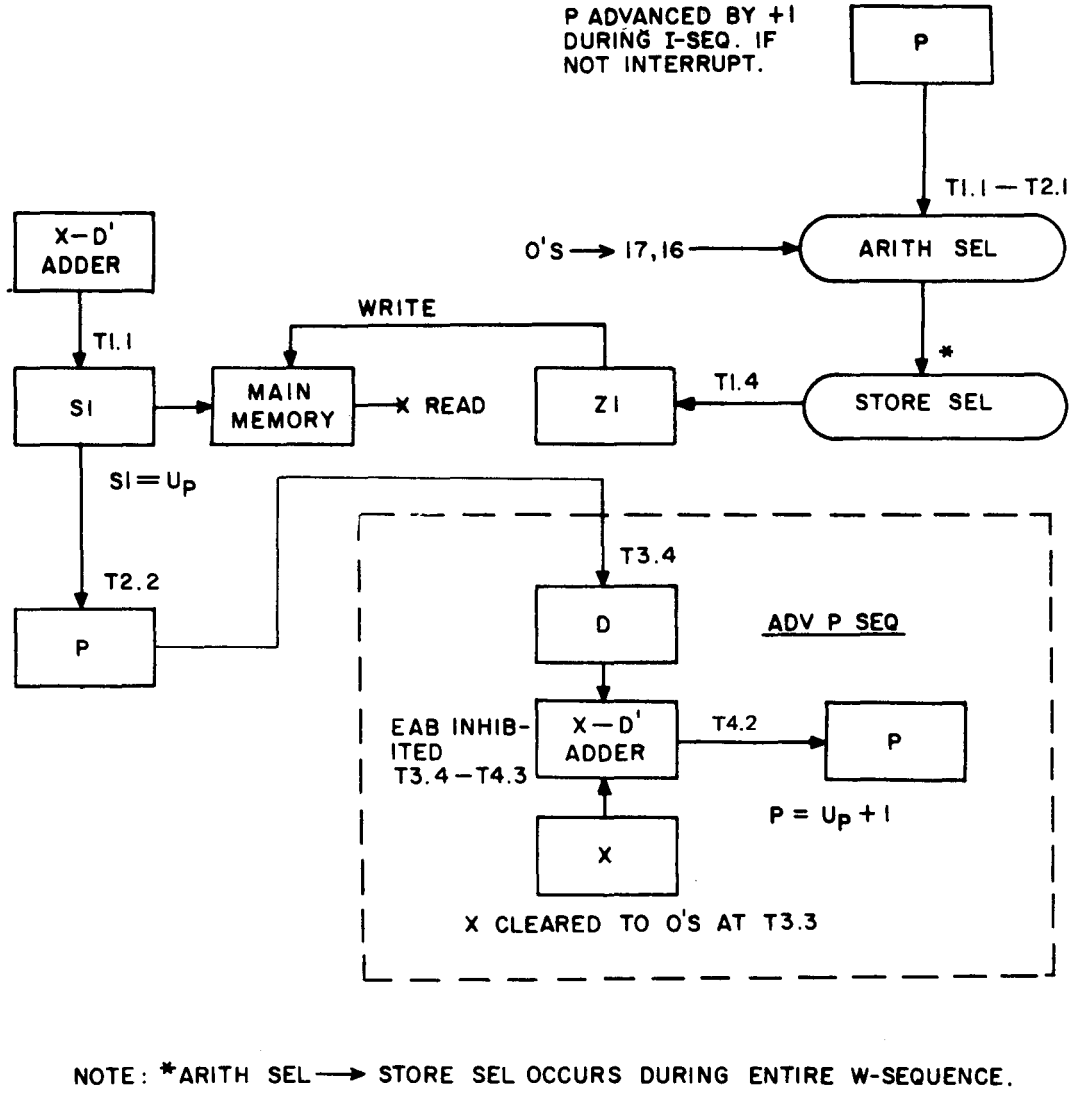
2-217. RJP (f = 76). The RJP instruction causes a jump from the main program, to perform a subroutine for example, and if desired a later direct return jump to the main program at the termination of the subroutine. Executing this instruction stores the contents of the P-register (the next sequential address of the main program) at address Up, and then jumps at address Up + 1, which is the entrance address of the subroutine.

2-218. IRJP (f = 30). The IRJP instruction is similar to a RJP instruction in that it causes a jump from the main program and a later indirect return jump if desired. The difference occurs in the method of obtaining the storage address for the P-register. In an RJP instruction, the contents of the P-register are stored at the Up address. In an IRJP instruction, the contents of the P-register are stored at a storage address obtained by the Up address. The storage address is then placed in the P-register and incremented by one. The program will then jump to the address contained in the P-register.

2-219. IRJPB (f = 31). The IRJPB instruction is similar to an IRJP instruction except for the way in which the storage address for the P-register is obtained. For the IRJPB instruction the storage address is obtained by address Up+B.

2-220. R1-Sequence Data Flow for f = 54, 55, W-Sequence Data Flow for f = 76, and R1 and W-Sequence Data Flow for f = 30, 31. Refer to figure 2-38 for a block diagram description of the execution of jump instructions f = 54, 55, figure 2-39 for the execution of f = 76, and figure 2-40 for the execution of f = 30, 31. Refer to table 2-42 for a list of R1-sequence essential commands for f = 54, 55, table 2-43 for a list of W-sequence essential commands for f = 76, and table 2-44 for a list of R1 and W-sequence essential commands for f = 30, 31. Those commands that have not been previously described in a preceding sequence are described in table 2-45 in detail.

2-221. Execution of Skip (SKP, SKPNBO, SKPOV, and SKPNOV) Instructions (f = 50:50, 50:51, 50:52, and 50:53). Like jump instructions, skip instructions provide greater flexibility in the execution of a program. Each time a nonconditional skip instruction is executed, it allows a skip of the next sequential instruction to occur, provided certain conditions are met. A conditional skip instruction depends upon some logical decision the computer must make before the skip can be executed. Instructions 50:50 through 50:53 are used to conditionally or unconditionally perform a program skip of the next sequential instruction.



NOTE: *ARITH SEL → STORE SEL OCCURS DURING ENTIRE W-SEQUENCE.

Figure 2-39. W-Sequence Data Flow For f = 76

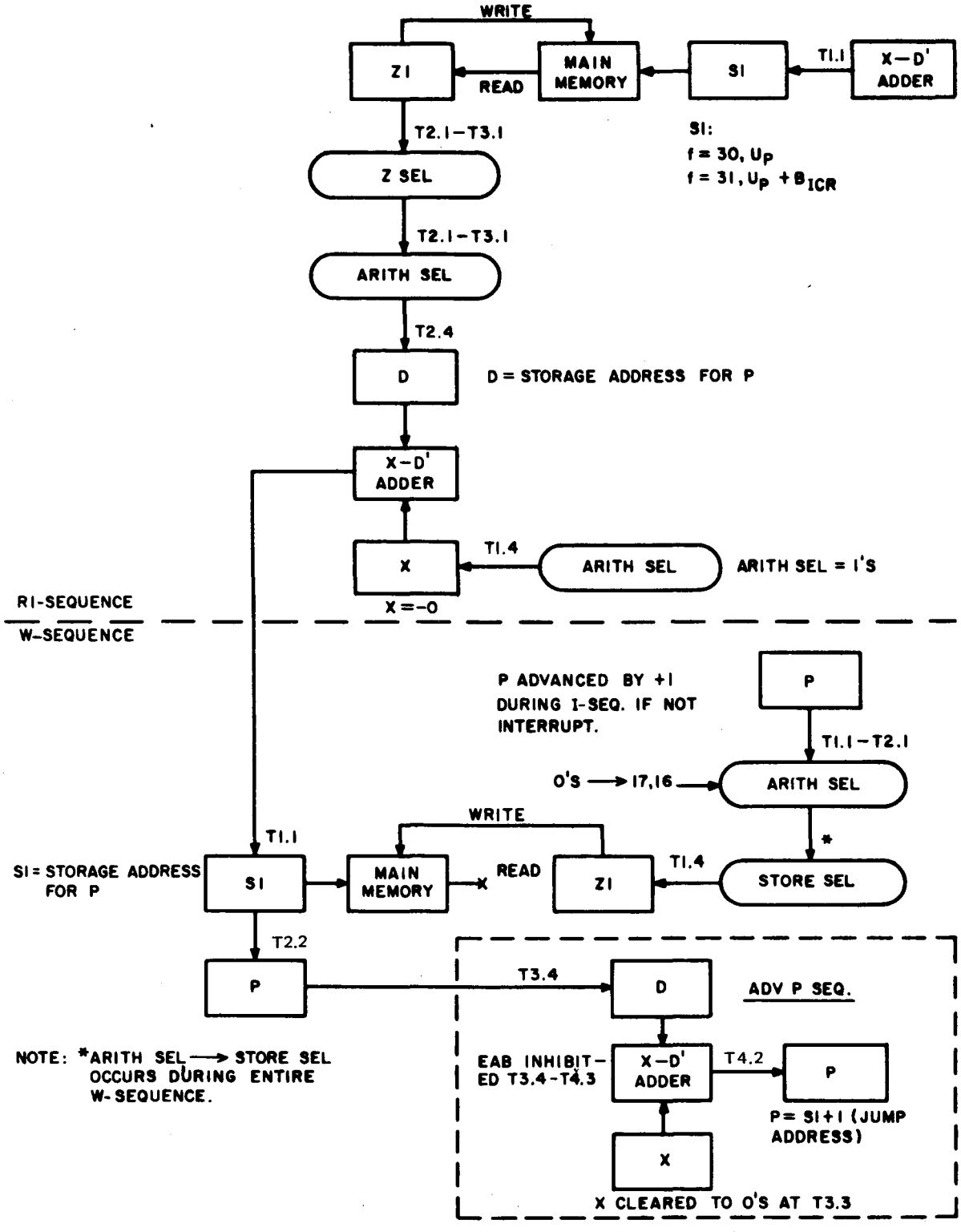


Figure 2-40. R1 and W-Sequence Data Flow For f = 30, 31

TABLE 2-42. R1-SEQUENCE ESSENTIAL COMMANDS FOR f = 54, 55

Time Notation	Commands
T4.4	Clear S1
T1.1	Adder → S1, Init Memory
T1.3	Clear Z1, clear X
T1.4	Arith Sel → X
T2.1	Z1 → Z Sel, Z-Sel → Arith Sel
T2.3	Clear D
T2.4	Arith Sel → D
T3.1	Drop Z1 → Z-Sel, drop Z-Sel → Arith Sel
T4.2	Clear P, Adder → P

TABLE 2-43. W-SEQUENCE ESSENTIAL COMMANDS FOR f = 76

Time Notation	Commands
T4.4	Clear S1
T1.1	Adder → S1, Init memory, P → Arith Sel, 0's → Arith Sel$_{17, 16}$
T1.3	Clear Z1
T1.4	*Store Sel → Z1, disable Mem → Z1
T2.1	Drop P → Arith Sel, drop 0's → Arith Sel$_{17, 16}$
T2.2	Clear P, S1 → P
T3.1	**Set Incr P ff
T3.3	**Clear D, **clear X, **set OXL11 ff
T3.4	**P_L → D_L, **P_U, → D_U, *set Inhib EAB ff
T4.1	**Clear Incr P ff
T4.2	**Clear P, **Adder → P
T4.3	**Clear OXL11 ff, **clear Inhib EAB ff

* Arith Sel → Store Sel occurs during entire W-sequence.
**These events are concerned with or are controlled by the Advance-P subsequence.

TABLE 2-44. R1 AND W-SEQUENCE ESSENTIAL COMMANDS FOR f = 30, 31

Time Notation	Commands
	R1-SEQUENCE
T4.4	Clear S1
T1.1	Adder → S1, Init Memory
T1.3	Clear Z1, clear X
T1.4	Arith Sel → X
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel
T2.3	Clear D
T2.4	Arith Sel → D
T3.1	Drop Z1 → Z-Sel, drop Z-Sel → Arith Sel
	W-SEQUENCE
T4.4	Clear S1
T1.1	Adder → S1, Init Memory, P → Arith Sel, 0's → Arith Sel$_{17, 16}$
T1.3	Clear Z1
T1.4	*Store Sel → Z1, disable Mem → Z1
T2.1	Drop P → Arith Sel, drop 0's → Arith Sel$_{17, 16}$
T2.2	Clear P S1 → P
T3.1	**Set Incr P ff
T3.3	**Clear D, **clear X, **set OXL11 ff
T3.4	**$P_L \rightarrow D_L$, **$P_U \rightarrow D_U$, **set Inhib EAB ff
T4.1	**Clear Incr P ff
T4.2	**Clear P, *Adder → P
T4.3	**Clear OXL11 ff, **clear Inhib EAB ff

*Arith Sel → Store Sel occurs during entire W-sequence.

**These events are concerned with or are controlled by the Advance-P subsequence.

TABLE 2-45. INSTRUCTIONS/DESCRIPTIONS FOR f = 54, 55, 76, 30, and 31
(NOT DESCRIBED IN A PREVIOUS SEQUENCE)
(REFER TO TABLES 2-21 AND 2-22 FOR COMMANDS NOT DESCRIBED)

Instruction	Description
IJPEI, IJP (f = 54, 55)	Jumps for f = 54, 55 use an I-sequence and an RI-sequence. During the I-sequence, the instruction word is called out of memory, translated, Up formulated, and if f = 54, the All Interrupt Lockout flip-flop is cleared at time T3.4 (refer to plate P-28). Inputs to AND-OR gate 00E70 are L ⇒ T34 I-Seq, L ⇒ f = 54, 55, and L ⇒ f = Even, which qualify it and OR gate 01E70 to place a low at pin 14 of the All Interrupt Lockout flip-flop. A ∅4 is ANDed with this low to clear the flip-flop OXG70. During the R1-sequence, the 54, 55 instructions proceed normally until time T3.1 when the Z1 → Z Sel and the Z Sel → Arith Sel signals are dropped as described in the R1-Sequence. Then at time T4.2 AND-OR gate 10N07 (refer to plate P-21) is qualified by a L ⇒ f = 54, 55 and a L ⇒ T42 R1, W-Seq. The output is applied to OR inverters 07N07 and 18N07 to Clear P and gate Adder → P at time T4.2.
RJP (f = 76)	This jump instruction uses an I and a W-sequence. The I-sequence formulates Up and increments P by 1. At time T1.1 of the W-sequence the adder outputs address Up. A memory cycle is initiated, and at time T1.4, the incremented contents of the P-register are stored at the address as designated by Up. The P-register is cleared, and at time T2.2, Up is gated into the P-register from the S1-register. At time T3.1 an Advance-P subsequence is initiated to increment the P-register by +1. The actual jump occurs as the I-sequence of the next instruction, whose address was obtained from the P-register (Up+1). When the part of the sub-program to which the computer jumps is completed and f = 30, referencing Up may be used to return to the main program. Note that an f = 76 instruction is useful only in intra bank jumps (as designated by Up).
IRJP, IRJPB (f = 30, 31)	These jump instructions make use of an I, R1, and W-sequence in that order. I/O permits inter-bank and B-modifiable jumps to be made. At time T1.1 of the R1-sequence, the adder outputs address Up or Up+B and initiates a memory cycle to obtain from memory the storage address for the contents of the P-register. At time T1.1 of the W-sequence, the adder outputs the storage address to the S1-register and initiates a memory cycle. At time T1.4, the contents of the P-register, as incremented during the I-sequence, are written into memory at the storage address designated by the contents of Up or Up+B contained in the S1-register. At time T2.2, the contents of the S1-register (storage address) are transferred to the P-register; an advance subsequence is initiated at time T3.1. The P-register now contains the storage address +1. The program jump is actually performed as the I-sequence of the next instruction whose address was obtained from the P-register (storage address +1). After completion of the subroutine, another f = 30 instruction referencing the storage address may be used to return to the main progra

2-222. SKP (f = 50:50). The SKP instruction performs a program skip if bit 5 of the instruction word equals a 1, or if any bit position from 4 through 0 of the instruction word equals a 1, and the corresponding control panel Skip key is set.

2-223. SKPNBO (f = 50:51). The SKPNBO instruction performs a program skip if the Borrow Test flip-flop is clear (refer to plate P-30). This flip-flop would have set if, during the execution of a double add or double subtract instruction, an End Around Borrow (EAB) condition had developed. Since the double add or double subtract instruction actually prohibits the production of an EAB, erroneous arithmetic computations can result. The 50:51 instruction will detect an EAB condition and allow the program to compensate for it.

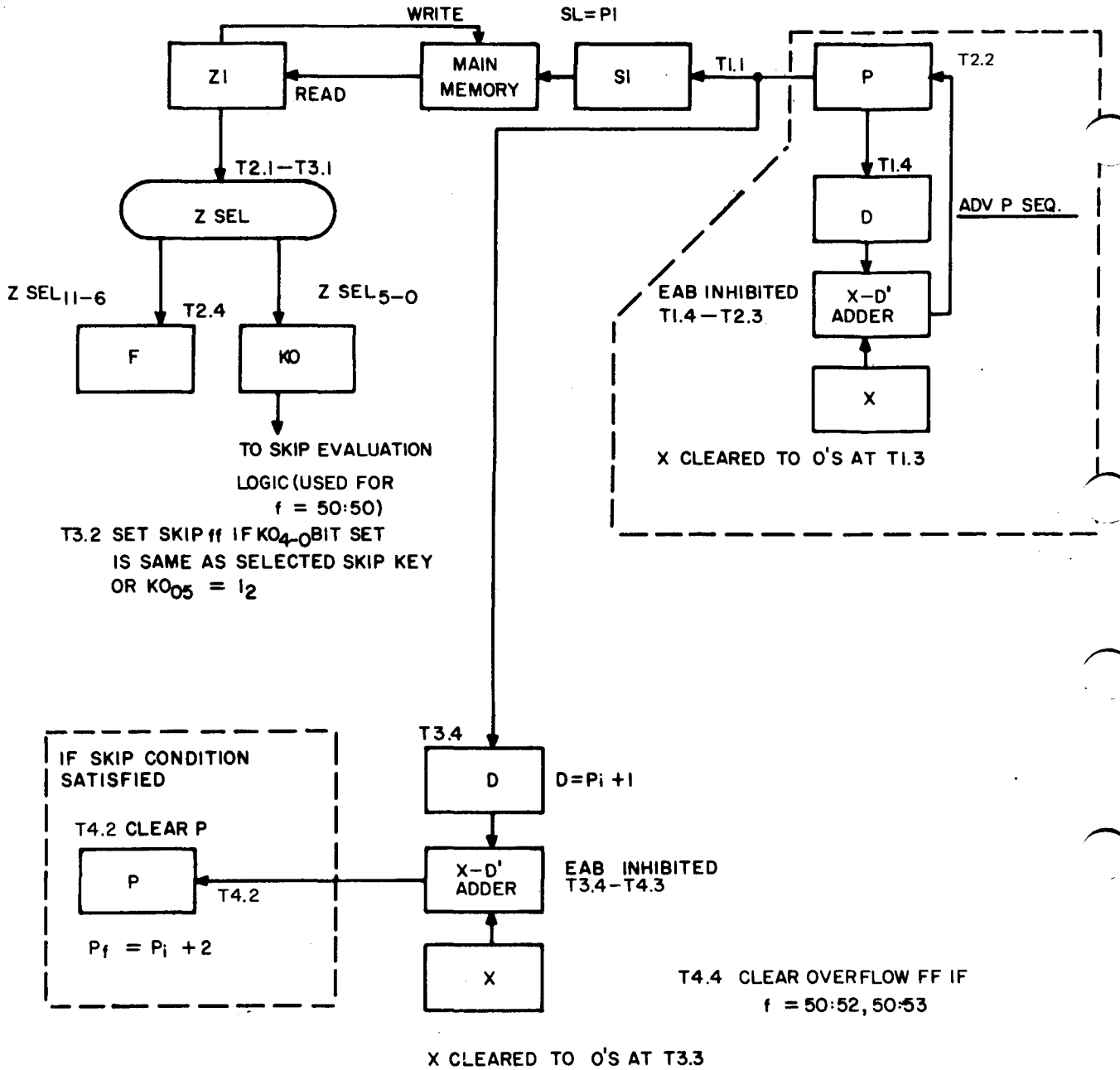
2-224. SKPOV (f = 50:52). The SKPOV instruction provides a program skip if the Overflow flip-flop is set (refer to plate P-30). The Overflow flip-flop will be cleared during the execution of this instruction.

2-225. SKPNOV (f = 50:53). The SKPNOV instruction provides a program skip if the Overflow flip-flop is clear.

2-226. I-Sequence Data Flow for f = 50:50 through 50:53. Refer to figure 2-41 for a block diagram description of the execution of f = 50:50 through 50:53, and table 2-46 for a list of I-Sequence essential commands for f = 50:50 through 50:53. Those commands that have not been described in a preceding sequence are described in table 2-47 in detail.

2-227. Execution of Skip (BSK) Instruction (f = 56). The BSK instruction performs a program skip if the contents of the B-register are equal to an operand obtained from memory. If they are not equal, the B-register is incremented by +1.

2-228. R1 and Next I-Sequence Data Flow for f = 56. Refer to figure 2-42 for a block diagram description of the execution of BSK instruction (f = 56). The I-sequence, which obtained the instruction word from memory and formulated Up (the address of the operand in memory), is not shown in this figure. The R1-sequence, which is shown in the figure references main memory, obtains the operand, and places the complement of the operand into the D-register. A control memory reference is also initiated, and the contents of the B-register are placed into the X-register. The adder is used for the comparison. If they are equal to each other, (B) = (Y), the Equal flip-flop (refer to plate P-29) is set and an additional advance-P subsequence is initiated. Under these circumstances, the next sequential instruction is skipped. A second control memory reference is used to again obtain the contents of the B-register. This value is placed into the BU-register and incremented by the +1 network. This value is used only if the Equal flip-flop is clear, indicating that the B-register contents are not equal to the operand (B) $\neq$ (Y). Under these circumstances, another control memory is initiated during the I-sequence of the next instruction. This allows the incremented value (B+1) to be stored in the B-register. Refer to table 2-48 for a list of the R1 and next I-sequence essential commands used if f = 56. Those commands that have not been described in preceding sequences are described in table 2-49 in detail.



NOTE: * SKIP IF f = 50:50, SKIP FF SET
f = 50:51, BORROW TEST FF CLEAR
f = 50:52, OVERFLOW FF SET
f = 50:53, OVERFLOW FF CLEAR

Figure 2-41. I-Sequence Data Flow for f = 50:50 through 50:53

TABLE 2-46. I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:50 through 50:53

Time Notation	Commands
T4.4	Clear S1
T1.1	P → S1, Init memory, *set Incr P ff
T1.3	*Clear D, *clear X, clear Z1, clear F, *set OXL11 ff
T1.4	$*P_L \rightarrow D_L$, $*P_U \rightarrow D_U$, clear KO, *set Inhib EAB ff
T2.1	Z1 → Z-Sel, *clear Incr P ff
T2.2	*Clear P, *Adder → P
T2.3	*Clear OXL11 ff, *clear Inhib EAB ff
T2.4	$Z\text{-}Sel_{11-6} \rightarrow F$, set OXF06 ff, $Z\text{-}Sel_{5-0} \rightarrow KO$
T3.1	Clear Skip ff, drop Z1 → Z-Sel
T3.2	Set Skip ff if $KO_{05} = 1$ or KO_{04-00} bit set same as selected skip key
T3.3	Clear D, clear X
T3.4	$P_L \rightarrow D_L$, $P_U \rightarrow D_U$, set Inhib EAB ff
T4.2	Clear P if skip condition satisfied**
	Adder → P if skip condition satisfied**
T4.3	Clear Inhib EAB ff
T4.4	Clear Overflow ff if f = 50:52, 50:53

* These events are concerned with or are controlled by the Advance-P subsequence.
**Skip condition satisfied if: f = 50:50, Skip ff set
f = 50:51, Borrow Test ff clear
f = 50:52, Overflow ff set
f = 50:53, Overflow ff clear

TABLE 2-47. INSTRUCTIONS/DESCRIPTIONS FOR f = 50:50 through 50:53
(NOT DESCRIBED IN A PREVIOUS SEQUENCE)
(REFER TO TABLE 2-19 FOR COMMANDS NOT DESCRIBED)

Instruction	Description
SKP (f = 50:50)	This instruction is similar to a Stop instruction (refer to plate P-32). If Skip keys are set, one or more of the XXG50 gates will feel a high and output a low to AND-OR gate 16G50 (refer to plate P-31). If the corresponding bit of the KO-register is set 16G50 will output a high to invert 17G50. (For example, if Skip key 4 is set & KO = $20_{(8)}$.) A KO = $40_{(8)}$ alone will also cause OR-inverter 17G50 to output a low. This low is ANDed with Ø3 T32 and Ø2 to set the Skip flip-flop OXG50. A H ⇒Key Set from 01G50 will disqualify pin 14 of 20G50 (refer to plate P-30). Only pin 13 of 20G50 will feel a low (f = 50:50) and all other AND gates will be disabled by the lack of the proper function, allowing 20G50 to output a L ⇒ Skip. This signal will be ANDed with a L⇒f = 50:21 through 23, 50 through 55, 57, and T42 I-Seq by 10N07 in order to clear P and gate Adder → P in the same manner as a normal Advance-P subsequence (refer to plate P-21 This extra subsequence increments P once more (the first increment took place at time T2.2) affecting a skip of one instruction location.
SKPNBO (f = 50:51)	With the exception of pins 9 and 10, AND-OR gate 20G50 is disqualified by this instruction (refer to plate P-30). Pin 9 feels a L ⇒f = 51, so if the Borrow Test flip-flop OXG51 is set, 20G50 will output a high. However, if OXG51 is clear, 20G50 will be completely disabled, outputting a L ⇒ Skip to 10N07 which initiates the second Advance P-subsequence.
SKPOV (f = 50:52)	If the Overflow flip-flop OXG52 is set, a low from its set side is ANDed with a L ⇒ f = X2, X3 by 10G52, producing a high which is inverted by 11G52 and applied to 12G52 (refer to plate P-30). A L ⇒ f = Even causes 12G52 to output a high, disabling pin 6 of 20G50. Since only pin 5 of 20G50 is enabled, the gate outputs a L ⇒ Skip to 10N07 to Advance-P once more, and accomplishes the skip.
SKPNOV (f = 50:53)	When OXG52 is clear, 10G52 is fully disabled and outputs a low which is applied to pin 11 of 12G52 (refer to plate P-30). A L ⇒f = Odd then causes the gate to output a high to pin 6 which disables 20G50. A L ⇒ Skip effects the skip via 10N07. If a SKPOV or a SKPNOV is affected, a L ⇒ T44 I-Seq Format II is ANDed with a L ⇒ f = 52, 53, by 30E52. OR-inverter 31E52 then outputs a low which is ANDed with a Ø4 to clear the Overflow flip-flop OXG52.

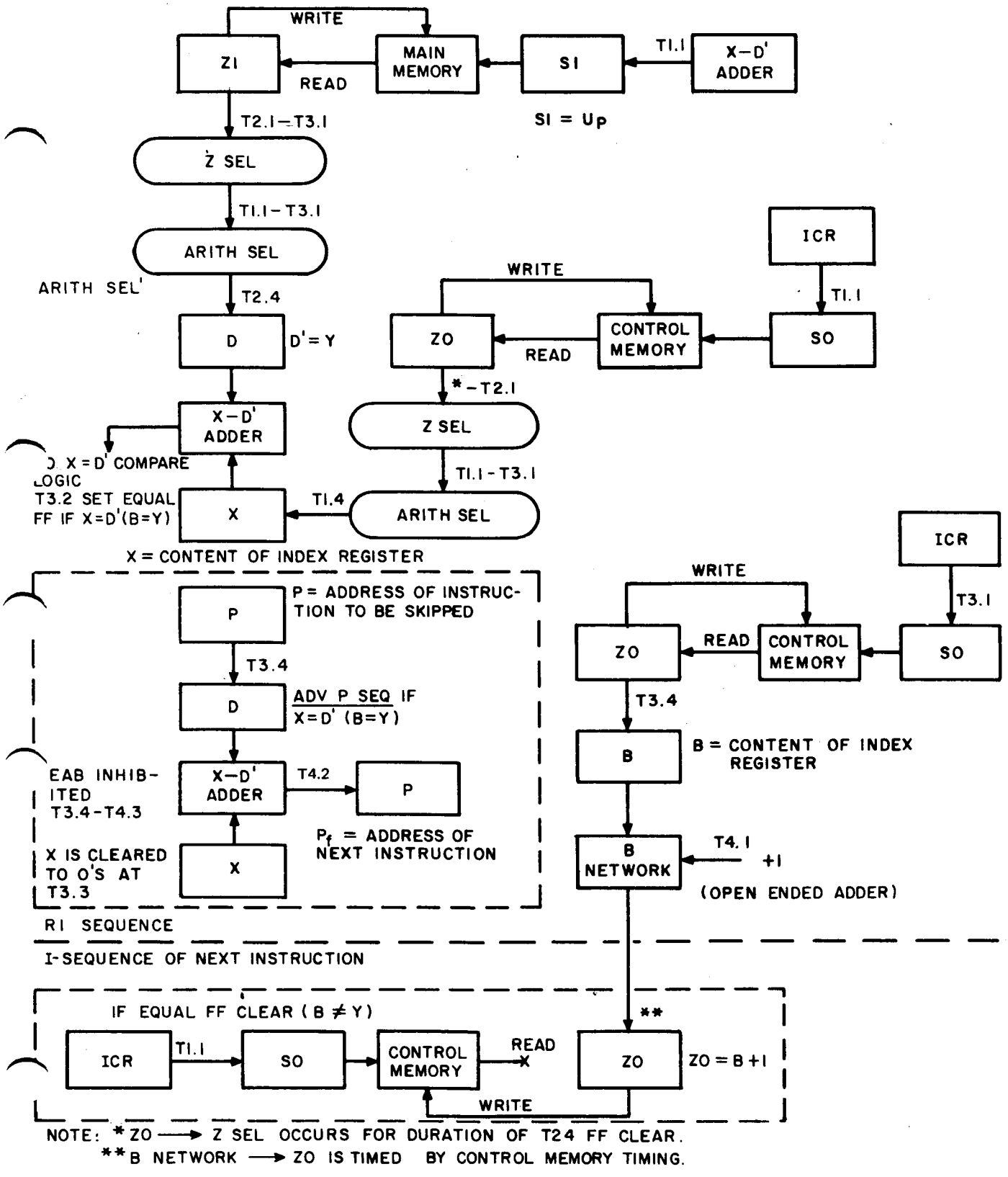


Figure 2-42. R1 and Next I-Sequence Data Flow for f = 56

TABLE 2-48. R1 AND NEXT I-SEQUENCE ESSENTIAL COMMANDS FOR f = 56

Time Notation	Commands
	R1-SEQUENCE
T4.4	Clear S1
T1.1	Adder → S1, Init Memory, ICR → SO, Init CM, *Z-Sel → Arith Sel
T1.3	Clear Z1, clear X
T1.4	Arith Sel → X
T2.1	Z1 → Z-Sel, *drop ZO → Z-Sel
T2.3	Clear D
T2.4	Arith Sel' → D
T3.1	ICR → SO, Init CM, set Incr P ff, drop Z1 → Z-Sel, drop Z-Sel → Arith Se
T3.2	Clear B, set Equal ff if X = D', clear Incr P ff if X ≠ D'
T3.3	**Clear D, **clear X, **set OXL11 ff
T3.4	$**P_L \rightarrow D_L$, $**P_U \rightarrow D_U$, **set Inhib EAB ff, ZO → B
T4.1	**Clear Incr P ff, clear B±1 ff (-1 B Network)
T4.2	**Clear P, **Adder → P
T4.3	**Clear OXK11 ff, **clear Inhib EAB ff
T4.4	Disable CM → ZO**
	I-SEQUENCE OF NEXT INSTRUCTION
T1.1	ICR → SO & Init CM if Equal ff clear
T1.4	Drop disable CM → ZO

* ZO → Z-Sel occurs for duration of T21 ff clear.

** These events are concerned with or are controlled by the Advance-P subsequence which is disabled if X = D' (B = Y) at T3.2 time.

***B-network → ZO is timed by control memory timing.

TABLE 2-49. COMMANDS/DESCRIPTIONS FOR THE EXECUTION OF BSK INSTRUCTION (f = 56)
(NOT DESCRIBED IN A PREVIOUS SEQUENCE)
(REFER TO TABLE 2-21 FOR COMMANDS DESCRIBED PREVIOUSLY)

Time Notation	Command/Description
T1.1	Z-Sel → Arith Sel. Gate 40N01 (refer to Plate P-17) is qualified by a L ⇒ 42, 43, 56 and a L ⇒ T14 R1, W-Seq and the high output is inverted by 48N01 to produce a L ⇒ Z → Select via 49N01.
T3.1	Set Incr P ff. A L ⇒ R1-Seq is ANDed with a L ⇒ f = 56 and a low from ∅3 T31 to qualify 10L10, and start an Advance-P subsequence (refer to plate P-35).
T3.2	Seq Equal ff if X = D'. A L ⇒ R1-Seq is ANDed with a ∅2 and a ∅1 T32 to enable the Equal flip-flop (OXG34) via 10E34. If X' = D (X = D'), then X-D', the output of the adder will be zero. Each bit pair of X and D is compared by the 10AXX gates of the adder (refer to plate P-94 through P-96) and if $X_N = D'_N$, a low is gated to the 80AXX gates (refer to plate P-29). All low inputs will cause the 80AXX gates to output highs to 81A00 which, when qualified, output a L ⇒ D = X'. This low will set the Equal flip-flop at time T3.2. Clear Incr P ff if X ≠ D'. The high from 81A00 (refer to plate P-29) is inverted by 82A00 and applied to AND gate 20L10 as a L ⇒ D ≠ X, (refer to plate P-35). The enables which were used to set the Increment P flip-flop at time T3.1 (L ⇒ f = 56, L ⇒ R1-Seq) are also applied to 20L10. If X ≠ D', the gate is qualified and enables the clearing of the increment P flip-flop at time T3.2 via 21L10.
T4.4	Disable CM → Z0. Same as described for a 36, 37 instruction.
T1.1	ICR → SO and Init CM if Equal ff Clear. If the Equal flip-flop (OXG34) was not set, a high from its set side will disqualify pin 5 of 15G34 (refer to plate P-29). Since only pin 6 is enabled by instruction L ⇒ f = 56, the gate will output a L ⇒ Jump P. This signal will be applied to 40N10 (refer to Plate P-24), and together with a L ⇒ I-Seq, L ⇒ f = 32, 33, 36, 37, 56, 73 and ∅3 T11, will produce a high. The lack of a L ⇒ Format II furnishes an additionally required enable for 48N10, which outputs a low to be ANDed with a ∅1 by 49N10, to produce a L ⇒ ICR → SO,

TABLE 2-49. COMMANDS/DESCRIPTIONS FOR THE EXECUTION OF BSK INSTRUCTION (f = 56) (NOT DESCRIBED IN A PREVIOUS SEQUENCE) (Cont)

Time Notation	Command/Description
T1.1 (cont)	Init CM. This occurs only if the Equal flip-flop has not been set ((B) $\neq$ (Y)).
T3.1	Drop Z-Sel → Arith Sel. Gate 40N01 (refer to Plate P-17) which was qualified at T1.1 by a L ⇒ 42, 56 and a L ⇒ T14 R1, W-Seq and kept qualified at T2.1 by a L ⇒ I, R or W-Seq and a low from 03T24, is now disabled by the loss of the low from 03T24.
T1.4	Drop disable CM → Z0. Same as described for a 36, 37 instruction.

2-229. Execution of Skip (ISK) Instruction (f = 57). The ISK instruction will perform a program skip if the operand, which was obtained from memory, is equal to a positive zero. If the operand does not equal a positive zero it will be decremented by one and replaced in memory.

2-230. R1 and W-Sequence Data Flow for f = 57. Refer to figure 2-43 for a block diagram description of the execution of ISK instruction (f = 57). The I-sequence that called the instruction word from memory and formulated the address of the operand (Up) is not shown in the figure. During the R1-sequence shown in the figure, the operand (Y) is obtained from memory and placed in the D-register. The X-register is set to all the ones so that an EAB cannot be generated. If the operand (Y) is a positive 0, the adder output is a zero.

$X = 777777_8 = Y'$

$D' = 777777_8$

000000 = adder output

If the operand (Y) is not a positive zero, the Insert EAB flip-flop (refer to plate P-27) is set and a -1 is subtracted from the operand. The operation is as follows, if the operand is a +2:

$X = 777777$

$= Y'$

$D' = 777775$

000002

1 EAB flip-flop Set
adder output

000001

During the W-sequence, the adder output is stored in memory. If the adder output is a positive zero, an Advance-P subsequence is initiated to skip, the next sequential instruction. Refer to table 2-50 for a list of R1 essential commands and table 2-51 for a list of W-sequence essential commands for the execution of ISK. Those commands that have not been described in preceding sequences are described in table 2-52 in detail.

2-231. Execution of Skip (SKPODD, SKPEVN) Instructions (f = 50:54, 50:55). The SKPODD, SKPEVN instructions perform a logical product (AND function) of the contents of the AU and AL-registers. The parity of the results is evaluated, and will cause a program skip of the next sequential instruction if the parity is odd (f = 50:54) or even (f = 50:55).

> NOTE: Parity refers to the number of ones that are present in a word. If the number of ones is even, an even parity is indicated. If the number of ones is odd, an odd parity is indicated. The parity evaluation function of the computer allows the program to check the parity of a particular word to determine whether or not it reflects the previously established odd or even parity.

2-232. I-Sequence Data Flow For f = 50:54, 50:55. Refer to figure 2-44 for a block diagram description of the execution of SKPODD, and SKPEVN instructions (f = 50:54, 50:55) and to table 2-53 for a list of I-sequence essential commands for the execution of f=50:54, 50:55. Those commands that have not been described in preceding sequence are described in table 2-54 in detail.

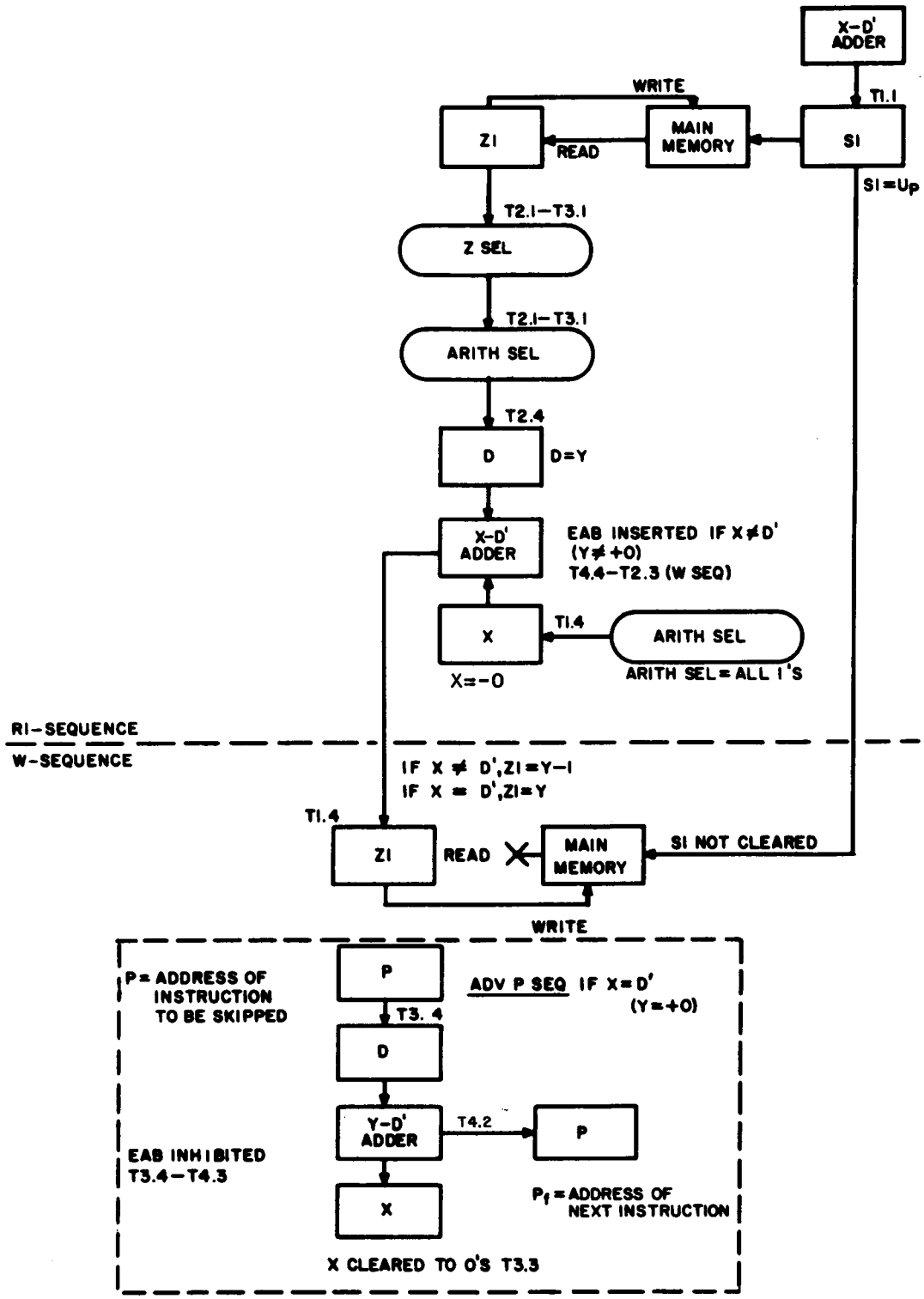


Figure 2-43. R1 and W-Sequence Data Flow For f = 57

TABLE 2-50. R1 SEQUENCE ESSENTIAL COMMANDS FOR f = 57

Time Notation	Commands
	R1-SEQUENCE
T4.4	Clear S1
T1.1	Adder → S1, Init Memory
T1.3	Clear Z1, clear X
T1.4	Arith Sel → X
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel
T2.3	Clear D
T2.4	Arith Sel → D
T3.1	Drop Z1 → Z-Sel, drop Z-Sel → Arith Sel
T4.4	Set Insert EAB ff if $X \neq D'$

TABLE 2-51. W-SEQUENCE ESSENTIAL COMMANDS FOR f = 57

Time Notation	Commands
T1.1	Init Memory
T1.3	Clear Z1
T1.4	Adder → Z1, disable Mem → Z1
T2.3	Clear Insert EAB ff
T3.1	*Set Incr P ff if X = D'
T3.3	*Clear D, *clear X, *set OXL11 ff
T3.4	*$P_L \rightarrow D_L$, *$P_U \rightarrow D_U$, *set Inhib EAB ff
T4.1	*Clear Incr P ff
T4.2	*Clear P
T4.2	*Adder → P
T4.3	*Clear OXL11 ff, *clear Inhib EAB ff

*These events are concerned with or are controlled by the Advance-P subsequence which is initiated only if X = D' (Y_i = +0).

TABLE 2-52. COMMANDS/DESCRIPTIONS FOR THE EXECUTION OF ISK INSTRUCTION (f = 57)
(NOT DESCRIBED IN A PREVIOUS SEQUENCE)
(REFER TO TABLES 2-22 AND 2-23 FOR COMMANDS NOT DESCRIBED)

Time Notation	Command/Description
T4.4	Set Insert EAB ff if X ≠ D'. If X ≠ D', a high from the adder disqualifies one of the 80AXX gates and subsequently disqualifies 81A00 (refer to plate P-29). A high from 81A00 is inverted by 82A00 to apply a L ⇒ X ≠ D' to 12L10 (refer to plate P-35), and is then ANDed with a L ⇒ f = 57 and the output inverted by 13L10 (refer to plates P-29, P-35, and P-27). The resulting L ⇒ f = 57 • $\overline{\text{Skip}}$ enables pin 7 of 20E32 (refer to plate P-27). The gate is qualified by additional inputs of L ⇒ W-Seq and T11 to produce a high which , via OR-inverter 21E32, enables OXG32 to be set at time T4.4.
T1.4	Adder → Z1. (Refer to plate P-23). A L ⇒ W-Seq • Run • NDRO Bootstrap Address, a L ⇒ f = 57, and a T14 qualify 30N13, and the resulting high output enables 39N13 via OR-inverter 38N13. A Ø4 then causes 39N13 to output a L ⇒ Adder → Z1. (Refer to plate P-35.)
T2.3	Clear Insert EAB ff. Same as described in I-Sequence.
T3.1	Set Incr P ff if X = D'. An Advance-P subsequence is started at T3.1 of the W-sequence with the setting of the Increment P flip-flop. If 12L10 is disqualified by the lack of a L ⇒ D ≠ X', a low is output to pin 14 of 10L10. Other enables are: A L ⇒ W-Seq; and a L ⇒ f = 30, 31, 57, 76 and T31. A high from 10L10 is inverted by 11L10 and combined with lows from the DISC ADV P switch and Ø1 to set OXL10.

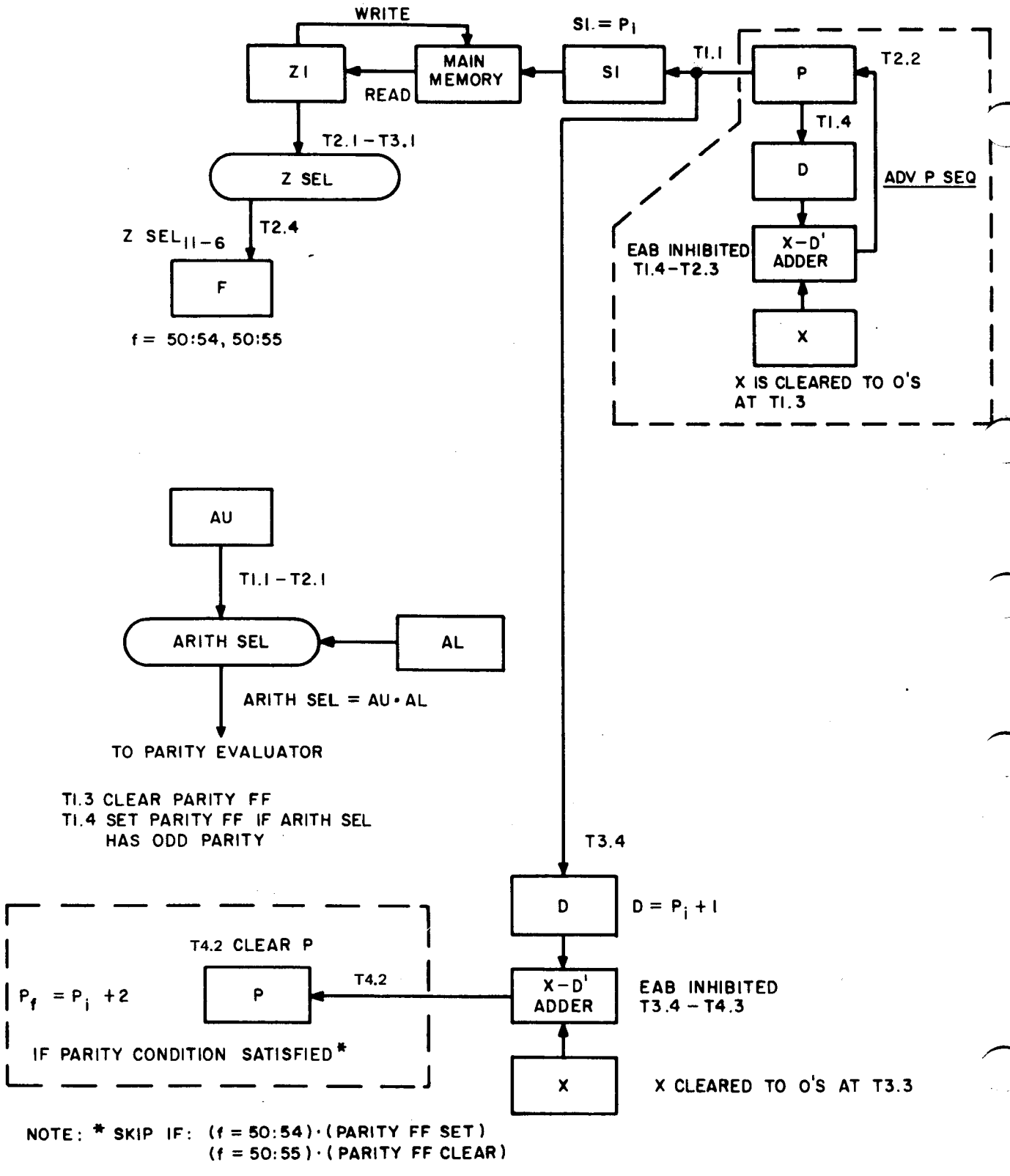


Figure 2-44. I-Sequence Data Flow for f = 50:54, 50:55

TABLE 2-53. I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:54, 50:55

Time Notation	Commands
T4.4	Clear S1
T1.1	P → S1, Init Memory, *set Incr P ff, AU → Arith Sel, AL → Arith Sel
T1.3	*Clear D, *clear X, clear Z1, clear F, *set OXL11 ff, clear Parity ff
T1.4	*P_L → D_L, *P_U → D_U, *set Inhib EAB ff, set Parity ff if Arith Sel = odd parity
T2.1	Z1 → Z-Sel, *clear Incr P ff, drop AU → Arith Sel, drop AL → Arith Sel
T2.2	*Clear P, *Adder → P
T2.3	*Clear OXL11 ff, *clear Inhib EAB ff
T2.4	Z-Sel_{11-6} → F, set OXF06 ff
T3.1	Drop Z1 → Z-Sel
T3.3	Clear D, clear X
T3.4	P_L → D_L, P_U → D_U, set Inhib EAB ff
T4.2	Clear P if skip satisfied** Adder → P if skip satisfied**
T4.3	Clear Inhib EAB ff

* These events are concerned with or are controlled by the Advance-P subsequence.

**Skip condition is satisfied if: (f = 50:54) • (Parity ff set)
(f = 50:55) • (Parity ff clear)

TABLE 2-54. COMMANDS/DESCRIPTIONS FOR THE EXECUTION OF SKPODD, SKPEVN INSTRUCTIONS (f = 50:54, 50:55) (NOT DESCRIBED IN A PREVIOUS SEQUENCE) (REFER TO TABLE 2-24 FOR COMMANDS NOT DESCRIBED)

Time Notation	Command/Description
T1.1	AU → Arith Select, AL → Arith Sel. Gates 18N01 & 28N01 (refer to Plate P-17) are both qualified by the H ⇒ T14 I-Seq. The low output of 18N01 provides a L ⇒ AU → Select via 19N01. The low from 28N01 is applied to 29N01, which also has a low from 90N01, to provide a L ⇒ AL → Select.
T2.1	Drop AU → Arith Select, Drop AL → Arith Sel. Gates 18N01 & 28N01 (Plate P-17) are now disabled due to the loss of the H ⇒ T14 I-Seq signal dropping the AU → Arith Sel & the AL → Arith Sel.
T3.3	Clear D. Gate 30N02 is enabled by a L ⇒ T34 I-Seq Format II and a L ⇒ 50:55 to clear D as previously described.
T3.4	$P_L \rightarrow D_L$, $P_U \rightarrow D_U$, Set Inhib EAB ff. The high from 30N02 sends $P_L \rightarrow D_L$, $P_U \rightarrow D_U$ and sets Inhibit EAB slip-flop as previously described.
T4.2	Clear P if Skip Satisified. For an f = 50:54, if parity flip-flop OXG54 is set, gate 10G52 is enabled by L ⇒ f = X4, X5, and by the low from the flip-flop (refer to plate P-30). The high output is inverted by 11G52, and with a L ⇒ f = Even, 12G52 provides a high to 20G50. The output of 20G50 is low and is applied to 10N07, and along with a L ⇒ T42 I-Seq and a L ⇒ f = 50:21 through 50:23, 50:55, and 50:57, outputs a high to clear P as previously described (refer to plate P-21). If f = 50:55 and parity flip-flop is clear, and if gate 10G52 produces a low output to 12G52 with a L ⇒ f = Odd, a high is applied to 20G50 to clear P as previously described for f = 50:54. Adder → P if Skip Satisfied. The high from 10N07 is applied to 18N07 to provide a L ⇒ Adder → P from 19N07 with Ø2 (refer to plate P-21).

Section 2-3. Detailed Functional Description (Arithmetic Section)

2-233. ARITHMETIC SECTION

2-234. ADDERS. The adders used in the arithmetic section of the computer are: X-D' adder, B adder, and the KO-1 adder.

2-235. X-D' Adder. The X-D' adder is an 18-bit End-Around-Borrow (EAB), subtractive type adder used by the computer for mathematical calculations. It receives inputs from both the X- and D-registers, and can output to the AU-, AL-, P-, S1-, or Z1-registers. The adder consists of three unique sections designated as the half-subtractor, the borrow generator, and the full adder. Four possible states of the X- and D-register flip-flops are presented in equation form as follows:

1.	X=1 (set) D=1 (set) 0	X=1 D'=0 1	Borrow is not required and can be satisfied
2.	X=0 (clear) D=0 (clear) 0	X=0 D'=1 1	Borrow is required and cannot be satisfied
3.	X=1 (set) D=0 (clear) 1	X=1 D'=1 0	Borrow is not required and cannot be satisfied
4.	X=0 (clear) D=1 (set) 1	X=0 D'=0 0	Borrow is not required and cannot be satisfied

2-236. Half-Subtractor. (Refer to plate P-94.) The half-subtract gates, 10AXX, perform a half-subtract (exclusive OR function) on the contents of the X- and D-registers, while the output of the 11AXX gates represents the half-subtract of X-D'.

2-237. Borrow Generator. The borrow generator gates, 12AXX and 13AXX are used to detect and propagate borrow requests (refer to plate P-94). If a stage in the adder requires a borrow, these gates provide the means to achieve it. If both the X- and D- registers are cleared, (X=0, D'=1), this stage requires a borrow. If both the X- and D- registers are set (X=1, D'=0), a borrow can be satisfied upon reaching this stage. If the registers are neither both set nor both cleared, (X=1, D'=1, X=0, D'=0) a borrow is not required, and cannot be satisfied. Therefore, if a borrow exists, it will only be propagated within the adder in an effort to satisfy it.

2-238. The adder is divided into three 6-bit sections to increase the speed of borrow propagation. Each section is subdivided into two 3-bit subsections. These sections and subsections are connected so that borrows will be propagated through the subsection in which they originated, and then to the first succeeding subsection that can satisfy the borrow request. This capability is illustrated in figure 2-45.

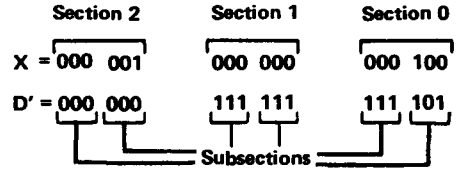


Figure 2-45. Section and Subsection Borrow Capability.

2-239. Bit position 2^0 requires a borrow (X=0, D'=1). This borrow request must be propagated to Section 2, bit position 2^{12} (X=1, D'=0) before it can be satisfied. Propagation in this instance is first from 2^0 through 2^2, and then to Section 2, bit position 2^{12} where it is satisfied. The upper subsection of Section 0 (2^3 -2^5) and all of Section 1 (2^6 -2^{11}) are by-passed since they are incapable of satisfying the borrow, and since time is wasted by propagating through these areas. The output of the adder is taken from the 14AXX (full subtractor) gates. (Refer to plate P-94.)

2-240. Full adder. The full adder is shown on logic prints P-94 through P-96. To better understand its operation, consider the following example. The actual gate logic is defined as positive logic, however, the interface between gates within the computer is actually mixed logic. The gate outputs are as follows: (In this example 4 + 2 is added; the 4 in the X-register and the 2 in the D-register.)

2^{17} 2^0
X= 000 000 000 000 000 100 =000004

D= 000 000 000 000 000 010 =00000

D'= 111 111 111 111 111 101 =

10AXX= 000 000 000 000 000 110

11AXX= 111 111 111 111 111 001

12AXX= 001 000 001 000 001 000

13AXX= 110 111 110 111 110 111

14AXX= 000 000 000 000 000 110 =00000

Bit position 2^0 indicates a borrow request of X=0, D'=1. Since none of the higher bit positions is capable of satisfying this borrow, gate 12A00 outputs a high which implies that an End-Around-Borrow (EAB) has been generated.

2-241. The adder speed-up circuits are shown on logic prints P-91 through P-93. The 20A00 and 20A03 gates in plate P-91 determine if the 00-02 or 03-05 stages are capable of satisfying a borrow (develop and enable). The 30A00 gate detects any borrow request applied to stage 03 from

stages 00-02, and the 30A03 gate checks for borrows applied to stage 06 from stages 03-05.

2-242. The 31A00 gate applies a borrow request from stages 00-02 and/or 03-05 to other sections of the adder. The 22A00 gate notifies other sections of the adder that stages 00-05 are available to satisfy a borrow request if one is applied. A low output from 22A00 indicates that stages 00-05 cannot satisfy a borrow request.

2-243. The other two sections of speed-up circuits (refer to plates P-92 and P-93) generate exactly the same type of borrow requests and enable signals in their own particular sections. Section 12-17 (refer to plate P-93) involves two additional factors in developing its output. The section is affected by the Inhibit EAB and Insert EAB flip-flops. If the Insert EAB flip-flop is set, a simulated borrow request is generated. If the Inhibit EAB flip-flop is set, a borrow request (other than the simulated request) is inhibited, and this section is forced to indicate that it contains an enable (satisfy) condition. In this case, a borrow request from either of the other two sections will be satisfied and an EAB will be inhibited. The method used by the adder to apply intersection borrow is shown in tables 2-55, 2-56, and 2-57.

2-244. B Adder. The B adder is an open-ended adder used to update the CACW in I/O operations and provide counting actions in BSK and BJP instructions as well as for the Real Time Clock. The complete network is shown in plates P-118 through P-120. The adder network is connected to the B-register to provide an output representing the value in the B-register ±1. The factor which determines whether a one will be added to or subtracted from the B-register is the function of the B ± flip-flop (refer to plate P-27). A cleared flip-flop causes the network to output a value representing a B +1; a set flip-flop causes the network to output a value representing B -1.

2-245. If the B-register contains 000001_8, pin 8 of 0XB00 will output a high to pin 8 of 10B01. (Refer to plate P-118.) If a L⇒ +1 → B enable is received, a high will be felt on pin 5 of 10B01, and 10B01 will output a one (low). This low output, and/ored with the output of cleared flip-flop 0XB01, causes 12B01 to output a low. The number now present at the adder output is an octal 000002.

2-246. If the B-register contains a 000001_8 a low is felt on pin 6 of 10B01. If L⇒ -1 → B enable is received, pin 5 of 10B01 will also go low and the gate will output a high. This high is inverted by 11B01 resulting in a low to pin 6 of 12B01. Since flip-flop 0XB01 is now clear, a low occurs on pin 5 of 12B01 which outputs a zero (high). A one has effectively been subtracted from B and the adder network now outputs all zeroes.

2-247. The 12BXX gates reflect the output of the adder with the exception of bit zero. Bit zero is taken from the zero side (pin 8) of the 0XB00 flip-flop. The output of the adder is fed only to the Z0-register. The method of incrementing or decrementing the contents of the B-register (refer to figure 2-46) is as follows:

1. Initially, a number is entered into the B-register. The set or clear condition of the B ±1 flip-flop determines the input applied to the ±1 network. This input, when acted upon by the network, develops an output representing B ±1, which is fed to the

TABLE 2-55. INTERSECTION BORROWS TO SECTION 05-00

Section 17-12	Section 11-06	Section 05-00	Input To 12A00
Brw Req			31A12 = L
No Enables	Brw Req		31A06 & 22A12 = L
No Enables	No Enables	Brw Req	22A12, 31A00 & 20A06 = L

TABLE 2-56. INTERSECTION BORROWS TO SECTION 11-06

Section 17-12	Section 11-06	Section 05-00	Input To 12A06
		Brw Req	31A12 = L
Brw Req		No Enables	31A12 & 22A00 = L
No Enables	Brw Req	No Enables	22A00, 31A06 & 22A12 = L

TABLE 2-57. INTERSECTION BORROWS TO SECTION 17-12

Section 17-12	Section 11-06	Section 05-00	Input To 12A12
	Brw Req		31A06 = L
	No Enables	Brw Req	31A00 & 22A06 = L
Brw Req	No Enables	No Enables	22A06, 31A12 & 22A00 = L

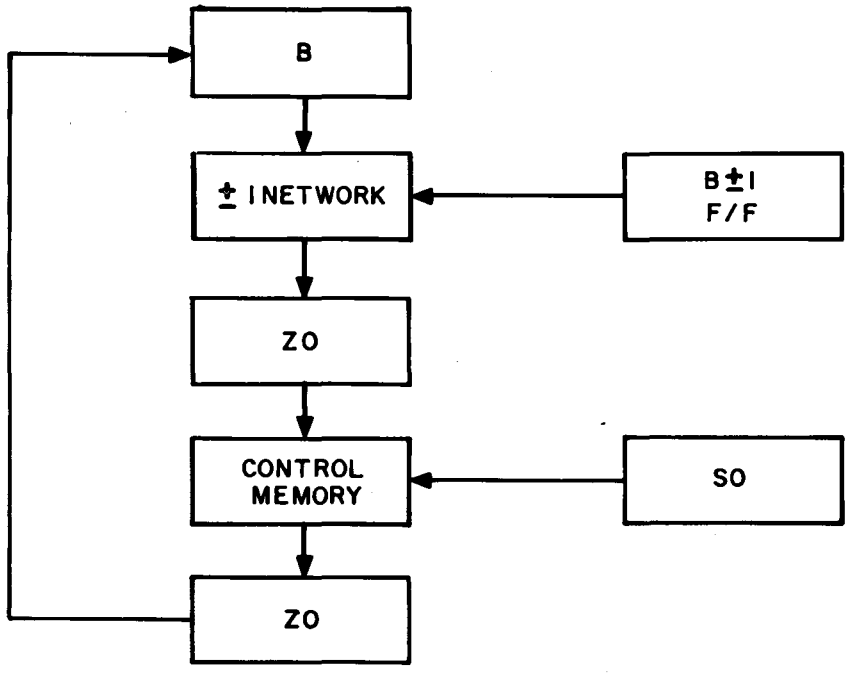


Figure 2-46. Method of Incrementing or Decrementing the Contents of the B-Register

Z0-register and placed in control memory at an address specified by the S0-register. As the program continues, and if it again becomes necessary to examine the B-register, the S0-register will call out of memory a number representing B +1 and place it into the B-register via the Z0-register.

2. The output of 11B10 to 11E18 (refer to plate P-119) is used to time the Resume Fault Special Interrupt. A high level from 11B10 applies a Carry/Borrow request to bit position 10 (1-second RTC bit). The fault condition is used during Real Time Clock operation in the following manner: Two consecutive carry indications from 11B10 mean that a 1-second time period has elapsed. This is the period in which the intercomputer resume condition must be established; otherwise, the resume fault interrupt is generated.

3. During I/0 operation, the B-register is used to increment or decrement the CACW. To prevent modification of bits 16 and 17 of CACW, which specify buffer direction and monitor interrupts, the Carry-Borrow request to these bits is disabled.

4. A high from 11B16 (refer to plate P-120) applies a Carry-Borrow request to bit position 16. This gate, however, if in the I/0 sequence and not using the RTC, is inhibited by a high on pin 10. When disabled that gate also prevents modification of bit 17. Thus, the B-network transfers bits 16 and 17 from B and Z0 without change.

2-248. K0-1 Adder. The K0-1 adder is a 6-bit, open-ended adder. It, along with the K0 and K1 registers, comprise the K Counter (refer to figure 2-47), which is used to control the number of repeated operations used to shifting, scaling, multiplying, or dividing.

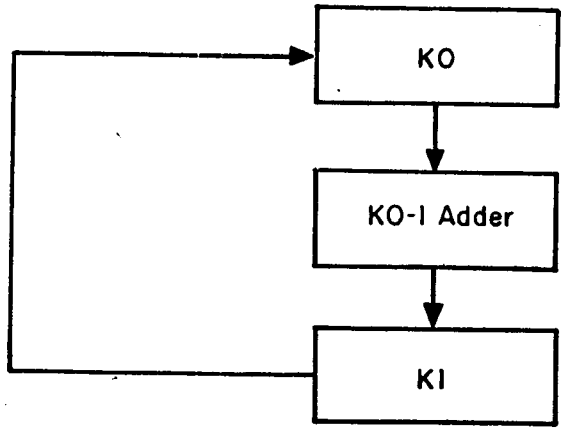


Figure 2-47. K0-1 Adder Block Diagram

2-249. Under normal circumstances, the specific count required to perform an operation is entered into the K0-register which is displayed on Control Panel 2, drawer A4 as the K register. The K0-1 adder subtracts one from K0 and enters the result in the K1-register. The decremented value is then placed in the K0-register, and one of the repeated operations required by a particular function is complete. When the content of the K0-register reaches zero count, the controlled operation is terminated.

2-250. The K0-1 adder logic is shown in plate P-37. The lower rank registers (0XK00 through 0XK05) are the K0 portion of the adder, and the upper rank registers (1XK00 through 1XK05) comprise its K1 portion.

1. Effect of Borrow Request. If a request for a borrow is applied to bit position of K0, the result of this subtraction produces the complement of the bit. Refer to the following examples:

0	0	K0 BIT
-1	-1	BORROW REQUEST APPLIED TO K0 BIT
0	1	RESULT (1's COMPLEMENT OF K0 BIT)

2. Adder Stage 00. Refer to logic diagrams, plate P-37 for the adder logic. The subtraction of K0-1 always affects $K0_{00}$ to produce its complement. This complemented value is outputted by inverter 02K00 and its placed in $K1_{00}$ during the "K0-1→ K1" command.

3. Generation of Borrow Requests. If a borrow request is applied to a particular bit position, the complement of that bit is the result. If that same K0 bit position contains 0_2, the borrow request is propagated to the next higher bit because the 0_2 cannot supply the 1_2 that the borrow requires. All borrow requests originate from the subtraction of $K0_{00}$ - 1. A borrow

request is applied to a particular bit position if all of the less significant K0 bit positions contain 0's. Refer to table 2-58 for the conditions necessary to apply borrow requests to the K0 bits. Adder bits 05 through 01 have logic which tests all of the less significant K0 bits for the binary configurations described above. If a borrow request is applied, the complement of the K0 bit is the adder output as was shown for bit 00. The adder request logic (figure 2-48) is a portion of that shown in the logic diagrams, plate P-37. Gate 08K00 is used to test K0 for all 0's. This condition terminates the operation being controlled by the K-counter.

-251. LOGIC NETWORKS. The logic networks used in the arithmetic section of the computer are: AU, AL, borrow, overflow, parity, K-zero, and B-zero.

-252. AU, AL Positive, Negative. Bits A35 and A17 (upper bits of AU and AL respectively) are monitored by AND/OR gate 21G34 (refer to plate P-29). If AU is negative, flip-flop 0XA35 (refer to plate P-98) will be set and a high from its clear side will disable in 6 of 21G34. Similarly, if AL is negative, 0XA17 (refer to plate P-100) will be set and pin 8 of 21G34 will be disqualified.

2-253. AU, AL Zero. Similar schemes are employed to determine whether AU or AL is zero, therefore, only the AL network will be described (refer to plates P-99 and P-100). AND gates 90A00, 90A04, 90A09, and 90A13 each monitor a portion of the AL flip-flops. If any flip-flop is set, a high from its clear side will disable one of the AND gates, causing it to output a low. The outputs of the gates are ANDed by 91A00 (refer to plate P-29), and any low input will disqualify that gate, causing it to output a high. Conversely, if all inputs are highs, 91A00 will output a L $\Rightarrow$ AL = 0. AND gate 91A18 monitors the outputs of the AU-register AND gates.

2-254. Borrow. (Refer to plate P-30.) When a L $\Rightarrow$ T42 R1, W-Seq, and a L $\Rightarrow$ f = 20-23 are felt by 10E51, it outputs a high which is inverted and applied to the set gate of the Borrow Test flip-flop 0XG51. If the correct borrow enable (EAB required) is generated by 30A15 (refer to plate P-93) or by 10A17 and 13A17 (refer to plate P-96) via 20E51, OR-inverter 21E51 will output an enable which will allow the setting of 0XG51.

2-255. Overflow. (Refer to plate P-30.) If it is assumed that two negative numbers

TABLE 2-58. K0-1 ADDER BORROW REQUEST CONDITIONS

Borrows Applied	K0 bit positions				
	04	03	02	01	00
Brw 01					0_2
Brw 02				0_2	0_2
Brw 03			0_2	0_2	0_2
Brw 04		0_2	0_2	0_2	0_2
Brw 05	0_2	0_2	0_2	0_2	0_2

NOTE: "Brw XX" means a borrow request is being applied to bit position XX.

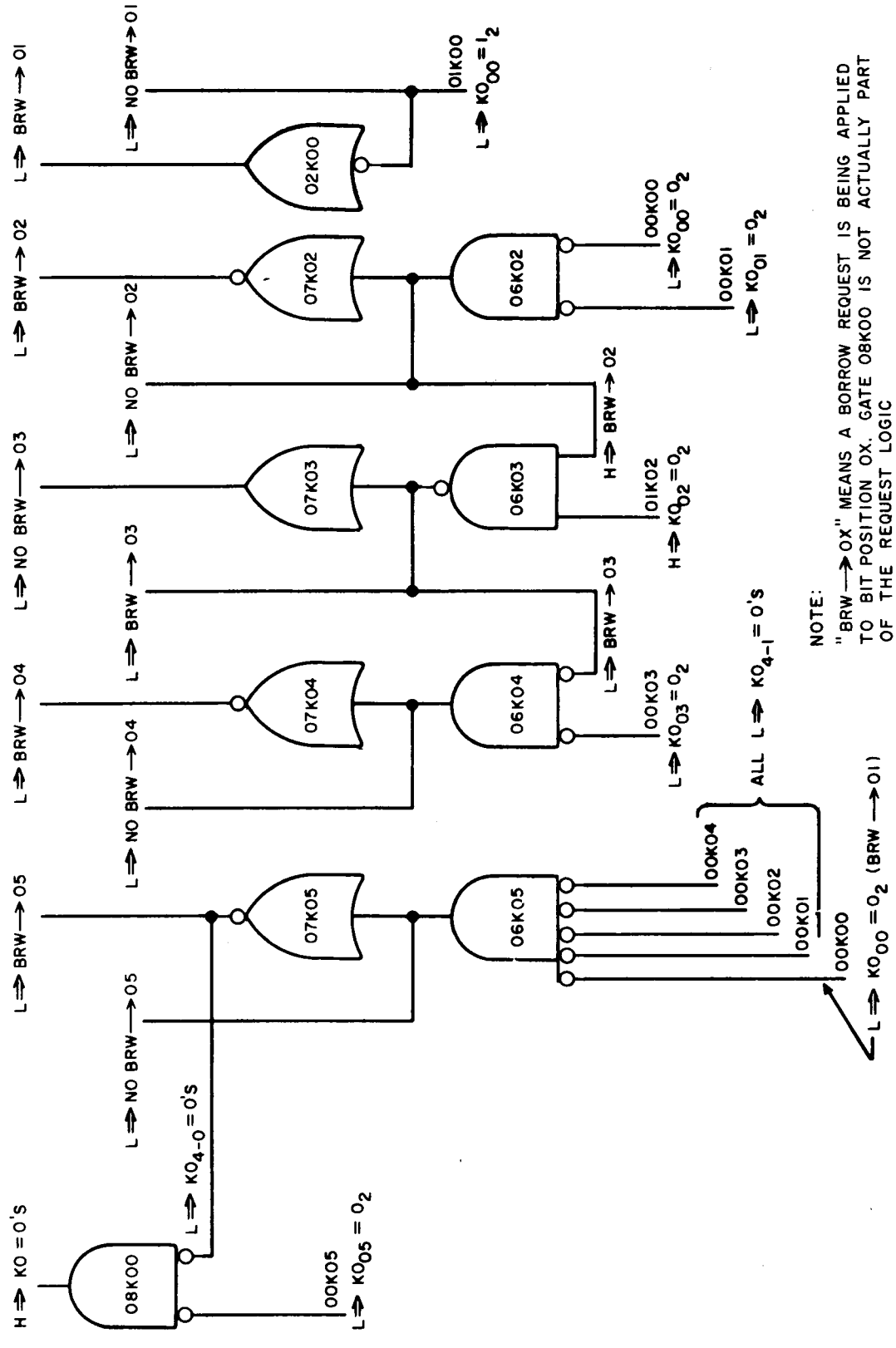


Figure 2-48. K0-1 Adder Borrow Request Generation Logic

are to be added, the results will be a negative number, bit 17 of the adder being a one. Since all three numbers are negative, bit 17 of both X- and D-registers will also be ones (set). When corresponding bits of both X and D are set, a borrow may be satisfied; but satisfying a borrow would make bit 17 of the adder output a zero, thereby changing the sign of the result. Similarly, if both X_{17} and D_{17} are zero (positive numbers), an EAB will be propagated; if it is satisfied, and no borrow request is generated to bit 18, bit 18 of the adder will be a one, thus signifying a negative number as a result of the addition of two positive numbers. In such situations the Overflows flip-flop will be set so the result can be checked and corrected.

2-256. (AND gates 10E52 and 20E52 (refer to plate P-30) output highs when conditions in which an overflow might occur are present.) AND gate 20G52 checks for overflow for positive numbers and gate 21G52 checks negative numbers. Gate 20G52 outputs a high if both X and D are positive numbers and no borrow is required. Gate 21G52 outputs a high if both X and D negative and a borrow is required. If an overflow condition exists, 22G52 and 11E52 will output lows which will be ANDed with a $\emptyset 2$ to set 0XG52. If the A-register is divided by too small a number, and the quotient is larger than AL, 21E52 will output a low which is ANDed with $\emptyset 2$ and a L $\Rightarrow$ Adder $\rightarrow$ AU to set 0XG52, thus signifying an overflow on a Divide instruction. (The entire A-register is divided by the contents of memory and the quotient is displayed in AL with the remainder in AU. If the divisor is too small, AL cannot contain the entire quotient and an overflow into the AU-register will take place. A more detailed explanation of the division process is provided under the Multiply/Divide Instructions portion of this section.)

2-257. Parity. The parity evaluator has hard-wired inputs from arithmetic-select. Parity logic is comprised of several stages. The initial stages separate the arithmetic-select inputs into groups of two bits each. The parity indications from these groups are then combined into sections of three groups each. Total parity is then evaluated from the combination of the three sections. (Refer to figure 2-49 below for an example.)

2-258. Group Parity. Each group is comprised of two bits. Bits 1 and 0 are used for an example. (Refer to figure 2-50 and logic diagram, plate P-101.) Inverter 15X00 tests bits 1 and 0 of arithmetic select for the two possible even parity conditions. (Refer to table 2-59 for the four possible configuration of these bits.) The logic for the other groups is the same except for the bits being sensed.

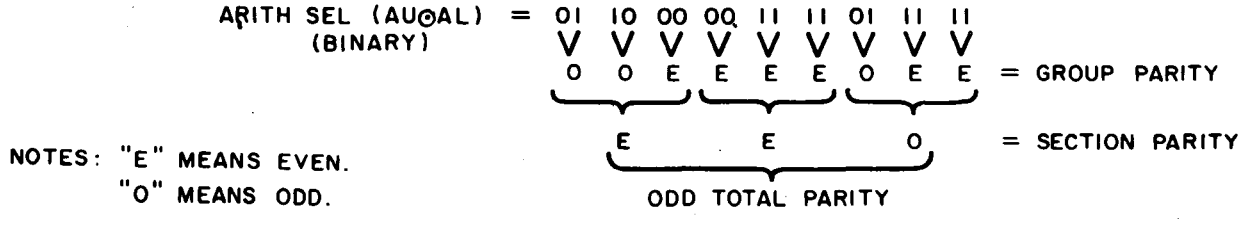
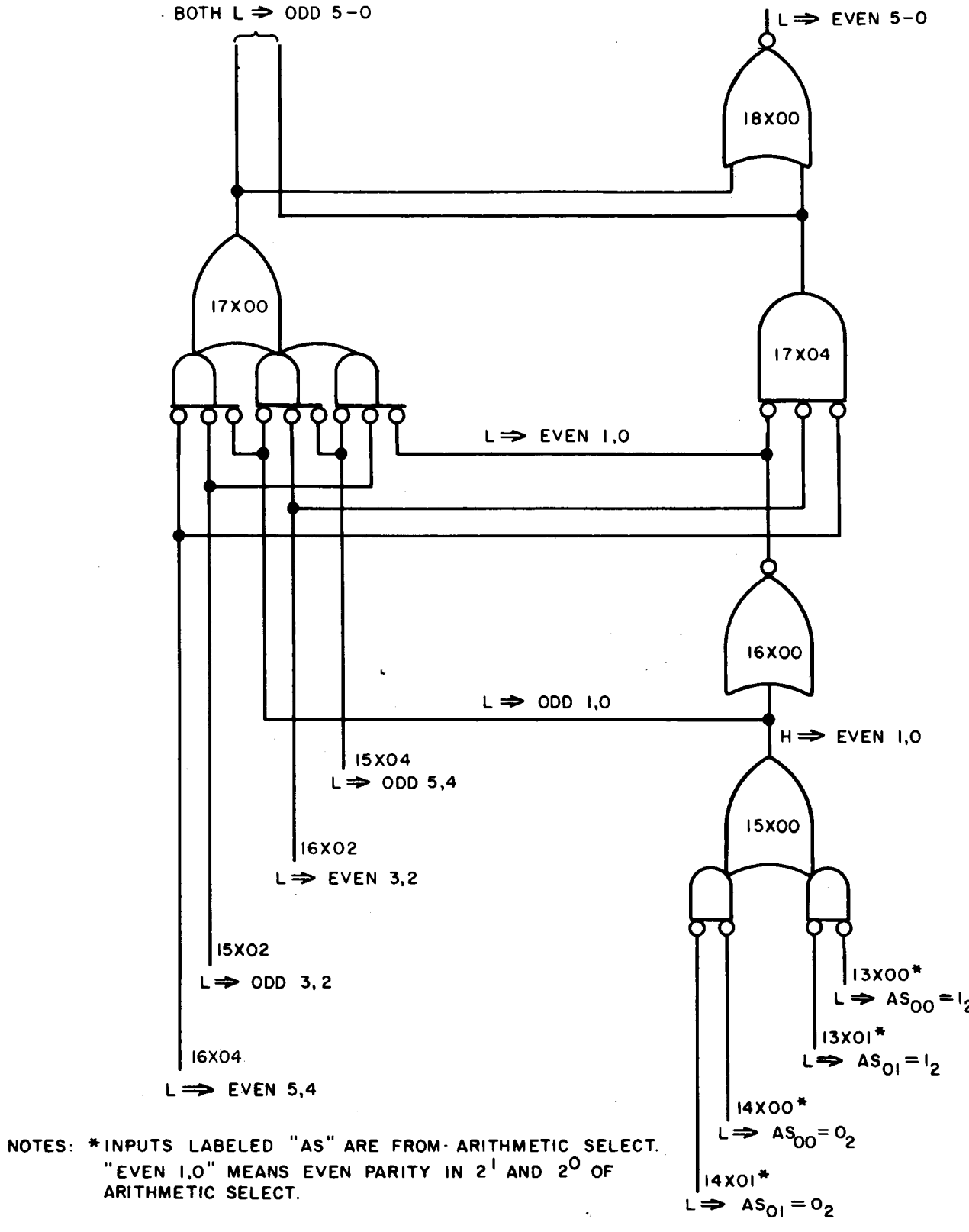


Figure 2-49. Parity Evaluation Example



NOTES: *INPUTS LABELED "AS" ARE FROM ARITHMETIC SELECT.
"EVEN 1,0" MEANS EVEN PARITY IN 2^1 AND 2^0 OF ARITHMETIC SELECT.

Figure 2-50. Parity Evaluator, Section 5-0

TABLE 2-59. PARITY EVALUATOR, GROUP 1 AND 0 CONDITIONS

2^1	2^0	Group Parity	15X00 Output
0	0	Even	H
0	1	Odd	L
1	0	Odd	L
1	1	Even	H

TABLE 2-60. PARITY EVALUATOR, SECTION 5-0 CONDITIONS

Groups			Section	Output Levels		
5,4	3,2	1,0	Parity	17X04	17X00	18X00
Odd	Odd	Odd	Odd	L	L	H
Odd	Odd	Even	Even	L	H	L
Odd	Even	Odd	Even	L	H	L
Odd	Even	Even	Odd	L	L	H
Even	Odd	Odd	Even	L	H	L
Even	Odd	Even	Odd	L	L	H
Even	Even	Odd	Odd	L	L	H
Even	Even	Even	Even	H	L	L

2-259. Section Parity. Each Section is comprised of three groups (six bits). Section 5-0 is used for an example. (Refer to figure 2-50 and logic diagram plate P-101.) Inverters 18X00, 17X00, and 17X04 test the parity conditions of the three groups (5, 4; 3, 2; and 1, 0). (Refer to table 2-60 above for the eight possible parity configurations.) The logic for the other sections is the same except for the bits being sensed.

2-260. Total Parity. Total Parity is the combined evaluation of the three sections. (Refer to figure 2-51 and logic diagram,

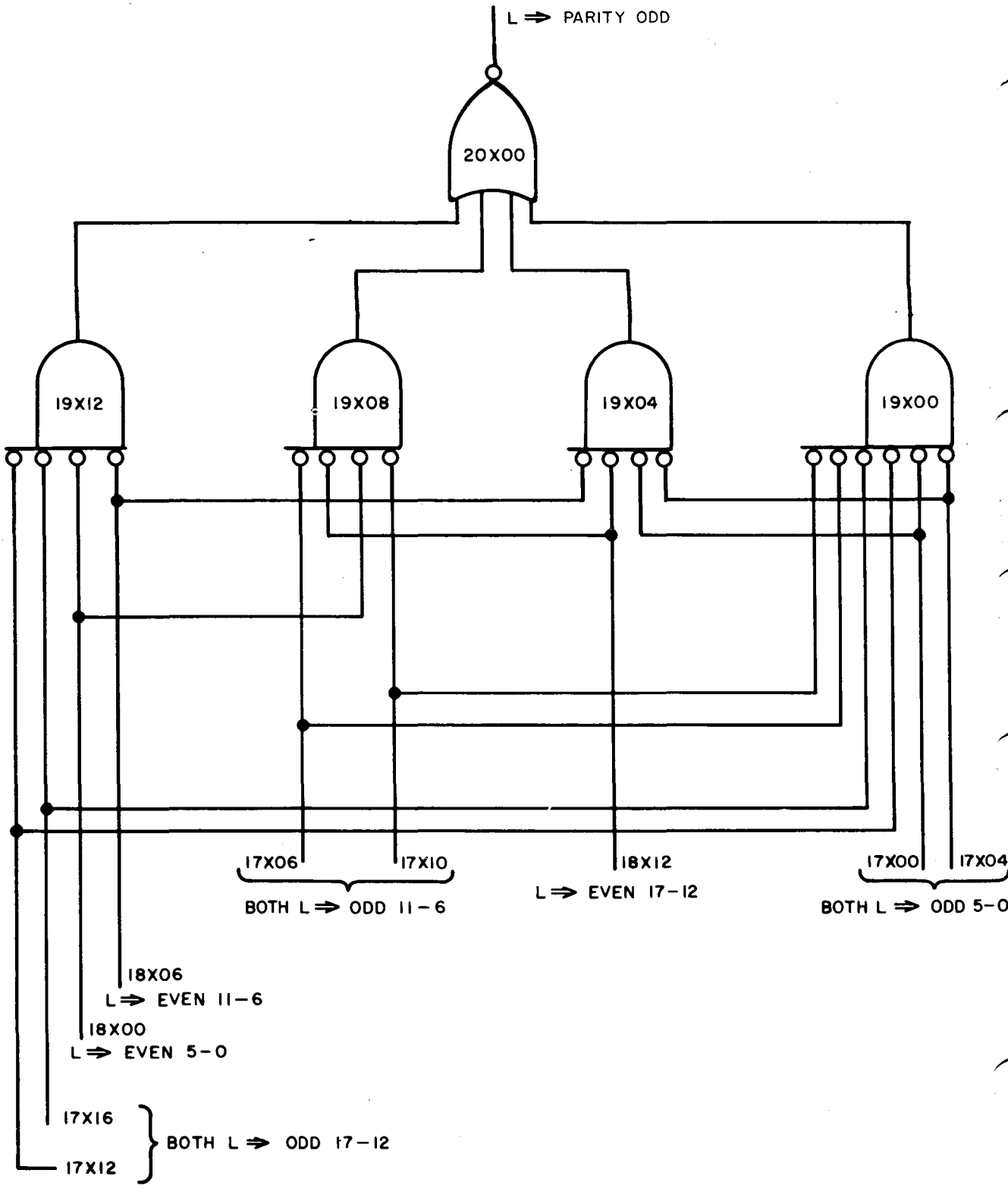


Figure 2-51. Parity Evaluator, Final Stage

TABLE 2-61. PARITY EVALUATOR, TOTAL PARITY CONDITIONS

Sections			Total	Output Levels				
17-12	11-6	5-0	Parity	19X00	19X04	19X08	19X12	20X00
Odd	Odd	Odd	Odd	H	L	L	L	L
Odd	Odd	Even	Even	L	L	L	L	H
Odd	Even	Odd	Even	L	L	L	L	H
Odd	Even	Even	Odd	L	L	L	H	L
Even	Odd	Odd	Even	L	L	L	L	H
Even	Odd	Even	Odd	L	L	H	L	L
Even	Even	Odd	Odd	L	H	L	L	L
Even	Even	Even	Even	L	L	L	L	H

plate P-101.) The gates shown in figure 2-51 test the parity conditions of the three groups for the four possible odd parity configurations. (Refer to table 2-61 for the eight possible parity configurations.)

2-261. K-Zero. The K-zero network is part of the interconnecting AND/OR logic between the K0 and K1 portions of the K Counter. Gates 08K00 and 06K05 (refer to plate P-37) monitor the clear side of the K0-register flip-flops. When all of these flip-flops are clear, the clear side outputs a low. These lows enable 06K05 whose high output is inverted by 07K05 to partially enable 08K05. 08K05 is fully enabled by the low from 00K05. The output from 08K00 is a H $\Rightarrow$ K0 = 0. This condition terminated the operation being controlled by the K-counter.

2-262. B-Zero. (Refer to plate P-107.) The clear side of each flip-flop in the B-register (refer to plates P-118 and P-119 is monitored by AND gates 90B00, 90B04, 90B09, and 90B13. All zeroes (lows) from the flip-flop cause the gates to output highs to AND gate 91B00 which then produces a L $\Rightarrow$ B = 0. If any B-register flip-flop is set, its clear side will be high, disqualifying its associated gates 90BXX and 91B00, causing it to output a high signifying B $\neq$ 0.

2-263. ARITHMETIC AND LOGICAL INSTRUCTIONS. The arithmetic and logical instructions (refer to table 2-62 for a list of these instructions) consist of add-subtract instructions, logical instructions, shift instructions, and multiply-divide instructions.

2-264. Add-Subtract Instructions. The Add-Subtract instructions allow the computer to add or subtract the contents of memory to or from the AU and/or AL-registers. These instructions consist of ADDAL (f = 14), ADDALB (f = 15), SUBAL (f = 16), SUBALB (f = 17), ADDA (f - 20), ADDAB (f = 21), SUBA (f = 22), and SUBAB (f = 23), which are all SR sensitive. Also included under the add-subtract instructions are ADDALK (f = 71), ENTBKB (f = 37), and RND (f = 50:60). ADDALK and ENTBKB are not discussed here, but are discussed under the Command Instructions in the Control Section.

2-265. ADDAL (f = 14). The ADDAL instruction adds the contents of the AL-register (AL) to the contents of a designated memory location (Y) and places the results of the addition in the AL-register.

2-266. ADDALB (f = 15). The ADDALB instruction performs the same service as an f = 14 instruction except that the address of the operand in memory becomes Y + B.

2-267. SUBAL (f = 16). The SUBAL instruction subtracts the contents of the designated memory location (Y) from the contents of the AL-register (AL) and places the result in the AL-register.

2-268. SUBALB (f = 17). The SUBALB performs the same service as an f = 16 instruction, except that the address of the operand in memory becomes Y + B.

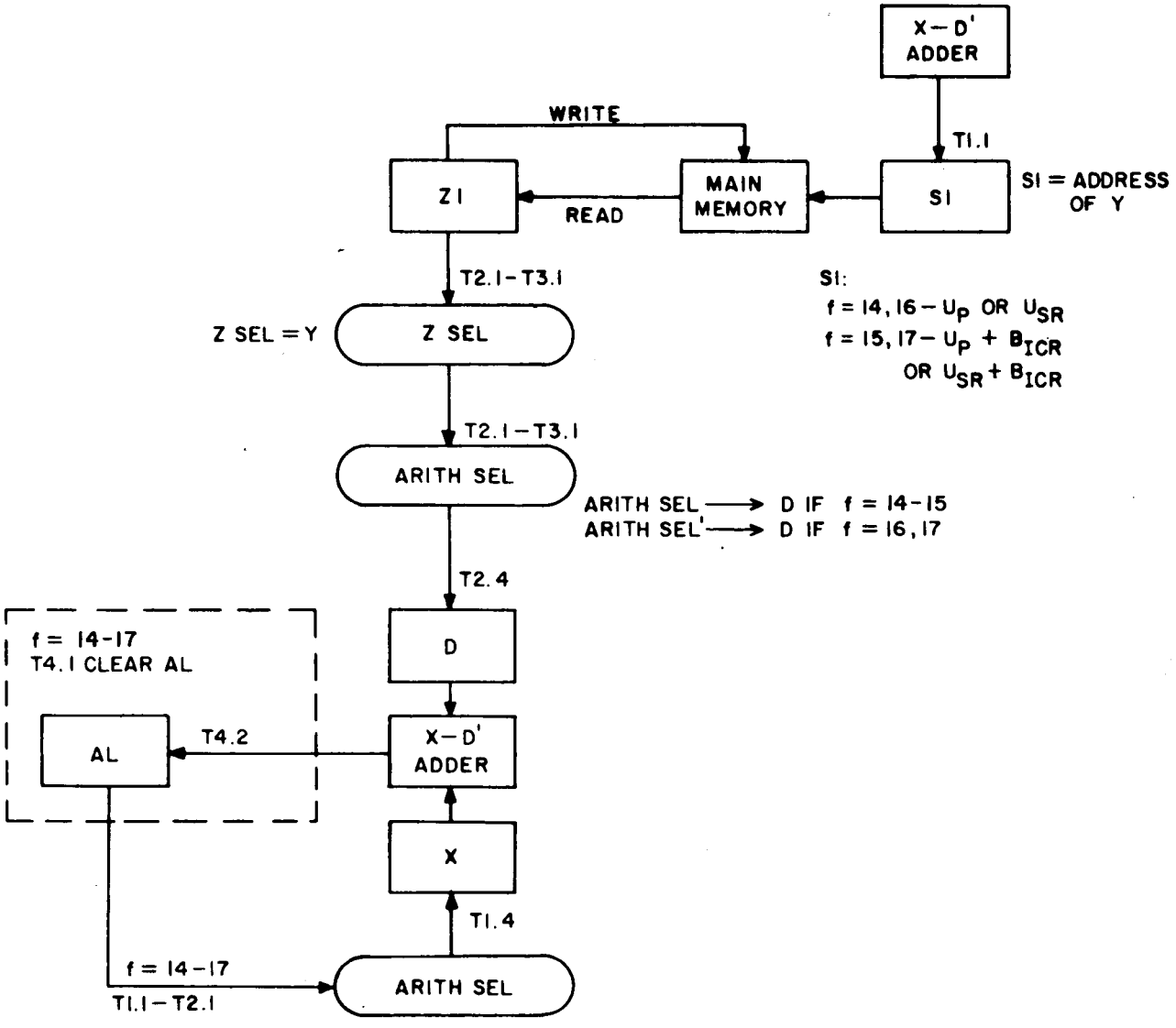
2-269. R1-Sequence Data Flow for f = 14 through 17. As shown in figure 2-52, a memory cycle is initiated and an operand is obtained from memory. (Not shown in figure 2-52 is the I-sequence that obtained the instruction word from memory and developed the address of the operand.) If an Add instruction (f = 14, 15) is being executed, the operand itself is transferred to the D-register. If a subtract instruction (f = 16, 17) is being executed, the complement of the operand is transferred to the D-register. The contents of the AL-register are transferred, via the Arithmetic Selector to the X-register. Essential commands necessary to accomplish the R1-sequence are listed in table 2-63. Some of these commands have been described previously in detail in the R1-sequence. Those commands not previously described are described in table 2-64 in detail.

2-270. ADDA (f = 20). The ADDA instruction is used to add large numbers the sum of which would exceed the normal 18-bit capacity of the arithmetic registers. The instruction requires that the AU- and AL-registers be combined into one large 36-bit register. Even though a 36-bit word is being manipulated, the computer is still only capable of operating with 18 bits at a time. Therefore, two distinct add operations must be performed: (1) the AL-register contents are involved with one memory location (Y) and (2) and AU-register contents are involved with another memory location (Y + 1). The computer is hard wired to perform this instruction automatically without special attention by the operator. It is necessary, however, that the operands be stored in memory with the least significant half of the number at an even address (Y) and the most significant half of the number at an odd address (Y + 1). An example of this is as

TABLE 2-62. ARITHMETIC SECTION - ARITHMETIC AND LOGICAL INSTRUCTIONS

Format Code	Instruction
ADD (REGISTER + MEMORY → REGISTER)	
14	ADDAL
15	ADDALB
20	ADDA
21	ADDAB
SUBTRACT (REGISTER → MEMORY → REGISTER)	
16	SUBAL
17	SUBALB
22	SUBA
23	SUBAB
SELECTIVE (LOGICAL OPERATION WITH REGISTER AND MEMORY)	
04	SLSU
05	SLSUB
51	SLSET
52	SLCL
53	SLCP
COMPLEMENT (REGISTER, NO MEMORY)	
50:61	CPAL
50:62	CPAU
50:63	CPA

Format Code	Instruction
ROUND (REGISTER, NO MEMORY)	
50:60	RND
MULTIPLY/DIVIDE (REGISTER AND MEMORY)	
24	MULAL
25	MULALB
26	DIVA
27	DIVAB
SHIFT AND SCALE FACTOR (NO MEMORY)	
50:41	RSHAU
50:42	RSHAL
50:43	RSHA
50:44	SF
50:45	LSHAU
50:46	LSHAL
50:47	LSHA
COMPARE (REGISTER AND MEMORY)	
02	CMAL
03	CMALB
06	CMSK
07	CMSKB



T3.2 SET OVERFLOW FF IF $(f = 14-17) \cdot$ (SIGN OF $AL_f \neq$ SIGN OF AL_i)*

NOTE: * THE OVERFLOW CONDITION IS INDICATED FROM THE X − D' ADDER BY:
($X_{17} = O_2 \cdot D_{17}' = I_2 \cdot$ NO BORROW REQUEST ⟶ BIT 17) +
($X_{17} = I_2 \cdot D_{17}' = O_2 \cdot$ BORROW REQUEST ⟶ BIT 17)

Figure 2-52. R1-Sequence Data Flow For f = 14 through 17

TABLE 2-63. R1-SEQUENCE ESSENTIAL COMMANDS FOR f = 14 THROUGH 17

Time Notation	Commands	f = 14, 15	16, 17
T4.4	Clear S1	X	X
T1.1	Adder → S1, Init memory	X	X
	AL → Arith Sel	X	X
T1.3	Clear X, clear Z1	X	X
T1.4	Arith Sel → X	X	X
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel	X	X
	Drop AL → Arith Sel	X	X
T2.3	Clear D	X	X
T2.4	Arith Sel → D	X	
	Arith Sel' → D		X
T3.1	Drop Z1 → Z-Sel, drop Z-Sel → Arith Sel	X	X
T3.2	Set Overflow ff if sign of $AL_f \neq$ sign of AL_i	X	X
T4.1	Clear AL	X	X
T4.2	Adder → AL	X	X

follows:

$(005000) = Y = 000001_8$

$(005001) = Y + 1 = 000000_8$

$(AL) = 000312_8$

$(AU) = 000461_8$

2-271. The first part of the problem is solved by subtracting the complement of the address 005000 (777776) from the contents of the AL-register. This is accomplished as follows:

$$000312 = (AL)_i$$

$$\underline{777776 = (005000)'}$$

$$000314 = (AL)_f$$

2-272. During the operation, an end-around-borrow is generated. This borrow is propagated into the second portion of the problem, which is solved by subtracting the complement of

TABLE 2-64. COMMANDS/DESCRIPTIONS FOR f = 14 THROUGH 17
(REFER TO TABLE 2-21 FOR COMMANDS NOT DESCRIBED)

Time Notation	Command/Description
T3.2	Set Overflow ff if Sign of $AL_f \neq$ Sign of AL_i. Overflow flip-flop is set if the sign of $AL_f \neq$ the sign of AL_i. (Gate 11E52 (refer to plate P-30) is enabled by a H ⇒ f = 14 - 17, and a high from 20E52 which is enabled by a L ⇒ R1-Seq, absence of a f = 20 - 23, and a low from 01T32. The low output along the low from 22G52 at Ø2 sets overflow flip-flop OXG52.

address 005001 (777777) from the contents of the AU-register. This is accomplished as follows:

$000460 = (AU)_i$ with borrow inserted

$\underline{777777 = (005001)}^{1}$

$000461 = (AU)_f$

2-273. It should be noted that this operation also requires an end-around-borrow, but the machine cannot propagate the borrow any further. This EAB would be noted by the setting of the Borrow Test flip-flop. The program would have to be designated to take this problem into consideration and subtract a + 1 from the contents of the A-register to make the answer correct. This is accomlished by a SUBA instruction as follows:

AU	AL
000461	$000314\ (A)_i$
-	1
000461	$000313\ (A)_f$

2-274. ADDAB (f = 21). The ADDAB instruction performs the same service as an f = 20 instruction except that the addresses of the operands in memory are developed by Y + B for the least significant half and by Y + B + 1 for the most significant half.

2-275. SUBA (f = 22). The SUBA instruction subtracts the 36-bit number contained in memory cells Y and Y + 1 from the 36-bit number contained in the A-register. As with an ADDA instruction, two passes of the computer are required for execution, since the computer can only operate with 18-bit numbers at any one time. The least significant operand must be stored in an even-numbered address and will be subtracted from the contents of the AL-register. The most significant operand must be stored in an odd-numbered address and is subtracted from the contents of the AU-register.

2-276. SUBAB (f = 23). The SUBAB instruction performs the same service as an f = 22 instruction except that the addresses of the operands in memory are now developed by Y + B for the least significant half, and by Y + B + 1 for the most significant half.

2-277. R1 and R1/R2-Sequence Data Flow for f = 20 through 23. As shown in figure 2-53, a memory cycle is initiated and the least significant operand is called out of memory. The I-sequence that called the instruction word from memory and developed the address of the least significant operand is not shown. Depending upon the arithmetic operation (Add or Subtract), either the operand or its complement is transferred to the D-register. The contents of the AL-register

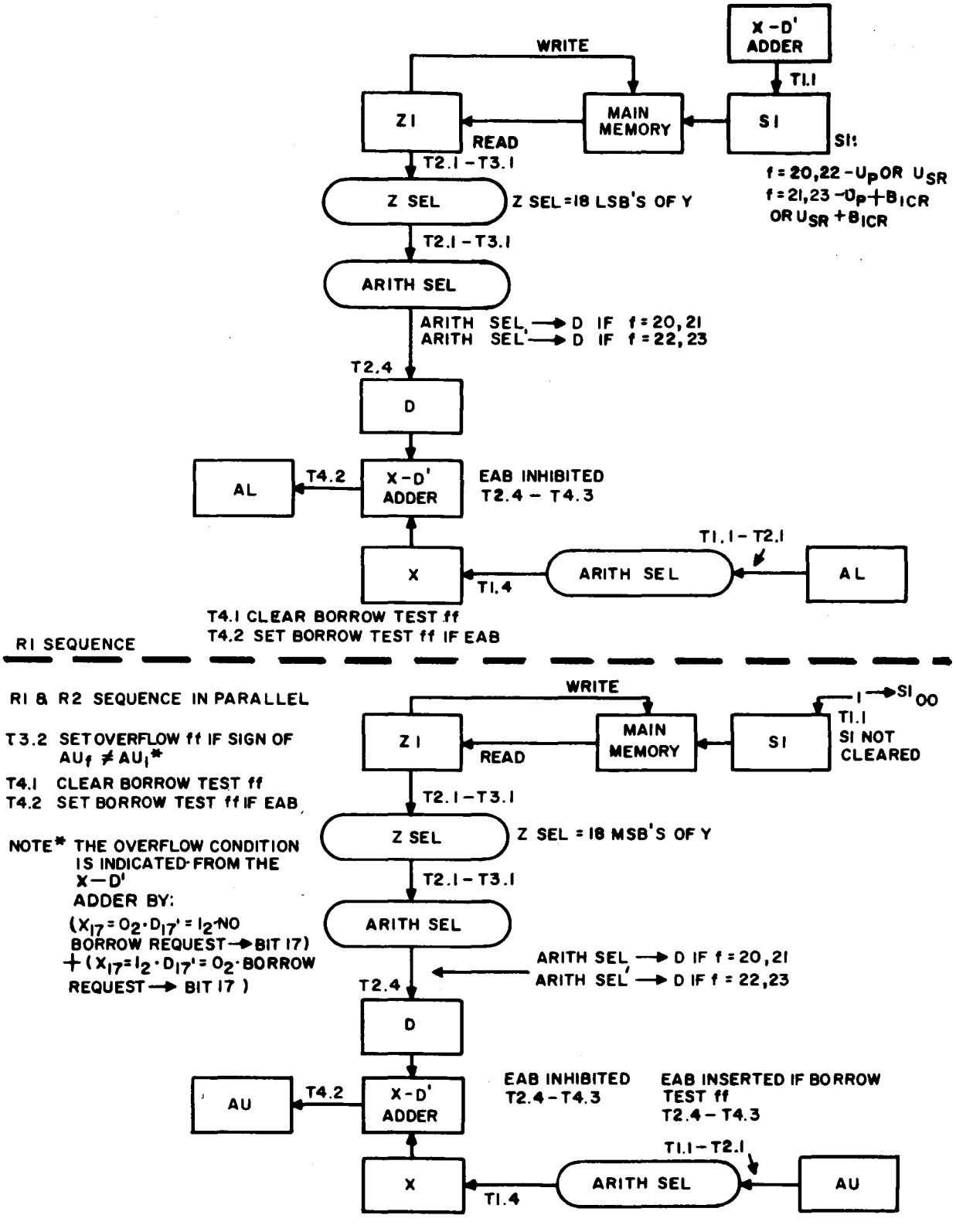


Figure 2-53. R1 And R1/R2-Sequence Data Flow For f = 20 through 23

are transferred to the X-register; the AL-register is cleared and the adder outputs to the AL-register. If an end-around-borrow occurs during this operation, the Borrow Test flip-flop (refer to plate P-30) will be set, so that this borrow can be propagated from the AL-register to the AU-register in the next sequence.

2-278. The R1 and R2-sequences now run in parallel to obtain the most significant half of the operand from memory. The address of the operand is obtained by not clearing the S1-register and setting S_{00} to a +1. The arithmetic operations performed are the same as before except that the AU-register is now involved. An end-around-borrow is inserted into the adder if the Borrow Test flip-flop was set during the previous sequence involving the A register. If an end-around-borrow occured during this operation, the borrow will be inhibited, but recorded by setting Borrow Test flip-flop. An f = 50:51 instruction should be executed to sense the end-around-borrow that was inhibited during the AU operation, allowing the program to make a correction. If an overflow occurs during this operation, the Overflow flip-flop (refer to plate P-30) will be set. This condition can be sensed with a 50:52 or 50:53 instruction, allowing the program to initiate corrective action. Essential commands for f = 20 through 23 in the R1 and R1/R2-sequence are contained in table 2-65. Commands that have not been described previously in the R1-sequence are described in table 2-66 in det

TABLE 2-65. R1 AND R1/R2-ESSENTIAL COMMANDS FOR f = 20 THROUGH 23

Time Notation	Commands
	R1-SEQUENCE
T4.4	Clear S1
T1.1	Adder→S1, Init memory, AL→Arith Sel
T1.3	Clear X, clear Z1
T1.4	Arith Sel→X
T2.1	Z1→Z-Sel, Z-Sel→Arith Sel, drop AL→Arith Sel
T2.3	Clear D
T2.4	Arith Sel→D if f = 20, 21, Arith Sel'→D if f = 22, 23 set Inhib EAB ff
T3.1	Drop Z1→Z-Sel, drop Z-Sel→Arith Sel
T4.1	Clear AL, clear Borrow Test ff
T4.2	Adder→AL, set Borrow Test ff if EAB
T4.3	Clear Inhib EAB ff
	R1 AND R2-SEQUENCES IN PARALLEL
T1.1	$1 \rightarrow S1_{00}$, Init memory, AU→Arith Sel
T1.3	Clear X, clear Z1
T1.4	Arith Sel→X
T2.1	Z1→Z-Sel, Z-Sel→Arith Sel, drop AU→Arith Sel
T2.3	Clear D
T2.4	Arith Sel→D if f = 20, 21, Arith Sel'→D if f = 22, 23 set Insert EAB ff if Borrow Test ff set, set Inhib EAB ff
T3.1	Drop Z1→Z-Sel, drop Z-Sel→Arith Sel
T3.2	Set Overflow ff if sign of $AU_f \neq$ sign of AU_i
T4.1	Clear AU, clear Borrow Test ff
T4.2	Adder→AU, set Borrow Test ff if EAB
T4.3	Clear Insert EAB ff, clear Inhib EAB ff

TABLE 2-66. COMMANDS/DESCRIPTIONS FOR f = 20 THROUGH 23
(REFER TO TABLE 2-21 FOR COMMANDS NOT DESCRIBED)

Time Notation	Command/Description
	R1-SEQUENCE
T2.4	Set Inhib EAB ff. Inhib EAB flip-flop is set. Gate 10E33 (refer to plate P-27) is enabled by a L ⇒ R1-Seq, L ⇒ f = 20-23, and a low from 03T24. The high output is inverted by 11E33 and, at Ø4, Inhibit EAB flip-flop OXG33 is set.
T4.1	Clear Borrow Test ff. Borrow Test flip-flop is cleared, Gate 10E51 (refer to plate P-30) is enabled by a L ⇒ T42 R1-Seq, and a L ⇒ f = 20-23. The high output is inverted and, at Ø1, Borrow Test flip-flop is cleared.
T4.2	Set Borrow Test ff if EAB. Borrow Test flip-flop is set if end-around-borrow. Given an end-around-borrow, 21E51 produces a low output, and with the low from 11E51 at Ø2, Borrow Test flip-flop OXG51 is set.
T4.3	Clear Inhib EAB ff. Inhibit EAB flip-flop is cleared. Gate 01E32 (refer to plate P-27) produces a low from the high from 00T43 and, at Ø3, the high from 02E32 clears Inhibit EAB flip-flop OXG33.
	R1/R2-SEQUENCE
T1.1	$1 \Rightarrow S1_{00}$. Gate 50N12 (refer to plate P-22) produces a high due to a L ⇒ R1-Seq, L ⇒ R2-Seq and a low from 03T11. The high enables 58N12 to provide a L ⇒ Set bit 2^0 to the S1-register and, at Ø1, to set 1X500 (refer to plate P-104). AU → Arith Sel. Gate 10N01 (refer to plate P-17) produces a high due to the L ⇒ T14 R1-Seq, and a L ⇒ f = 20-23 R2-Seq. The high is inverted by 18N01 and 19N01 produces a L ⇒ AU → Select.
T2.1	Drop AU → Arith Sel. AU → Arith Sel is dropped. Gate 10N01 now outputs a low due to the absence of a L ⇒ T14 R1-Seq.
T2.4	Set Insert EAB ff if Borrow Test ff Set. Insert EAB flip-flop is set if Borrow Test flip-flop is set. Gate 10E33 (refer to plate P-27) is enabled and produces a high output by a L ⇒ R1-Seq, L ⇒ f = 20-23, and a low from 03T24. Gate 10E32 is enabled by a L ⇒ R2-Seq and a L ⇒ Borrow, and outputs a high to 11E32. Gate 11E32 provides a low to Insert EAB flip-flop OXG32 and, at Ø4, EAB flip-flop is set.

TABLE 2-66. COMMANDS/DESCRIPTIONS FOR f = 20 THROUGH 23 (Cont)
(REFER TO TABLE 2-21 FOR COMMANDS NOT DESCRIBED)

Time Notation	Command/Description
T3.2	Set Overflow ff. if Sign of $AU_f \neq$ Sign of AU_i. Overflow flip-flop is set if sign of $AU_f \neq$ sign of AU_i. The same gates as previously described are used except that 11E52 (refer to plate P-30) is enabled by a H ⇒ f = 20-23, and 20E52 is enabled by being a R1 and R2-Seq.
T4.1	Clear AU. Same as described for a f=10 through 13 instruction.
T4.2	Adder AU. Same as described for a f=10 through 13 instruction.
T4.3	Clear Insert EAB ff, Clear Inhib EAB ff. Insert EAB flip-flop and Inhibit EAB flip-flop are clear. The high input to 01E32 (refer to plate P-27) from 00T43 provides a low at 02E32 and, at Ø3, Inhibit EAB flip-flop OXG33 and Insert EAB flip-flop OXG32 are cleared.

2-279. RND (f = 50:60). The RND instruction will round the 36-bit number contained in both the AU- and AL-registers in order to formulate an 18-bit rounded number which is placed in the AL-register. The round operation involves modification of the most significant half of the 36-bit number (contained in the AU-register) by bit-17 of the AL-register. The four possible combinations of the AU-register, and bit-17 of the AL-register, are as per table 2-67.

2-280. I-Sequence data Flow For f = 50:60. Figure 2-54 represents the I-sequence data flow for f = 50:60. Most of the I-sequence is similar to that previously described. At time T3.4, the D-register receives the contents of the AU-register. At time T2.3, the X-register is cleared (set to a + 0) and remains in the state unless $AL_{17} = 0_2$. If $AL_{17} = 0_2$, the X-register is set to all ones (-0). Refer to table 2-68 for a list of I-sequence essential commands for f = 50:60. Commands that have not been previously described in the I-sequence are described in table 2-69 in detail.

TABLE 2-67. X-D' ADDER CONDITIONS FOR f = 50:60

AU	AL_{17}	D	X	Other	Adder Output	AL_f
pos	0	AU	-0	none	(-0) - (AU)'	AU + 0
pos	1	AU	+0	EAB inhibited	(+) - (AU)' +1	AU +1
neg	0	AU	-0	EAB inhibited* EAB inserted	(-0) - (AU)' -1	AU - 1
neg	1	AU	+0	none	(+0) - (AU)'	AU - 0

* The inhibiting of the EAB with these conditions is of no use, since no EAB will occur.

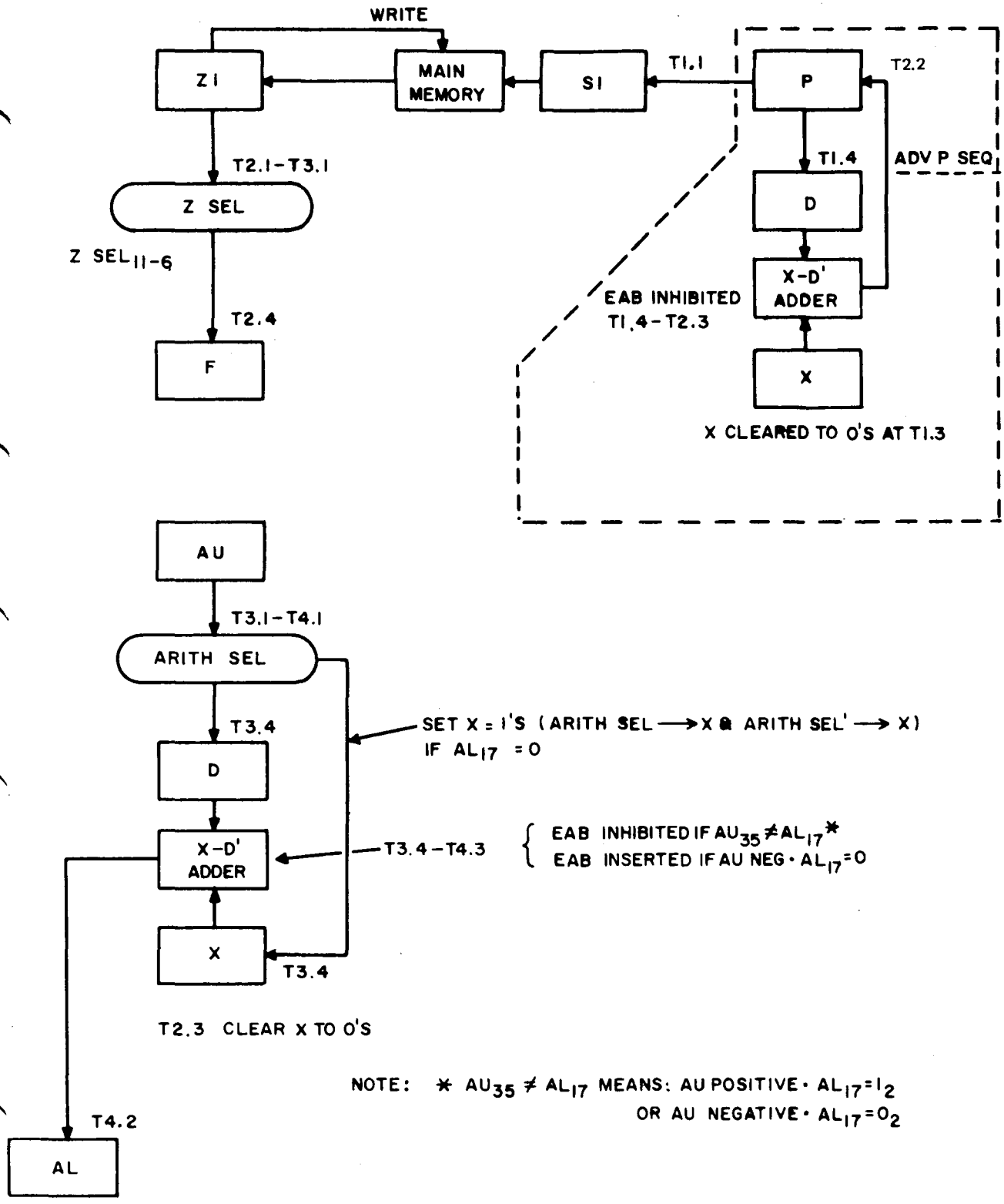


Figure 2-54. I-Sequence Data Flow For f = 50:60

TABLE 2-68. I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:60

Time Notation	Commands
T4.4	Clear Sl
T1.1	P → S1, Init memory, *set Incr P ff
T1.3	*Clear D, *clear X, clear Z1, clear F, *set OXL11 ff
T1.4	*P_L → D_L, *P_U → D_U, *set Inhib EAB ff
T2.1	*Z1 → Z-Sel, *clear Incr P ff
T2.2	*Clear P *Adder → P
T2.3	Clear X, *clear OXL11 ff, *clear Inhib EAB ff
T2.4	Z-Sel_{11-6} → F, set OXF06 ff
T3.1	AU → Arith Sel, drop Z1 → Z-Sel
T3.3	Clear D
T3.4	Arith Sel → D, Arith Sel → X & Arith Sel' → X (Set X = 1's) if $AL_{17} = 0$
	Set Inhib EAB ff if $AU_{35} \neq AL_{17}$, set Insert EAB ff if AU neg · $AL_{17} = 0$
T4.1	Clear AL, drop AU → Arith Sel
T4.2	Adder → AL
T4.3	Clear Inhib EAB ff, clear Insert EAB ff

*These events are concerned with or are controlled by the Advance-P subsequence.

TABLE 2-69. COMMANDS/DESCRIPTIONS FOR f = 50:60
(NOT DESCRIBED IN A PREVIOUS I-SEQUENCE)
(REFER TO TABLE 2-19 FOR COMMANDS NOT DESCRIBED)

Time Notation	Command/Description
T3.1	AU ⇒ Arith Sel Gate 10N01 (refer to Plate P-17) is qualified by a L ⇒ f = 50:60, 62, 63 & a L ⇒ T34 I Seq Format 2. The high output is inverted by 18N01 & 19N01 to provide a L ⇒ AU Select.
T3.3	Clear D. The D-register is cleared as previously described except that gate 10N02 (refer to plate P-18) is enabled by a L ⇒f = 50:60 and a L ⇒ T34 I-Seq Format II.
T3.4	Arith Sel → D Gate 10N02 (refer to Plate P-18) is qualified by a L ⇒f = 50:60 & a L ⇒ T34 I Seq Format 2. The high output is inverted by 18N02 & applied to 19N02. Gate 17N02 is disabled & the low output is applied to 19N02 which produces a L ⇒Select → D at ∅4. Arith Sel → and Arith Sel^1 → X if $AL_{17} = 0$. Flip-flop OXA17 (refer to plate P-100) produces a high output to gate 95A00. With a L ⇒ f = 50:60, gate 95A00 produces a L ⇒Set X = 1's as previously described. Set Inhib EAB ff if $AU_{35} \neq AL_{17}$. If $AU_{35} \neq 0$ and $AL_{17} = 1$, flip-flop OXA35 (refer to plate P-98) will produce a high output from the set side to 60A17 (refer to plate P-27). The high is also inverted by 02A35 (refer to plate P-98) and goes to 60A17 as a low. With AL_{17}=1, flip-flop OXA17 (refer to plate P-100) is set and produces a low which is inverted by 02A17 and sent to 60A17 as a high. The high from the clear side of OXA17 is inverted by 03A17 and sent to 60A17 as a low. If $AU_{35} = 1$ and $AL_{17} = 0$, 60A17 will also be enabled. The low is applied to 12E33 (refer to plate P-27) and with L ⇒ f = 50:60 and L ⇒ T34 I-Seq Format II, 12E33 produces a high at 11E33 and, at ∅4, Inhibit EAB flip-flop OXG33 is set. Set Insert EAB ff if AU Neg $AL_{17} = 0$. With AU Neg flip-flop OXA35 (refer to plate P-98) set, a low is applied to 60A17

TABLE 2-69. COMMANDS/DESCRIPTIONS FOR f = 50:60 (Cont)

Time Notation	Command/Description
	(refer to plate P-27). This low from flip-flop OXA35 is also inverted by 02A35 and applied as a high to 60A17. With $AL_{17} = 0$, flip-flop OXA17 (refer to plate P-100) is cleared and the low from the clear side is inverted by 03A17 and applied to 60A17. The high from the set side is inverted by 02A17 and also applied to 60A17. 12E33 is enabled as described previously. Gate 95A00 (refer to plate P-29) is disabled by the high from the set side of flip-flop OXA17 and applies a low to 10E32 (refer to plate P-27) which is enabled by a L → f = 50:60. The high output from 10E32 and 12E33 enables 11E32 and, at ∅4, Insert EAB flip-flop OXG32 is set.
T4.1	Clear AL. Same as described for a f = 70, 71 instruction.
T4.1	Drop AU → Arith Sel Gate 10N01 (refer to Plate P-17) is now disabled due to T34 going high.
T4.2	Adder → AL. Same as described for a f = 70, 71 instruction.

2-281. Logical Instructions. The logical instructions consist of: CMAL (f = 02), CMALB (f = 03), CMSK (f = 06), and CMSKB (f = 07), which are all SR sensitive. Also included in the logical instructions are SLSU (f = 04), SLSUB (f = 05), SLSET (f = 51), SLCL (f = 52), SLCP (f = 53), CPAL (f = 50:61), CPAU (f = 50:62), and CPA (f = 50:63). The f = 02, 03, 06, and 07 instructions compare an operand obtained from memory with either the contents of the AL-register or the logical product of the AU- and AL-registers. The f = 04, 05, and 51 through 53 instructions control the value placed in each bit position of the AL-register. The f = 50:61 through f = 50:63 instructions complement the value in either AL-, AU-, or A-registers.

2-282. CMAL (f = 02). The CMAL instruction compares an operand (Y) with the contents of the AL-register. The Compare flip-flop is always set by this instruction. The Equal flip-flop is set if the operand is equal to the contents of the AL-register. The Greater Than flip-flop is set if the content of the AL-register is greater than the operand. Under normal circumstances, a conditional jump instruction (60-67) will follow a Compare instruction so that the results of the comparison can be used for some purpose. If a 60-67 instruction does not follow a Compare instruction, the results of the comparison are lost because of the clearing of the Compare logic.

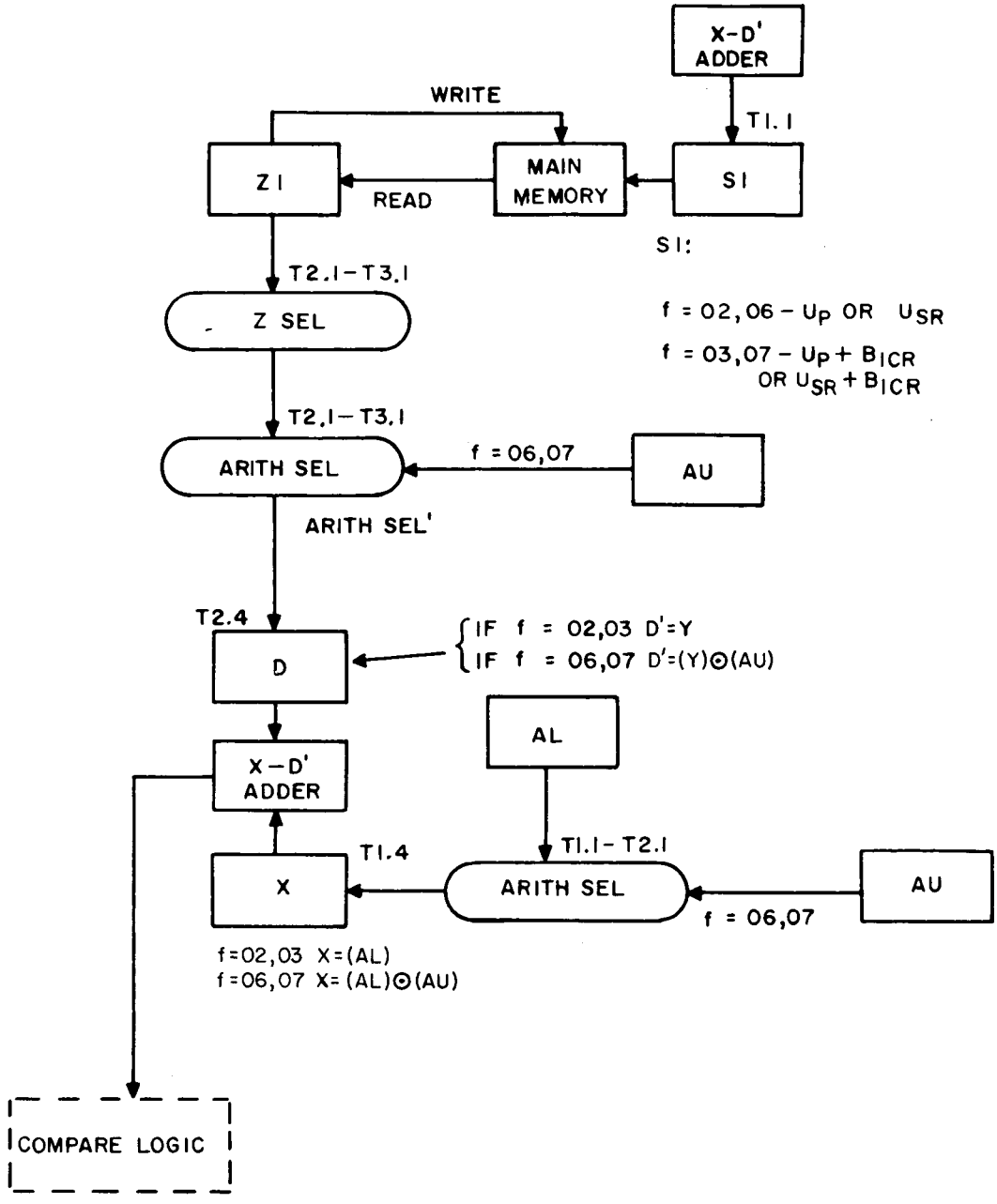
2-283. CMALB (f = 03). The CMALB instruction performs the same service as an f = 02 instruction except that the address of the operand is obtained by address Y + B.

2-284. CMSK (f = 06). The CMSK instruction will compare selected bits of the AL-register with selected bits of an operand obtained from memory after using the contents of the AU-register as a mask. The contents of the X- and D-registers are then compared as with a normal Compare (f = 02) instruction.

2-285. CMSKB (f = 07). The CMSKB instruction performs the same service as an f = 06 instruction except that the operand is obtained from memory address Y + B.

2-286. R1-Sequence Data Flow for f = 02, 03, 06, and 07. Figure 2-55 illustrates the R1-Sequence Data Flow for f = 02, 03, 06, and 07. During the I-sequence, which is not shown in this figure, the instruction word was obtained from memory and the address of the operand was formulated. As shown in the figure, the R1-sequence obtains the operand from memory. The X-register receives either the contents of the AL-register or the logical product of AL and AU via the arithmetic selector at T1.4. The D-register receives either the complement of the operand or the complement of the logical product of the operand and the AU-register. The adder will perform the comparison by subtracting the X- and D'-register contents and outputting the result to the comparison logic. Refer to table 2-70 for a sequential list of essential R1-sequence events. Commands that have not been previously described in the R1-sequence are described in table 2-71 in detail.

2-287. Clearing of Comparison Flip-Flops. The Greater, Compare, and Equal flip-flops (refer to plate P-29) are cleared by a high level from gate 00E34. Gate 00E34 clears these flip-flops at time T4.1 of the I-sequence for any instruction other than f = 60-67. The f = 60-67 instructions test the state of these flip-flops. Therefore, these instructions must immediately follow the f = 02, 03, 06, and 07 instructions in order to detect the result of the comparison.



T3.2 SET COMPARE ff
SET EQUAL ff IF X = D' *
SET GREATER ff IF X ≥ D' ALGEBRAICALLY **

NOTES: * X = D' MEANS: f = 02,03 (AL=Y)
f = 06,07 (AL·AU)=(Y·AU)
** X ≥ D' MEANS: f = 02,03 (AL ≥ Y)
f = 06,07 (AL·AU) ≥ (Y·AU)

Figure 2-55. R1-Sequence Data Flow For f = 02, 03, 06, and 07

TABLE 2-70. R1-SEQUENCE ESSENTIAL COMMANDS FOR f = 02, 03, 06, and 07

Time Notation	Commands
T4.4	Clear S1
T1.1	Adder → S1, Init memory, AL → Arith Sel
	AU → Arith Sel if f = 06, 07
T1.3	Clear Z1, clear X
T1.4	Arith Sel → X
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel, drop AL → Arith Sel, drop AU → Arith Sel, AU → Arith Sel if f = 06, 07
T2.3	Clear D
T2.4	Arith Sel' → D
T3.1	Drop Z1 → Z-Sel, drop Z-Sel → Arith Sel, drop AU → Arith Sel
T3.2	Set Compare ff, set Equal ff if X = D'
	Set Greater ff if X ≥ D' algebraically

TABLE 2-71. COMMANDS/DESCRIPTIONS FOR f = 02, 03, 06, AND 07 (REFER TO TABLE 2-21 FOR COMMANDS NOT DESCRIBED IN A PREVIOUS R1-SEQUENCE)

Time Notation	Command/Description
T3.2	Set Compare ff, Equal ff if X = D'. The Compare flip-flop (refer to plate P-29) is set by gate 10E34 unconditionally. At the same time, gate 81A00 sets the Equal flip-flop is X = D'. Gate 81A00 is conditioned by the AND function of the outputs of gates 80A00, 80A04, 80A09, and 80A13. These gates use the outputs of certain X-D' adder gates to determine the equality of a portion of the X and D' values. For example, the 10A00 input to 80A00 is at a low level if $X_{00} = D'_{00}$. (Refer to logic diagrams, plate P-94, for the analysis of the operation of gate 10A00.) Gate 10A00 receives input from bit position 00 of the X- and D-registers. The output, however, is described in terms of X and D'. A low-level output of 10A00 indicates $X_{00} = D'_{00}$. Refer to table 2-72 for the conditions which produce a low level from 10A00. Each of the other 10A-gates of the adder is used to determine equality in its particular bit position of X and D'. Set Greater ff if X ≥ D algebraically. When the Compare flip-flop is set, the Greater flip-flop is also set if X is more positive than D', as indicated by a low-level output of gate 70A17. Gate 70A17 compares X

TABLE 2-71. COMMANDS/DESCRIPTIONS FOR f = 02, 03, 06, AND 07 (Cont)

Time Notation	Command/Description
	and D' by sensing the signs of these values from the adder bit position 17 and the end-around-borrow condition involved in the X-D' subtraction. Input 10A17 to 70A17 is like that previously discussed for 10A00. A high level is output by 10A17 if $X_{17} = D'_{17}$. The condition for this high level is best expressed as X and D' (signs alike). Refer to table 2-73 for conditions which produce a low level from 70A17.

TABLE 2-72. CONDITIONS FOR LOW LEVEL OUTPUT OF 10A00

Condition	Input To 10A00			
	00D00	00X00	01D00	01X00
$X_{00} = 0$ & $D'_{00} = 0$	H	L	L	H
$X_{00} = 1$ & $D'_{00} = 1$	L	H	H	L

TABLE 2-73. CONDITIONS FOR LOW LEVEL OUTPUT OF 70A17 (SET GREATER FF)

Condition	Inputs to 70A17			
	12A00	11A17	13A00	10A17
Unlike signs · EAB (X pos & D'neg)	H	L	L	H
Like signs · No EAB (/X/≥/D'/)	L	H	H	L

2-288. SLSU (f = 04). The SLSU instruction selectively substitutes the bits of the operand Y for the bits of AL as controlled by AU. The origin of Y is memory at address U_p, if SR is inactive, or U_{SR} if SR is active. Where AU is 1_2, the corresponding bit of Y is put in AL in place of the initial bit of AL. Where AU is a 0_2, the corresponding bit of AL is retained unchanged, and AU is not disturbed. The logical term for AL_f is $(\overline{AU}) \odot (AL_i) + (AU) \odot (Y)$.

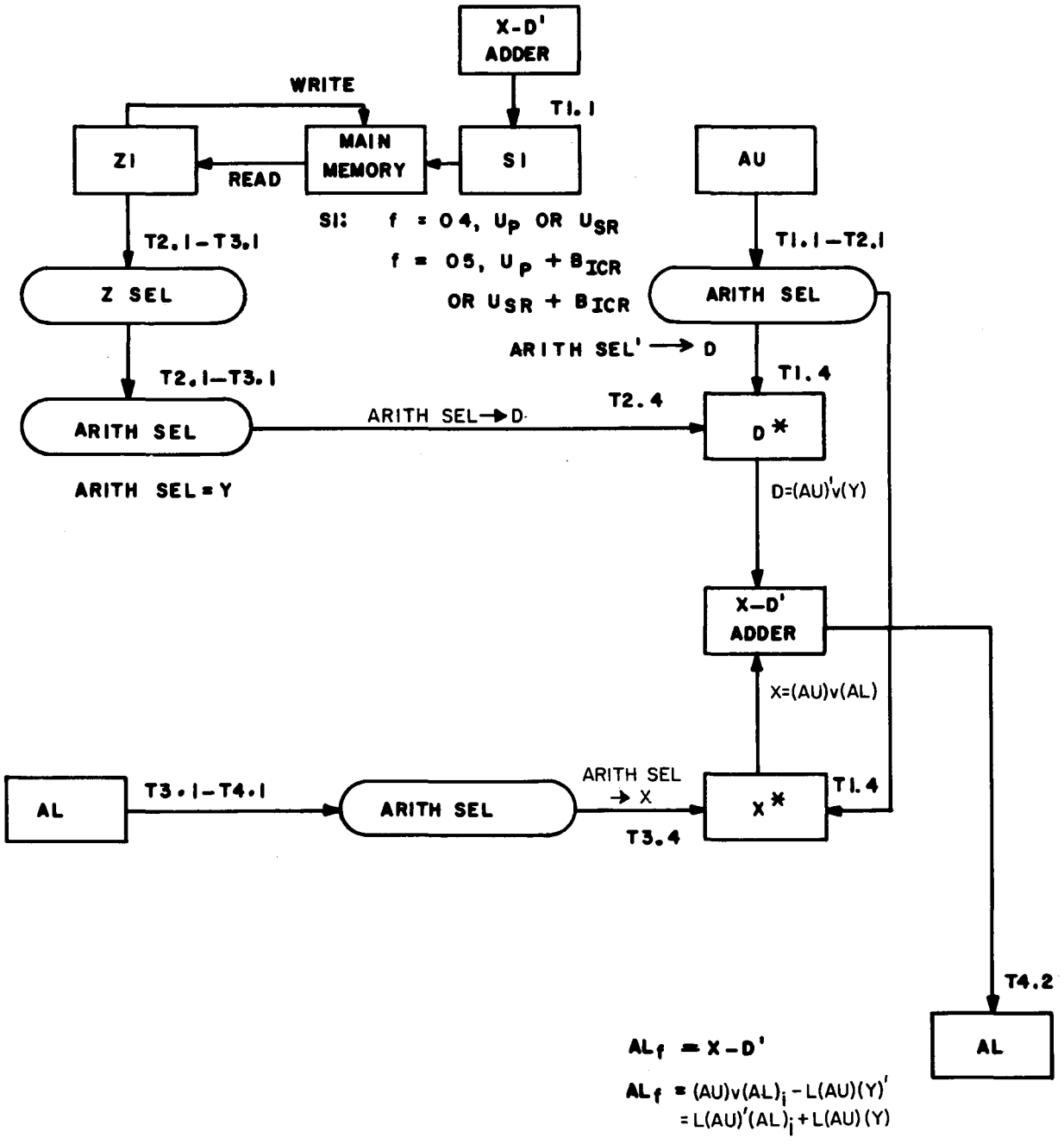
2-289. SLSUB (f = 05). Except for the memory address, the SLSUB instruction is the same as f = 04. The address of Y is either U_p + B or U_{SR} + B depending upon the activeness of SR. The B-register is specified by ICR.

2-290. SLSET (f = 51). The SLSET instruction selectively sets the bits of AL as controlled by the operand Y, and will effectively perform an inclusive OR. The origin of Y is memory at address U_p. Where Y is 1_2, the corresponding bit of AL is set to 1_2. Where U is 0_2, the corresponding bit of AL is retained unchanged. The logical term for AL_f is $(AL)_i$ V(Y).

2-291. SLCL (f = 52). The SLCL instruction selectively clears the bits of AL as controlled by the operand Y and will effectively perform a logical AND function. The origin of Y is memory at address U_p. Where Y is 0_2, the corresponding bit of AL is cleared to a 0_2. Where Y is 1_2, the corresponding bit of AL is retained unchanged. The logical term for AL_f is $(AL)_i \odot (Y)$.

2-292. SLCP (f = 53). The SLCP instruction selectively complements the bits of AL controlled by the operand Y, and will effectively perform an exclusive OR. The origin of Y is memory at address U_p. Where Y is 1_2, the corresponding bit of AL is complemented. Where Y is 0_2, the corresponding bit of AL is retained unchanged. The logical term for AL_f is (AL) $\oplus$ (Y).

2-293. R1-Sequence Data Flow for f = 04, and 05. Refer to figure 2-56 for the R1-sequence flow for f = 04 and 05. The I-sequence that obtained the instruction word from memory and formulated the operand address is not shown on this figure. The R1-sequence used a memory reference to obtain the operand Y. D receives the complement of AU from arithmetic select at time T1.4. Without clearing, D also received Y at time T2.4. D applies the inclusive OR function value (AU)' v (Y) to one side of the X-D' adder. X receives AU at time T1.4 and also, without clearing, receives AL at time T3.4. Consequently, X applies the inclusive OR function value of (AU) v (AL) to the other side of the X - D' adder. AL is cleared and set to the value of X - D' from the adder which is actually (AU) v (AL) - (AU)'v (Y). This term can best be expressed as (AU) v (AL) - L(AU) (Y)! This final result in AL should be such that each bit position is set to the corresponding bit of either AL_i or Y depending upon the value in AU. No bit position value is in any way dependent upon any other bit position. To use the arithmetic adder, interaction among the bit positions must be prevented by ensuring that borrow conditions will not occur. A borrow condition exists when, in the same bit position, the subtraction is $0_2 - 1_2$. For these instructions the adder subtracts (AU) v $(AL)_i$ - L(AU) (Y)! If the value of (AU) v $(AL)_i$ is equal to 0_2, AU must be equal to 0_2 and, therefore the value of L(AU) (Y)' must also be equal to 0_2. Thus, whenever X is equal to 0_2, D'is also equal to 0_2. A borrow condition can never exist during this logical operation, and no bit position can be affected by another position. To



NOTES: * X AND D ARE NOT CLEARED AFTER T1.4 ENTRY

Figure 2-56. R1-Sequence Data Flow For f = 04, 05

analyze the value in AL_f of (AU) v $(AL)_i$ - L(AU) $(Y)'$ refer to table 2-74 (Truth Table). Develop the AL_f values in this table by considering each configuration of values for AU, AL_i, and Y. This same truth table also holds true for the AL_f value of $L(AU)'(AL) + L(AU)(Y)$. The term satisfies the expression "substitute Y for AL where AU is an 1_2." If AU is an 0_2, the value in the corresponding AL_f bit position is the same as the initial AL value. As will be discussed later, the operand can be obtained from bootstrap or control memory.

2-294. R1-Sequence Data Flow for f = 51 and 52. Refer to figure 2-57 for the R1-Sequence flow for f = 51 and 52. The I-sequence that obtained the instruction word from memory and formulated the address of the operand is not shown in the figure. The R1-sequence uses a memory reference to obtain the operand Y. Y is applied to arithmetic-select from Z-select. If f = 52, AL is also applied to arithmetic-select, which formulates the logical AND function value of AL_i and Y. This value is applied to one side of the X-D' adder, from D. The other side of the adder senses -0 from X. AL is cleared and receives the X - D' value of $(-0) - (L(AL_i)(Y))'$ which is effectively $L(AL)_i(Y)$. This term satisfied the expression "clear AL where Y is an 0_2". If Y is an 1_2, the value in the corresponding AL_f bit position is the same as the initial AL value. If f = 51, operations are very similar to those for f = 52. The logical AND function is not used. D simply receives Y and applies it to one side of the X - D' adder. AL is not cleared of its initial value and receives the X - D' value of $(-0) - Y'$, which is effectively Y. The inclusive OR function of $(AL)_i$ v (Y) is formulated in AL_f. This term satisfies the expression "set AL to 1_2 where Y is 1_2". If Y is an 0_2, the value in the corresponding AL_f bit position is the same as the initial AL value. As will be discussed later, the operand can be obtained from bootstrap or control memory.

2-295. R1-Sequence Data Flow for f = 53. Refer to figure 2-58, for the R1-sequence flow for f = 53. The I-sequence in which

TABLE 2-74. TRUTH TABLE OF (AU) v $(AL)_i$ - L(AU) $(Y)' = AL_f$

AU	AL_i	Y	AL_f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

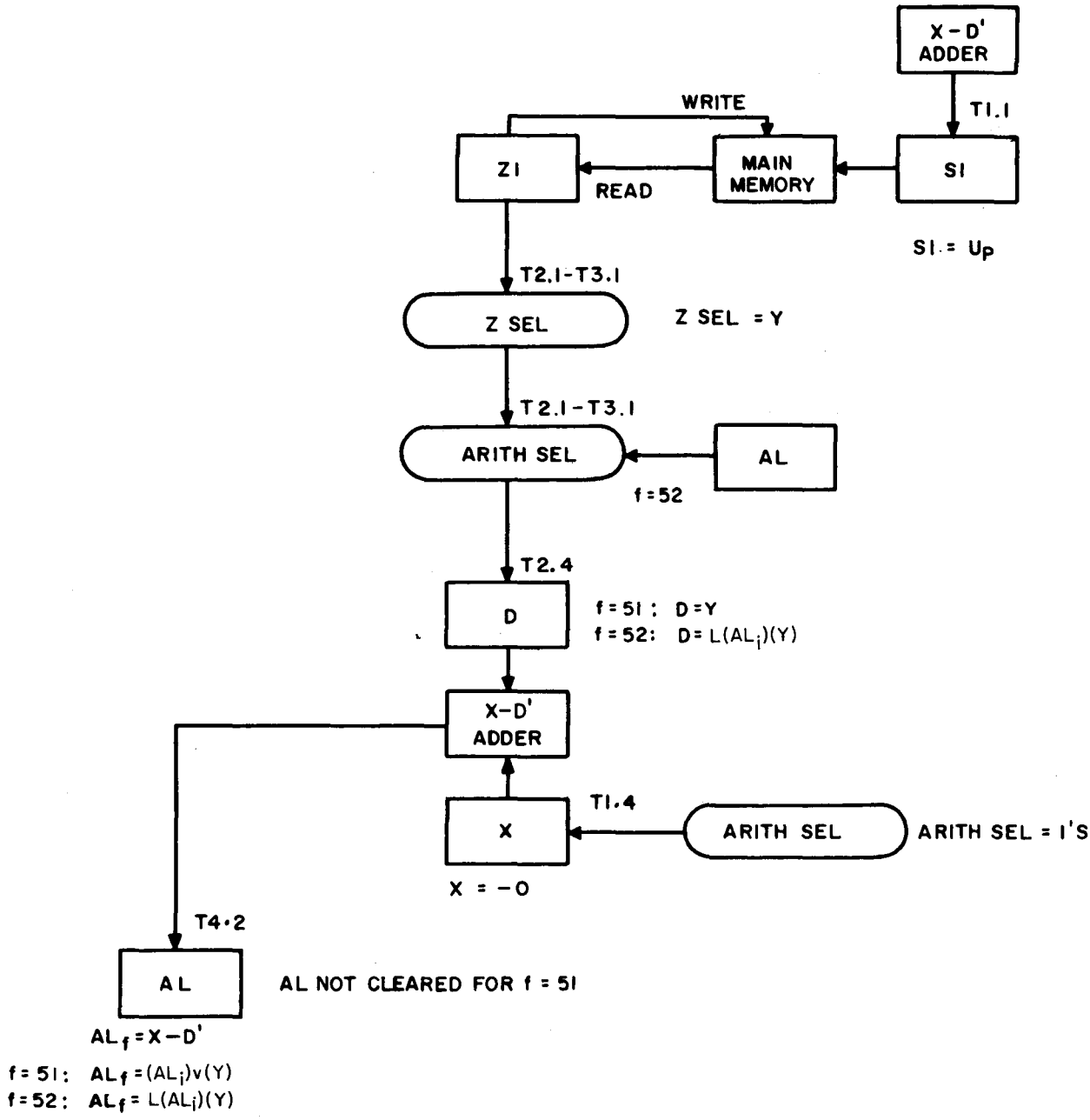
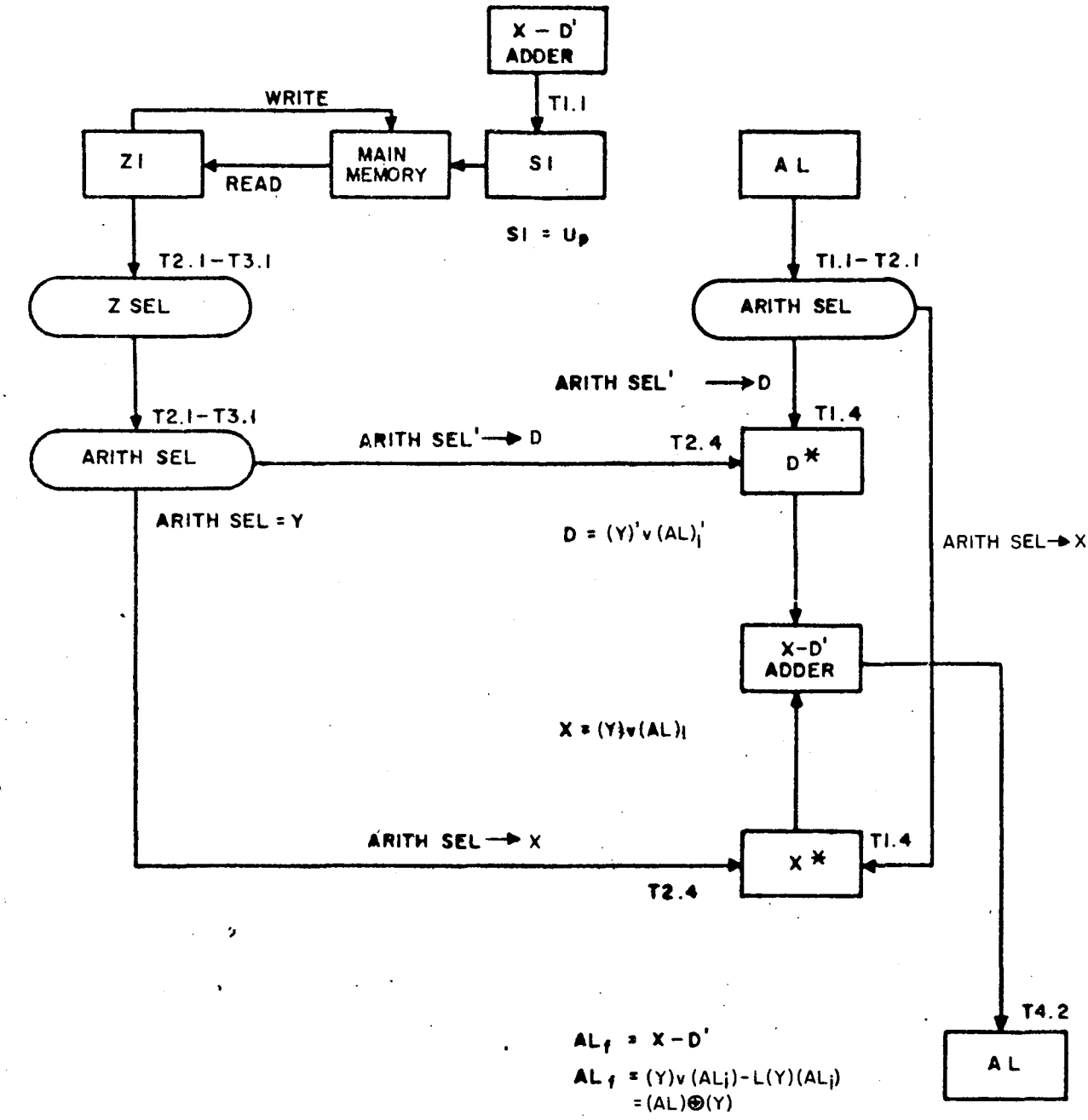


Figure 2-57. R1-Sequence Data Flow For f = 51, 52



NOTES: * X AND D ARE NOT CLEARED AFTER T1.4 ENTRY.

Figure 2-58. R1-Sequence Data Flow For f = 53

the instruction word was obtained and the operand address formulated is not shown on this figure. The R1-sequence uses a memory reference to obtain the operand Y. D receives the complement of AL_i from arithmetic-select at time T1.4. Without clearing, D also receives the complement of Y at time T2.4. Therefore, D applies the inclusive OR function value of Y′ and AL_1′ to one side of the X - D′ adder. Without clearing, X receives AL at time T1.4 and Y at time T2.4. Consequently, X applies the inclusive OR function value of Y and AL_i to the other side of the X - D' adder. AL is cleared and set to the value of X - D' from the adder which is actually (Y) v (AL_i) - L(Y) (AL). In the adder, borrow conditions are prevented to ensure that any AL bit position value is not affected by any other bit positions. A borrow condition exists when, in the same bit position, the subtraction is $0_2 - 1_2$. For this instruction, the adder subtracts (Y) v (AL_i) - L(Y) $(AL)_i$. If the value of Y + AL_i is equal to 0_2, AL_i must be equal to 0_2; and, therefore, the value of $L(AL_i)$ (Y) must also be equal to 0_2, D′ is also equal to 0_2. A borrow condition can never exist during this logical operation, and no bit position can be affected by another bit position. To analyze the value of AL_f of (Y) v (AL_i) - L(Y) (AL_i) refer to table 2-75. (Truth Table). Develop the AL_f values in this table by considering each configuration of values for AL_i and Y. This same truth table also holds true for the AL_f value of (AL) ⊕ (Y). This term satisfied the expression "complement AL where Y is 1_2". If Y is 0_2, the value in the corresponding AL_f bit position is the same as the initial AL value. As will be discussed later, the operand could be obtained from bootstrap or control memory.

2-296. R1-Sequence Essential Commands. Refer to table 2-76 for a sequential list of R1-sequence events. Commands not previously described in an R1-sequence are described in table 2-77 in detail.

2-297. CPAL (f = 50:61). The CPAL instruction will perform a complement of the contents of the AL-register. If the origina content of the AL-register is a positive zero, it will not be complemented. For example:

$$(AL)_i = 345710_8 \rightarrow (AL)_f = 432067_8$$

$$(AL)_i = 000000 \rightarrow (AL)_f = 000000$$

2-298. CPAU (f = 50:62). The CPAU instruction will perform a complement of the contents of the AU-register. As in the 50.61 instruction, a positive zero will not be complemented.

2-299. CPA (f = 50:63). The CPA instruction will perform a complement of the contents of the A-register (combined AU- and AL-registers). As in a 50:61 or 50:62 instruction, a positive zero cannot be complemented.

2-300. I and Next I-Sequence Data Flow f f = 50:61, 50:62, and 50:63. Refer to figure 2-59 for the I and Next I-Sequence data flow for f = 50:61, 50:62, and 50:63. Most of the I-sequence operations are as previously described in a typical I-sequence. If f = 50:62 or 50:63, at time T3.4, the D-register receives the complement of the AU-register via the arithmetic selector. This value is applied to one side of the adder. If the initial content of the AU-register is not a positive zero, the X-register is set to all ones, and the adder will output the complement of the AU-register. The X-register is cleared at time T2.3, and will remain cleared if the AU-register initially contained a positive zero. The adder will not output

TABLE 2-75. TRUTH TABLE OF (Y) v (AL_i) - L(Y) $(AL)_i = AL_f$

AL_i	Y	AL_f
0	0	0
0	1	1
1	0	1
1	1	0

TABLE 2-76. R1-SEQUENCE ESSENTIAL COMMANDS FOR f=04, 05, 51 THROUGH 53

Time Notation	Commands	f = 04, 05	51	52	53
T4.4	Clear S1	X	X	X	X
T1.1	Adder → S1, init memory	X	X	X	X
	AU → Arith Sel	X			
	AL → Arith Sel				X
T1.3	Clear Z1, clear X	X	X	X	X
	Clear D	X			X
T1.4	Arith Sel → X	X	X	X	X
	Arith Sel' → D	X			X
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel	X	X	X	X
	AL → Arith Sel			X	
	Drop AU → Arith Sel	X			
	Drop AL → Arith Sel				X
T2.3	Clear D		X	X	
T2.4	Arith Sel → D	X	X	X	
	Arith Sel' → D, Arith Sel → X				X
T3.1	AL → Arith Sel	X			
	Drop Z1 → Z-Sel, drop Z-Sel → Arith Sel	X	X	X	X

TABLE 2-76. R1-SEQUENCE ESSENTIAL COMMANDS FOR f = 04, 05, 51 THROUGH 53 (Cont)

Time Notation	Commands	f = 04, 05	51	52	53
T3.1	Drop AL → Arith Sel			X	
T3.4	Arith Sel → X	X			
T4.1	Clear AL	X		X	X
	Drop AL → Arith Sel	X			
T4.2	Adder → AL	X	X	X	X

TABLE 2-77. COMMANDS/DESCRIPTIONS FOR f = 04, 05, 51, 52 AND 53 (REFER TO TABLE 2-19 FOR COMMAND NOT DESCRIBED PREVIOUSLY IN AN R1-SEQUENCE)

Time Notation	Command/Description
T2.1	AL → Arith Sel. This event is the same as described at T1.1 of the R1-sequence except that gate 20N01 (refer to plate P-17) is enabled by a L ⇒ f = 52 and a L ⇒ T24 R1-Seq. Drop AU → Arith Sel. Gate 10N01 (refer to plate P-17) is disabled by the absence of a L ⇒ T14 R1-Seq.
T2.4	Arith Sel → X. The event is the same as described at T1.4 of the R1-sequence except that gate 10N03 (refer to plate P-19) is enabled by a L ⇒ f = 53 and a L ⇒ T24 R1-Seq.
T3.1	AL → Arith Sel. The event is the same as described at T1.1 of the R1-Sequence except that gate 20N01 (refer to plate P-17) is enabled by a L ⇒ f = 04, 05 and a L ⇒ T34 R1-Seq. Drop AL → Arith Sel. Gate 20N01 (refer to plate P-17) is disabled by the absence of a L ⇒ T24 R1-Seq.
T3.4	Arith Sel → X. Gate 10N03 (refer to plate P-19) is enabled by a L ⇒ f = 04, 05 and a L ⇒ T34 R1-Seq.
T4.1	Drop AL → Arith Sel. Gate 20N01 (refer to plate P-17 is diasbled by the absence of a L ⇒ T34 R1-Seq.

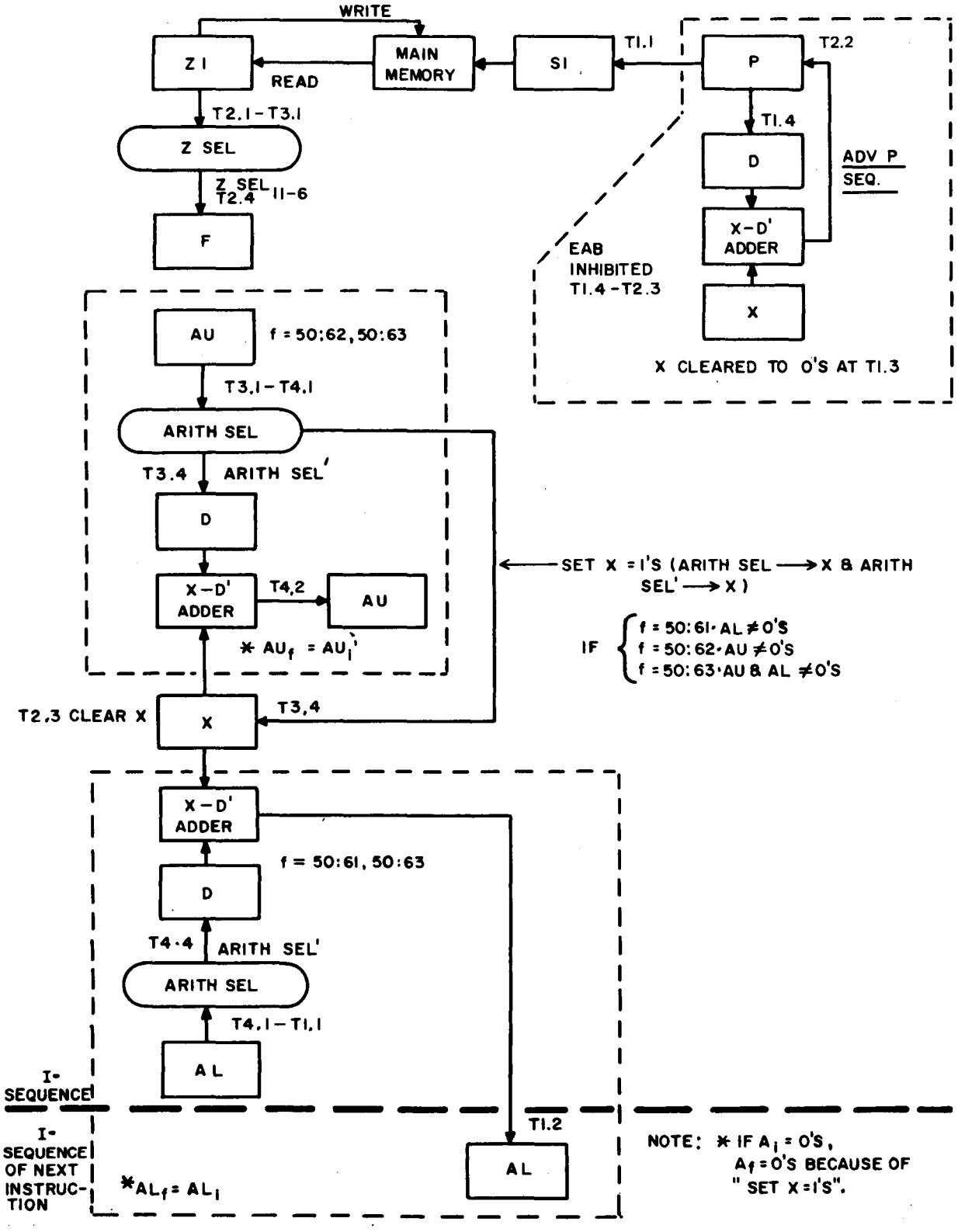


Figure 2-59. I and Next I-Sequence Data Flow For f = 50:61, 50:62, 50:63

a negative zero to the AU-register. The complementing of the AL-register for f = 50: 61 or 50:63 is essentially the same as for complementing the AU-register. An additional main trimming flip-flop (T52) is used for f = 50:61, 50:63 to provide the timing for the clear-AL and adder → AL commands. Refer to plate P-11 for a logic diagram of this flip-flop. Refer to table 2-78 for a sequential list of essential I-sequence events. Commands that have not been described previously in an I-sequence are described in table 2-79 in detail.

2-301. Shift Instructions. The shift instructions consists of RSHAU (f = 50:41), RSHAL (f = 50:42), RSHA (f = 50:43), SF (f = 50:44), LSHAU (f = 50:45), LSHAL (f = 50:46) and LSHA (f = 50:47). These instructions, with the exception of SF, shift the contents of the AU-, AL- or A-registers either left or right. The SF instruction is used to normalize the contents of the A-register.

2-302. RSHAU (f = 50:41). The RSHAU instruction shifts the contents of the AU-register to the right by the number contained in the six least significant bits of the instruction word. The high-order bits of the contents of AU are replaced by the original sign bit, and the low-order bits are lost. For example, if the octal number 016471 is inserted into the AU-register and a 504106 instruction is executed, the number contained in the AU-register after instruction execution will be 000164.

2-303. RSHAL (f = 50:42). The RSHAL instruction performs the same function as the 50:41 on the contents of the AL-register.

2-304. RSHA (f = 50:43). The RSHA instruction shifts the contents of the A-register (combination of AU- and AL-registers) to the right by the number contained in the least significant bit positions of the instruction word. The high-order bits are replaced by the original sign bit and the low-order bits are lost. For example, if the A-register contains the numbers 710626 and 124106, and a 504302 instruction is executed, the numbers contained in the A-register after instruction execution will be 762145 and 425021.

2-305. SF (f = 50:44). The SF instruction is used to normalize the contents of the A-register, either fully or for a specified count contained in the K Counter. The contents of the A-register are shifted circularly to the left until bits A_{34} and A_{35} are not equal, or until the quantity has been shifted to the left, the number of places specified by the count in the K Counter. Upon completion of the scale factor operation, the value remaining in the KO register is stored at address 00017 in control memory. For example, if the numbers 124601 and 241602 are entered into the A-register and instruction 504406 is executed, the numbers contained in the A-register after instruction execution will be 251402 and 502144. It should be noted, that even though K is 06, only one shift to the left is allowed before $A_{35} \neq A_{34}$. Therefore, the number 5 will be stored at address 000017 (6-1= 5). If the numbers 000012 and 460124 are entered into the A-register and instruction 504406 is again executed, the number contained in the A-register will be 001246 012400. Note that the maximum programmed shift count of 6 will be allowed this time because $A_{35} \neq A_{34}$ during the shift process. Therefore, a zero will be stored at address 000017 (6-6=0).

2-306. LSHAU (f = 50:45). The LSHAU instruction shifts the contents of the AU-register to the left by the number contained in the six least significant bit positions of the instruction word. This is a circular shift, where the most significant bit is shifted to

TABLE 2-78. I AND NEXT I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:61, 50:62, 50:63

Time Notation	Commands f =	50:61	50:62	50:63
	I-SEQUENCE			
T4.4	Clear S1	X	X	X
T1.1	P → S1, Init Memory, *set Incr P ff	X	X	X
T1.3	Clear Z1, *set 0XL11 ff	X	X	X
	*Clear D, *Clear X, Clear F	X	X	X
T1.4	*P_L → D_L, *P_U → D_U	X	X	X
	*Set Inhib EAB ff	X	X	X
T2.1	Z1 → Z-Sel	X	X	X
	*Clear Incr P ff	X	X	X
T2.2	*Clear P, Adder → P	X	X	X
T2.3	Clear X, *clear Inhib EAB ff	X	X	X
	*Clear 0XL11 ff	X	X	X
T2.4	Z Sel_{11-6} → F, set 0XF06 ff	X	X	X
T3.1	Drop Z1 → Z-Sel	X	X	X
	AU → Arith Sel		X	X
T3.3	Clear D		X	X
T3.4	Arith Sel' → D		X	X
	**Arith Sel → X & Arith Sel' → X if AL ≠ 0's	X		
	**Arith Sel → X & Arith Sel' → X if AU ≠ 0's		X	
	**Arith Sel → X & Arith Sel' → X if AU & AL ≠ 0's			X
T4.1	Clear AU, drop AU → Arith Sel		X	X
	AL → Arith Sel	X		X
T4.2	Adder → AU		X	X
T4.3	Set T52 ff (Main Timing), Clear D	X		X
T4.4	Arith Sel' → D	X		X
T1.1	Clear AL, drop AL → Arith Sel	X		X
T1.2	Adder → AL	X		X
T1.3	Clear T52 ff (Main Timing)	X		X

*These events are concerned with or are controlled by the Advance-P subsequence.
**Arith Sel → X and Arith Sel' → X occurring simultaneously through the function of set X = 1's.

TABLE 2-79. COMMANDS/DESCRIPTIONS FOR f = 50:61, 50:62, 50:63
(REFER TO TABLE 2-19 FOR COMMANDS NOT DESCRIBED
IN A PREVIOUS R1-SEQUENCE)

Time Notation	Command/Description
T3.1	AU → Arith Sel. Same as described for a f = 50:60 instruction.
T3.3	Clear D. Gate 20N02 (refer to plate P-18) is enabled by a L ⇒ f = 50:62, 50:63 and a L ⇒ T34 I-Seq-Format II.
T3.4	Arith Sel → D. The high out of 20N02 is used to perform the command. Arith Sel → X and Arith Sel' → X if AL ≠ 0's. If AL ≠ 0's, gate 91A00 (refer to plate P-29) produces a high output to 95A00 which outputs a L ⇒ Set X = 1's to gate 20N03 (refer to plate P-19) producing a high output due to the other low inputs of L ⇒ f = 50:60, 50:63 and L ⇒ T34 I-Seq Format II. The high out of 20N03 is inverted by gates 18N03 and 28N03 and, at Ø4, 19N03 and 29N03 produce a L ⇒ Select → X and a L ⇒ $\overline{\text{Select}}$ → X. Arith Sel → X and Arith Sel' → X if AU ≠ 0's. If AU ≠ 0's, gate 91A18 (refer to plate P-29) produces a high output to 95A00 and produces a L ⇒ Select → X or a L ⇒ $\overline{\text{Select}}$ → X as previously described. Arith Sel → X and Arith Sel' → X if AU and AL = 0's. The high output of gate 91A18 and 91A00 (refer to plate P129) disables gate 92A00, the low output is inverted by 93A00, and 95A00 produces a L ⇒ Select → X, and a L ⇒ $\overline{\text{Select}}$ → X as previously described.
T4.1	Clear AU. Gate 11N04 (refer to plate P-20) is enabled by a L ⇒ f = 50:62, 50:63 and a L ⇒ T42 I-Seq, and the high output is inverted by 07N04 to produce a H ⇒ Clear AU as previously described. Drop AU → Arith Sel. Gate 10N01 (refer to plate P-17) produces a low output due to the absence of a L ⇒ T34 I-Seq Format II. AL → Arith Sel. Gate 20N01 (refer to plate P-17) produces a high output due to a L ⇒ f = 50:61, 50:63 and a L ⇒ T44 I-Seq Format II. Gate 29N01 is enabled as previously described to produce a L ⇒ AL → Select.
T4.2	Adder → AU. The high out of 11N04 (refer to plate P-20) is used to produ a L ⇒ Adder → AU as previously described.
T4.3	Set T52 ff. Flip-flop OXT52 (refer to plate P-11) is set by a L ⇒ f = 50:61, 50:63, L ⇒ T44 I-Seq Format II and a Ø3.

TABLE 2-79. COMMANDS/DESCRIPTIONS FOR f = 50:61, 50:62 and 50:63 (Cont)

Time Notation	Command/Description
T4.3 (cont)	Clear D. The event is the same as previously described except that gate 20N02 (refer to plate P-18) is enabled by a L ⇒ f = 50:61, 50:63, and a L ⇒ T44 I-Seq Format II.
T4.4	Arith Sel' → D. The high out of 20N02 is used to produce a L ⇒ $\overline{\text{Select}}$ → D as previously described.
T1.1	Clear AL. The event is the same as previously described except gate 07N05 (refer to plate P-20) produces a low output due to a H ⇒ f = 50:61, 50:63 and T52 on input.
	Drop AL → Arith Sel. Gate 20N01 (refer to plate P-17) is now disabled by the absence of a L ⇒ T44 I-Seq Format II.
T1.2	Adder → AL. The event is the same as previously described except the input to gate 18N05 (refer to plate P-20) is a H ⇒ f = 50:61, 50:63 and T52.
T1.3	Clear T52 ff. Flip-flop 0XT52 (refer to plate P-11) is cleared by the low output from 0CT44 and Ø3.

the left, and will appear as the least significant bit on the right. For example, if the AU-register contains the number 710625, and a 504506 instruction is executed, the number contained in the AU-register after instruction execution will be 062571.

2-307. LSHAL (f = 50:46). The LSHAL instruction performs the same function as the f = 50:45 on the contents of the AL-register.

2-308. LSHA (f = 50:47). The LSHA instruction shifts the contents of the A-register circularly to the left by the number contained in the six least significant bit positions of the instruction word. For example, if the A-register contains the number 771062 451042 and a 504704 instruction is executed, the number contained in the A-register after instruction execution will be 621451 221057.

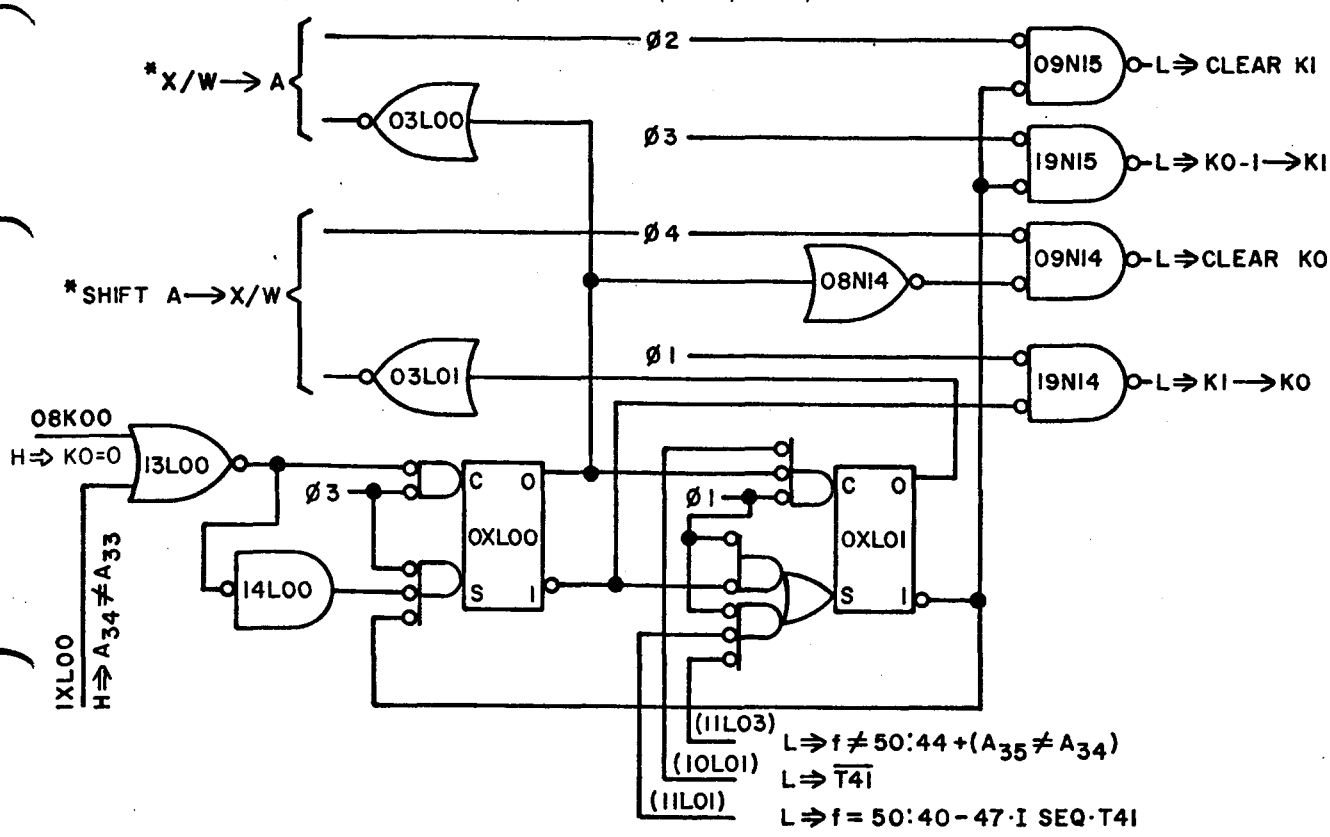
2-309. Instruction Sequence. The shift instructions with the exception of SF, require both an I and an extended sequence for completion. During the I-sequence, the instruction word is called out of memory, the function code (m) is placed into the F-register, and the six least significant bits (K) are placed into the K Counter as the shift count. In order to allow time for the shifting operations, the I-sequence must be extended. This is accomplished by setting the Hold I and Hold II flip-flops. These flip-flops keep the machine in the I-sequence and prevent the reading of the next instruction until the shift operation is completed. Refer to plate P-28 for the HOLD 1 and HOLD 2 flip-flops. These flip-flops both have outputs to the logic on plate P-14. This logic supplies timing and enables for the I-sequence operations necessary to obtain the next instruction. When the HOLD

flip-flops are set, their high level outputs disable these I-sequence operations. Among the events which are prevented are initiate-memory, advance-P subsequence, clear-F, Z-select → F, clear-KO and Z-Select → KO. As long as the HOLD flip-flops are set, the shift function code and the remaining shift count are retained in F and KO, respectively. As shown in figure 2-60, when shift sequence flip-flops (OXLOO and OXLO1) are set, shifting commands are generated via the 03L00 and 03L01 gates and enabling commands are sent to the K1 adder. The only other inputs necessary for K1 adder operation are phase outputs generated by the Master Clock. For each cycle of the Master Clock (4 phases) a shift of one place is executed. For complete logic diagrams of the shift sequence flip-flops, and the K1 adder, refer to plates P-36 and P-37.

2-310. The execution of the SF instruction requires an I, an extended, and a W-sequence. The I-sequence is used to call out of memory the instruction word, load K into the K1 adder, and normalize (shift) the contents of the A-register. The W-sequence is active throughout the normalize process but is prevented from being effective by the setting of the Hold I and Hold II flip-flops. When these flip-flops are cleared at the end of the normalize process, the W-sequence is allowed to store the difference between the programmed shift count and the actual shift count, which is contained in the KO-register.

2-311. The actual shifting is accomplished by using the AU-register in conjunction with the X-register, and the AL-register in conjunction with the W-register. For an example using the AL- and W-register, refer to figure 2-61 (Shift Operations Involving the AL-register). Notice in step 2 that when A is shifted to the right one place to W, the sign bit (AL_{17}) is transferred to both W_{17} and W_{16}. On the subsequent W to AL transfer in step 4, a bit for bit transfer is made. This ensures that in a right shift operation, the original sign bit is retained. Notice in step 2 that when AL is shifted to the left one place to W (with the exception of AL_{17}, which is sent to W_{00},) the individual bit positions of AL are sent to the adjacent bit positions of W. On the subsequent W to AL transfer in step 4, a bit for bit transfer of W to AL occurs. This results in a circular shift in a left-shift operation.

2-312. Initial Shift Count Equal to Zero or $A_{35i} \neq A_{34i}$. If the shift count specified in the six least significant bits of the instruction word equals zero, or if f = 50:44 and $A_{35} \neq A_{34}$, no shifts are performed. The Hold flip-flops are set but are cleared before any shifting is accomplished. Table 2-80 conta a sequential list of essential I and Shift sequence commands when the initial shift count is zero or when $A_{35i} \neq A_{34i}$ and F = 50:44. As shown, no shifting is performed. Two main timing cycles of the I-sequence are used before the HOLD flip-flops are cleared. With the initial shift count equal to zero or with $A_{35i} \neq A_{34i}$, the instruction time is four microseconds. Most of the commands listed in table 2-80 have been described previously in detail. Those commands that have not been previously describ are described in table 2-81 in detail.



SEQUENCE OF EVENTS (KOi ≠0/A35i=A34i)

T4.1 SET OXL01 FF (INITIATE SHIFT SEQUENCE)
T4.2 CLEAR K1
T4.3 SET OXL00 FF, KO-1→K1* CLEAR X & W
T4.4 CLEAR KO, * SHIFT A→X/W
TX.1 KI KO, * CLEAR A
TX.2 CLEAR K1, * X/W→A
→TX.3 KO-1 K1 *CLEAR X & W
TX.4 CLEAR KO, * SHIFT A→X/W
TX.1 K1→KO, * CLEAR A
TX.2 CLEAR K1, * X/W→A
(CONTINUE UNTIL K1=0 or A34≠A35 FOR a f = 50:44
TX.3 KO-1→K1 (K1=0 or K1=MAXIMUM SHIFT COUNT-NUMBER OF SHIFTS)
*CLEAR X/W
TX.4 CLEAR KO * SHIFT A→X/W
TX.1 K1→KO (KO=0 or KO= MAXIMUM SHIFT COUNT-NUMBER OF SHIFTS)
* CLEAR A
TX.2 CLEAR K1 (NOT USED) * X/W→A (Af)
TX.3 CLEAR OXL00 FF, KO
* CLEAR X & W (NOT USED)
TX.4 *SHIFT A→X/W (NOT USED)
TX.1 CLEAR OXL01 FF

NOTE: * THESE EVENTS PERTAIN TO ACTUAL SHIFTING OF AU AND AL AND ARE DISCUSSED IN PARAGRAPH 2-311.

Figure 2-60. Shift Sequence Simplified Logic

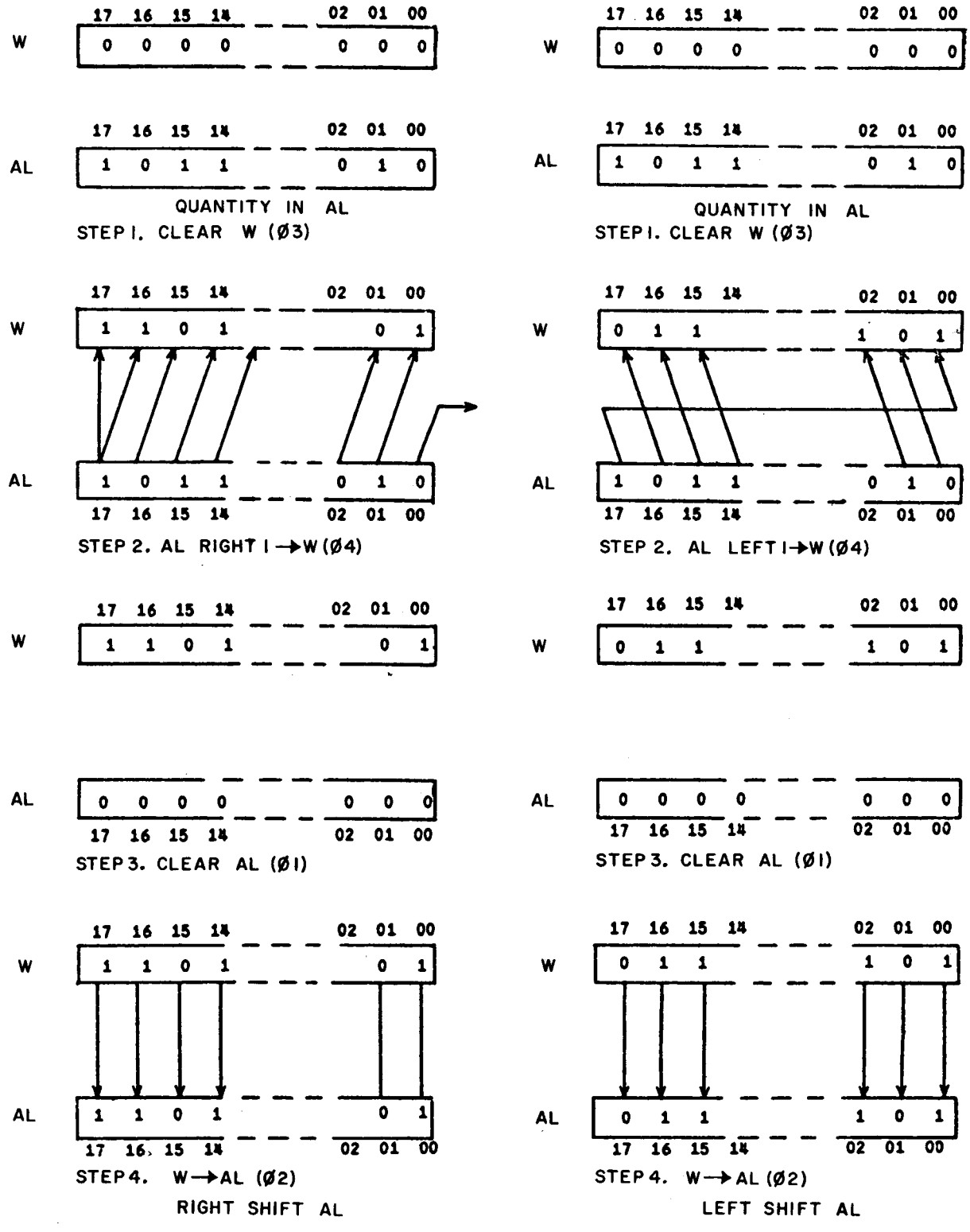


Figure 2-61. Shift Operations Involving the AL-Register

TABLE 2-80. I AND SHIFT SEQUENCE ESSENTIAL COMMANDS WITH INITIAL SHIFT COUNT = $0/A_{35i} \neq A_{34i}$

Time Notation	Command	Initial Shift Count 0	$A_{35} \neq A_{34}$ Initially
	I-SEQUENCE		
T1.4	Clear KO	X	X
T2.3	Clear 0XL00 ff	X	X
T2.4	Z Sel_{5-0} → KO (KO = 0 for 50:41, 42, 43, 45, 46, 47)	X	X
T3.4	Set HOLD 1 ff, Clear Scale Factor ff	X	X
T4.1	Set 0XL01 ff	X	
T4.3	Set 0XL02 ff	X	
	Set HOLD 2 ff	X	X
T1.1	Set Clear HOLD ff Clear 0XL01	X	X
T1.3	Clear 0XL02	X	X
T4.2	Clear HOLD 1 ff	X	X
T4.3	Clear the Clear HOLD ff	X	X
	I-SEQUENCE FOR NEXT INSTRUCTION		
T1.3	Clear HOLD 2 ff	X	X

TABLE 2-81. COMMANDS/DESCRIPTIONS FOR INITIAL SHIFT COUNT EQUAL TO ZERO (REFER TO TABLE 2-19 FOR COMMANDS NOT DESCRIBED) (NOT DESCRIBED IN A PREVIOUS SEQUENCE)

Time Notation	Command/Description
T2.3	Clear 0XL00 ff. The high input into 13L00 H⇒K0 = 0 (refer to plate P-36) produces a low and, at ∅3, 0XL00 is cleared.
T3.4	Set Hold 1 ff. The Hold 1 flip-flop (refer to plate P-28) is set by a L⇒ T34 I-Seq Format 2, a L⇒f = 50:26, 27, 40-47 and a ∅4. Clear Scale Factor ff (1XL00). The Scale Factor ff is cleared via 11L01 and 10L01 (refer to plate P-36). A L→I-Seq, L→50:40 through 50:47, and a low from 01T41 enables 10L01. The high output from 10L01 is inverted by 11L01 and, when ANDed with a ∅4, clears the Scale Factor ff.

TABLE 2-81. COMMANDS/DESCRIPTIONS FOR INITIAL SHIFT COUNT EQUAL TO ZERO (Cont)

Time Notation	Command/Description
T4.1	Set 0XL01 ff. Gate 11L03 (refer to plate P-36) is disabled by the absence of a L ⇒ f= 50:44 and outputs a low to 0XL01. With the low from gate 11L01, 0XL01 is set. If $A_{35} \neq A_{34}$ initially, 10L03 (refer to plate P-36) outputs a low. This low, along with a L ⇒ f = 50:44, enables 11L03 and prevents the setting of 0XL01 for a f = 50:44 and $A_{35} \neq A_{34}$.
T4.3	Set 0XL02 ff. The low from 0XL01 and from 13L00, at ∅3, sets 0XL02. Set Hold 2 ff. Hold 2 flip-flop 2XG38 (refer to plate P-28) is set by the low from Hold 1 flip-flop IXG38 and a ∅3.
T1.1	Set Clear Hold ff (for $KO_i = 0$). The high from 00L02 (refer to plate P-36) is inverted by 05G38 and, at ∅1, Clear Hold flip-flop 0XG38 (refer to plate P-28) is set. Set Clear Hold ff (for $A_{35i} \neq A_{34i}$). The high from 11L03 (refer to plate P-36) is inverted by 05G38 and, at ∅1, the Clear Hold flip-flop 0XG38 (refer to plate P-28) is set. Clear 0XL01 ff. Gate 10L01 (refer to plate P-36) is disabled by the absence of a low from 0IT41. The low output, along with the low from 0XL00, is applied to 0XL01 which, at ∅1, is cleared.
T1.3	Clear 0XL02 ff. Flip-flop 0XL02 is cleared, at ∅3, by the low from 0XL01 flip-flop.
T4.2	Clear Hold 1 ff. Hold 1 flip-flop 1XG38 (refer to plate P-28) is cleared by the low from Clear Hold flip-flop 0XG38, the low from 05T42, and a ∅2.
T4.3	Clear Clear Hold ff. The low output from Hold 1 flip-flop is applied to clear Clear Hold flip-flop at ∅3.
T1.3	Clear Hold 2 ff. The low from Hold 1 flip-flop is applied to Hold 2 flip-flop, and with the low from 13T14, is cleared at ∅3.

2-313. Initial Shift Count Not Equal to Zero or $A_{35i} = A_{34i}$ for f = 50:44. If the shift count specified in the six least significant bits of the instruction is not equal to zero, or if $A_{35} = A_{34}$ initially, shifts are performed. Table 2-82 contains a sequential list of essential I and Shift sequence commands when the initial shift count is not equal to zero or when $A_{35i} = A_{34}$. Most of the commands listed in the table have been described previously in detail. Those commands that have not been previously described are described in table 2-83 in detail.

2-314. W-Sequence Data Flow For SF Instruction (f = 50:44). Figures 2-62 and 2-63 provide block diagram descriptions of final scale W and last W-sequence data flow, and scale sequence once place shift. Tables 2-84 and 2-85 list the essential commands for termination of scale sequence by K1=0 and $AU_{34} \neq AU_{33}$.

2-315. Multiply Instructions. The multiply instructions consist of MULAL (f = 24) and MULALB (f = 25) which effectively multiply the contents of the AL-register by an operand obtained from memory. These instructions are SR sensitive.

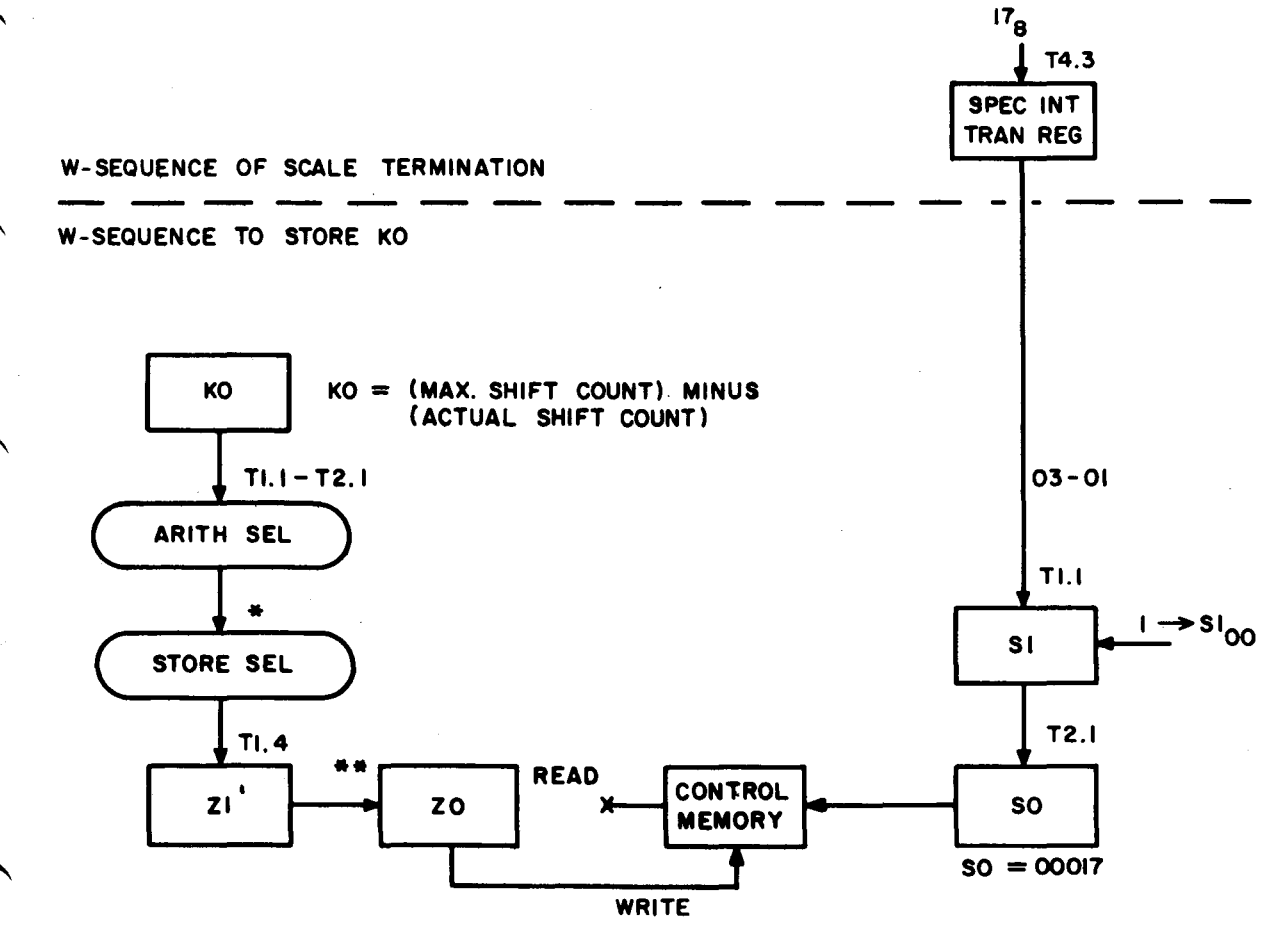


Figure 2-62. Final Scale W And Last W-Sequence Data Flow

2-316. MULAL (f = 24). The MULAL instruction multiplies the contents of the AL-register (AL), by the contents of a designated memory location (Y) and places the results into the A-register. The most significant bits are placed into the AU-register and the least significant bits are placed into the AL-register. Before the operation starts, both the multiplier and the multiplicand must be changed to positive numbers, but the computer remembers the signs of the original numbers for later conversion of the product. The multiplier is held in the AL-register while the multiplicand is held in the D-register. Since multiplication is only a series of additions, the computer examines the least significant bit (LSB) of the multiplier $(AL_{00})_i$; if this bit is a one, it adds the multiplicand to the partial product contained in the AU-register and shifts the A-register right one place. If the LSB of the multiplier is a zero, only the shift is performed. After 23_8 shifts have been completed, the multiplication process is terminated with the final result contained in the A-register. It should be noted that overflow can never occur during a multiply instruction, as the shift is always to the right.

2-317. MULALB (f = 25). The MULALB instruction provides the same service as the MULAL instruction, except that the address of the operand (multiplicand) in memory is now Y + B.

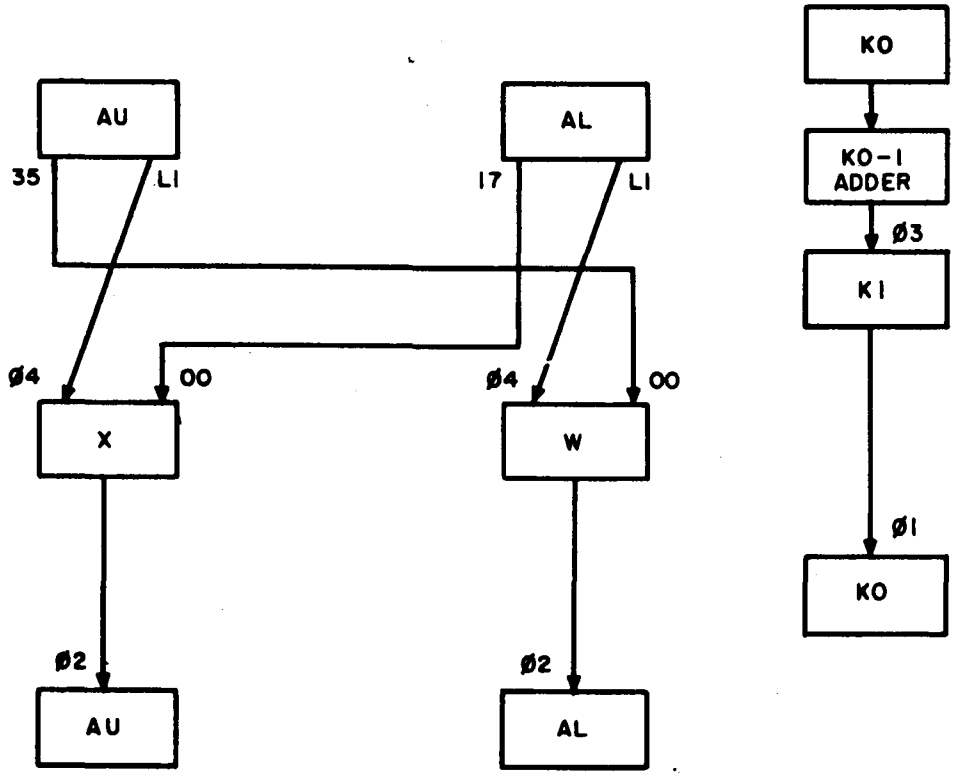


Figure 2-63. Scale Sequence, One Place Shift

TABLE 2-82. I AND SHIFT SEQUENCE ESSENTIAL COMMANDS WITH INITIAL SHIFT COUNT $\neq 0$ OR $A_{35} = A_{34}$ FOR A f = 50:44

Time	Command	f = 50:	41	42	43	44	45	46	47
	I-SEQUENCE								
T1.4	Clear KO		X	X	X	X	X	X	X
T2.3	Clear 0XL00		X	X	X	X	X	X	X
T2.4	Z Sel_{5-0} → KO		X	X	X	X	X	X	X
T3.4	Set HOLD 1 ff		X	X	X	X	X	X	X
	Clear Scale Factor ff		X	X	X	X	X	X	X
T4.1	Set 0XL01 ff		X	X	X	X	X	X	X
T4.2	Clear K1		X	X	X	X	X	X	X
4.3	Set 0XL00 ff, Set HOLD 2 ff		X	X	X	X	X	X	X
	Clear X, Clear W, KO-1 → K1		X	X	X	X	X	X	X
T4.4	Clear KO		X	X	X	X	X	X	X
	AUR1 → X		X	X	X				
	AU_{35} → X_{17} *(Set X_{17} = 1 if AU_{35} = 1)		X	X	X				
	ALR1 → W		X	X	X				
	AU_{18} → W_{17} *(Set W_{17} = 1 if AU_{18} = 1)		X		X				
	AL_{17} → W_{17} *(Set W_{17} = 1 if AL_{17} = 1)			X					
	AUL1 → X					X	X	X	X
	AU_{35} → X_{00} *(Set X_{00} = 1 if AU_{35} = 1)					X	X		
	AL_{17} → X_{00} *(Set X_{00} = 1 if AL_{17} = 1)					X		X	X
	ALL1 → W					X	X	X	X
	AU_{35} → W_{00} *(Set W_{00} = 1 if AU_{35} = 1)					X	X		X
	AL_{17} → W_{00} *(Set W_{00} = 1 if AL_{17} = 1)							X	
T1.1	K1 → KO		X	X	X	X	X	X	X
	Clear AU		X		X	X	X		
	Clear AL			X	X	X		X	X
T1.2	Clear K1		X	X	X	X	X	X	X
	X → AU		X		X	X	X		X
	W → AL			X	X	X		X	X

TABLE 2-82. I AND SHIFT SEQUENCE ESSENTIAL COMMANDS WITH INITIAL SHIFT COUNT $\neq 0$ OR $A_{35} = A_{34}$ FOR A f = 50:44 (Cont)

Time	Command	f = 50:	41	42	43	44	45	46	47
TX. 3	KO - 1 → K1, Clear X, Clear W		X	X	X	X	X	X	X
TX. 4	Clear KO		X	X	X	X	X	X	X
	AUR1 → X		X	X	X				
	$AU_{35} \rightarrow X_{17}$ *(Set $X_{17} = 1$ if $AU_{35} = 1$)		X	X	X				
	ALR1 → X		X	X	X				
	$AU_{18} \rightarrow W_{17}$ *(Set $W_{17} = 1$ if $AU_{18} = 1$)		X		X				
	$AL_{17} \rightarrow W_{17}$ *(Set $W_{17} = 1$ if $AL_{17} = 1$)			X					
	AUL1 → X					X	X	X	X
	$AU_{35} \rightarrow X_{00}$ *(Set $X_{00} = 1$ if $AU_{35} = 1$)					X	X		
	$AL_{17} \rightarrow X_{00}$ *(Set $X_{00} = 1$ if $AL_{17} = 1$)					X		X	X
	ALL1 → W					X	X	X	X
	$AU_{35} \rightarrow W_{00}$ *(Set $W_{00} = 1$ if $AU_{35} = 1$)					X	X		X
	$AL_{17} \rightarrow W_{00}$ *(Set $W_{00} = 1$ if $AL_{17} = 1$)							X	
TX. 1	K1 → KO		X	X	X	X	X	X	X
	Clear AU		X		X	X	X		X
	Clear AL			X	X	X		X	X
TX. 2	Clear K1		X	X	X	X	X	X	X
	X → AU		X		X	X	X		X
	W → AL			X	X	X		X	X
	(Continue until K1 = 0)		X	X	X	X	X	X	X
TX. 3	KO - 1 → K1 (K1 = 0), Clear X, Clear W		X	X	X	X	X	X	X
TX. 4	Clear KO		X	X	X	X	X	X	X
	Shift AUR1 → X, $AU_{35} \rightarrow X17$								
	ALR1 → W, $AU_{18} \rightarrow W_{17}$, $AL_{17} \rightarrow W_{17}$								
	AUL1 → X, $AU_{35} \rightarrow X_{00}$, $AL_{17} \rightarrow X_{00}$								
	ALL1 → W, $AU_{35} \rightarrow W_{00}$, $AL_{17} \rightarrow W_{00}$								
	(Same as above)		X	X	X	X	X	X	X

TABLE 2-82. I AND SHIFT SEQUENCE ESSENTIAL COMMANDS WITH INITIAL SHIFT COUNT $\neq 0$ OR $A_{35} = A_{34}$ FOR A f = 50:44 (Cont)

Time	Command f = 50:	41	42	43	44	45	46	47
TX.1	K1 → KO (KO = 0) Set Clear HOLD ff	X	X	X	X	X	X	X
	Clear AU	X		X	X	X		X
	Clear AL		X	X	X		X	X
TX.2	Clear K1	X	X	X	X	X	X	X
	X → AU	X		X	X	X		X
	W → AL		X	X	X		X	X
TX.3	KO-1 → K1	X	X	X	X	X	X	X
	Clear 0XL00 ff	X	X	X	X	X	X	X
	Set 0XL02 ff	X	X	X	X	X	X	X
	Clear X, Clear W	X	X	X	X	X	X	X
TX.4	Shift A → X/W (not used)							
TX.1	Clear 0XL01 ff	X	X	X	X	X	X	X
TX.3	Clear 0XL02 ff	X	X	X	X	X	X	X
next T4.1	Clear Special Interrupt Translator Register				X			
T4.2	Clear HOLD 1 ff	X	X	X	X	X	X	X
T4.3	Clear the Clear HOLD ff	X	X	X	X	X	X	X
	Set Spec Int Trans Reg = 17_8				X			
T4.4	Clear S1				X			
	I-Seq for next instruction	X	X	X		X	X	X
	W-Seq				X			
T1.1	Spec Int Trans Reg_{03-01} → $S1_{03-01}$				X			
	1 → $S1_{00}$, KO → Arith Sel				X			
T1.3	Clear HOLD 2 ff	X	X	X	X	X	X	X
	Clear Z1				X			
T1.4	Disable CM → Z0				X			
	Store Sel → Z1				X			
T2.1	S1 → S0, Drop KO → Arith Sel				X			
T2.4	Drop Disable CM → Z0				X			

TABLE 2-83. COMMANDS/DESCRIPTIONS FOR INITIAL SHIFT COUNT NOT EQUAL TO ZERO OR $A_{35} = A_{34}$ FOR A 50:44 (NOT DESCRIBED IN A PREVIOUS SEQUENCE) (REFER TO TABLE 2-19 FOR COMMANDS NOT DESCRIBED)

Time Notation	Command/Description
T2.3	*Clear 0XL00
T3.4	*Set Hold 1 ff *Clear Scale factor ff
T4.1	*Set 0XL01 ff
T4.2	Clear K1. The low from 0XL01 flip-flop (refer to plate P-36) is applied to 09N15 (refer to plate P-37) and, at ø2, 1XK00 through 1XK05 are cleared. K1 will clear every ø2, as long as 0XL01 ff remains set.
T4.3	Clear X, Clear W. For a 50:41, 50:42, or 50:43 instruction, gate 30N03 (refer to plate P-19) is enabled by a L ⇒ f = 40 through 43 and a L ⇒ Mult/Div Seq and the high is applied to 08N03. The low at ø3 enables 09N03 and 08N06 to produce a L ⇒ clear X and a H ⇒ clear W. For a 50:45, 50:46, or 50:47, gate 30N03 applies a low to 40N03 which provides a high to 08N03 to clear X and W. The clear X, clear W command will occur on every ø3 as long as the 0XL01 flip-flop remains set. KO-1 → K1. The L ⇒ KO-1 → K is applied to 19N15 (refer to plate P-37) and, at ø3, 19N15 provides a low to 1XK00 through 1XK05. This command is applied to 1XK00 through 1XK05 on every ø3 as long as the 0XL01 ff remains set. Set 0XL00 ff. With the Scale Factor ff 1XL00 cleared (refer to plate P-36) the low from the clear side is inverted by 13L00 and 14L00 and applied to 0XL00. The other inputs are a low from 0XL01, which was set at T4.1, and a ø3 which will set the ff.
T4.4	Clear KO. The high from the clear side of 0XL00 (refer to plate P-36) is applied to 08N14 (refer to plate P-37) and KO is cleared as previously described. This command will occur at every ø4 as long as the 0XL00 ff is set.

TABLE 2-83. COMMANDS/DESCRIPTIONS FOR INITIAL SHIFT COUNT NOT EQUAL TO ZERO OR $A_{35} = A_{34}$ FOR A 50:44 (Cont)

Time Notation	Command/Description
	Shift A → X/W. The high from the clear side of 0XL01 is inverted by 03L01 (refer to plate P-36) and applied to 30N03 as a L ⇒ Mult/Div Seq. With the other input (a L ⇒ 40-43), 30N03 outputs a high which is inverted by 38N03 and, at ∅4, 39N03 produces a L ⇒ AURI → X and gate 39N06 produces a L ⇒ ALRI → W. If the instruction is a 50:45, 50:46, 50:47 the absence of a L ⇒ 40-43 at gate 30N03 produces a low enable to 40N03. The high output of 40N03 is inverted by 48N03 and, at ∅4, 49N03 produces a L ⇒ AULI → X and 49N06 a L ⇒ ALLI → W. The Shift A → X/W command will occur on every ∅4 as long as 0XL01 remains set.
0XL01 ff Set·∅4	Set $X_{17} = 1$ if $AU_{35} = 1$ ($AU_{35} \rightarrow X_{17}$). The low from 0XA35 (refer to plate P-98) is inverted by 02A35 and applied to gates 31W00, 31W17, 31X00, and 31X17 (refer to plate P-33). Gate 31X17 is enabled because of the high from 02A35 and a high resulting from the absence of a L ⇒ f = 24, or 25, and produces a low into 01X17 (refer to plate P-88). The other input into 01X17 is a L ⇒ AURI → X. Set $W_{17} = 1$ if $AU_{18} = 1$ ($AU_{18} \rightarrow W_{17}$). The high from the clear side of 0XA18 (refer to plate P-97) is applied to 31W17 which, along with the high due to the absence of a L ⇒ f = 42, outputs a low to 01W17 (refer to plate P-90). The other output to 0XW17 is a L → ALRI, which was developed in an earlier description. Set $W_{17} = 1$ if $AL_{17} = 1$ ($AL_{17} \rightarrow W_{17}$). Gate 31W17 is now enabled by a high from 02A17 (refer to plate P-100), because of 0XA17 being set, and a H ⇒ f = 42, to set $W_{17} = 1$ if $AL_{17} = 1$. Set $X_{00} = 1$ if $AU_{35} = 1$ ($AU_{35} \rightarrow X_{00}$). Gate 31X00 (refer to plate P-33) is enabled by a H ⇒ f = 45 and the high from 02A35. The low output from 31X00 is applied to 01X00 and is set to 1. Set $X_{00} = 1$ if $AL_{17} = 1$ ($AL_{17} \rightarrow X_{00}$). Gate 31X00 (refer to plate P-33) is enabled by a high from 02A17 (refer to plate P-100) because 0XA17 is set and also because a high resulting from an absence of L ⇒ f = 45 and a high from the set side of a Neg flip-flop 0XG30, which was cleared at time T4.2 of a previous I-sequence.

TABLE 2-83. COMMANDS/DESCRIPTIONS FOR INITIAL SHIFT COUNT NOT EQUAL TO ZERO OR $A_{35} = A_{34}$ FOR A 50:44 (Cont)

Time Notation	Command/Description
	Set $W_{00} = 1$ if $AU_{35} = 1$ ($AU_{35} \rightarrow W_{00}$). Gate 31W00 (refer to plate P-33) is enabled by the high from 02A35 and the absence of a L ⇒ f = 46 and produces a low which is applied 01W00 (refer to plate P-89). With a L ⇒ ALRI → W, 01W00 is set to 1.
	Set $W_{00} = 1$ if $AL_{17} = 1$ ($AL_{17} \rightarrow W_{00}$). Gate 31W00 (refer to plate P-33) is now enabled by the high from 02A17 (refer to plate P-100) and a H ⇒ f = 46.
∅1	KI → KO. The low from the set side of 0XL00 (refer to plate P-36) is applied to 19N14 (refer to plate P-37) and, at ∅1, 19N14 provides a low to 0XK00 through 0XK05. The output of the set side of the KO-register (IXK00 through IXK05) is also applied to 0XK00 through 0XK05 to provide KI → KO. This command is applied to the flip-flops at every ∅1 as long as 0XL00 remains set.
	Clear A. For a 50:41, 50:43, 50:45, or 50:47 instruction the low from gate 03L00 (refer to plate P-36) is applied to gate 20N04 (refer to plate P-20). With the absence of a H ⇒ f = 42, and a H ⇒ f = 46, 20N04 produces a high which is inverted by 07N04. At ∅1, 08N04 produces a H ⇒ clear AU, via 09N04. For a 50:42, 50:43, 50:46, or 50:47 instruction, the low from 03L00 is applied to gate 20N05 with a L ⇒ f ≠ 40, 41 and the absence of an f = 45. 20N05 produces a high, inverted by a 07N05, and applied to 08N05. Gate 00N05 is disabled by the absence of a proper sequence or instruction and, at ∅1, 08N05 produces a H ⇒ clear AL, via 09N05. This command will occur at every ∅1 as long as 0XL00 remains set.
∅2	X/W → A. For a 50:41, 50:43, 50:45, or 50:47 instruction, the high from 20N04 is inverted by 28N04 and applied to 29N04. Gate 10N04 is disabled by the lack of a proper command. At ∅2, 29N04 produces a L ⇒ X → AU. For a 50:42, 50:43, 50:46, or 50:47 instruction, the high from 20N05 is inverted by 28N05 and applied to 29N05 and at ∅2, produces a L ⇒ W → AL. The X/W → A command will occur at every ∅2 as long as 0XL00 remains set.
TX.1	Set Clear-Hold ff. With K1 - 0, the low from the clear side of flip-flops IXK00 through IXK05 (refer to plate P-37) is applied to gate 04G38 (refer to plate P-28). The other outputs are the low from 0IL00 and the H ⇒

TABLE 2-83. COMMANDS/DESCRIPTIONS FOR INITIAL SHIFT COUNT NOT EQUAL TO ZERO OR $A_{35} = A_{34}$ FOR A 50:44 (Cont)

Time Notation	Command/Description
	f = 50:40 through 50:47 via 89F40. The high out of 04G38 is inverted by 05G38 and, at ø1, sets Clear-Hold flip-flop 0XG38.
TX.3	Set 0XL02 ff. With KO = 0, 13L00 (refer to plate P-36) outputs a low to 0XL02 and with the low from 0XL01, 0XL02 will set at ø3. (Note that gate 13L00 will also output a low when the Scale Factor ff 1XL00 is set).
TX.1	*Clear 0XL01 ff.
TX.3	*Clear 0XL02 ff.
T4.1	Clear Special Interrupt Translator Register. Gate 00E00 (refer to plate P-49) is enabled by a low from 05T42 and by not being in I/O. The high output is inverted by 01E00 and, at ø2, 02E00 outputs a high via 04E00, to clear flip-flops 2XG15 through 2XG18 (refer to plate P-55).
T4.3	Set Special Interrupt Translator Register = 17_8. Gate 22E15 (refer to plate P-55) is enabled by a L ⇒ f = 50:44, a low from IXG23 (refer to plate P-13) (which is not set at this time) and a low from 03T43. The high is inverted by 23E15 and, at ø3, 20E15 sets 2XG15 through 2XG18.
T1.1	Spec Int Trans Reg 03-01 → S1. Gate 70N12 (refer to plate P-22) is qualified by a L ⇒ W Seq, L ⇒ f = 50:44 and a low from 03T11. The high is inverted by 36N12 and applied to 37N12 which at ø1 produces a L ⇒ Translator → S1 + Special Interrupt Reg → S1 to 1XS01 through 1XS03 (plate P-104). The high from 70N12 is also inverted by 78N12 and applied via 79N12 as a high (H ⇒ $\overline{\text{Translator} \rightarrow \text{S1}}$) to gates 25G16 through 25G18 (plate P-55) which are now qualified due to the high from ff's 2XG16 through 2XG18. The low outputs from 25G16 through 25G18 are applied to 1XS01 through 1XS03 (plate P-104) which will now set.
T1.1	$\overline{1 \rightarrow S1_{00}}$. Gate 50N12 (refer to plate P-22) is qualified by a L ⇒ f = 50:44, L ⇒ W Seq and a low from 03T11. The high output qualifies 58N12 which produces a L ⇒ Set Bit 2° to set 1XS00 at ø1 (plate P-104).
T1.1	KO → Arith Sel. Gate 90N01 (refer to plate P-17) is qualified by a L ⇒ 50:44 and a L ⇒ T14 R1, W Seq. The high output is inverted by 91N01 to produce a L ⇒ Scale Factor Count → X and by 58N01 to produce a L ⇒ SR + ICR + K → Select via 59N01. The output of 91N01 is applied

TABLE 2-83. COMMANDS/DESCRIPTIONS FOR INITIAL SHIFT COUNT NOTE EQUAL TO ZERO OR $A_{35} = A_{34}$ FOR A 50:44 (Cont)

Time Notation	Command/Description
	to 12X00 through 12X05 (plate P-39) which also has an input from KO. The outputs of these gates are applied to Arithmetic Select gates 13X01 through 13X05 (plate P-82).
T2.1	S1 → S0. All inputs into 90S09 and 90S06 (refer to plate P-107) are low due to the S1 Register ff's being clear. The high outputs qualify 91S00 which produces a L ⇒ S1 = CONTR OR BOOTSTRAP MEMORY ADDRESS into 30N10 (plate P-24) which is qualified by a L ⇒ I, R1, W or I/O and a low from 03T21. The high output is inverted by 38N10 and at ∅1, 39N10 produces a L ⇒ S1 → S0.
T2.1	Drop KO →Arith Sel. Gate 90N01 (plate P-17) which was qualified at T1.1 is now disabled due to the loss of the low from T14.
	* Described previously in TABLE 2-81.

TABLE 2-84. TERMINATION OF SCALE SEQUENCE BY $AU_{34} \neq AU_{33}$ (W-SEQUENCE)

Time Notation	Commands
	W-SEQUENCE
TX. 3	K0-1 → K1
TX. 4	Clear K0, AUL1 → X, ALL1 → W, **$AU_{35} \rightarrow W_{00}$, **$AL_{17} \rightarrow X_{00}$
TX. 1	K1 → K0 , clear AU, clear AL
TX. 2	Clear K1, X → AU ($AU_{34} \neq AU_{33}$), ***W → AL
TX. 3	K0-1 → K1
TX. 4	Set Scale Factor ff (1XL00)
TX. 1	K1 → K0
TX. 2	Clear K1, X → AU ($AU_{35} \neq AU_{34}$), ***W → AL
TX. 3	K0-1 → K1, clear 0XL00 ff, set 0XL02 ff
TX. 4	AUL1 → X, ALL1 → W, $AU_{35} \rightarrow W_{00}$, **$AL_{17} \rightarrow X_{00}$ (not used)
TX. 1	Clear 0XL01 ff, set Clear Hold ff
TX. 3	Clear 0XL02 ff
xt T4. 1	Clear special interrupt translator Reg
T4. 2	Clear Hold 1 ff
T4. 3	Set Spec Int Trans Reg = 17_8
T4. 4	Clear S1
T1. 1	Spec Int Trans $Reg_{03-01} \rightarrow S1_{03-01}$, $1 \rightarrow S1_{00}$, K0 → Arith Sel
T1. 3	Clear Hold 2 ff, clear Z1
T1. 4	Disable CM → Z0, Store Sel → Z1**
T2. 1	S1 → S0, drop K0 → Arith Sel
T2. 4	Drop disable CM → Z0

**These commands are enabled by gates 31W00 and 31X00 in the logic diagrams, plate P-33, and are timed by the AUL1 → X, and ALL1 → W commands.

***The transmission of bit 00 for the W → AL command is through gate 83A00 in the logic diagrams, plate P-33.

TABLE 2-85. COMMANDS/DESCRIPTIONS FOR TERMINATION OF SCALE SEQUENCE BY $AU_{34} \neq AU_{33}$ (NOT DESCRIBED IN A PREVIOUS SEQUENCE) (REFER TO OTHER SHIFT SEQUENCES FOR COMMANDS DESCRIBED PREVIOUSLY)

Time Notation	Commands/Description
TX. 4	Set Scale Factor ff (1XL00). With $AU_{34} \neq AU_{33}$ gate 20L00 (refer to plate P-36) is qualified by either pins 11 and 12 or 13 and 14 being low due to either ff 0XA34 or 0XA33 (plate P-98) being set. The high out of 20L00 is applied to 21L00. Gate 10L01 (Plate P-36) outputs a low due to the absence of a W Seq input and the low output is inverted by 11L01 and applied to 21L00 which now outputs a low to 1XL00 which will not set due to the L ⇒ f = 50:44 input and a ϕ4.

2-318. Figure 2-64 is a simplified signal flow diagram of a Multiply instructions. If a small example is applied to this diagram it can easily be seen that the value of KOi must be one greater than the number of bits so that the complete answer will appear in A.

2-319. R1 and Multiply Sequence Data Flow. Refer to figure 2-65 for a block diagram description of the data flow during the R1 and Multiply sequences. In the I-sequence, which is not shown in the figure, the instruction word is called from memory and the address of the multiplicand is developed. As is shown in the figure, a memory reference is initiated to obtain the multiplicand from memory. The multiplicand must be a positive value before it is transferred to the D-register. Therefore, if the multiplicand is a negative value, its complement is transferred to the D-register. If the multiplicand is a positive value, it is transferred to the D-register unchanged. The multiplier contained in the AL-register must also be made positive if it was initially a negative value. This is part of the sign correction operation which makes both the multiplier and multiplicand positive before the actual multiplication process begins. The Y Neg and A Neg flip-flops (refer to plate P-33) are used to remember the original signs of the multiplier and multiplicand for possible conversion of the product at the end of the multiplication operation. The Hold 1 and Hold 2 flip-flops (refer to plate P-28) are set during the instruction execution to prevent an exit from the R1 sequence until the multiplication process h been completed.

2-320. The actual multiplication of D and AL occurs during the multiply sequence which runs in parallel with the R1-sequenc The multiplication commands are enabled by the setting of the 0XL00 and 0XL01 flip-flops (refer to plate P-36) and are timed by the clock pulses from the Master Clock. Multiply operation commences when the 36-bit value contained in the AU and AL-registers is right-shifted one place into the X- and W-registers respectively. The Multiply Store flip-flop (refer to plate P-27) is used to store the LSB of the multiplier contained in the AL-register (AL_{00}). This is the multiplier bit that is to be examined. If AL_{00} is a one, the Multiplier Store flip-flop sets. This causes the multiplicand, contained in the D-register, to be added to the partial product in the X-register, the

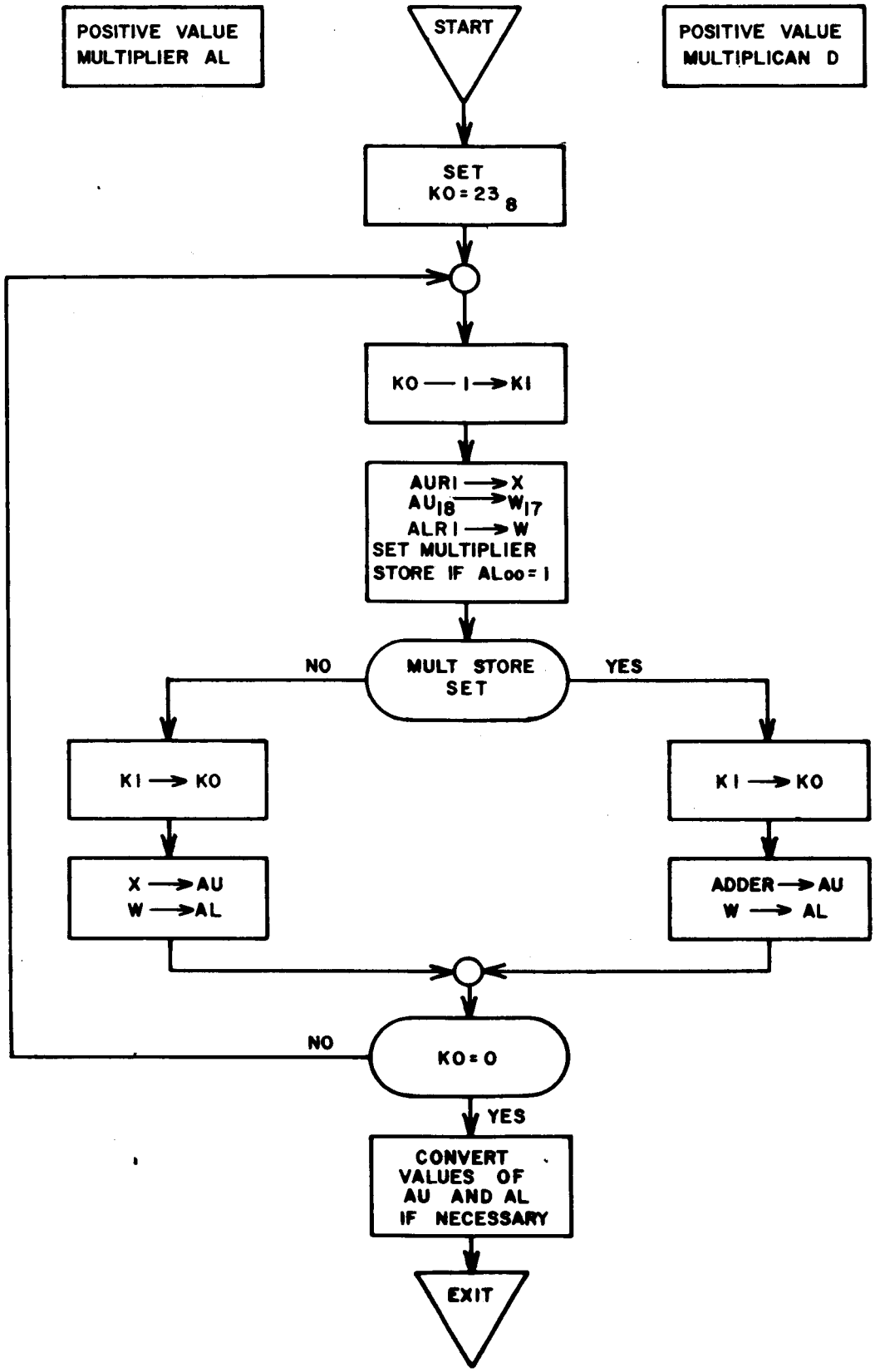


Figure 2-64. Simplified Multiply Sequence.

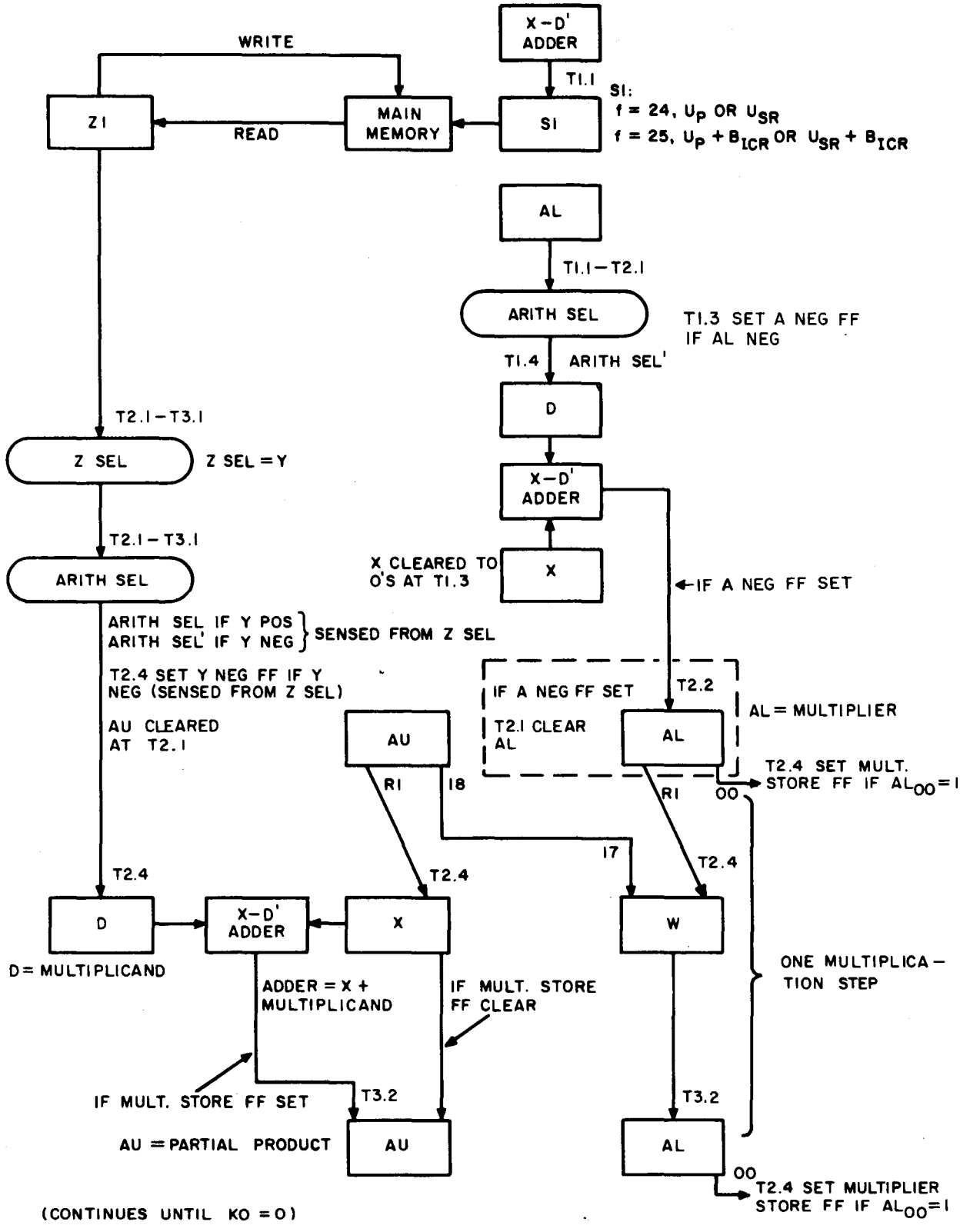


Figure 2-65. R1 and Multiply Sequence Data Flow

result being placed into the AU-register. If AL_{00} is a zero, the Multiplier Store flip-flop remains clear. Nothing is added to the partial product in the X-register and the unchanged partial product is transferred to the AU-register. During each succeeding multiplication step AU and AL are right shifted one place. Depending upon AL_{00}, either the multiplicand or nothing is added to the partial product in the X-register. Initially, a count of 23_8 is placed into the K counter. Each time a multiplication step occurs, the adder is decremented by one. When the adder contains a zero, the multiplication process is completed and the A-register contains a positive product. If the product should be negative, as determined by the stored signs of the initial quantities, the contents of the A-register are complemented. Figure 2-66 shows the data flow of final sign correction. Table 2-86 lists the essential commands for a f = 24 or 25 instruction. Those commands that have not been described previously are described in table 2-87 in detail.

2-321. Divide Instructions. The divide instructions consist of DIVA (f = 26) and DIVAB (f = 27), which divide the contents of the A-register by an operand obtained from memory. Both of these instructions are SR sensitive.

2-322. DIVA (f = 26). The DIVA instruction divides the dividend in the A-register (A) by the divisor (Y) obtained from memory. The quotient is contained in the AL-register and the remainder, if there is one, is contained in the AU-register. The computer performs division by systematically attempting to subtract the divisor (Y) from the dividend (A). Each subtraction is followed by left-shifting the A-register one place, decreasing the K counter by one, and testing to see if an End-Around-Borrow (EAB) resulted from the trial subtraction. If an EAB has occurred, a zero is inserted into AL_{00} and the A-register is left-shifted one place. If an EAB did not occur, a one is loaded into AL_{00}, the output of the adder is gated into the AU-register, and the A-register is left-shifted one place. This cycle is repeated 22_8 times, once for each bit in the divisor. Initially, the signs of the divisor and dividend are stored. The dividend is always converted to a positive number and the divisor is always converted to a negative number. It should be noted that overflow can occur in a division problem because of the left-shift after each trial subtraction.

2-323. DIVAB (f = 27). The DIVAB instruction provides the same service as f = 26 instruction except that the address of the operand in memory is developed by Y + B. Figure 2-67 is a simplified signal flow diagram of Divide instruction.

2-324. R1 and Divide Sequence Data Flow. Refer to figure 2-68 for a block diagram description of the data flow during the R1 and divide sequences. The figure does not show the I-sequences in which the instruction word was obtained from memory and the address of the operand was developed. However, at the end of the I-sequence, the adder outputs the address of the operand. As is shown in the figure, a memory reference is initiated and the operand (divisor) is called out of memory. At time T2.4, the D-register receives either the divisor or its complement. If the divisor is a positive number, it will be complemented and appear in the D-register in its negative form, or if it is a negative number, it will appear in the D-register unchanged. The dividend in the AU-register is made positive at time T2.2 if its initial value was negative. This is part of the initial sign correction operation which makes the divisor negative and the dividend positive prior to the actual divide operation. The Y Neg and A Neg flip-flops (refer to plate P-33) are used to remember the initial signs of the numbers for possible sign correction at the end of the division process. The Hold 1 and Hold 2 flip-flops (refer to plate P-28) are set

during the division operation to prevent an exit from the R1 sequence until the operation has been completed.

2-325. The actual division of A and D' is affected by the Divide sequence which runs in parallel with the R1-sequence. The divide commands are enabled by the setting of the 0XL00 and 0XL01 flip-flops (refer to plate P-36) and are timed by the clock phases generated by the Master Clock. At time T2.4, the 36-bit word in the AU- and AL-registers is left-shifted one place to the X- and W-registers respectively. The partial dividend in the X-register is compared with the divisor in the D-register. If X is greater than or equal to D', the division of this step can occur and is indicated by the lack of an EAB ($\overline{EAB}$). The divisor is then subtracted from the partial dividend and the difference is placed in the AU-register. The least significant bit of AL (AL_{00}) is set to a one which is the quotient bit value for this division step. If X is less than D', as is indicated by an EAB, the division cannot occur. Zeroes are subtracted from the partial dividend and it is transferred unaltered from the X to the AU-register. In this case, AL_{00} remains a zero. During each succeeding division step, the operations are as previously described. Initially the K1 adder is set to 22_8 and each time a division step occurs, the K1 adder is decremented by one. When the K1 adder contains a zero, the division operation is terminated. The quotient is contained in the AL-register, and the remainder, if there is one, is contained in the AU-register. If the signs of the original quantities were unlike, as determined by the Y Neg and A Neg flip-flops, the contents of the AL-register are complemented to provide a negative quotient. See table 2-88 for sign correction. The sign of the remainder is always adjusted so that it affects the sign of the original dividend. Figure 2-69 shows the data flow of final sign conversion. Table 2-89 lists the essential commands for a f=26 or 27 instruction. Those instructions not described previously are described in table 2-90 in detail.

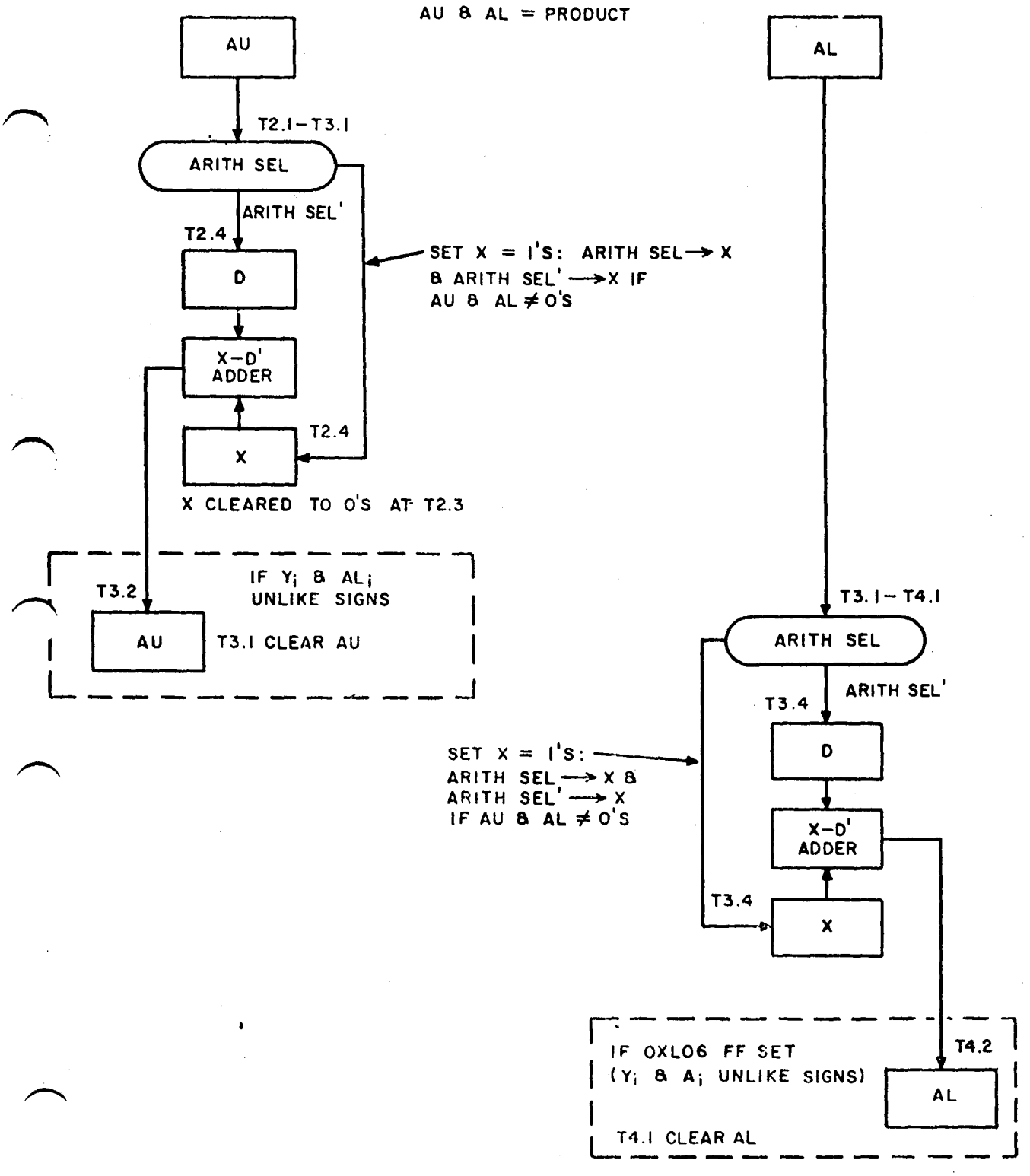


Figure 2-66. Multiply Final Sign Correction Data Flow

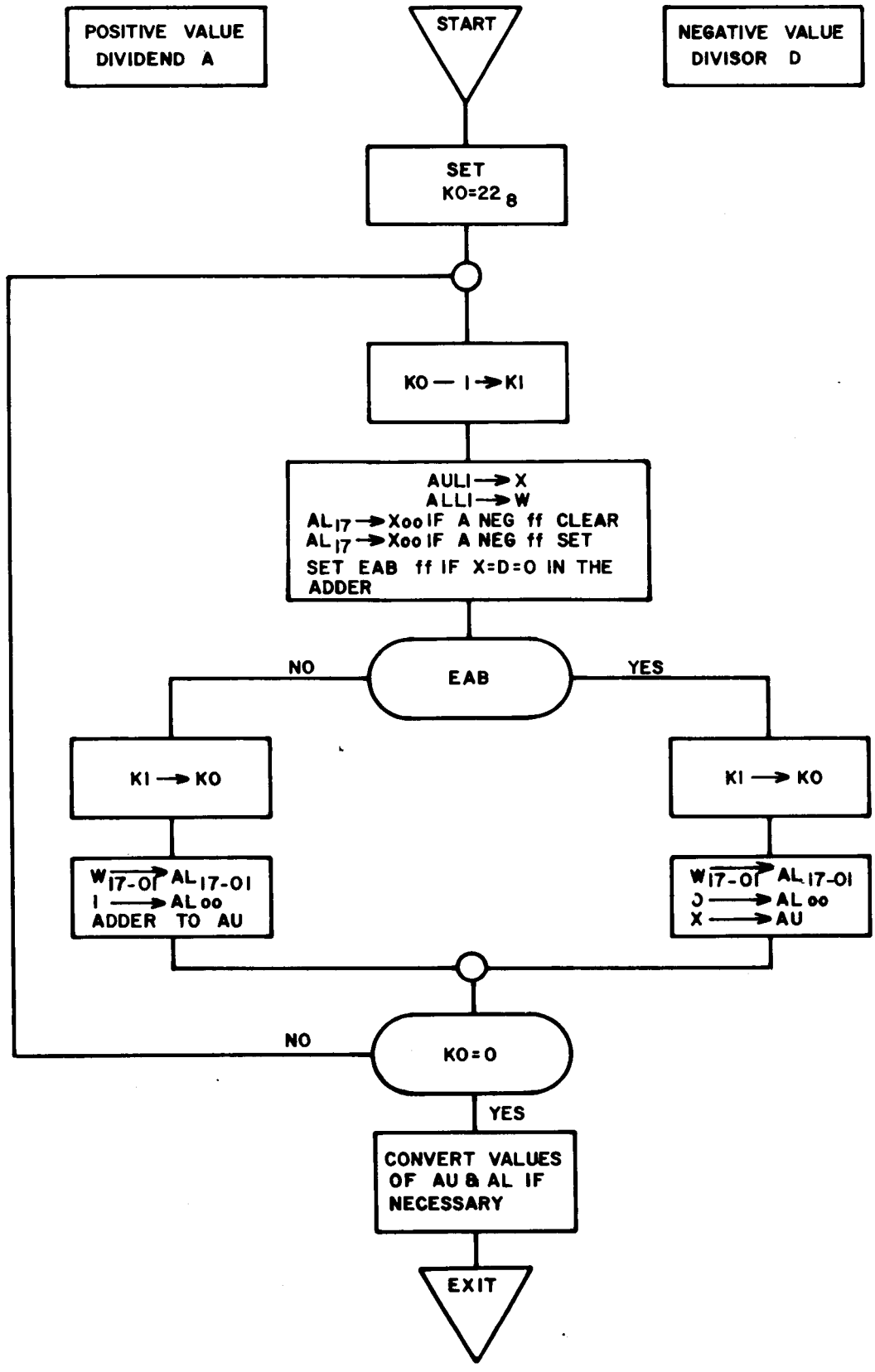


Figure 2-67. Simplified Divide Sequence

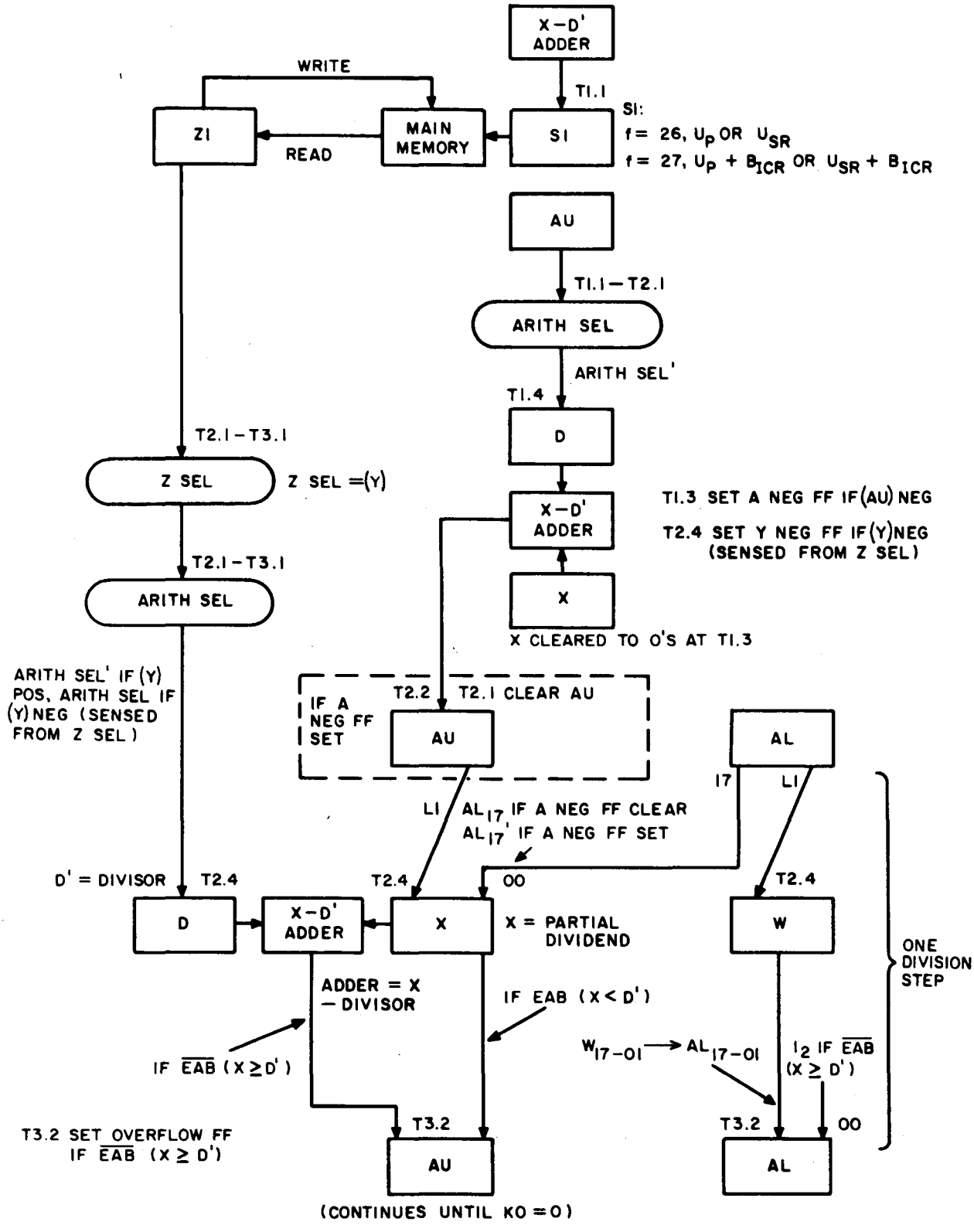


Figure 2-68. R1 and Divide Sequence Data Flow

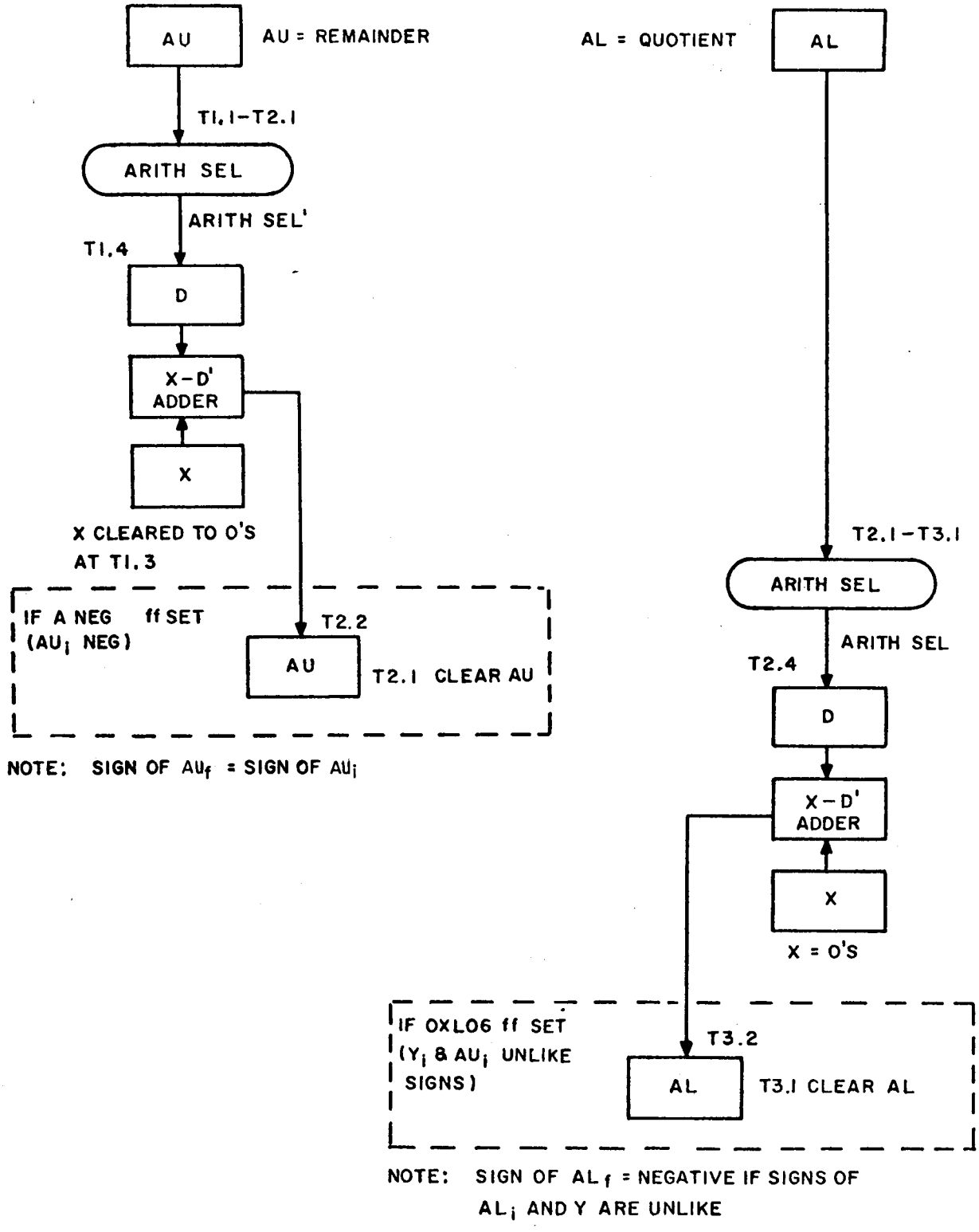


Figure 2-69. Divide Final Sign Correction Data Flow

TABLE 2-86. R1, MULTIPLY AND NEXT I-SEQUENCE ESSENTIAL COMMANDS

Time Notation	Commands
	R1-SEQUENCE
T4.4	Clear S1
T1.1	AL → Arith Sel, Adder → S1, Init Memory
T1.3	Set A Neg ff if AL neg, clear D, clear X, clear Z1
T1.4	Arith Sel' → D, clear K0, clear Scale Factor ff (1XL00)
T2.1	Clear AU, clear AL if A Neg ff set, set K0 = 23_8, set 0XL01 ff, Z1 → Z Sel, Z Sel → Arith Sel, drop AL → Arith Sel
T2.2	Adder → AL if A Neg ff set, Clear K1
T2.3	Clear D, set 0XL00 ff, clear X, clear W, clear Mult Store ff, K0-1 → K1
T2.4	Set Y Neg ff if Y neg*, Arith Sel → D if Y pos*, Arith Sel' → D if Y neg*, clear K0, AUR1 → X, ALR1 → W ***AU_{18} → W_{17}, set Mult Store ff if AL_{00} = 1
T3.1	K1 → K0 (K0 = 18), clear AL, clear AU
T3.2	Set Hold 1 ff, **W → AL, Adder → AU if Mult Store ff set X → AU if Mult Store ff clear, clear K1
T3.3	Set Hold 2 ff, K0-1 → K1, clear X, clear W
T3.4	Clear K0, AUR1 → X, ALR1 → W, ***AU_{18} → W_{17}, set Mult Store ff if AL_{00} = 1
TX.	K1 → K0, clear AL, clear AU
TX.2	Clear K1, **W → AL, adder → AU if Mult Store ff set, X → AU if Mult Store ff clear
TX.3	K0-1 → K1, clear X, clear W
TX.4	Clear K0, AUR1 → X, ALR1 → W, ***AU_{18} → W_{17}, set Mult Store ff if AL_{00} = 1
	(Continue until K0 = 0)

TABLE 2-86. R1, MULTIPLY AND NEXT I-SEQUENCE ESSENTIAL COMMANDS (Cont)

Time Notation	Commands
T1.1	Clear AL, clear AU, K1 → K0 (K0 = 0)
T1.2	W → AL, Adder → AU if Mult Store ff set
	X → AU if Mult Store ff clear, clear K1
T1.3	Set 0XL02 ff, clear 0XL00 ff, K0-1 → K1, clear X, clear W, clear Mult Store ff
T1.4	AUR1 → X, ALR1 → W, ***AU_{18} → W_{17}, set Mult Store ff if AL_{00} = 1
T2.1	Clear 0XL01 ff, set 0XL03 ff, AU → Arith Sel, set Clear-Hold ff
T2.3	Set 0XL04 ff, clear 0XL02 ff, clear D, clear X
T2.4	Arith Sel' → D, Arith Sel → X and Arith Sel' → X if AU and AL ≠ 0's (Set X = 1's)
T3.1	Set 0XL05 ff, if Y_i and AL_i unlike signs, clear 0XL03 ff, ****AL → Arith Sel, drop AU → Arith Sel, clear AU if Y_i and AL_i unlike signs
T3.2	Adder AU if Y_i and AL_i unlike signs
T3.3	****Set 0XL06 ff, clear 0XL04 ff, ****clear D
T3.4	****Arith Sel' → D, Arith Sel → X and Arith Sel' → X if AU and AL ≠ 0's (Set X = 1's)
T4.1	Clear 0XL05 ff, clear AL if 0XL06 ff set, drop AL → Arith Sel
T4.2	Adder → AL if OXL06 ff set, clear Hold 1 ff
T4.3	Clear the Clear-Hold ff, clear OXL06ff
	I-SEQUENCE OF NEXT INSTRUCTION
T1.3	Clear Hold 2 ff

*Sign of Y is sensed from Z select.

**The transmission of bit 00 for the W → AL command is through gate 83A00 in the logic diagrams, plate P-33.

***AU_{18} → W_{17} data flow is through gate 31W17 in the logic diagrams, plate P-33, and enabled by the ALR1 → W command.

****These events occur only if the 0XL05 ff is set to perform final sign correction.

TABLE 2-87. COMMANDS/DESCRIPTIONS FOR EXECUTION OF f = 24, 25
(NOT DESCRIBED IN A PREVIOUS SEQUENCE)
(REFER TO THE R SEQUENCE OR SHIFT SEQ FOR COMMANDS DESCRIBED PREVIOUSLY)

Time Notation	Command/Description
T1.3	Set A Neg ff if AL Neg. Gate 00E30 (refer to plate P-33) is enabled by a L ⇒ f = 24, 25 and the low from the set side of 0XA17 (refer to plate P-100). The high output is inverted by 01E30, and with the L ⇒ T14 R1 Seq, at ∅3, 0XG30 is set.
T1.4	Clear Scale Factor ff (1XL00). Gate 10L01 (refer to plate P-36) is enabled by a L ⇒ f = 24-27, L ⇒ R1-Seq and a low from 03T21. It produces a high output to clear flip-flop 1XL00 as previously described.
T2.1	Clear AU, Clear AL if A Neg ff set. Gate 00N04 (refer to plate P-20) is qualified by a L ⇒ f = 24, 25 and a L ⇒ T22 R1 Sequence and produces a high output which is inverted by 07N04 and applied to 08N04 to produce a H ⇒ Clear AU at ∅1 via 09N04. (Note that gate 11N04 is also qualified if the A Neg ff is set and will output a high to 07N04.) Gate 10N05 is qualified by the L ⇒ A Neg. (A Neg ff set) L ⇒ f = 24, 25 and a L ⇒ T22 R1 Seq and the high output is inverted by 07N05 and applied to 08N05. The other low inputs to 08N05 are from 00N05 which is disabled, and a ∅1 which will cause 08N05 to produce a H ⇒ Clear AL via 09N05.
T2.1	Set 0XL01 ff. The high from 10L01 (refer to plate P-36) is inverted by 11L01 and applied to 0XL01 which will now set with the low from 11L03 and ∅1.
T2.2	Adder → AL if A Neg ff set. The high from 10N05 developed at T2.1 (refer to plate P-20) is inverted by 18N05 and qualifies 19N05 at ∅2 to produce a L ⇒ Adder → AL.
T2.3	Clear Mult Store ff. With 0XL01 (refer to plate P-36) set, the low from 03L01 is applied to 0XG39 (refer to plate P-27), and, at ∅3, the Mult Store flip-flop is cleared.
T2.4	Set Y Neg ff if Y Neg. The sign of Y is sensed from Z-select. If 10X17 (refer to plate P-81) outputs a high because of the low from the Z1-register 1XZ17 (refer to plate P-113), it is inverted by 12X17 (refer to plate P-81) and applied to Y Neg flip-flop 1XG30 (refer to plate P-33). Gate 10E30 is enabled by a L ⇒ F = 24-27 and a L ⇒ T24 R1-Seq, and the high is inverted by 11E30. At ∅4, 1XG30 is set.

TABLE 2-87. COMMANDS/DESCRIPTIONS FOR EXECUTION OF f = 24, 25 (Cont)

Time Notation	Command/Description
	Arith Sel ⇒ D if Y Pos. Gate 27N02 (refer to plate P-18) is enabled by a L ⇒ Y Pos, L ⇒ f = 24, 25 and a L ⇒ T24 R1 Seq. The high output disables 29N02 and 17N02. The low from 17N02 with the low from 18N02, at ∅4, enables 19N02 to provide a L ⇒ Select ⇒ D.
	Arith Sel' ⇒ D if Y Neg. Gate 27N02 is disabled by the absence of a L ⇒ Y Pos, and with the low input from 28N02, at ∅4, 29N02 provides a L ⇒ $\overline{\text{Select}}$ ⇒ D.
	Set Mult Store ff if AL_{00} = 1. Multiplier Store flip-flop 0XG39 (refer to plate P-27) is set by a L ⇒ Mult Div from a 03L01 due to 0XL01 being set (refer to plate P-36), a low from the set side of 0XA00 (refer to plate P-99) and ∅4.
T3.2	Adder ⇒ AU if Mult Store ff Set. The L ⇒ AL Bit 0 = 1 (refer to plate P-27) is applied to 10N04 (refer to plate P-20) and, with the low from 03L00 (refer to plate P-36) due to 0XL00 being set, 10N04 produces a high output. With 00N04 disabled by the absence of a L ⇒ T22 R1-Seq, 19N04 produces a L ⇒ Adder ⇒ AU as previously described.
	X ⇒ AU if Mult Store ff Clear. With Mult Store flip-flop clear, gate 10N04 (refer to plate P-20) is disabled by the absence of a L ⇒ AL Bit 0 = 1 and a L ⇒ X ⇒ AU is produced as previously described.
	Set Hold 1 ff. Hold 1 ff 1XG38 (refer to plate P-28) is set at ∅2 by the low from 01L00 and the absence of a H ⇒ 50:40-47.
T2.1	Set 0XL03 ff. Flip-flop 0XL03 (refer to plate P-36) is set by the low from 0XL02, the absence of a f = 50:40-49, and ∅1.
T2.3	Set 0XL04 ff. Flip-flop 0XL04 (refer to plate P-36) is set by the low from 0XL03 and ∅3.
T3.1	Set 0XL05 ff, if Y_i and AL_i Unlike Signs. If Y_i and AL_i have unlike signs, either Y Neg flip-flop or A Neg flip-flop will be set (but not both). If Y Neg ff 1XG30 (refer to plate P-33) is set and A Neg flip-flop 0XG30 is clear, gate 20G30 is enabled by the high from the clear side of 1XG30 and the high from the set side of 0XG30. The L ⇒ Signs Unlike is applied to 0XL05 (refer to plate P-36), and with the low from 0XL04, 0XL05 is set at ∅1.

TABLE 2-87. COMMANDS/DESCRIPTIONS FOR EXECUTION OF f = 24, 25 (Cont)

Time Notation	Command/Description
	Clear AU if Y_i and AL_i Unlike Signs. The low from 20G30 (refer to plate P-33) is also applied to 21G30, and with L ⇒ f = 24, 25, 21G30 produces a high that is inverted by 24G30 and applied as a L ⇒ Comp → AU to 11N04 (refer to plate P-20). The other input to 11N04 is from the set side of 0XL04 (refer to plate P-36), and AU is cleared as previously described.
T3.2	Adder → AU if Y_i and AL_i Unlike Signs. Uses same inputs as clear AU.
T3.3	Set 0XL06 ff. Flip-flop 0XL06 is set at ∅3 by the low from 0XL05 (refer to plate P-36).
	Clear D. The high from the clear side of 0XL05 (refer to plate P-36) is inverted by 03L03 and applied to 20N02 (refer to plate P-18) to clear D as previously described.
T4.1	Clear AL if 0XL06 ff Set. The high from the clear side of 0XL06 (refer to plate P-36) is applied to L ⇒ Clear AL and Adder → AL to 07N05 (refer to plate P-20) to clear AL as previously described.
T4.2	Adder → AL if 0XL06 ff Set. Uses input H ⇒ Clear AL, Adder → AL into 18N05 (refer to plate P-20).

TABLE 2-88. DIVIDE SEQUENCE SIGN CORRECTION

DIVIDEND	DIVISOR	QUOTIENT	REMAINDER
Positive	Positive	Positive	Positive
Positive	Negative	Negative	Positive
Negative	Positive	Negative	Negative
Negative	Negative	Positive	Negative

TABLE 2-89. R1, DIVIDE, AND NEXT I-SEQUENCE ESSENTIAL COMMANDS

Time Notation	Commands
	R1-SEQUENCE
T4.4	Clear S1
T1.1	AU → Arith Sel, Adder → S1, Init Memory
T1.3	Set A Neg ff if AU neg, clear D, clear X, clear Z1
T1.4	Arith Sel' → D, clear K0, clear Scale Factor ff (1XL00)
T2.1	Clear AU if A Neg ff set, set K0 = 22_8, set 0XL01 ff, Z1 → Z-Sel, Z-Sel → Arith Sel, drop AU → Arith Sel
T2.2	Adder → AU if A Neg ff set, clear K1
T2.3	Set 0XL00 ff, K0-1 → K1, clear D, clear X, clear W
T2.4	Arith Sel' → D if Y pos*, Arith Sel → D if Y neg*, clear K0, set Y Neg ff if Y neg*, AUL1 → X, ALL1 → W, $**AL_{17} \rightarrow X_{00}$ if A Neg ff clear, $**AL'_{17} \rightarrow X_{00}$ if A Neg ff set
T3.1	K1 → K0, clear AU, clear AL, drop Z1 → Z-Sel, drop Z-Sel → Arith Sel
T3.2	Set Hold 1 ff, $W_{17-01} \rightarrow AL_{17-01}$, $***1_2 \rightarrow AL_{00}$ if $\overline{EAB}$, X → AU if EAB, Adder → AU if $\overline{EAB}$, clear K1, set Overflow ff if $\overline{EAB}$
T3.3	Set Hold 2 ff, K0-1 → K1, clear X, clear W
T3.4	Clear K0, AUL1 → X, ALL1 → W, $**AL_{17} \rightarrow X_{00}$ if A Neg ff clear, $**AL'_{17} \rightarrow X_{00}$ if A Neg ff set
TX.1	K1 → K0, clear AU, clear AL
TX.2	Clear K1, $W_{17-01} \rightarrow AL_{17-01}$, $***1_2 \rightarrow AL_{00}$ if $\overline{EAB}$, X → AU if EAB, Adder → AU if $\overline{EAB}$
TX.3	K0-1 → K1, clear X, clear W
TX.4	Clear K0, AUL1 → X, ALL1 → W, $**AL_{17} \rightarrow X_{00}$ if A Neg ff clear, $**AL'_{17} \rightarrow X_{00}$ if A Neg ff set
	(Continue until K0 = 0)

TABLE 2-89. R1, DIVIDE, AND NEXT I-SEQUENCE ESSENTIAL COMMANDS (Cont)

Time Notation	Commands
T4.1	Clear AU, clear AL, K1 → K0 (K0 = 0)
T4.2	$W_{17-01} \rightarrow AL_{17-01}$, $***1_2 \rightarrow AL_{00}$ if $\overline{EAB}$, X → AU if EAB, Adder → AU if $\overline{EAB}$, clear K1
T4.3	Set 0XL02 ff, clear 0XL00 ff, K0 = 1 → K1, clear X, clear W
T4.4	AUL1 → X, ALL1 → W, $**AL_{17} \rightarrow X_{00}$ if A Neg ff clear, $**AL'_{17} \rightarrow X_{00}$ if A Neg ff set
T1.1	Clear 0XL01 ff, set 0XL03 ff, AU → Arith Sel, set Clear Hold ff
T1.3	Set 0XL04 ff, clear 0XL02 ff, clear D, clear X
T1.4	Arith Sel' → D
T2.1	Set 0XL05 ff if Y_i and AU_i unlike signs, clear 0XL03 ff, ****AL → Arith Sel, drop AU → Arith Sel, clear AU if A Neg ff set
T2.2	Adder → AU if A Neg ff set
T2.3	****Set 0XL06 ff, clear 0XL04 ff, ****clear D
T2.4	****Arith Sel' → D
T3.1	Clear 0XL05 ff, clear AL if 0XL06 ff set, drop AL → Arith Sel
T3.2	Adder → AL if 0XL06 ff set
T3.3	Clear 0XL06 ff
T4.2	Clear Hold 1 ff
T4.3	Clear the Clear-Hold ff I-SEQUENCE OF NEXT INSTRUCTION
T1.3	Clear Hold 2 ff

*Sign of Y is sensed from Z-Select.

**$AL_{17} \rightarrow X_{00}$ and $AL'_{17} \rightarrow X_{00}$ data flow is through gate 31X00 in the logic diagrams, plate P-33, and is enabled by the AUL1 → X command.

***$1_2 \rightarrow AL_{00}$ data flow is through gate 83A00 in the logic diagrams, plate P-33, and is enabled by the W → AL command

****These events occur only if the 0XL05 ff is set to perform final sign correction.

TABLE 2-90. COMMANDS/DESCRIPTIONS FOR EXECUTION OF f = 26, 27
(NOT DESCRIBED IN A PREVIOUS SEQUENCE)
(REFER TO THE R, MULTIPLY OR SHIFT SEQ FOR COMMANDS PREVIOUSLY DESCRIBED)

Time Notation	Commands/Description
T1.1	AU → Arith Sel. Gate 10N01 (refer to plate P-17) is enabled by a L ⇒ f = 26, 27 and a L ⇒ T14 R1-Seq to provide a L ⇒ AU → Select as previously described via 18N01 and 19N01.
T1.3	Set A Neg ff if AU Neg. Same as f = 24, 25, except gate 00E30 (refer to plate P-33) is enabled by a L ⇒ f = 26, 27 and a low from the set side of 0XA35 (refer to plate P-98).
T2.4	Arith Sel' → D if Y Pos. Gate 27N02 (refer to plate P-18) is disabled by the absence of a L ⇒ Y Neg and gate 29N02 produces a L ⇒ Select → D as previously described. Arith Sel → D if Y Neg. Gate 27N02 (refer to plate P-18) is enabled by a L ⇒ f = 26, 27, a L ⇒ Y Neg, and a L ⇒ T24 R1-Seq to produce a L ⇒ Select > D as previously described. $AL_{17} \rightarrow X_{00}$ if A Neg ff Clear. If $AL_{17} = 1$, the low from 0XA17 (refer to plate P-100) is inverted by 02A17 and applied to 31X00 (refer to plate P-33). With 0XG30 clear, the high from the set side is also applied to 31X00, along with the absence of a L ⇒ f = 45. Gate 31X00 is enabled to produce $AL_{17} \rightarrow X_{00}$, as previously described. If $AL_{17} = 0$, the input from 02A17 is low and will disable 31X00, which outputs a high to produce $AL_{17} \rightarrow X_{00}$. $AL'_{17} \rightarrow X_{00}$ if A Neg ff Set. If $AL'_{17} = 1$, 31X00 (refer to plate P-33) is disabled by the low from 03A17 (refer to plate P-100), the low due to the absence of a H ⇒ f = 45, and the low from the set side of 0XG30 (refer to plate P-33). The output of 31X00 is now high and will produce $AL'_{17} \rightarrow X_{00}$. If $AL_{17} = 0$, 31X00 is enabled and will also produce $AL'_{17} \rightarrow X_{00}$.
T3.2	$W_{17-01} \rightarrow AL_{17-01}$. Flip-flop 0XA17 (refer to plate P-100) is set by a L ⇒ W → AL, if 0XW17 (refer to plate P-90) is set. $1_2 \rightarrow AL_{00}$ if $\overline{EAB}$. Gate 83A00 (refer to plate P-33) is enabled by a H ⇒ f = 26, 27 and the absence of a L ⇒ EAB. The low output is applied to 0XA00 (refer to plate P-99) and with the L ⇒ W → AL, 0XA00 is set. X → AU if EAB. Gate 10N04 (refer to plate P-20) is disabled by the absence of a L ⇒ $\overline{EAB}$ to produce a L ⇒ X → AU as previously described. Adder → AU if $\overline{EAB}$. Gate 10N04 (refer to plate P-20) is enabled by a L ⇒ $\overline{EAB}$ and a low from 03L00 to produce a L ⇒ Adder → AU as previously described. Set Overflow ff if $\overline{EAB}$. Gate 20E52 (refer to plate P-30) is enabled by a L ⇒ R1-Seq, the absence of a f = 20-23 · R2-Seq and a low from 01T32. The high output with H ⇒ f = 26-27 produces a low out of 21E52 and with a L ⇒ Adder → AU, 0XG52 is set at ∅2.

Section 2-3. Detailed Functional Description (Memory Section)

2-326. MEMORY SECTION

2-327. The memory section consists of main memory, control memory and bootstrap memory. The three memories differ in cycle time and storage capacity, but all have 18-bit addressable words and use magnetic cores as the storage media. (A magnetic core is a ring-shaped ferrite device capable of being magnetized in either of two directions. The core is considered to be storing a one when magnetized in one direction, and a zero when magnetized in the opposite direction.) Each core represents one bit position for one word location in memory. Storage capacities of the three memories are listed in table 2-91.

2-328. Three steps are necessary in the operation of memory: (1) selection of a discrete address (2) performance of a read cycle to read the word from the selected address, and (3) performance of a write cycle to store the word at the selected address. In main and control memories, the read cycle erases the addressed memory location. The word must, therefore, be restored or replaced with a new word during the write cycle. Bootstrap is a nondestructive read-out memory. A word is not erased during the read cycle and a word cannot be written into this memory. Figure 2-70 shows the memory section of the computer in block diagram form. It shows the registers, control, addressing, and other circuits necessary to operate the main, control, and bootstrap memories.

2-329. 16-BIT S_1-REGISTER. The 16-bit S_1-register is the address register for main memory. For every memory reference, it receives addresses from the P-register, the B-register, the I/O section, or the arithmetic section.

2-330. ADDRESS TRANSLATORS. The address translators consist mainly of switching circuits and read-and-write current diverters. These circuits translate the address from the S_1-register and generate drive currents for reading from, or writing into, memory at the selected address. During a read cycle,

TABLE 2-91. COMPUTER MEMORY SIZES

Memory Designation	Main Memory Addresses Available	Bootstrap Memory Addresses	Control Memory Addresses
16K	16,384	32	128
32K	32,768	32	128
49K	49,152	32	128
65K	65,536	32	128

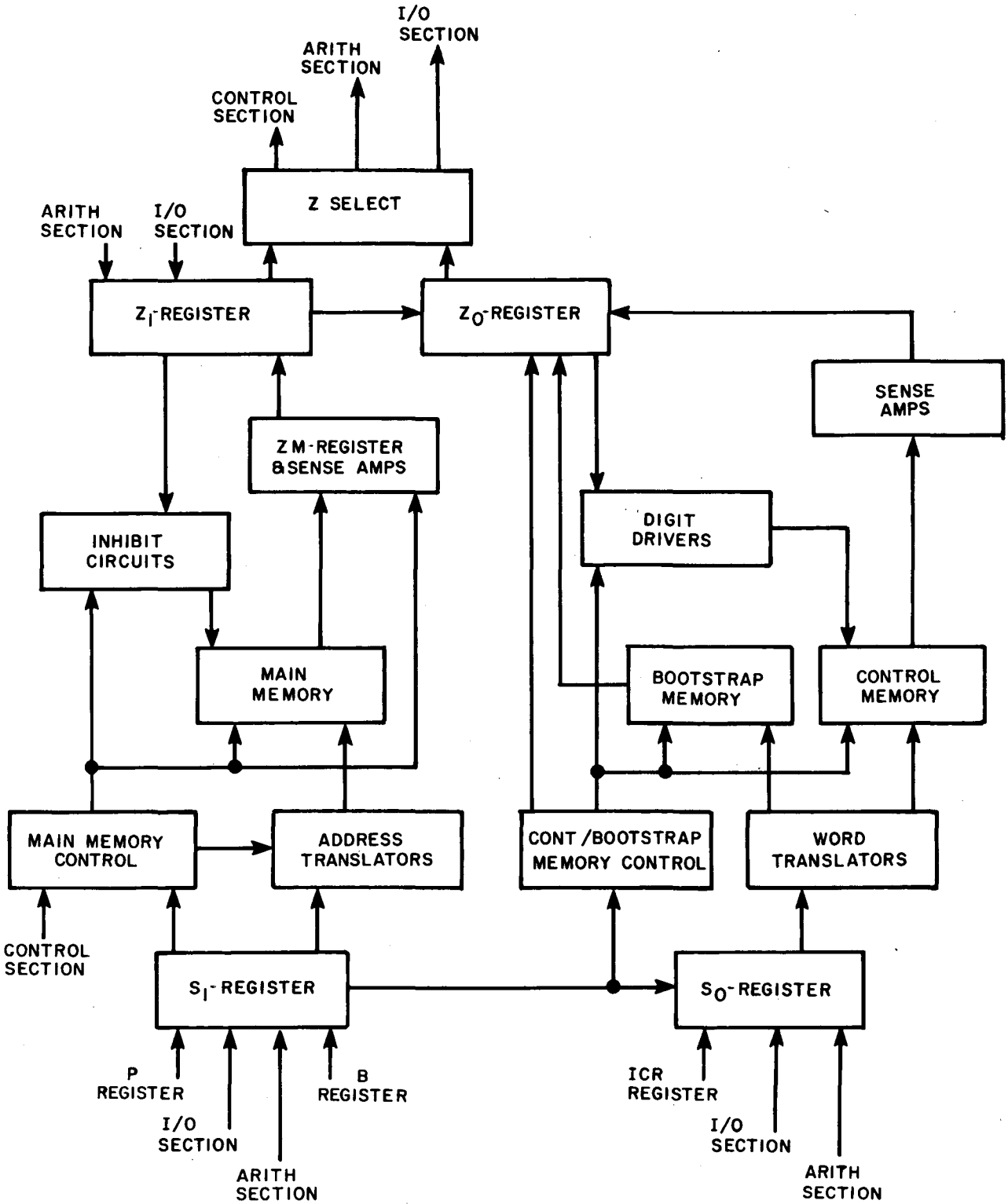


Figure 2-70. Memory Section Simplified Block Diagram

the information stored in the cores is detected by the sense amplifiers and temporarily stored in the 18-bit Z_1-register, which is the input-output register for main memory. Information in the Z_1-register is routed through Z-select circuits to the proper area in the computer. During a write cycle, the information in Z_1 is written into memory by means of the inhibit circuits. This may be information which was stored as the result of a read cycle, or it may be information received from the arithmetic or I/O sections.

2-331. 8-BIT S_0-REGISTER. The 8-bit S_0-register receives control and bootstrap addresses from the S_1-register, the index control register, the I/O section, or the arithmetic section. Translators decode the address and generate drive currents for read/write operations. When a control memory address has been decoded, an 18-bit word is read from control memory by the sense amplifiers and is temporarily stored in the Z_0-register. When a bootstrap address has been decoded, the information is read from bootstrap memory directly into the Z_0-register. Z-select circuits distribute the information in the Z_0-register to the proper computer section. During a write cycle, the information stored in Z_0 may be written into control memory by the digit drivers. Since bootstrap memory is hard-wired, it does not require rewrite circuitry to restore the information to its proper location.

2-332. MAIN MEMORY. Main memory is a random-access, coincident-current, bit-oriented, core memory with a two-microsecond cycle time and 750-nanosecond (maximum) access time. There are two memory chassis with a maximum of two memory banks per chassis. Memory size is determined by the number of memory banks in the computer. A memory bank has 16,384 eighteen-bit storage locations and is composed (functionally and physically) of four stacks. Each stack is made up of a group of core-planes. The number of planes in a stack corresponds to the number of bits in the words to be stored. The computer uses 18-bit words. Therefore, each memory stack contains 18 core-planes. The 4,096 cores on each plane establish the number of 18-bit storage locations (words) in a stack. Figure 2-71 illustrates the main memory layout of the computer.

2-333. During a memory cycle, the address of S_1 is translated to select the proper chassis, bank, stack, X-line, and Y-line. Select circuits provide enables for the read-write (R/W) switch circuit associated with the selected address. The R/W switches permit read and write currents to be developed by the R/W current diverters and fed to the selected core stack. The read current is developed in one direction and the write current in the other.

2-334. The sense lines are connected to common sense-amplifier circuits. All information read from memory passes through the Z_1-register, from the sense amplifiers. Information to be written into memory is placed into the Z_1-register, from which the inhibit circuits are controlled during a write process. The inhibit circuits develop enables to all the inhibit switch circuits of a selected stack, permitting inhibit drive currents to pass through. The inhibit drive currents come from the + 10V regulator and are controlled by the current diverter circuits. The current diverter circuits receive their enables from the Z_1-register. Any zero in the Z_1-register enables its current diverter circuit. Current from the current diverter is passed through the inhibit switch circuits through all the cores on its plane to prevent writing in that plane.

2-335. Main Memory Addressing. Figure 2-72 illustrates the method of addressing main

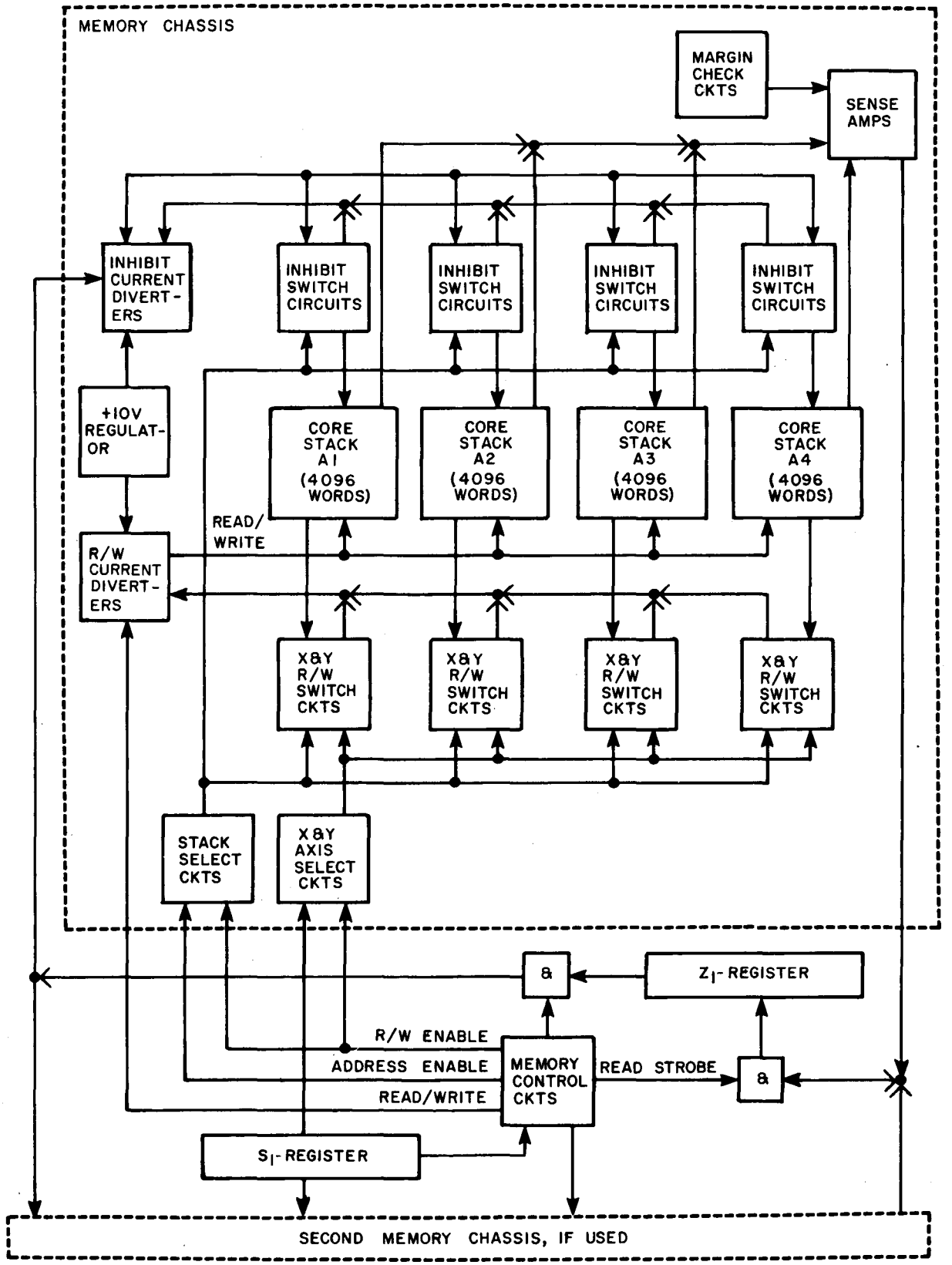


Figure 2-71. Main Memory Block Diagram

memory through translation of the 16-bit S_1-register. To select a single memory address, the bit configuration of the word contained in S_1 must specify one of four banks, one of four stacks in the selected bank, one of 64 X-lines in the selected stack, and one of 64 Y-lines in the selected stack. The 64 X-lines and 64 Y-lines in each stack are divided into eight groups of eight lines per group. Therefore, line selection is accomplished by specifying one of eight groups and one of eight lines within that group. The bit configuration of S_1 is shown in figure 2-73.

2-336. As shown in plates P-104 and P-105 (refer to Chapter 9), bits are gated into the S_1-register from the I/O translator, B-register, P-register, or adder. Also, bits 0, 6, or 8 may be set during phase 1 if special external selections have been made. Outputs from S_1-register, bits 0 through 13, are routed to level changers (refer to plates P-137 and P-138) where they are converted into logic levels useable in memory. Bits 14 and 15 are decoded to select a 16k bank (refer to plate P-134) and the bank selections are converted to memory logic by level changers 74Yg0 and 74Yg1 (refer to plate P-136).

> NOTE: Level changers are used to convert -4.5 and 0.0 voltage levels from other sections of the computer to the +3.0 and 0.0 voltage levels used in main memory. Similarly, they convert information read from memory back into the -4.5 and 0.0 levels used by other sections of the computer.

2-337. The bank-select signals are converted into bank enables (refer to plate P-136). They are also gated by the proper timing to produce read-address enables or write-address enables for banks 0 through 3. These enables are routed to the selected banks in order to enable stack, group, and line selection for that bank.

2-338. Core Magnetization. Two conductors (X and Y) threaded through the center of the ferrite core provide a means for magnetizing the core in either direction (refer to a in figure 2-74). The direction of current through these two conductors determines the magnetic polarity induced in the core. The operating current is of such magnitude that pulses must be passed simultaneously through both conductors to magnetize the core.

2-339. A core is read (refer to b in figure 2-74) magnetizing it to the zero state. If the core was storing a one, the changing polarity induces a sense pulse in a third wire (sense-inhibit) through the core. This sense pulse, during read time, is detected as a one. If the core was already storing a zero, no sense pulse is induced in the sense-inhibit line. The lack of a sense pulse, at read time, is detected as a zero.

2-340. A read cycle places all cores in the zero state prior to writing. During the write cycle (refer to c in figure 2-74), if a one is to be written, coincident pulses on the X and Y-lines switch the core to the one state. If a zero is to be written, an inhibit pulse is passed through the sense-inhibit line. The direction of the inhibit pulse induces magnetism that counteracts the effect of the X and Y-pulse and results in the core not being switched.

2-341. Memory Stack Operation. The example in figure 2-75 illustrates the typical operation for a five-bit stack of 6 X 6 planes. This stack has a capacity of 36 five-bit words.

> NOTE: In actual operation, main memory uses 18-bit stacks of 64 X 64 core planes.

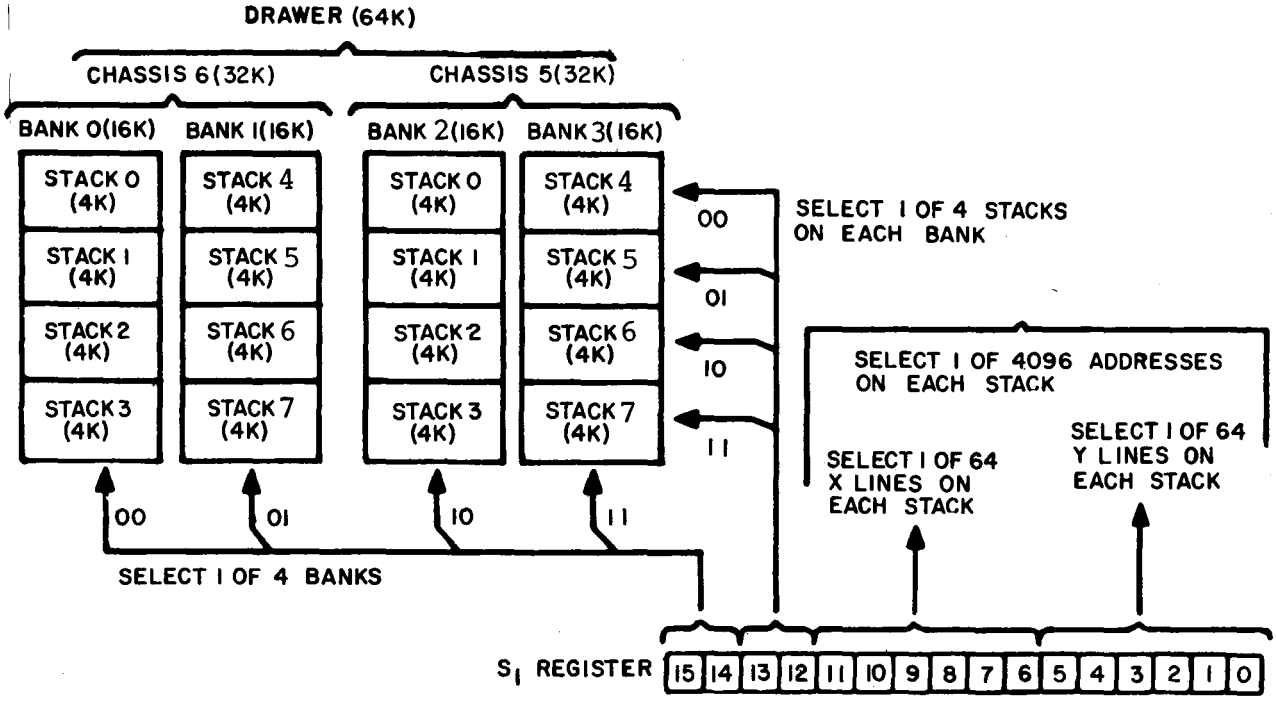
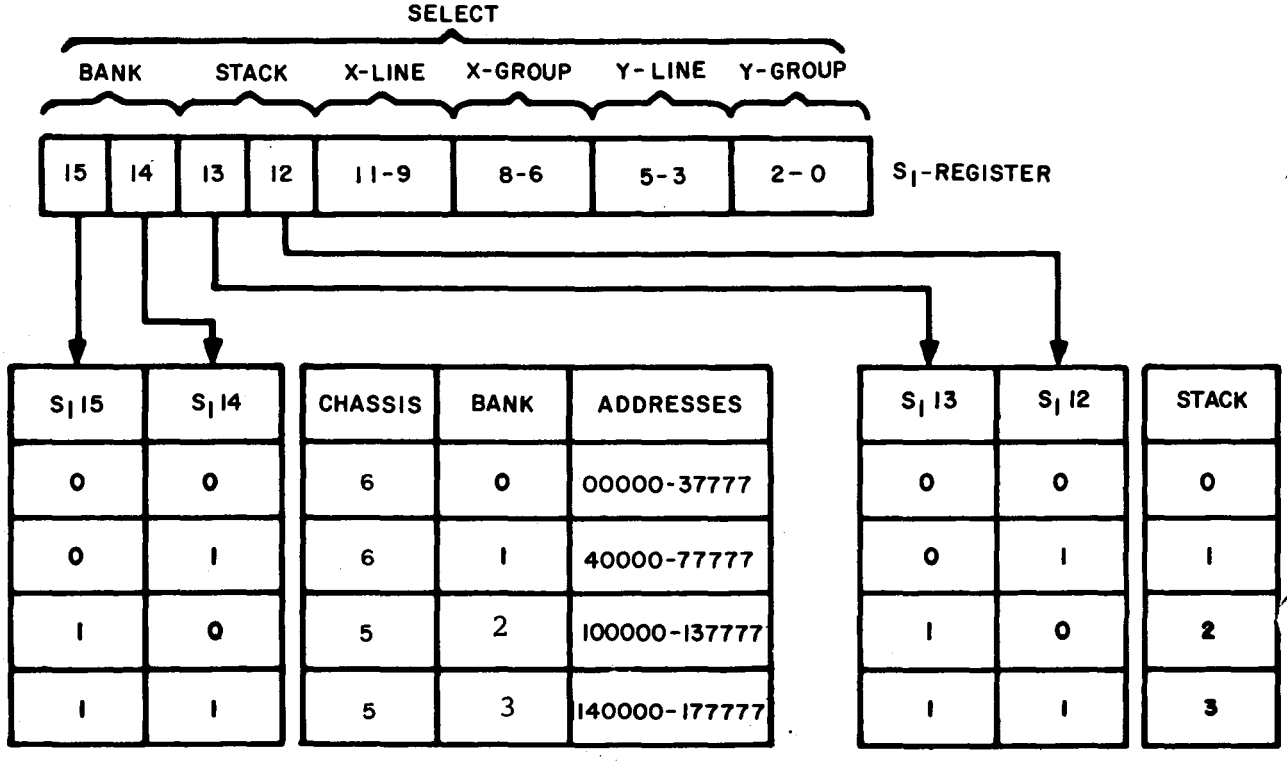


Figure 2-72. Main Memory Addressing



S_1 15	S_1 14	CHASSIS	BANK	ADDRESSES
0	0	6	0	00000-37777
0	1	6	1	40000-77777
1	0	5	2	100000-137777
1	1	5	3	140000-177777

S_1 13	S_1 12	STACK
0	0	0
0	1	1
1	0	2
1	1	3

Figure 2-73. S_1 Bit Translation

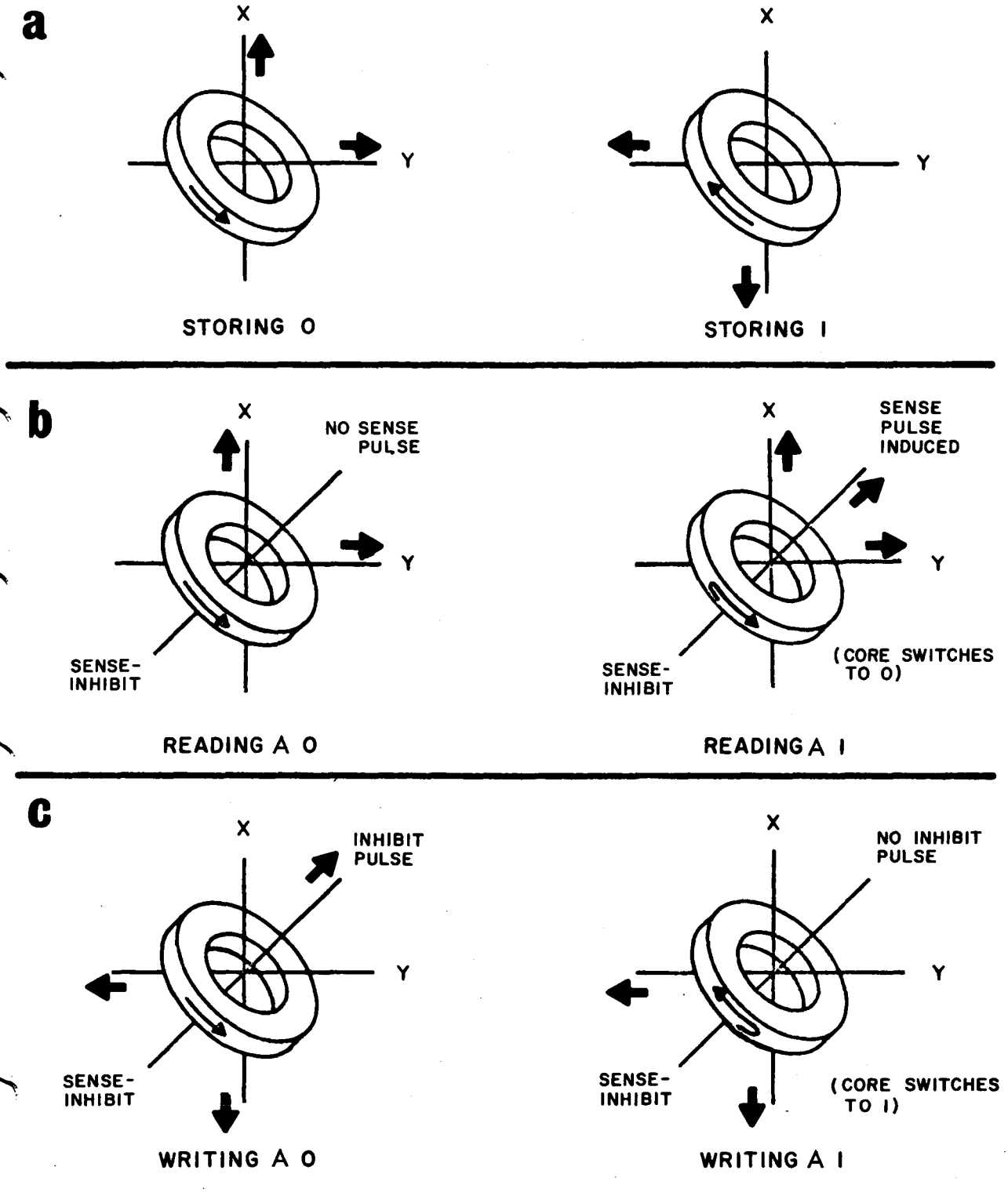


Figure 2-74. Core Magnetization

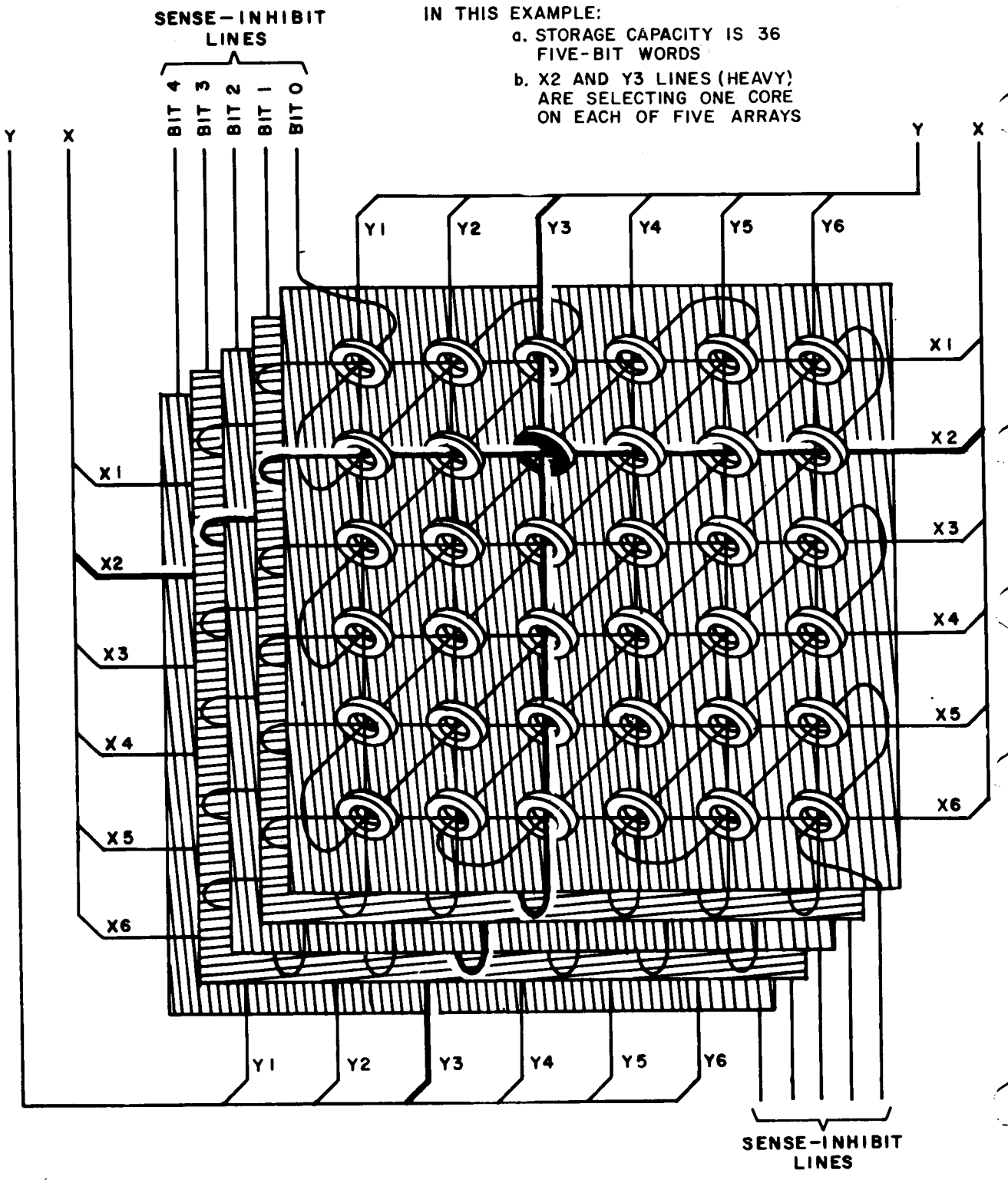


Figure 2-75. Memory Stack Operation

2-342. Each X-line passes through a single horizontal row of cores on each plane. Each Y-line passes through a single vertical row of cores on each plane. Applying simultaneous current pulses to a single X-line and a single Y-line selects the one core (on every plane) at the junction of the two lines. The example in figure 2-75 illustrates selection of a core at the junction of the X2 and Y3-lines (heavy lines). A separate sense-inhibit line is used for each plane. Each line passes through every core on the plane. The five-array stack in figure 2-75 has five sense-inhibit lines (one for each bit in a word). During a read cycle, X and Y-lines are energized in the direction necessary to magnetize a selected core on each plane to zero. If the selected core contains a one, a sense pulse is induced into the sense-inhibit line for that plane when the core switches to the zero state. If a core already contains a zero, it does not change state and no sense pulse is induced into the sense-inhibit line. Any sense pulse on a sense-inhibit line is interpreted as a one, regardless of polarity.

2-343. As a result of the read cycle, the selected cores are magnetized to zero. During the write cycle, X and Y-lines are pulsed to write a one in the selected core of each plane. If a zero is to be stored at any bit position, an inhibit pulse is placed on the sense-inhibit line at the same time the X and Y-lines are pulsed. This inhibit pulse is of such polarity and magnitude that it counteracts the effect of the X and Y-pulses and prevents the core from being switched.

2-344. X and Y-Line Selection. The selection of a particular 18-bit location during a memory reference is accomplished by pulsing one X-line and one Y-line with approximately 400 mA of current each. For example, figure 2-76 illustrates an X-line or Y-line selection for a single stack. The heavy line in the figure shows the write current path for the selection of line 1 in group 1. The word-line translator decodes S_1 bits 9, 10 and 11 and turns on one of eight transistor switches (Q1) in the selected bank. The group selector decodes S_1 bits 6, 7 and 8 and turns on one of eight transistor switches (Q2) in the same bank. Transistors Q1 and Q2 provide a ground return for both write current pulses from the secondaries of T2 and T3 and read current pulses from T1 and T3 secondaries. A write circuit is completed through CR4, Q2, CR6, one of eight secondary windings on T3, one line of cores in the stack, CR8, one of eight secondary windings in T2, Q1, CR1, and ground. A read circuit is completed through CR2, Q1, one of eight windings on T1, CR7, one line of cores in the stack, one of eight secondary windings on T3, CR5, Q2, CR3, and ground.

2-345. The primary circuit for the stack is enabled by decoding the stack select bits (S_1 bits 12 and 13), gating them with the proper bank-select signal, and turning on transistor switch Q5. Transistor Q6 is normally conducting, thereby clamping the collector of Q5 to -15 volts and disabling the primary circuit. When a read enable or write enable turns Q6 off, +15 volts is provided, thereby allowing generation of a pulse on the primary line. A read enable or write enable is gated with the selected bank enable signal to produce a read pulse or write pulse, respecitvely. A write pulse turns on transistor Q4 and completes a primary circuit from -15 volts, thru Q4, T2 primary, T3 primary pins 4-3, Q5, and L1 to +15 volts. A read pulse turns on transistor Q3 and completes a primary circuit from -15 volts, thru Q3, T1 primary, T3 primary pins 1-2, Q5, and L1 to +15 volts.

2-346. As shown by the arrows in figure 2-76 a write current pulse is induced in the secondary circuit in one direction and a read current pulse is indicated in the opposite direction.

The difference in current direction is determined by the difference between the winding direction of T1 and T2 secondaries, and between the two halves of the T3 primaries.

2-347. X and Y-line selection is identical for each 16k bank of memory except that the read/write address-enables and bank-enables depend on the bank being selected. Refer to simplified diagram figure 2-77 along with associated logic diagrams for the following discussion of X and Y-line selection for bank 0. Logic diagrams for the 0400, 0421, and 0431 Cards are shown in Chapter 7 (figures 7-72, 7-75, and 7-77).

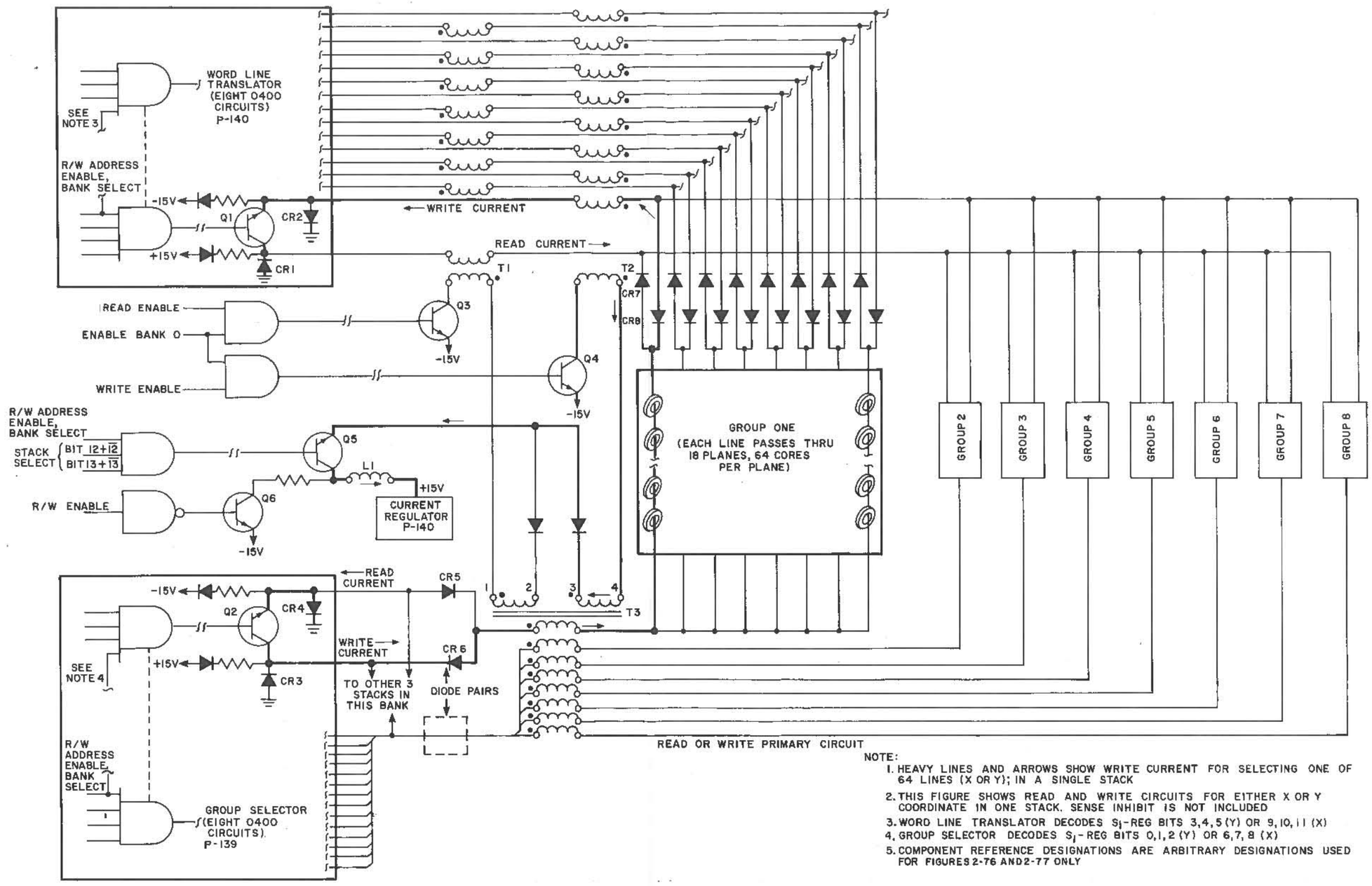
2-348. X-Axis Group and Stack Selection (refer to plate P-139). Depending upon the configuration of S_1-register bits 6 through 8, and AND circuit on one of the eight type 0400 cards receives three highs. When the R/W address enable for bank 0 goes high, both output pins are switched to ground level. The ground level signals are routed to the associated diode pairs on four type 0431 cards. In the 0431 cards, each diode pair is connected to one of eight secondary windings on the X-azis transformer. In this manner, a ground is provided for one of eight secondary windings on the X-axis transformers. At the same time, depending upon the configuration of S_1-register bits 12 and 13, an input AND circuit on one of the four 0431 cards is enabled, turning on the driver transistor whose collector is connected to a current-diverter 80R00 (refer to plate P-140). (The X-line current-diverter for bank 0 is 80R00. Refer to table 2-92 for a complete listing of current-diverters.) One of the X-axis group transformers is thus selected to drive one stack.

2-349. X-Axis Word Selection and Drive (refer to plate P-140). The R/W address enable that selects X-axis group, enables one of eight 0400 cards, depending upon the configuration of S_1-register bits 9 through 11. Both output pins of the enabled 0400 card are switched to ground level, providing ground returns to one of eight secondaries on transformers 82R00 and 82W00. (82R00 and 82W00 are the X-line read and write select transformers for bank 0. Refer to table 2-93 for a complete listing of read/write select transformers.) The other ends of the secondaries are connected to the read and write diodes associated with the eight X-lines in each group. Thus, one of the eight lines in the group is selected.

2-350. Y-Axis Group, Stack and Line Connection (refer to plates P-142 and P-143). Selection of the Y-axis drive line is similar to X-axis selection. S_1-register bits 0 through 2 enable one group, and bits 3 through 5 enable one line. The same bits (12 and 13) select one transformer associated with a stack.

2-351. Read/Write (R/W) Operation. A memory cycle is initiated by setting 0XG80 (refer to plate P-134). This occurs during time T1.1 of an I, R1, W, or I/O sequence. Initiate memory (refer to plate P-136) is sent via inverters 74Yg2 and 54MT01 to 50MT20 (refer to plate P-135) where it is used to initiate main memory timing. Initiate memory is also sent to the Z_M-register flip-flops. The R/W address-enables for a selected bank will allow selection of one X and one Y drive-line, as previously discussed. Refer to the simplified diagram, figure 2-77, along with associated logic diagrams for the following discussion of read/write operations of a selected X drive-line in bank 0.

2-352. During a read or write operation, the read/write enables for bank 0 cut off current-diverter 80R00 (refer to plate P-140), removing the loads from current-regulator 85X00. (85X00 is the X-line current regulator for bank 0. See table 2-94 for a complete listing of current-regulators.) The



NOTE:
1. HEAVY LINES AND ARROWS SHOW WRITE CURRENT FOR SELECTING ONE OF 64 LINES (X OR Y); IN A SINGLE STACK
2. THIS FIGURE SHOWS READ AND WRITE CIRCUITS FOR EITHER X OR Y COORDINATE IN ONE STACK. SENSE INHIBIT IS NOT INCLUDED
3. WORD LINE TRANSLATOR DECODES S_i-REG BITS 3,4,5 (Y) OR 9,10,11 (X)
4. GROUP SELECTOR DECODES S_j-REG BITS 0,1,2 (Y) OR 6,7,8 (X)
5. COMPONENT REFERENCE DESIGNATIONS ARE ARBITRARY DESIGNATIONS USED FOR FIGURES 2-76 AND 2-77 ONLY

Figure 2-76. X or Y-Line Selection Circuit

THIS PAGE INTENTIONALLY LEFT BLANK

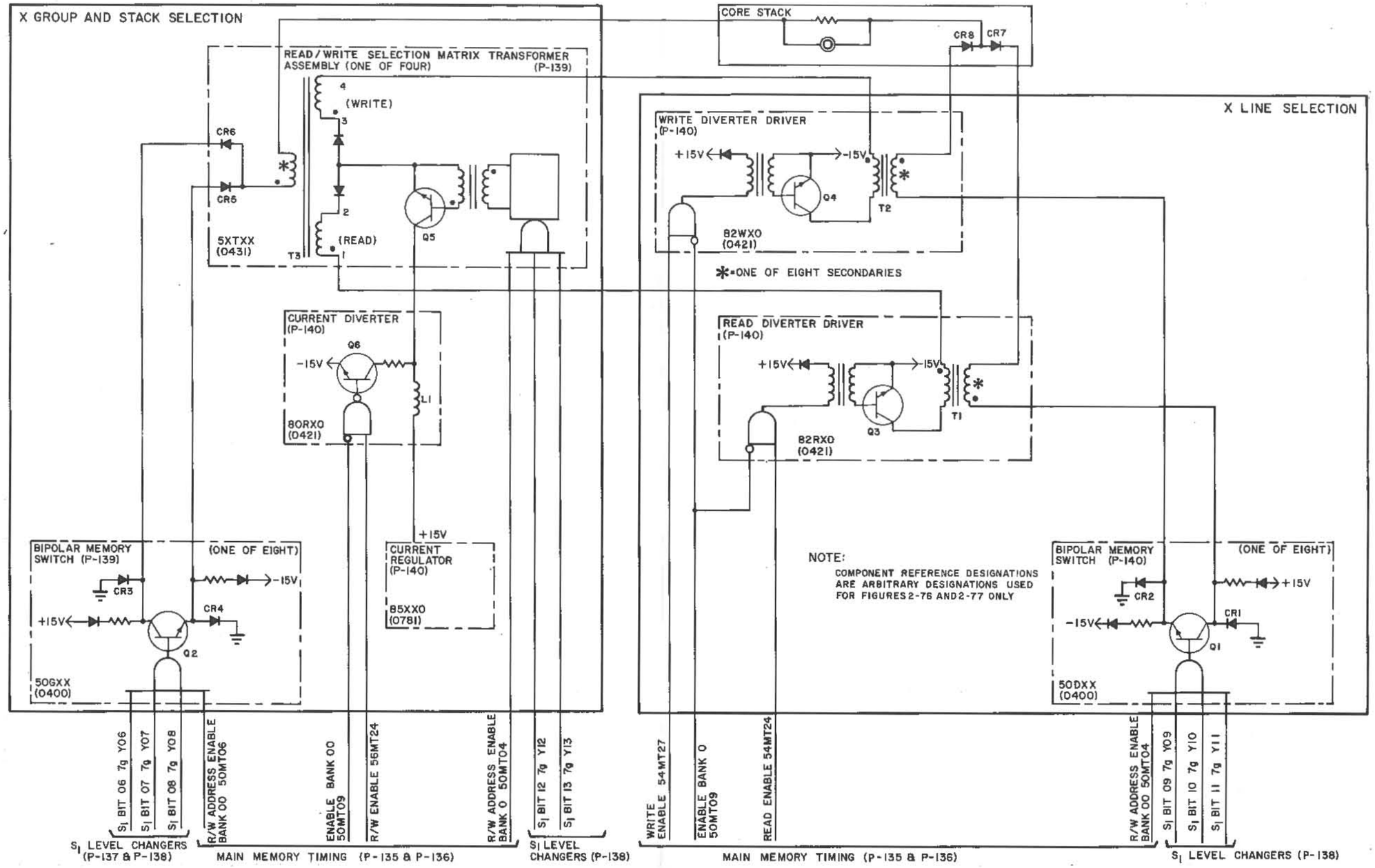


Figure 2-77. X and Y-Line Selection

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE 2-92. CURRENT DIVERTER DESIGNATORS

Bank	X-Line Diverter	Plate No.	Y-Line Diverter	Plate No.
00	80R00	P-140	81R00	P-143
01	80R10	P-148	81R10	P-151
02	80R20		81R20	
03	80R30		81R30	

TABLE 2-93. READ/WRITE SELECT TRANSFORMER DESIGNATORS

	Read				Write			
Bank	X-Line Trans.	Plate No.	Y-Line Trans.	Plate No.	X-Line Trans.	Plate No.	Y-Line Trans.	Plate No.
00	82R00	P-140	83R00	P-143	82W00	P-140	83W00	P-143
01	82R10	P-148	83R10	P-148	82W10	P-148	83W10	P-151
02	82R20		83R20		82W20		83W20	
03	82R30		83R30		82W30		83W30	

TABLE 2-94. CURRENT REGULATOR DESIGNATORS

Bank	X-Line Regulator	Plate No.	Y-Line Regulator	Plate No.
00	85X00	P-140	85Y00	P-143
01	85X10	P-148	85Y10	P-151
02	85X20		85Y20	
03	85X30		85Y30	

read-enable to 82R00 switches -15 volts through the driver primary of 82R00, through one of the primary windings of the 0431 transformer circuit for the selected stack (refer to plate P-139) and through a transistor switch in the 0431 card turned on by R/W Enable and S_1 bits 12 and 13 to the regulated +15 volts from 85X00 (refer to plate P-140). The actual current will depend upon stack temperature as determined by the heat-sensing resistors in the stacks. Read current is induced in the transformer secondaries in the proper direction to switch the 18 selected cores to the zero state.

2-353. Write current is produced in a similar manner. The write enable to 82W00 (refer to plate P-140) switch -15 volts through the driver primary of 82W00, through the opposite primary winding of the 0431 transformer circuit (refer to plate P-139) and the same transistor switch in the 0431 card used in the read operation to the regulated +15 volt supply (refer to plate P-140). Write current is induced in the transformer secondaries in the direction necessary to switch the selected cores to the 1 state.

2-354. The read/write operation for a Y-axis drive-line is similar to the X-axis operation. Refer to logic diagrams (plates P-142 and P-143) for Y-axis drive line operation in bank 0.

2-355. Sense/Inhibit Line Operation. Refer to figure 2-78 for an illustration of the sense/inhibit line wiring for a 4 X 4 (16 core) plane. The sense/inhibit line is actually two wires threaded through alternate rows of cores. At one end, the sense/inhibit wires are connected and go to one side of the inhibit driver transformer secondary (point A). The other ends of the lines (points B and C) are connected to the sense amplifier through a noise-cancelling balun transformer. These same ends are connected to the other side of the inhibit driver transformer secondary through an anti-balun transformer and a pair of diodes. In effect, the two lines are series-connected, as seen by the sense amplifier and parallel-connected, as seen by the inhibit driver.

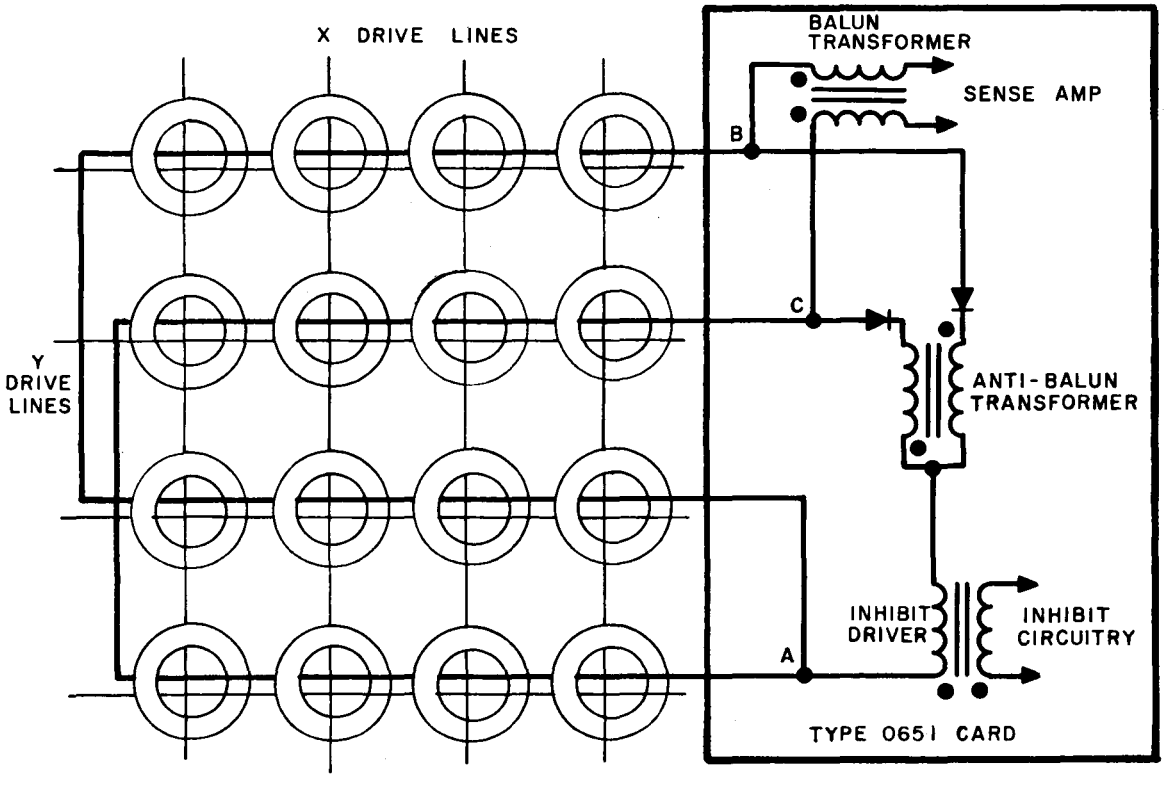


Figure 2-78. Typical Core Matrix

2-356. The sense/inhibit line-circuitry for a single bit in bank 0 is illustrated in figure 2-79. Eighteen type 0651 cards are required for each 16k bank of memory. Each card contains a sense amplifier, four inhibit drivers (one for each stack), and a Z_M-register flip-flop for one bit position in the bank. Each 16k bank incorporates a Z_M-register which is the storage register for information read from main memory. The heavy lines in figure 2-79 depict the current path for an inhibit pulse to a plane in stack one, and the current path for a sense pulse from a plane in stack two.

2-357. As shown in figure 2-79, overload flip-flop Z17 is set by an input on pin 52 prior to receiving an inhibit pulse. An inhibit pulse at input pin 40 will enable the Z4 AND gate for stack one, on condition that there is no power failure (pin 52 high) and the corresponding Z_M-register flip-flop 10 is not set. With Z4 (1) enabled, a negative pulse is generated through the primary of transformer T3 and current regulator Q10. The pulse induced in T3 secondary turns on transistor switch Q7, and completes a circuit from -15 volts through resistor R7, reactor T1, Q7, and transformer T7 to +6 volts. The pulse induced in the secondary of T7 is the inhibit current pulse that passes through anti-balun transformer T11 and diodes CR18 and CR19 to all cores on a single plane of stack one. This inhibit pulse occurs during a write cycle. Since it is in opposition to the write pulse current, it prevents the writing of a one, leaving the associated core in the zero state. The inhibit pulse current through T1 causes T1 to saturate and increase the voltage drop across R7. This voltage, inverted by transistor Q1, clears flip-flop Z17. With flip-flop Z17 cleared, AND gate Z4 is disabled and T3 turns off Q7 to block any further inhibit current to stack one. If Z_M-register flip-flop Z10 is set (contains a one) prior to receiving an inhibit pulse, the inverter on the one output of the flip-flop disables the Z4 AND gates and allows a one to be written into the selected core.

2-358. As previously described, the sense/inhibit line is actually two wires threaded through alternate rows of cores. Thus, the 4,096 cores on a single plane of a memory stack are arranged so that one sense wire passes through half of the cores and a second sense wire passes through the other half of the cores. During a read cycle, selected X and Y drive-lines are pulsed in the proper direction to switch the addressed core to a zero state. Because of this pulsing, a certain amount of noise is induced on both sense lines. If the addressed core (at the junction of X and Y-lines) already contains a zero, it will not switch states and the noise pulses on both lines will be identical. Because of the phasing of the windings on balun transformer T16 (common mode rejection transformer), these noises are cancelled before they reach the sense amplifier. If the core being sensed contains a one, it will switch to zero and a pulse will be induced on one of the sense lines. T16 receives different inputs and the relative difference is passed through the transformer as an input to the sense amplifier. This small voltage difference (of either polarity) is detected by the differential-type sense amplifier as a stored one. The output of the sense amplifier is gated to set the Z_M-register flip-flop, Z10, by the strobe bank 0 signal. The flip-flop, representing a single bit of the Z_M-register, stores the one that was read from memory.

2-359. Because the two windings of T16 are in the same direction (note polarity dots on the same side for both windings), the transformer appears as a high impedance when both inputs are the same polarity. When the two inputs are different, the transformer appears as a very low impedance and allows the difference to pass. Figure 2-80 shows the difference between two noise pulses (straight line and dashed line) at the moment

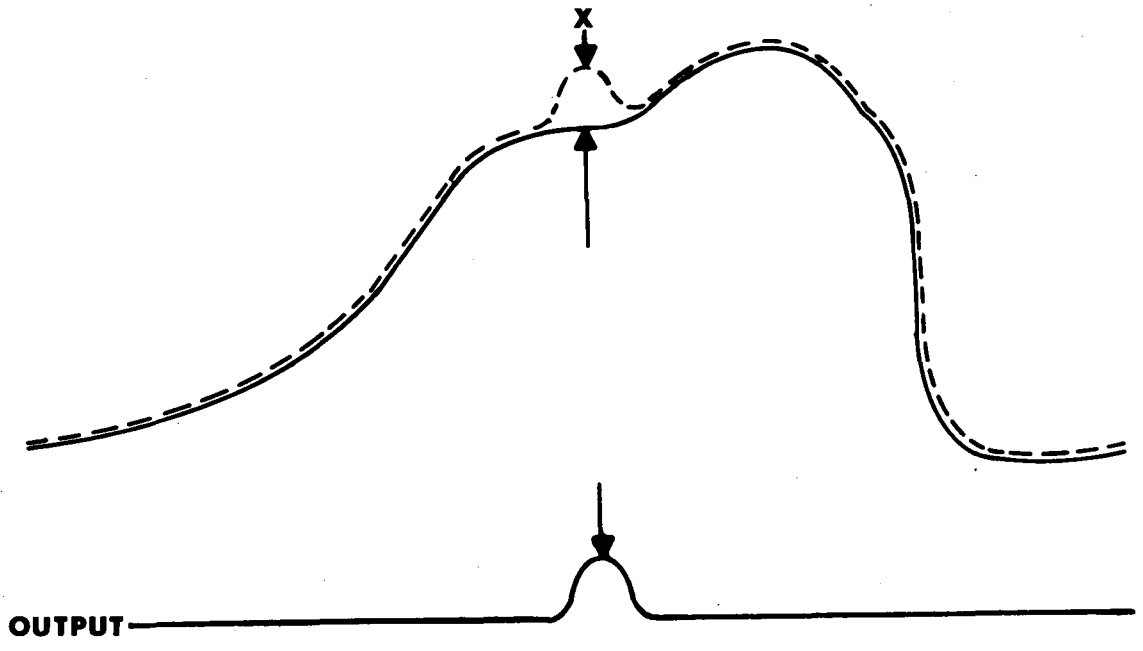


Figure 2-80. Signal Detection Through Rejection Transformer

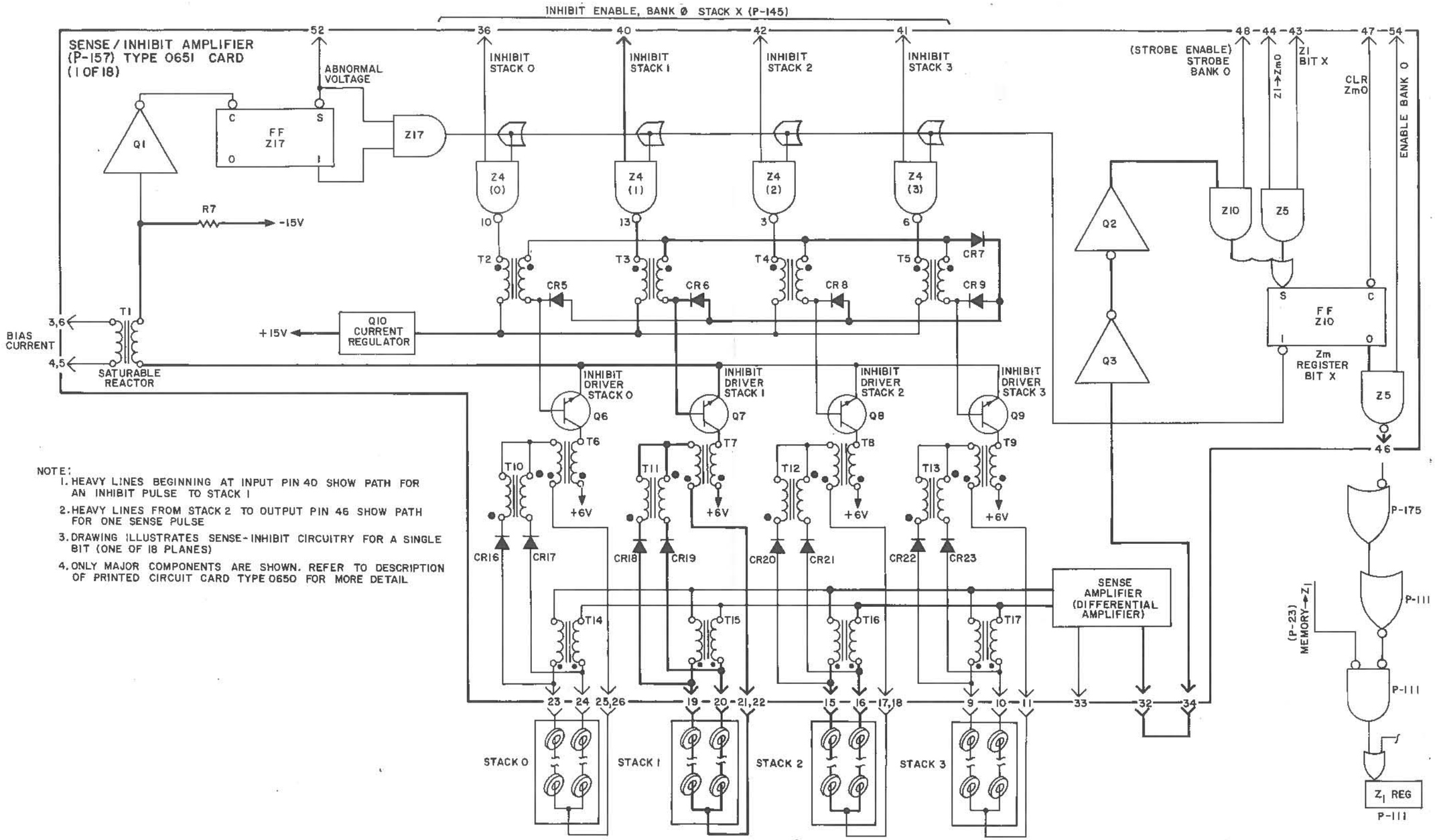
when the addressed core is switched (point X), and indicates the resulting transformer output.

2-360. A single 0651 module contains all circuits necessary for generating an inhibit-pulse, and for sensing and storing a sense-pulse for a single bit for all four stacks of a memory bank. Figure 2-81, along with associated logic diagrams, illustrates the current path for an inhibit pulse to a bit plane in stack one, and a sense pulse from a bit plane in stack two.

2-361. A negative pulse from 02VS00 (refer to plate P-145) is applied to pin 52 on the 0650 card (refer to plate P-157). This negative pulse sets flip-flop Z17 (refer to figure 2-81). When the input again goes positive, the flip-flop output is gated through an inverter-driver, to provide a partial enable for AND circuits Z4. The Z4 associated with current driver 71I00 is fully enabled by the Inhibit Stack 1 signal from 78I01 (refer to plate P-145). For the following description of current driver 71I00, refer to figure 2-81.

2-362. During inhibit current-drive operation, when Z4 is enabled, pin 13 of Z4 goes negative. This negative pulse passes through transformer T3 primary and transistor Q10 to +15 volts. Q10 and its associated bias resistors form a base drive current-limiter circuit for T3 primary. Transformer T3 provides a 1 to 3 current step-up to drive the base of transistor Q7. When Q7 is turned on, it completes the circuit from -15 volts, through R7, through saturable reactor T1, through Q7, and through the primary of transformer T7 to +6 volts. The current through T7 primary induces an inhibit pulse in T7 secondary which passes through T11 to all the cores in a single plane of the memory stack.

2-363. T7 is a 2 to 1 current step-up transformer which provides inhibit pulses to the



NOTE:
1. HEAVY LINES BEGINNING AT INPUT PIN 40 SHOW PATH FOR AN INHIBIT PULSE TO STACK 1
2. HEAVY LINES FROM STACK 2 TO OUTPUT PIN 46 SHOW PATH FOR ONE SENSE PULSE
3. DRAWING ILLUSTRATES SENSE-INHIBIT CIRCUITRY FOR A SINGLE BIT (ONE OF 18 PLANES)
4. ONLY MAJOR COMPONENTS ARE SHOWN. REFER TO DESCRIPTION OF PRINTED CIRCUIT CARD TYPE 0650 FOR MORE DETAIL

Figure 2-79. Sense-Inhibit Line Selections

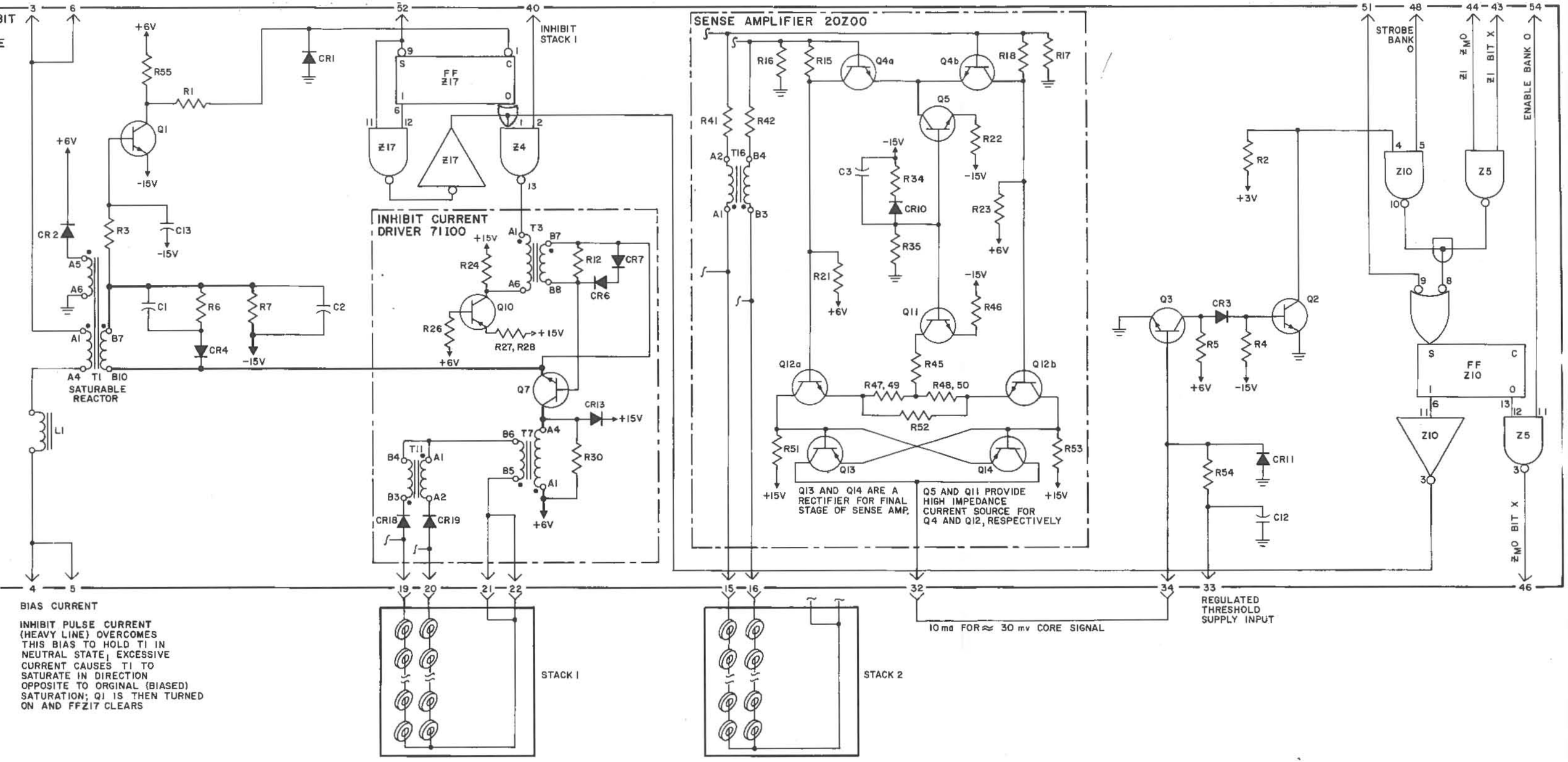


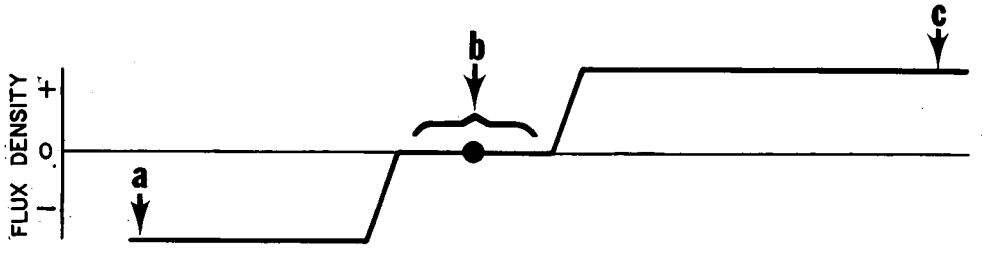
Figure 2-81. Sense-Inhibit Line Circuitry

THIS PAGE INTENTIONALLY LEFT BLANK

re circuits. R30 is a damping resistor for T7 and diode CR13 protects Q7 collector from excessive positive spikes at turn-off time. Diodes CR18 and CR19 are blocking diodes used to prevent oscillatory ringing in the inhibit loop and thus insure that secondary current returns to zero after each pulse. Transformer T11 is an anti-balun transformer that compensates for possible differences in voltage drops across the two diodes.

2-364. Saturable reactor T1, transistor Q1, and flip-flop Z17 monitor the inhibit pulse current and block the pulse when excessive current is detected. T1 will saturate when its internal magnetic flux density due to current flow in either winding exceeds a certain level; it will then present a low impedance to current flow. The bias current from pins A1 through A4 is supplied from 0781 current regulator cards (plate P-146) and is regulated further by diode pairs in 0660 cards connected across pins 5 and 6 of the 0651 card. The bias current produces a magnetic flux in opposition to the flux from the inhibit pulse current so that T1 is held below saturation (figure 2-82). The inhibit pulse current flows through R7, T1 pins B7 through B10, Q7, and T7. If the inhibit pulse duration is too long, or excessive current occurs because of a defective component, the flux from the inhibit pulse overcomes the bias and saturates T1. The voltage drop across R7 then increases enough to turn on Q1. The negative-going collector of Q1 clears flip-flop Z17. The output of Z17 disables the current drivers to block further inhibit current to the cores.

2-365. The current drivers may also be disabled by inverter-driver Z10 (the inverter-driver from the set side of Z_M flip-flop) (refer to Figure 2-81 and plate P-157) which produces a negative output when Z_M-register flip-flop Z10 contains a one. This negative voltage level disables Z4 and prevents the inhibit pulse from being generated. In this manner, a one may be written into the selected core during a write cycle. The sense/inhibit line from pin 19 passes through half of the 4,096 cores on one memory plane in stack 1 and returns to pins 21 and 22. A second sense/inhibit line from pin 20 passes through the other half of the cores and also returns to pins 21 and 22.



BIAS CURRENT FROM PIN A1 TO A4 PRODUCES MAGNETIC FLUX IN ONE DIRECTION (POINT a ON CURVE).
INHIBIT-PULSE CURRENT FROM PIN B7 TO B10 COUNTERACTS BIAS CURRENT, AND HOLDS T1 OUT OF SATURATION (POINT b ON CURVE). FLUX FROM EXCESSIVE CURRENT FROM PIN B7 TO B10 CAUSES T1 TO SATURATE IN A DIRECTION OPPOSITE TO BIASED DIRECTION (POINT c ON CURVE).

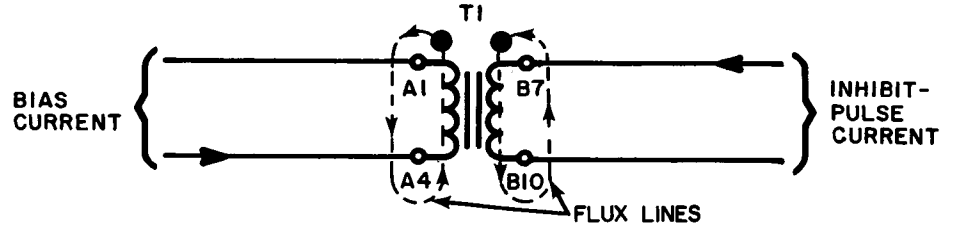


Figure 2-82. Operation of Saturable Reactor T1

2-366. When an X-line read pulse and a Y-line read pulse occurs in memory stack 2, noise pulses are induced on lines A2X4 and A2Y4 which enter the associated 0650 card at pins 15 and 16, respectively (refer to plate P-157). For the following discussion of sense amplifier operation, refer to figure 2-81.

2-367. During sense amplifier operation, the two sense lines at pins 15 and 16 go directly to common mode rejection transformer T16. The manner by which this transformer produces a voltage difference for a detected one has been discussed in paragraph 2-358. Any voltage difference, regardless of its polarity, is amplified by a two-stage differential amplifier consisting of Q4 a and b, and Q12 a and b. Resistors R15 and R18 are regenerative-feedback resistors for the first stage of the differential amplifier. They lower the differential input impedance so that four quadrants can be terminated into one amplifier.

2-368. Transistor Q5 provides a high-impedance constant current source for Q4, and transistor Q11 provides a high-impedance constant current source for Q12. A high-impedance current source aids in common-mode rejection of the differential amplifier. Transistors Q13 and Q14 comprise a rectifier with a positive output for the final stage of the sense amplifier, enabling detection of differential signals of either polarity. A regulated negative threshold supply is connected to pin 33. The value of this threshold supply, and the value of R54, determines the point at which Q3 begins to conduct.

2-369. Main Memory Timing. Main memory has a timing source separate from other computer timing. A delay line is initiated by the main timing chain to generate read and write pulses. A single main-memory timing operation is performed to read a single word from memory or to write a single word into memory. The first half of the operation (read cycle) controls reading a word from memory, and the second half (write cycle) controls writing a word into memory.

2-370. To read a word from memory, the word is taken out of memory and stored in the Z_1-register during the read cycle. During the write cycle, the word is restored to its original location in memory. To write a different word into memory, the read cycle is performed, but the word is not stored in Z_1. Thus, the selected memory location is cleared. The word to be stored is placed in Z_1-register and loaded into memory during the write cycle.

2-371. Refer to timing diagram, figure 2-[illegible], and associated logic diagrams for the following discussion of main memory timing. The initiate memory signal is generated by flip-flop 0XG80 (refer to plate P-134). This flip-flop is set during time T1.1 and is cleared during time T2.3, resulting in a set time of 750 nanoseconds. The L Initiate Mem is first sent to 74YG2 (refer to plate P-136. The initiate memory signal is applied to the 1-microsecond delay line 50MT21 (refer to plate P-135), via amplifie. 50MT20, to start the memory timing cycle. The negative-going leading edge of the initiate memory signal starts the read cycle, and the positive-going trailing edge starts the write cycle. Outputs from delay line 50MT21 (refer to plate P-135) are gated to provide the necessary sequential control signals for memory operation.

2-372. The read cycle requires R/W addre enable, read enable, and R/W enable signa to generate X-line and Y-line pulses through the selected address (refer to figure 2-76). The write cycle requires R/W address enable, write enable, and R/W enable signals.

2-373. Address decoding circuitry (refer plate P-134) provides bank selection gating

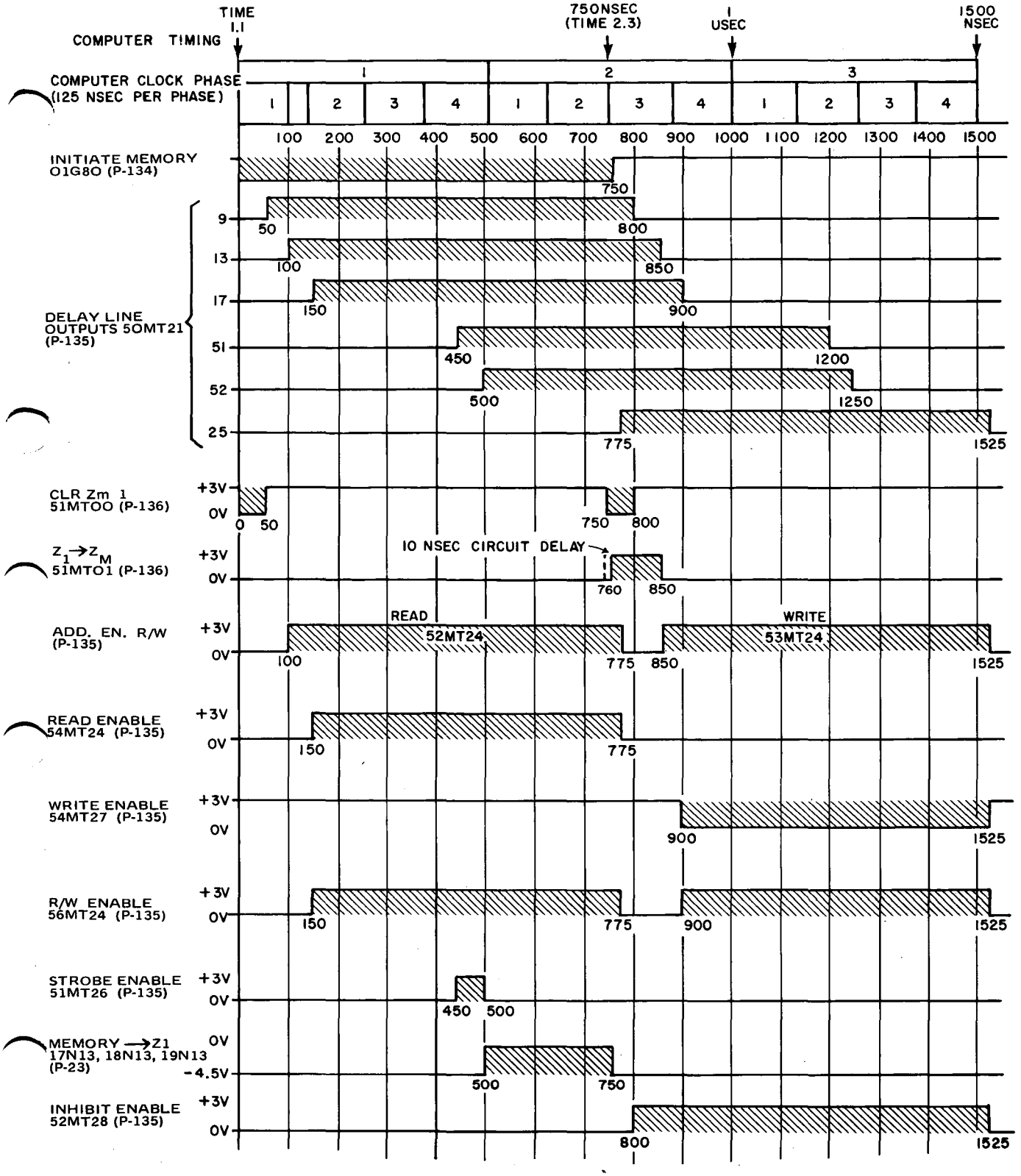


Figure 2-83. Main Memory Timing

the R/W address enable gates on plate P-136. These gates (50MT04, 50MT06, 51MT04, 51MT06) are fully enabled by address enable (read) or address enable (write) from plate P-135. Read enable, write enable, and R/W enable signals are provided by the delay-line decoding circuitry shown on plate P-135. A clear Z_M pulse is generated during the first 50 nanoseconds of both read and write cycles to clear the Z_M-register. This is accomplished as follows. (Refer to the timing diagram figure 2-83 and plates P-135 and P-136. The negative-going leading edge of the initiate memory pulse provides a partial enable to pin 10 of 50MT00 via 50MT20 and 50MT22, and disables pin 18 of 50MT00 via 50MT22. At this time, pin 9 of delay-line 50MT21 is low, pin 10 of 50MT00 is enable by the high from 50MT23, and 50MT00 produces a clear Z_M signal. After 50 nS, pin 9 of delay-line 50MT21 goes high, disabling 50MT00 via 50MT23. When the initiate memory signal goes high, pin 18 of 50MT00 receives an enable via 50MT20 and 51MT22. The left AND gate of 50MT00 is disabled by the high on pin 10 from 50MT22. Pin 11 of 50MT00 has a high at this time from the output of delay-line 50MT21 pin 9 via 50MT23 and 51MT23. The bank select signal on pin 16 of 50MT00 will enable the gate to produce the clear Z_M signal. After 50 nS, pin 9 of delay-line 50MT21 goes low, putting a low on 50MT00 pin 11 via 50MT23 and 51MT23 and disabling the clear Z_M signal.

2-374. During a read operation, the strobe enable pulse from 51MT26, generated by delay-line decoding (refer to plate P-135), is gated by bank enable signals (refer to plate P-136) and sent to the associated Z_M-register via 50MT05 and 50MT03. In this way, the read pulses induced in the sense lines and detected by the sense amplifiers are strobed into the Z_M-registers (refer to plates P-157 through P-165 and P-166 through P-174) for the selected bank. Z_M-register outputs are then routed through OR circuit level changes (refer to plate P-175) to the Z_1-register (refer to plates P-111 through P-113). The memory Z_1 signal necessary to transfer the read data from the Z_M-register to the Z_1-register is generated at time T2.1 (refer to plate P-23). If the programmed operation being performed is a write operation, the memory Z_1 signal is blocked. The conditions necessary to block this signal are shown on plate P-23. When 12N13 or 13N13 is enabled, its high level outputs disable AND gates 18N13 and 17N13, respectively. If any of the three input AND gates to 10N13 are fully enabled, the high output from 10N13 will disable AND gate 11N13. In this manner, memory data is not written into Z_1. Instead, data selected for memory storage is gated to the Z_1-register by one of the input gates (refer to plate P-111).

2-375. To gate Z_1 to Z_M, the positive output from delay-line 50MT21 pin 13 100 nS after the initiate memory pulse partially enables gate 10MT00 through 50MT25 and 51MT25. (See plates P-135 and P-136 and timing diagram figure 2-83). The positive-going trailing edge of the initiate memory signal at 750 nS (figure 2-83) satisfies gate 10MT00 via 50MT20, 50MT22, and 51MT22. These three gates provide a 10 nS circuit delay so that the clear Z_M pulse, which is also initiated by the initiate memory trailing edge, can clear the Z_M-register before the Z_1-register contents are gated in. 10MT00 enables 50MT01 and 51MT01 (Plate P-136) along with the proper bank select signal, to generate the $Z_1 \rightarrow Z_M$ signal. Each bit position of the Z_M-register that contains a zero is gated within the 0651 cards (figure 2-81) to its associated sense inhibit line as an inhibit pulse. The signal that accomplishes this originates with the inhibit enable signal (refer to plate P-135). This signal is gated by the associated bank select signal (refer to plate P-136) to the selected stack enable

(refer to plate P-145) to enable the associated inhibit current driver (see pin 40, figure 2-81).

2-376. CONTROL MEMORY. Control memory is a random-access, word-oriented core memory with a 500-nanosecond cycle time and 300-nanosecond (maximum) access time. The basic control memory has 128 eighteen-bit storage locations addressable from 00000 through 00077 and 00200 through 00277. Additional memory can be installed to provide 256 word locations. The storage elements use two ferrite cores for each bit position of each word location. (One core is magnetized in a particular direction to store a binary one while the other core is magnetized in the same direction to store a binary zero.) In extracting a word from memory, the read operation is destructive, i.e., word information is no longer retained. Thus, a write operation must necessarily follow a read operation to restore the information read-out or to store new information. The word line current used to write a word is opposite to that used for reading. Most of the control memory locations are assigned a special purpose, such as real time clock, index registers, and input/output address control words. Because information is more readily available for operations utilizing this information, the speed of the main computer is increased. However, no time is saved if control memory locations are referenced directly by the program.

2-377. Core Pair Operation. Refer to figure 2-84 for the following discussion of core pair operation involving one core pair for one particular bit position. The first drawing (a) illustrates the effect of the read operation upon the cores. Notice that both cores are magnetized in the same direction by the word line read current. The read operation is illustrated at this time only to show the cores being set to a common state.

2-378. Figure 2-84 (b) and (c) illustrates the method of writing a binary one or zero into the same core pairs. A second current-carrying line (digit-line) is introduced in these drawings. During a write operation, both digit-line and write current flows. The write current is always in the same direction (opposite that for reading). Current in the digit-line, however, is made to flow in either direction, in one direction to write a zero, and in the opposite direction to a write one. The effect of the digit-line current with word line current is to cause a switch in magnetization of one of the cores in the pair without affecting the other. The amplitude of the word-line current is sufficient to cause the cores to switch states. The function of the digit-line current is to negate the word-line current in one of the cores to prevent it from switching, while allowing the other core to be switched by the word-line current. In this way, the digit-line current direction determines the binary value written in the core pair.

2-379. Figure 2-84 (d) again shows the read operation. The digit-line is not used. The read current flows in the opposite direction to the write current and causes both cores to become magnetized in the same direction. One of the cores will already be magnetized in this direction so it will not switch. The other core will switch to this direction and the corresponding change in the magnetic lines of force will induce a voltage in the sense-line. The sense-line is used only during the read operation. The direction of current flow in the sense-lines depends on which core is switched. The direction of sense-line current indicates whether the core pair stored a binary one or zero. The sense-line feeds a polarity sensitive amplifier which can discriminate between a binary one and zero.

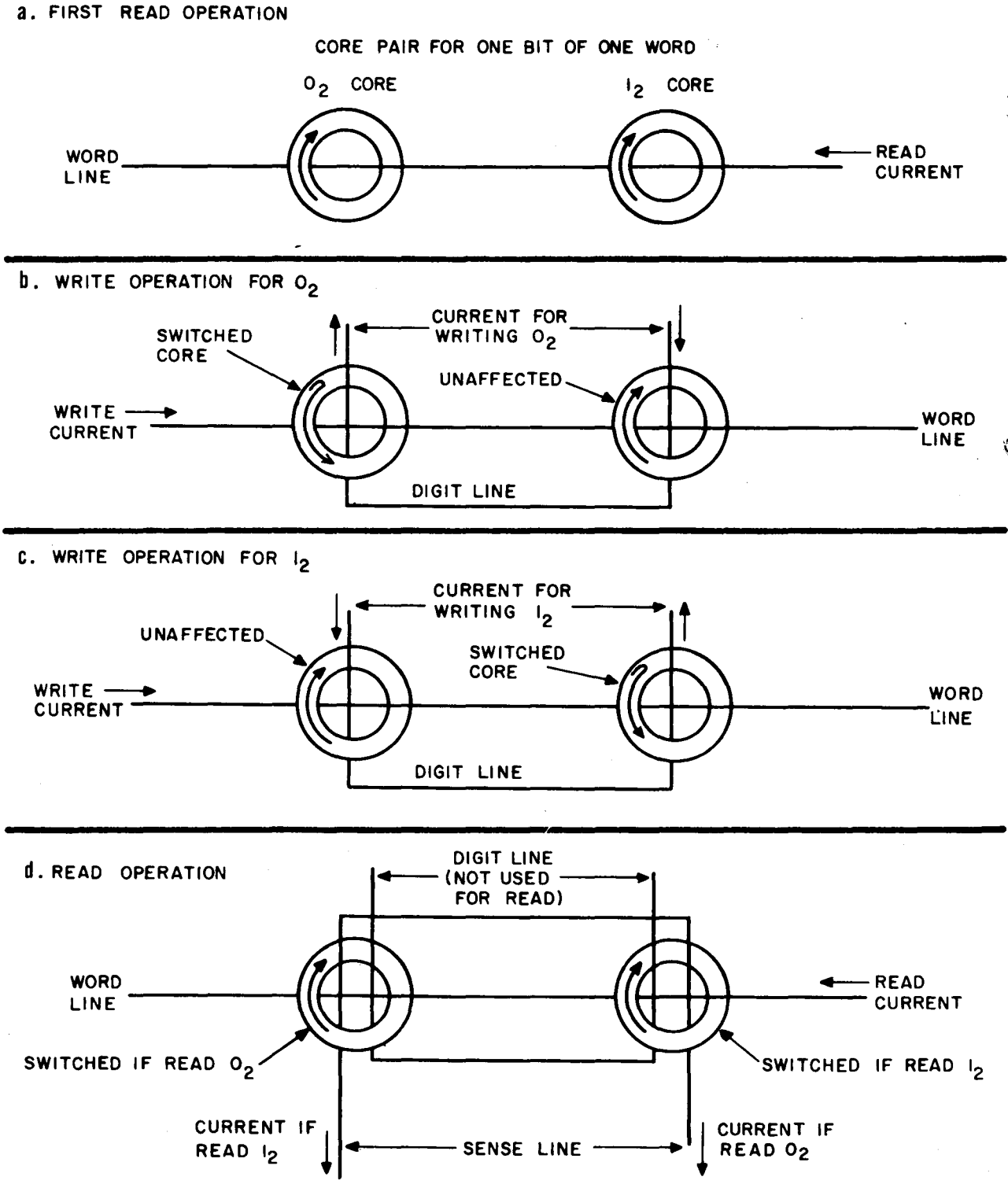


Figure 2-84. Control Memory Core Pair Operation

2-380. Physical Arrangement. Refer to figure 2-85 for an illustration showing control memory arrangement. There is a separate word-line for each of the 128 eighteen-bit word locations. Thus, each word location has its own word-line which passes through all of its 18 core pairs. When a word line is activated with current, the associated 18-bit word location is selected and no other cores are affected. Each word-line is actually threaded twice through its associated core pairs to increase the effect of the word-line current. (Notice that each of the sense and digit-lines passes through all 128 core pairs for a particular bit position.)

2-381. Control memory is separated into two sections of 64 locations each. Between these sections, the sense-lines are crossed to minimize induced noise voltages. This sense-line reversal causes the position of the 1_2 and 0_2 cores to be reversed in the two sections. That is, the left-most core of any core-pair in the lower section is the 0_2 core; but, in the upper section, the left-most core of any pair is the 1_2 core. Since there is no such crossing of the digit-lines between the two sections, the writing of a 1_2 in the upper section requires digit-line current to flow in the direction opposite to that required for writing a 1_2 in the lower section. Thus, the direction of digit-line current is not only dependent upon the binary value to be written, but also upon the storage address.

2-382. Control Memory Addressing. Control memory has circuitry to select one of 128 word drive lines during each memory control reference. Each word-line affects all 18 core pairs of only one memory location. The selected word drive lines carry current in one direction for approximately 80 nanoseconds during the read cycle, and in the opposite direction for approximately 120 nanoseconds during the write operation. There are 18 digit-lines, one for each bit position common to every address. The direction of digit drive line current determines the binary value written in the core pair by negating the effect of the write word-line current through one core, while allowing the other core to be switched. This current direction is dependent upon the binary value of the bit being stored as well as the storage address selected. Digit-lines are activated during the write cycle only.

2-383. The 128 word-lines are considered to be in 16 groups of eight lines each. Word selection is made by enabling one group and one line of that group. Line selection is also referred to as diode selection. Actually, the sixteen groups are divided into two sections of eight groups per section. Group selection is subdivided into group end-board selection (one of two) and group-pair selection (one of eight). The group end-board selection circuits enable one of two sections and the group-pair selection circuits enable one of eight groups within that section. Line (diode) selection is subdivided into diode end-board selection (one of two) and diode-pair selection (one of eight). Diode end-board selection is available as a provision for subsequently expanding control memory to a capacity of 256 words. To provide the additional 128 locations, a second core plane must be added.

2-384. The 8-bit SO-register is the address register for control memory. It receives addresses from the S_1-register, the index control register, the I/O translator, or the externally specified index. When a control memory address is detected in SO, the bit translation is decoded into group and line information. Refer to figure 2-86 for an illustration of SO-register bit translation when addressing control memory. A complete control memory word-line array

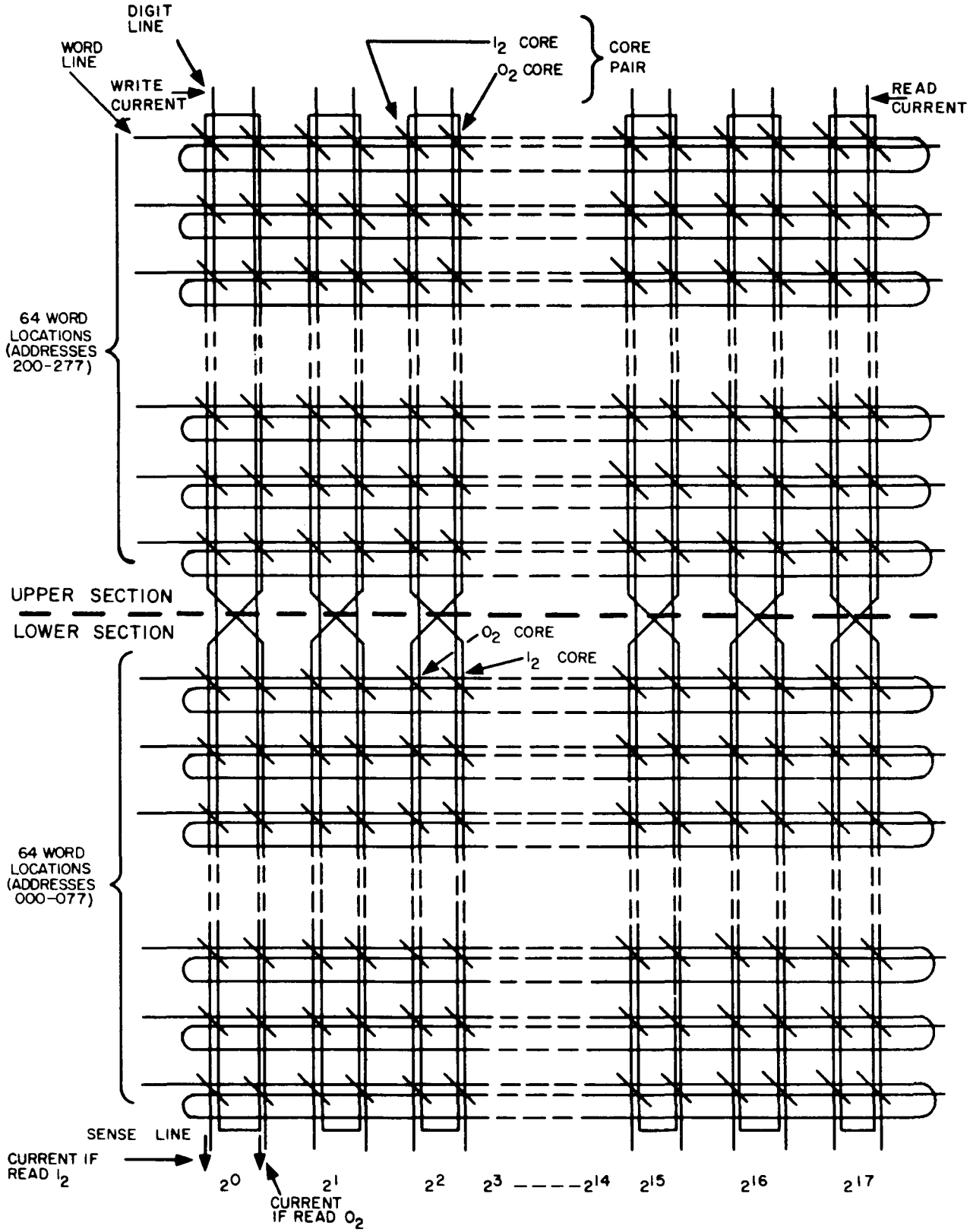
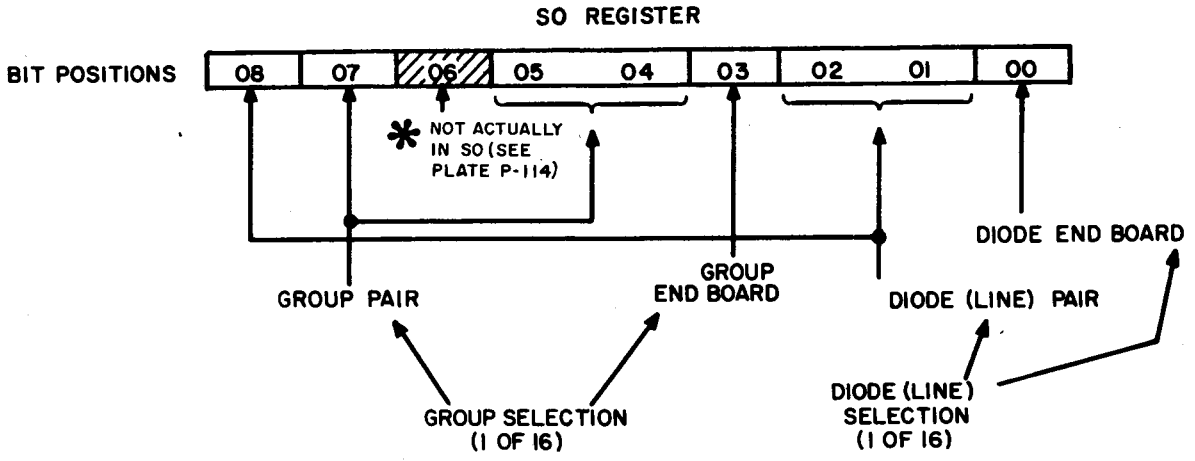


Figure 2-85. Control Memory Arrangement For 128 Word Locations



*BINARY BIT 06 IS NOT INCLUDED IN SO BECAUSE IT WOULD INDICATE ADDRESSES IN THE 100's, 300's, 500's, OR 700's. NONE OF THESE ADDRESSES ARE IN CONTROL MEMORY; THEREFORE, EXCEPTING ADDRESSES IN THE 500's, THEY ARE NOT ADDRESSED THROUGH THE SO REGISTER. ADDRESSES IN THE 500's ARE DISCUSSED UNDER THE SECTION ON BOOTSTRAP MEMORY.

Figure 2-86. Control Memory Address Translation From SO

is illustrated in figure 2-87. The direction of current flow and the SO translation levels indicated are for the read operation. The write operation performs the same selection, except that the SO selection levels are reversed and the resulting current through the selected word-line is in the direction opposite to that for read. As shown, group selection is made on the side where the lines are joined to form 16 groups. Line selection is performed at the other end. Each line is diode-coupled to the line-selection termination, which prevents any other current flow through the selected line. The bottom section does not exist for the 128-word memory used in the computer at present. It is illustrated here to show the necessary arrangement for providing an additional 128 word locations.

2-385. To further clarify word line-selection, refer to figure 2-88. This figure is a partial diagram showing only the read circuits for 64 lines of control memory. As shown in plate P-124 and figure 2-88, group selection is made by the 20WG0X and 10WG0X series gates. The enabling of a group is done by forward biasing the corresponding transistor switch (Q16 through Q2). This applies a ground level to the group. Selection of the transistor switch is made by current flow from the group end-board circuit through the associated transformer primary to the group-pair circuit. Diode (line) selection is performed in a similar manner with the 20WD0X and 10WD0X series gates. The selected transistor switch (Q31 through Q17) is forward biased and applies a positive voltage to the line from 00WR00 read current diverter. Word-line current flow is from group selection to line selection for the read operation. The write circuits are not shown. Separate transistor switches and transformers for the write operation cause word-line current flow in the opposite direction.

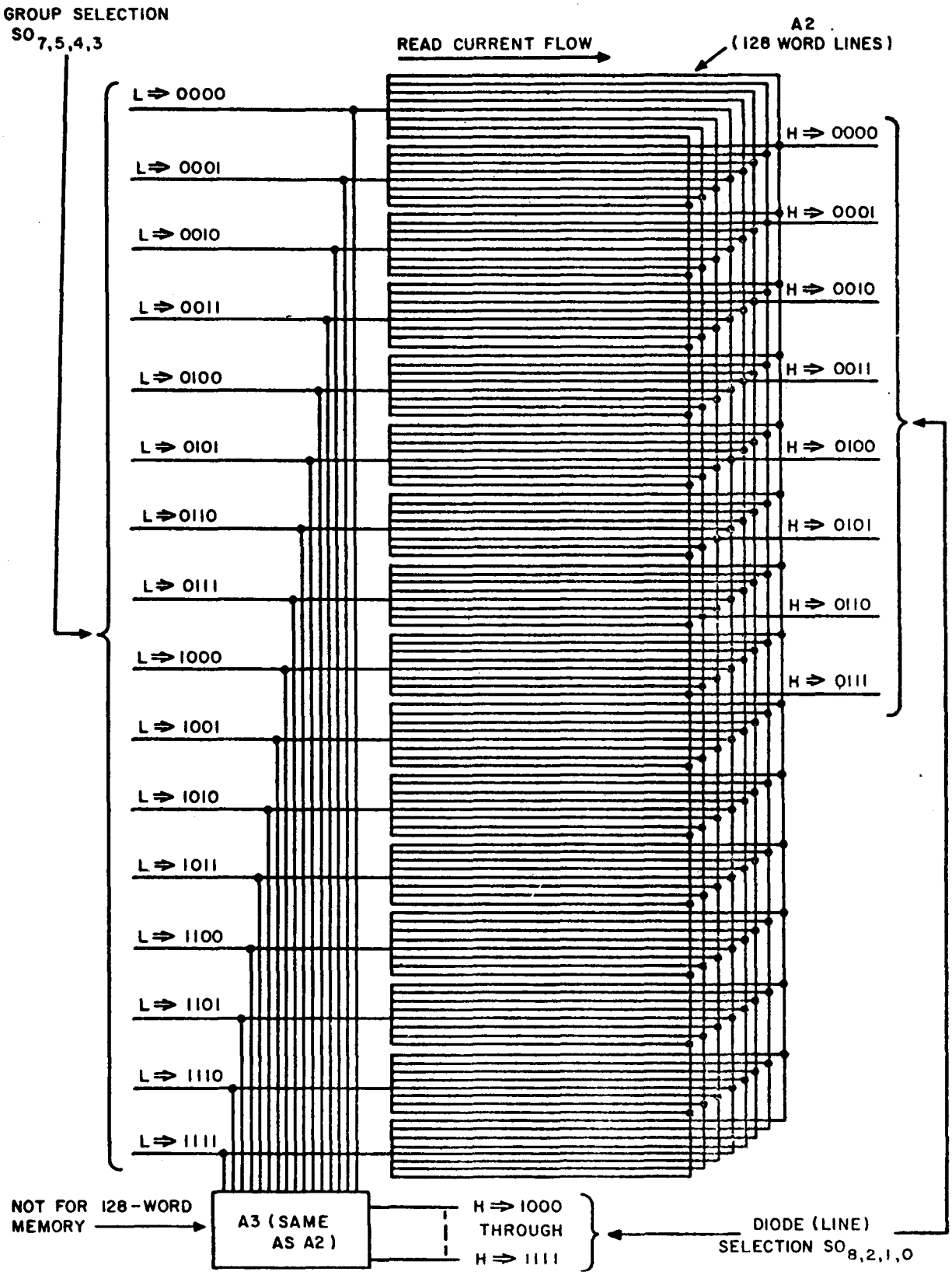


Figure 2-87. Simplified Diagram Of Complete Control Memory Word Line Array (Read Operation)

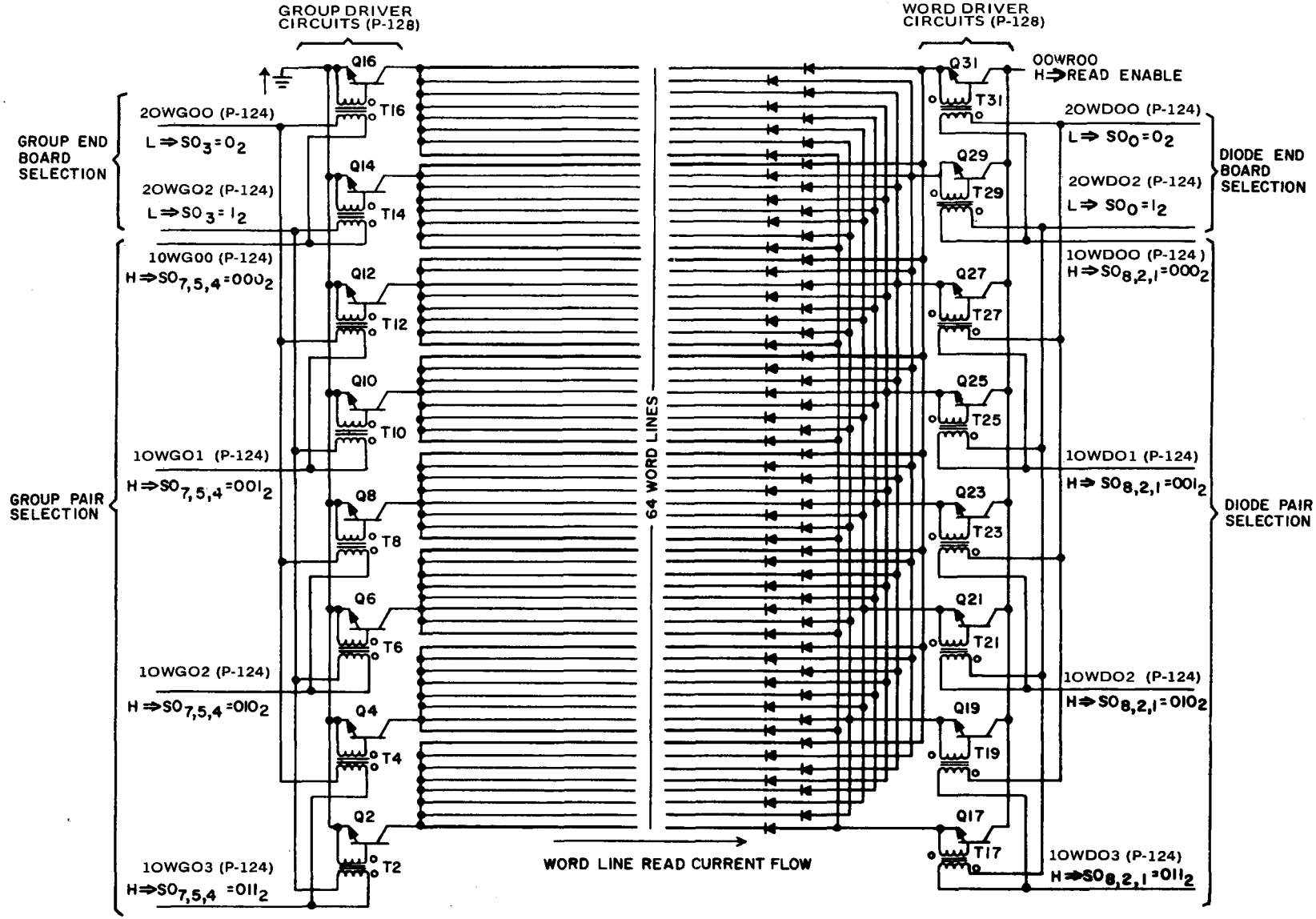


Figure 2-88. Detailed Diagram Of Partial Control Memory Word Line Connection (Read Operation)

2-386. Information in the SO-register (refer to plate P-114) is routed through the SO-register slaves (refer to plate P-123) to the control memory word translation circuits (refer to plate P-124). When a control memory address is translated, 20WG00 through 20WG03 select one of two group end-boards, depending upon the status of bit-3 in the SO-register. Circuits 20WG00 and 20WG02 are used during the read cycle, while circuits 20WG01 and 20WG03 are used during the write cycle. Circuits 10WG00 through 10WG07 select one of eight word groups, depending upon the status of bits 7, 5, and 4 of the SO-register.

2-387. When a particular group has been selected, the 20WG0X gate applies -15 volts to one side of the associated group transformer (refer to plate P-128), while the 10WG0X gate applies a relatively positive (-4.5 volts) voltage to the other side of the group transformer. The voltage induced in the secondary of the group transformer enables one of the group transistor select switches (refer to plate P-128) which applies either a ground level to the selected group (refer to plate P-127) for a read operation, or a +15 volts for a write operation.

2-388. Circuits 20WD00 through 20WD03 (refer to plate P-124) select one of two diode end-boards, depending on the status of bit-0 in the SO-register. Circuits 20WD00 and 20WD02 are used during the read cycle, while 20WD01 and 20WD03 are used during the write cycle. End-board selection is provided in case it is subsequently desired to expand control memory to a capacity of 256 words. At present, SO bit-8 will always contain a zero. Circuits 10WD00 through 10WD07 select one of eight word-lines, depending upon the status of bits 8, 2, and 1 of the SO-register. When a line has been selected, the 20WD0X gate applies -15 volts to one side of the associated diode (line) transformer (refer to plate P-128) while the 10WD0X gate applies a relatively positive (-4.5 volts) voltage to the other side of the line transformer. The voltage induced in the secondary of the line transformer enables one of the transistor select switches (refer to plate P-128) which applies either a ground level to the selected line (refer to plate P-127) for a write operation, or a +15 volt level for a read operation.

2-389. The source of word-line current is the +15 volt supply regulator shown on plate P-134. This voltage is routed directly to the current diverters shown on plate P-124. The read current diverter is 00WR00 and the write current diverter is 00WW00. These act as switches which are normally closed; that is, they supply a ground level load for the regulated +15 volts. When either a read-enable or write-enable signal occurs, the respective switch is open and the load is removed. With the ground level removed, the selected word-line becomes the major load for the regulated voltage supply. The two potentiometers, A2R37 and A2R38 (refer to plate P-124), are for read-and-write word-current adjustments.

2-390. Selected Circuit Analysis. A simplified illustration of the complete path for read current, when address 000 has been selected, is shown on figure 2-89. When group end-board 0 ($SO_3 = 0_2$) and group-pair 0 ($SO_{7,\ 5,\ 4,} = 000_2$) are selected, 20WG00 and 10WG00 cause current flow through the T16 group transformer primary in the direction shown. The resulting secondary voltage forward-biases transistor Q16, which applies a ground level to one end of the selected group of word-lines. Assuming diode end-board 0 and diode pair

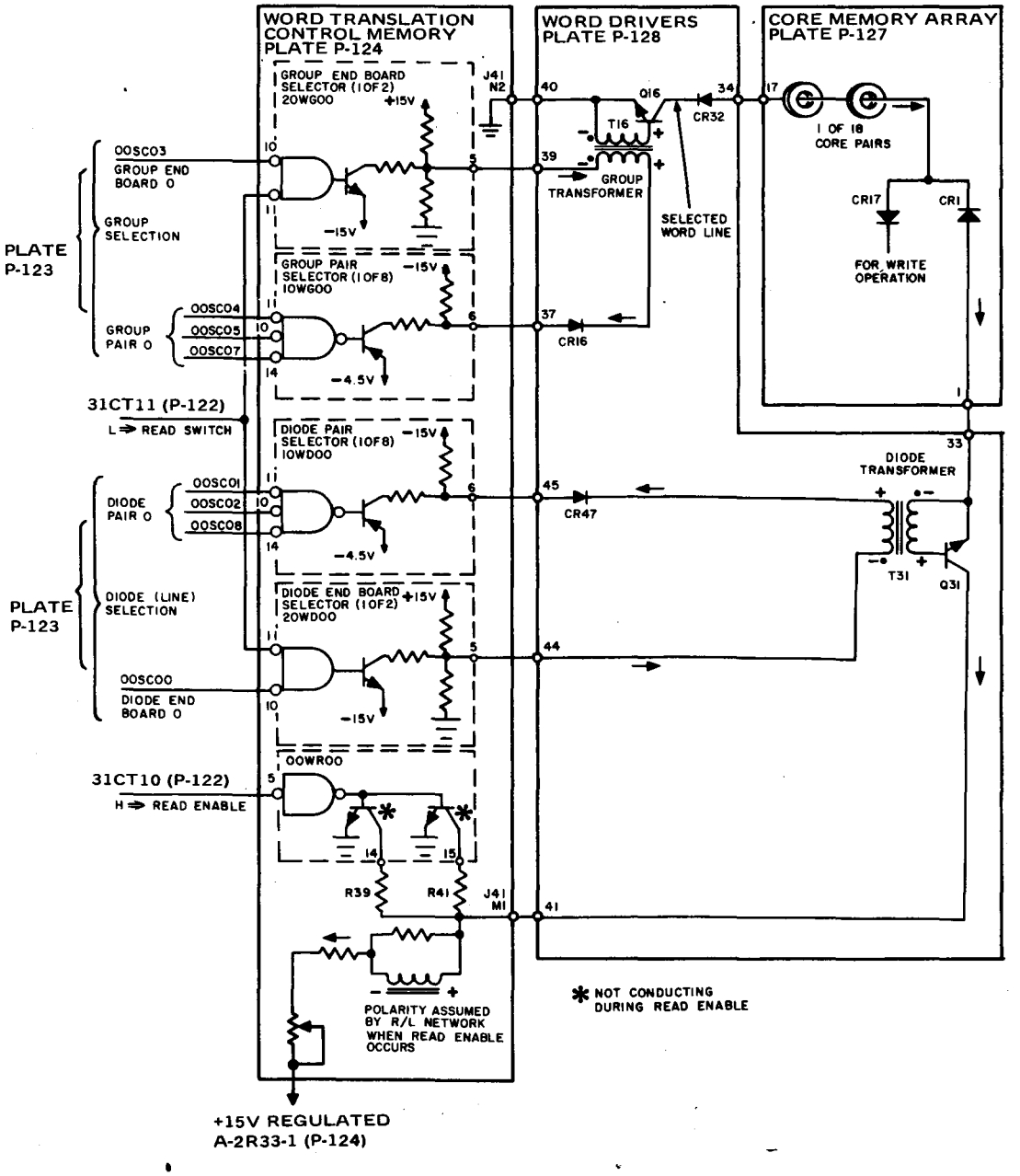


Figure 2-89. Simplified Diagram Of Control Word Drive Circuitry For Read Operation (SO = 000_8)

0 are selected, 10WD00 and 20WD00 cause current flow through the diode transformer primary in the direction shown. The resulting secondary voltage forward-biases transistor Q31, which completes the connection of the word-line to the current diverter 00WR00 and the +15 volt supply.

2-391. In the absence of the read-enable signal, the two output transistors of 00WR00 are conducting to provide a ground load and cause current to flow through R39 and R40 to the +15 volt supply. The occurrence of the read enable cuts off the output transistors of 00WR00. With the load removed, current flows to the +15 volt supply through the selected word-line in the direction shown (refer to figure 2-89).

2-392. Digit-Line Current. Digit-line current flows only during the write operation. There are 18 digit-lines, one for each bit. Each line intersects all 128 word-lines through the core-pairs associated with its particular bit. Only 18 core-pairs, threaded by the selected (current carrying) word-line, are affected by the current. The direction of digit-line current for writing a binary one is opposite that for writing a zero. The digit-line drivers are controlled by the contents of the ZO-register. Also, since the position of the 1 and 0 cores are reversed between each 64 word section, digit-line current for each function (writing a 1 or a 0) must be reversed for each section. Thus, digit-line driver control is also affected by bit-7 in the SO-register. Refer to plates P-129 and P-130 for the following discussion of digit-line selection and current direction, and to plate P-125 for the functional interconnection schematic.

2-393. An analysis of SO-register contents shows that bit-7 contains a zero when the lower section is being addressed (addresses 000-077), and a one when the upper section is being addressed (addresses 200-277). The bit-7 outputs are routed from the SO-register (refer to plate P-114), through the SO-register slaves (refer to plate P-123) to control memory-timing gates (refer to P-122). At the proper time, an enable signal for either the upper section or lower section, is gated to the digit drivers (refer to plates P-129 and P-130), depending upon the status of bit-7 in the SO-register. There are two digit drivers associated with each digit-line. Both bit-related drivers are on the same 3780 type card. When enabled, the drivers act as switches between their output pins. Thus, the 00DDXX circuits of the series connect their associated digit-lines to a negative regulated voltage source (pins 12 to 13) and the 10DDXX series circuits connect their digit-lines to a positive regulated voltage source (pins 14 to 1

2-394. The digit-driver output lines are threaded through their associated core pairs (refer to plate P-127) and return to ground via the driver circuits (refer to plates P-129 and P-130). The connections are through J41 and P1. Thus, the 00DDXX circuits cause current to flow out of pin 13, through the digit-lines to ground, while the 10DDXX circuits cause current to flow from ground, through the digit-lines, to pin 14. The following analysis of the input gating on plates P-129 and P-130 will show that the direction of current flow is dependent upon the contents of the ZO-register and upon the section of memory being addressed (as determined by the status of bit-7 in the SO-registe

2-395. When ZO-register bit-0 contains a zero and the storage address is in the lower section (000-077), 00DD00 is enabled. Current will flow from the -15 volt regulated

supply, through 00DD00 (pins 12 to 13), through the core pairs for control memory bit-0, and back to ground. (Refer to simplified diagram, figure 2-90.) If ZO bit-0 is a binary one and the storage address the same, 10DD00 is enabled and current will flow from ground, through the core pairs for bit-0, through 10DD00 (pins 14 to 15), to the +15 volt regulated supply. A similar analysis of addresses in the upper section (200-277) will show that the current flow generated for each function (writing a binary one or zero) is opposite that just described.

2-396. Sense Line Output. The sense-lines (refer to plate P-131) feed directly from the control memory stack into sense amplifiers 20SA00 through 20SA17. The sense amplifiers must be polarity-sensitive since the direction of current flow induced in the sense-line determines the binary value of the bit being read. Using 20SA00 as an example: a positive voltage applied to pin 15, with respect to pin 14, indicates a binary one has been read from memory. The output will be a positive level in this case. When a zero is read from memory, a positive (with respect to pin 15) voltage is applied to pin 14. This causes an open (relatively low level) output. Output sense amplifier 21SA00 inverts and amplifies the 20SA00 output for application to the ZO-register. Sense amplifier 20SA00 also performs a holding function by continuing to output the indication of the bit value read for a short period of time. This allows the read word-line current to be dropped before the read strobe signal occurs, to gate the output of the 21SAXX series circuits to the ZO-register (refer to plates P-115, P-116, and P-117).

2-397. Control Memory Timing. When a control or bootstrap memory address has been translated from the S1-register, 91S00 (refer to plate P-107) outputs a low. At time T2.1 of the proper sequence, 39N10 via 30N10 and 38N10 (refer to plate P-24) develops the S1 → SO command enable to set the 0XDT10 flip-flop (refer to plate P-122) and start control memory timing. Control memory and bootstrap memory use different portions of the same timing chain. Sinse a bootstrap address has not been translated, the Bootstrap flip-flop (refer to plate P-107) will not be set. When this flip-flop is clear, its output disables bootstrap and enables the control-memory portion of the timing chain (refer to plate P-122). Also, 10N00 (refer to plate P-107) is disabled by 91500 via 92500 to allow the ZO → Z Sel rather than the Z1 → Z Sel command enable. Refer to figure 2-91 for a block diagram description of the I-sequence operations used to obtain an instruction word from control memory.

2-398. A control memory reference can also be initiated by 19N10, 29N10 and 49N10 (refer to plate P-24). These gates enable I/O Translator → SO, ESI → SO, or ICR → SO, respectively. They also initiate control memory timing by setting the 0XDT10 flip-flop (refer to plate P-122).

2-399. If a store instruction uses a control memory address, the control memory cycle is used instead of main memory. Figure 2-92 is a block diagram description of the W-sequence operations for a store AL (f = 46, 47) in control memory. The Z_1 → ZO enable noted in figure 2-92 originates at 10N13 (refer to plate P-23), which outputs a high during time T2.3 of a W-sequence when f = 46 or 47. This high is inverted by 28N11 (refer to plate P-25) to partially enable 29N11. 29N11 is fully enabled by the output of 20N11 when both outputs from 31EF02 and 30EF04 (refer to plate P-122) are low. This occurs during the first 100 nanoseconds of the control-memory cycle.

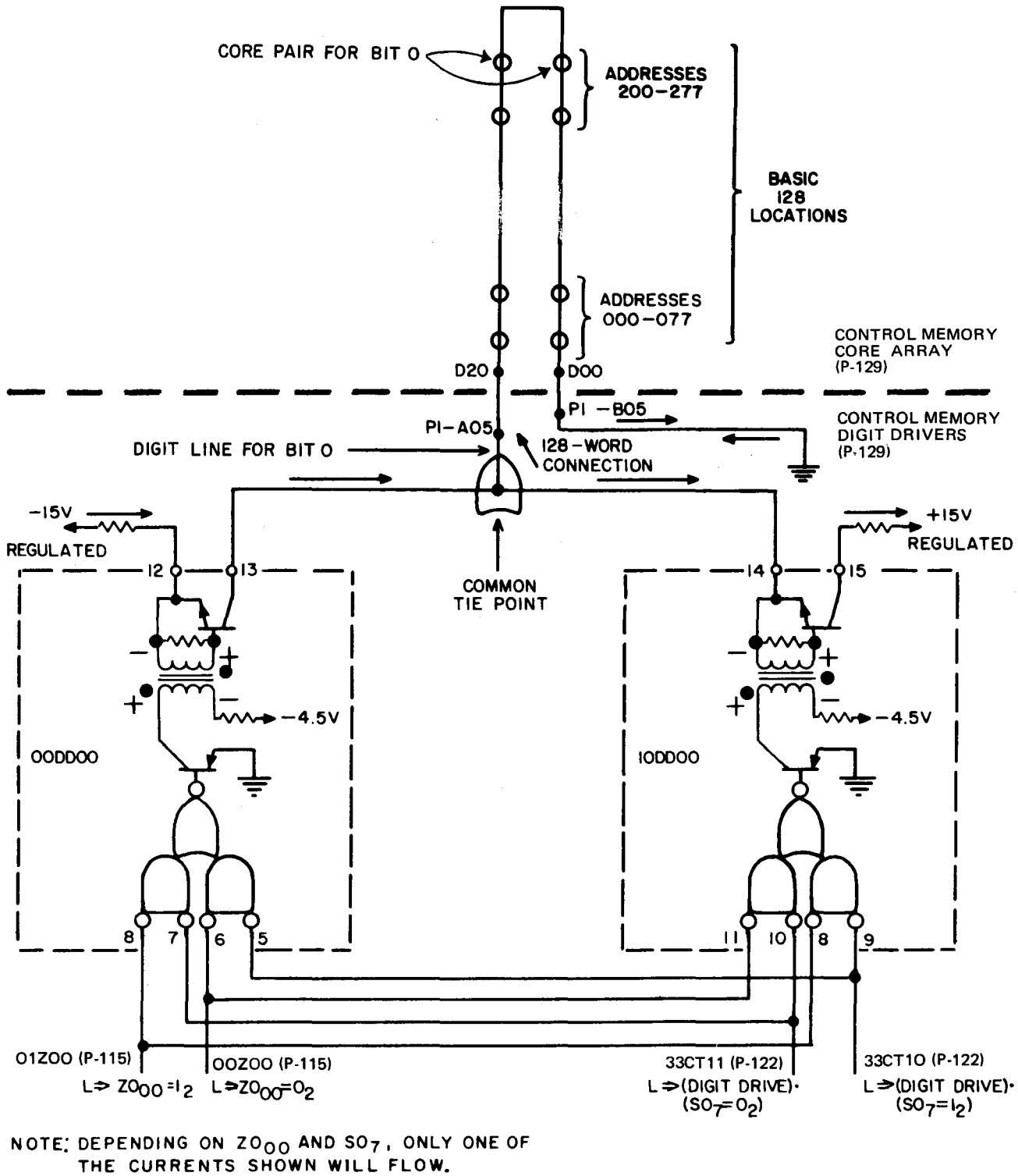
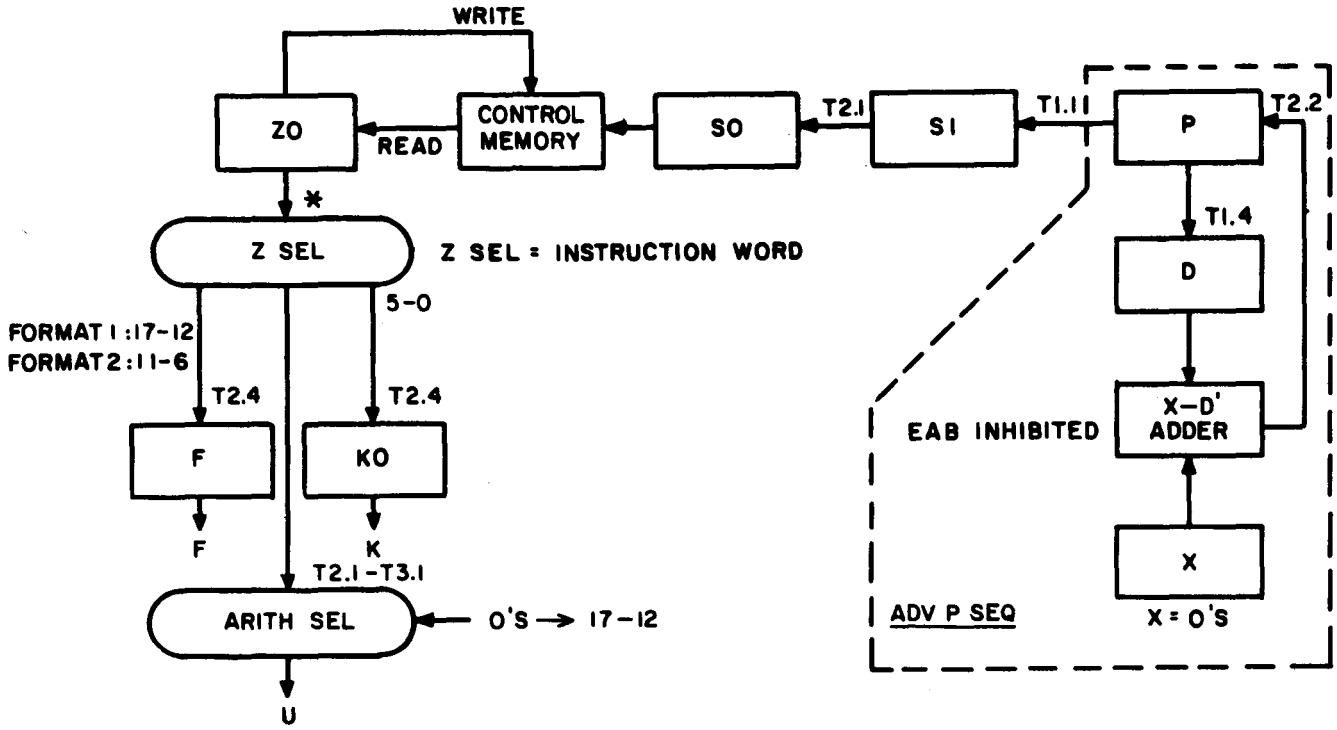
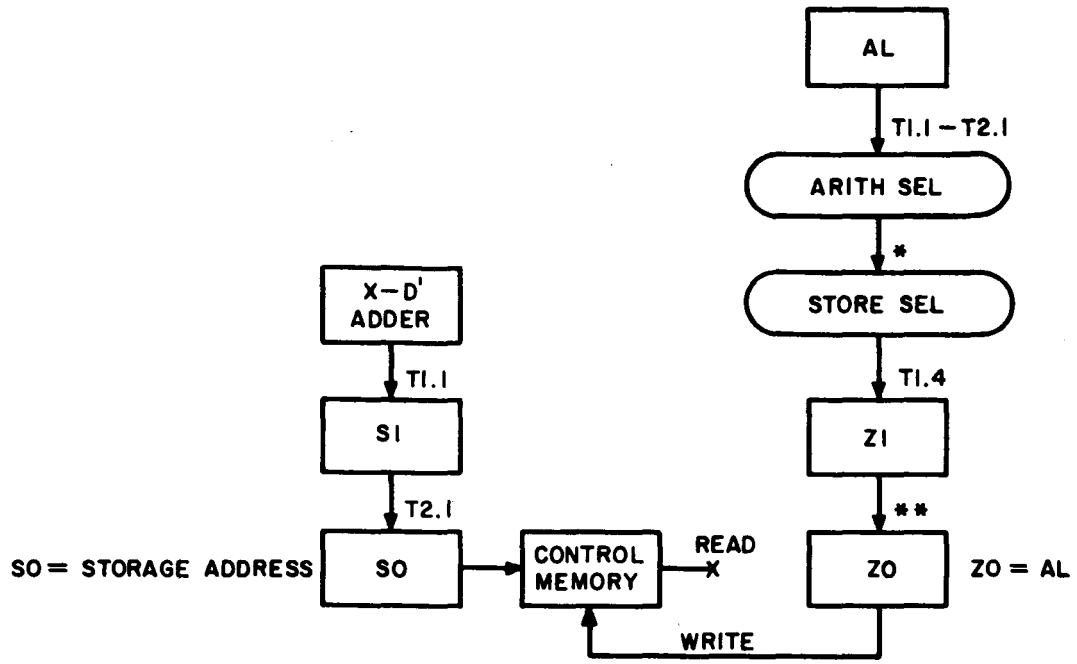


Figure 2-90. Simplified Diagram Of Control Memory Digit Line Circuitry (Bit Zero)



NOTE: * ZO → Z SEL OCCURS AS LONG AS SI = CM ADDRESS.

Figure 2-91. I-Sequence Data Flow Referencing Control Memory



NOTES: *ARITH SEL → STORE SEL OCCURS FOR ENTIRE W-SEQUENCE.
**ZI → ZO IS TIMED BY CONTROL MEMORY TIMING

Figure 2-92. W-Sequence Data Flow For F=46, 47 With Storage Address In Control Memory

2-400. Figure 2-93 illustrates the timing waveforms developed during a control-memory reference. (Refer to plate P-122 for timing logic.) Timing is initiated with the setting of 0XDT10 flip-flop. When this flip-flop sets, a low signal from 02DT10 propagates down delay lines 01DY10 and 01DY11, producing low pulses at 20-nanosecond intervals. The pulses provide enables and disables to the gates on plate P-122 to generate the timing waveforms shown on figure 2-93. After 120 nanoseconds, the input from 02DT10 is switched to a high by clearing flip-flop 0XDT10 from tap 11 of delay line 01DY10 via 30EF04 and 31EF04. A complete control-memory cycle is approximately 500 nanoseconds, with an access time of approximately 250 nanoseconds. Since all of the signals which set the 0XDT10 flip-flop are gated by phase 1, any control-memory reference is started at phase 1, and its cycle is completed before the occurrence of the next phase 1.

2-401. BOOTSTRAP MEMORY. Bootstrap memory has 32 storage locations for 18-bit words. The program contained therein is hard-wired to provide a short-load routine for a particular peripheral input device, such as a tape reader or a magnetic tape transport, and for automatic reloading of the program if the computer senses an invalid operation. Maximum access time is 300 nanoseconds. However, when bootstrap is referenced, the effective cycle time is 2 microseconds.

2-402. Storage Element. As storage elements, bootstrap memory used ferromagnetic core transformers. There are eighteen transformers, one for each bit position. Each transformer is shared by all 32 addresses. The transformer secondary winding is the sense-line. The primary winding is simply a current-carrying drive line. If a one is desired in a particular bit position, the drive line is threaded through the associated core to produce secondary current. If a zero is desired, the drive line bypasses the core. The sense-line (secondary winding) carries no current for the zero configuration. Refer to figure 2-94 for an illustration of one core transformer showing both the one and zero configurations.

2-403. Readout Operation. The bootstrap addresses are 00500_8 through 00537_8. There is one word-drive-line for each of the thirty-two, 18-bit addresses. During each bootstrap memory reference, the word-line corresponding to the selected address is activated with current for approximately 120 nanoseconds. Each core transformer through which the line passes develops a secondary voltage in the sense-line. This voltage represents a one readout of bootstrap. In this manner, each word-line determines the binary content of its address by the pattern in which it is threaded through and around the cores. Refer to figure 2-95 for an example showing the configuration for the first two bootstrap addresses. Only two of the 32 word-drive-lines (addresses) are shown. Notice that the sense-lines are gated directly to the ZO-register. The magnitude of the sense-line voltage is sufficient to set the ZO flip-flop without amplification. The -4.5 voltage and diodes are used to prevent the sense-line voltage from rising more negative than -4.5 volts. Since the read operation is not destructive to the information contained in bootstrap memory, no write operation is necessary.

2-404. Physical Arrangement. The entire memory is contained in one plug-in module comprised of four cards. The module is located in chassis 8. The transformer cores are attached to three of these cards and are threaded serially by the 32-word drive-lines according to the binary content.

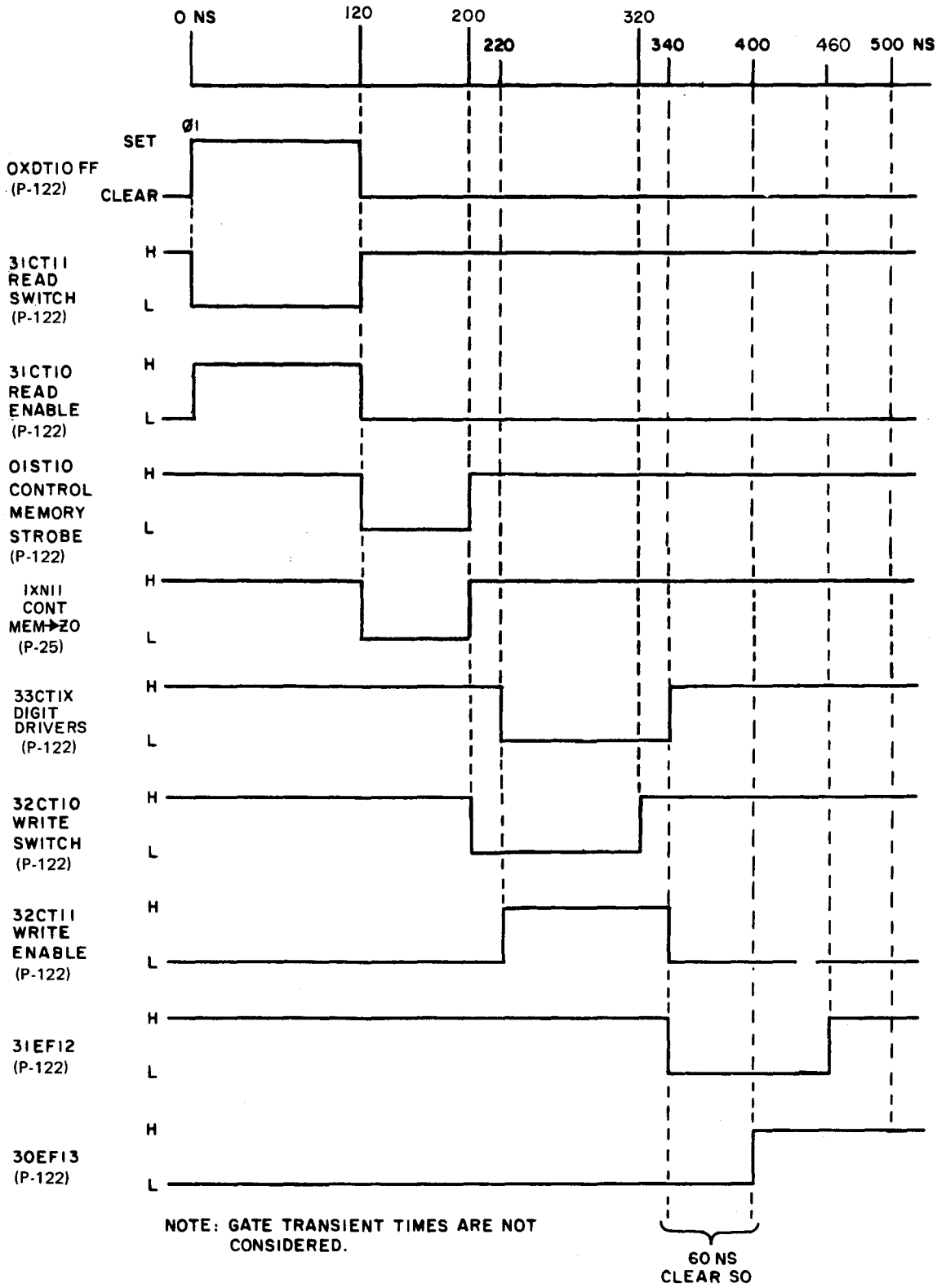


Figure 2-93. Control Memory Timing Waveforms

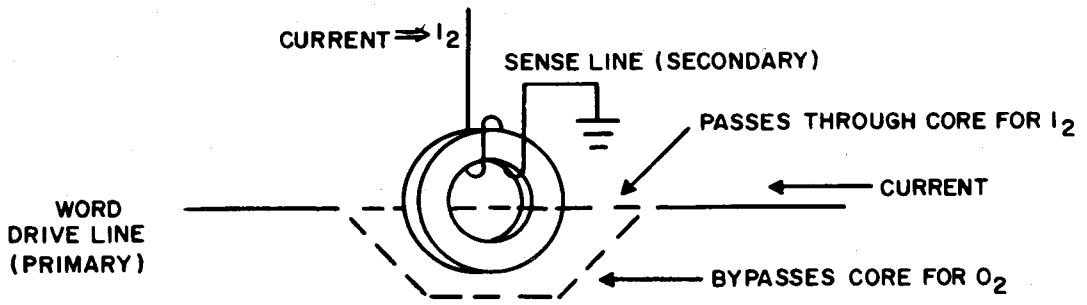


Figure 2-94. Core Transformer

The bootstrap load routine can be changed by inserting a different module.

2-405. Group and Line Selection. The drive-lines are considered to be in eight groups of four lines each. Drive-line selection is made by enabling one group and one line of that group. The bootstrap address is placed in SO, and the bit configuration of SO is translated into group and line. Refer to figure 2-96 for the translation of SO.

2-406. Refer to figure 2-97 for a simplified diagram of the drive-line array. The lines are connected at one side to form eight groups of four lines per group. Group selection is made by the 00BG0X series gates. The selected group of lines has a positive voltage applied from the 00BG0X gate. The connection at the other end forms the line selection. One of the 10BDX0 series gates is enabled to select one of the four lines by applying a negative voltage. The resulting current flow is from the line selection gate, through the diode, through the selected word-drive-line, which passes through or around the 18 core transformers, and into the group selection gate. The diodes prevent more than one word-line from carrying current.

2-407. The logic diagram for bootstrap memory word selection is shown on plate P-132. Gates 00BG00 through 00BG07 provide group selection. The common input enable bootstrap is a timing signal with a pulse duration of approximately 180 nanoseconds. This signal originates from the bootstrap portion of control memory timing (refer to plate P-122). The other inputs to the group-select gates are SO_{2-0} from the SO-register slaves (refer to plate P-123). When a bootstrap address has been decoded, one of the gates is fully enabled. When enabled, the gate acts as a switch between its output pins and applies +15 volts to one side of the selected group of drive-lines.

2-408. Gates 10BD00 through 10BD30 perform the line selections. These gates are timed by the same enable bootstrap signal as the group selectors. The other inputs are SO_{3-4}. When one of these gates is enabled, it switches -15 volts to the other side of the selected line. Refer to simplified circuit diagram, figure 2-98 and logic diagrams, plates P-132 and P-133, for the following discussion of drive-line current flow for address 500.

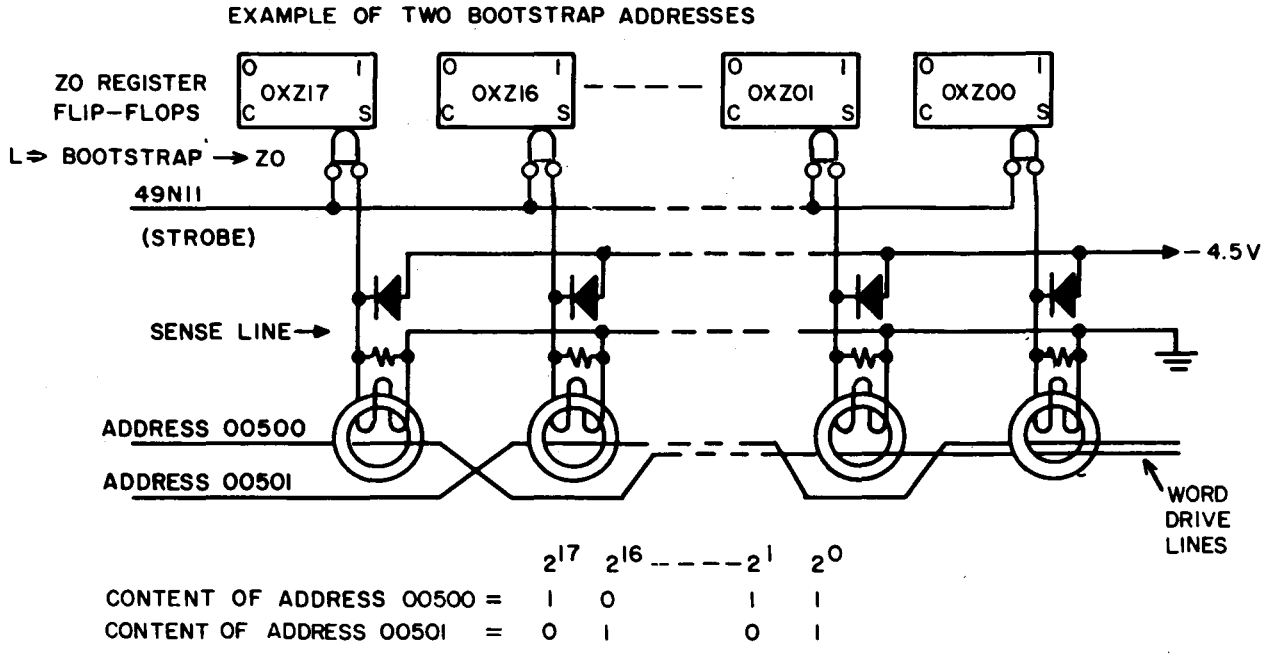
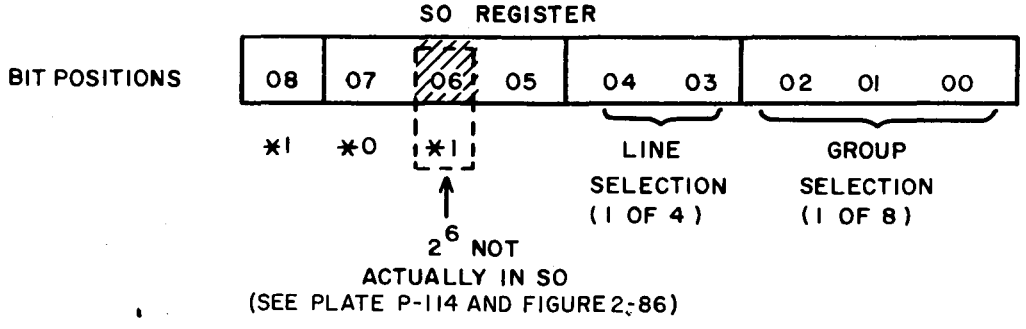


Figure 2-95. Example Of Two Bootstrap Addresses



NOTE: * THESE DIGITS ARE CONSTANTS OF THE BOOTSTRAP ADDRESSES ($500_8 - 537_8$)

Figure 2-96. Bootstrap Address Translation From SO

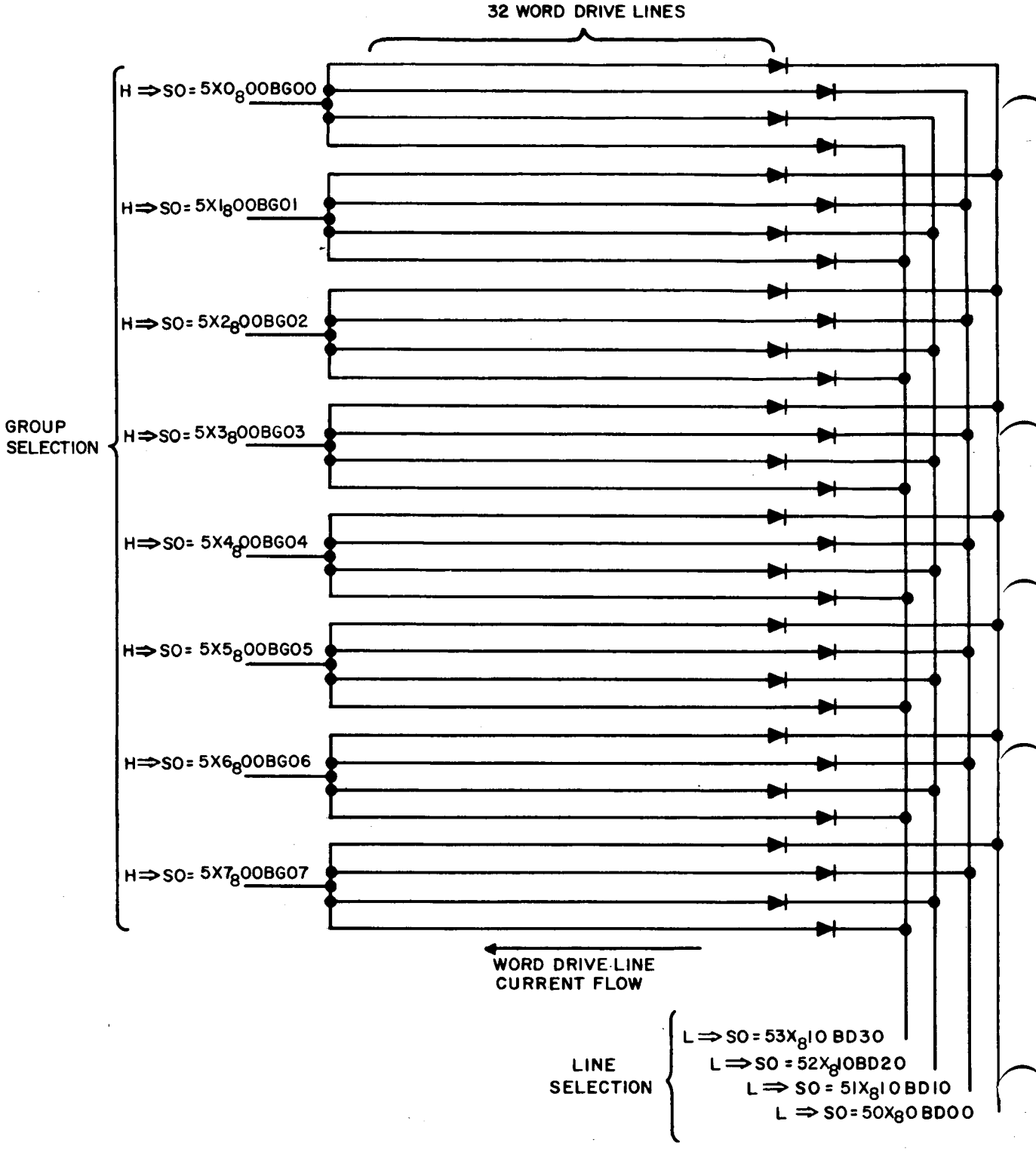


Figure 2-97. Simplified Diagrams Of Bootstrap Drive Lines

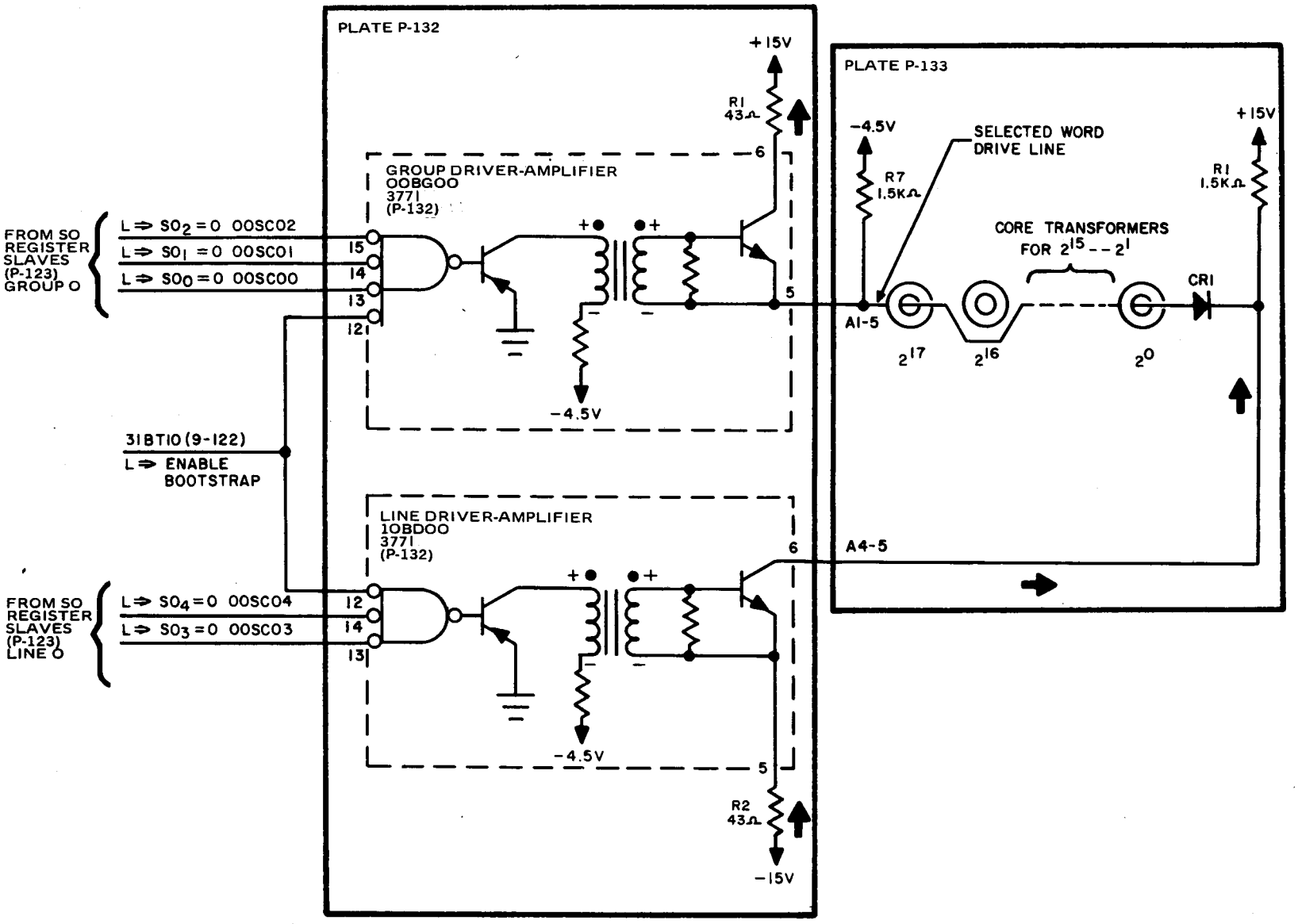


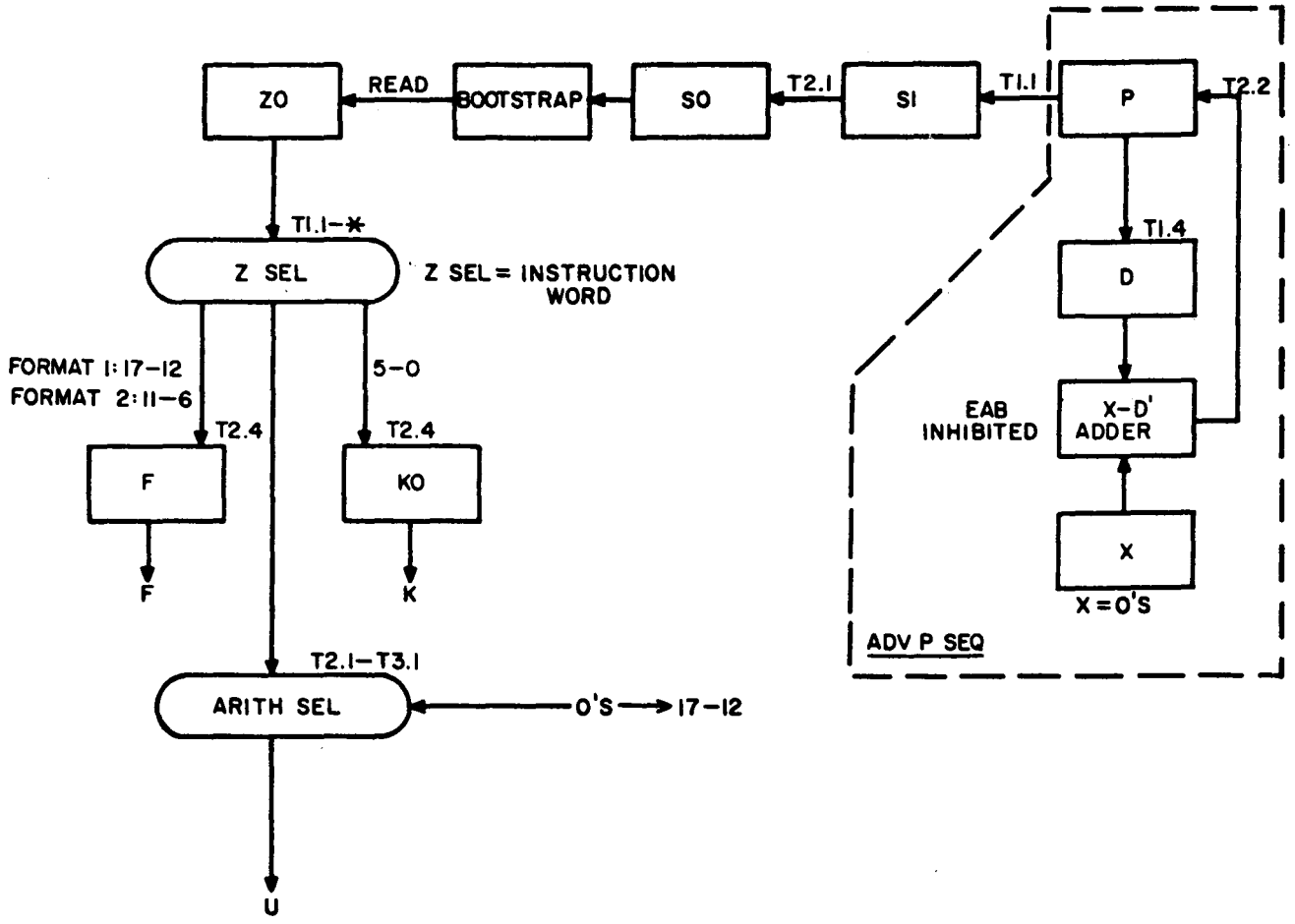
Figure 2-98. Simplified Circuit Diagram Of Bootstrap Drive Circuitry (Group O and Line O)

2-409. Group 0 has been selected ($SO_0 = 0$, $SO_1 = 0$, $SO_2 = 0$). 00BG00 applies +15 volts to the drive-line by enabling the conduction of the output transistor. Since line 0 has been selected ($SO_3 = 0$, $SO_4 = 0$), 10DB00 applies -15 volts on the other side of the drive-line. Current now flows in the direction shown on figure 2-98, either through, or around, each of the 18 core transformers, depending upon the desired binary content of this address.

2-410. Sense-Line Analysis. Refer to plates P-132 and P-133 for sense-line arrangement and outputs. Each of the 18 core transformers has a sense-line as its secondary winding. A common connection is made between one side of each sense-line and ground. A voltage is induced in the sense-line only if the current-carrying (selected) word-drive-line is threaded through the core. This is a negative voltage limited to -4.5 volts by its associated output diode. The 18 sense-line outputs are routed directly to the ZO-register (refer to plates P-115, P-116, and P-117) where they are gated by the Bootstrap $\Rightarrow$ ZO signal from 49N11 (refer to plate P-25).

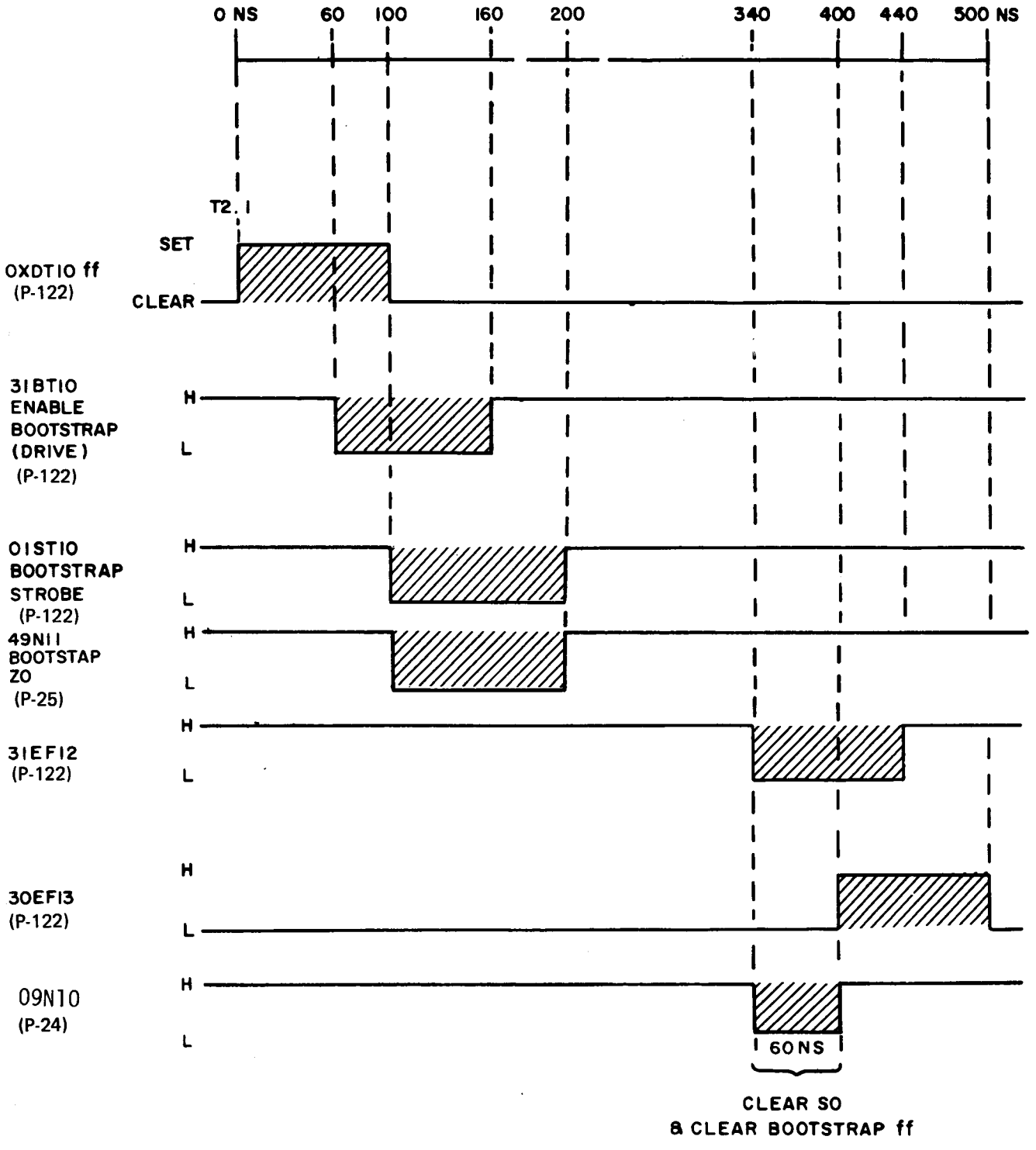
2-411. Initiate Bootstrap Memory. To initiate a bootstrap memory reference, 91S00 (refer to plate P-107) outputs a low when a control or bootstrap memory address has been translated from the S1-register. At time T2.1 of the proper sequence, 39N10 (refer to plate P-24) develops the S1 $\rightarrow$ SO command enable to set the 0XDT10 flip-flop (refer to plate P-122) and start control-memory timing. Since a bootstrap addres has been translated, the Bootstrap flip-fl (refer to plate P-107) sets to enable the bootstrap portion of the control-memory timing gate which references bootstrap instead of control-memory. Gate 10N00 (refer to plate P-107) is disabled to allow the ZO $\rightarrow$ Z Sel rather than the Z1 $\rightarrow$ Z Sel command enable. Refer to figure 2-99 for a simplified block diagram of the I-sequence operations used to obtain an instruction word from bootstrap memory.

2-412. Figure 2-100 illustrates the timing waveforms developed during a bootstrap memory reference. Timing is initiated with the setting of the 0XDT10 flip-flop (refer to plate P-122). Although bootstr memory uses the same timing source as control-memory, a different set of waveforms is developed because the Bootstrap flip-flop 0XS09 (refer to plate P-107) has been set. The operation of the timing chain was discussed in paragraph 2-400. A co plete bootstrap memory cycle is approximately 500 nanoseconds with an access time of approximately 250 nanoseconds. It is not necessary to develop a write pulse because bootstrap memory is not destroyed on readout.



NOTES: ONLY THE INSTRUCTION READ-OUT AND P ADVANCEMENT OPERATIONS ARE SHOWN. OTHER I SEQUENCE EVENTS ARE AS NORMAL.
*ZO → Z SEL OCCURS AS LONG AS SI = BOOTSTRAP ADDRESS.

Figure 2-99. I-Sequence Data Flow Referencing Bootstrap



NOTE: GATE TRANSIENT TIMES ARE NOT CONSIDERED.

Figure 2-100. Bootstrap Memory Theoretical Timing

Section 2-3. Detailed Functional Description (I/O Section)

2-413. INPUT/OUTPUT SECTION

2-414. The Input/Output (I/O) section of the computer provides communication between the computer and peripheral equipment. This communication consists of control and word transfers using specially formatted input and output logic. The computer uses separate input and output lines (channels) to communicate with a maximum of 16 peripheral devices. Logic for the 16 channels is distributed among four I/O chassis, each chassis controlling four channels. The number of I/O chassis in use at any one time is determined by the channels required for a given operation. The number of channels in use at any one time may be any multiple of four up to the maximum of sixteen. The I/O chassis are hard-wired to the peripheral equipments utilizing twisted pairs and ground lines. The pairs are assembled into separate input and output cables having grounded shields and protective covers.

2-415. INTERFACE. The term interface refers to the characteristics involved in mating the computer with peripheral equipment so that data and signals may be exchanged. All I/O factors such as voltage levels, signal duration, signal rate, and line impedance are considered under the topics of interface. Two types of interface (fast and slow) are used by the computer, and the particular type used in given installation is determined primarily by the requirements of the peripheral equipment and the lengths of the interconnecting cables. The principal interface characteristics are shown in table 2-95. Signals listed the table will be discussed later in this section. The type of interface (fast or slow) is selected by plug-in jumper cards. All four channels on a common chassis have the same interface for control signals; data signals may use either interface type.

2-416. DATA EXCHANGE RATE. The time necessary to complete one data-word exchange operation for a single channel is four microseconds. The maximum I/O rate is one word exchanged every two microseconds because of a two-microsecond overlap of consecutive word-exchange operations. The data-exchange rate is also affected by the type of interface employed. For example, the Input Data or External Interrupt (ID/EI) Acknowledge Signal timing is common to the four channels of a chassis and prevents any input-data operation on these channels while the signal is being generated. Similarly, the Output Data or External Function (OD/EF) Acknowledge Signal timing will disable any output-data generation on the four channels of a chassis. If dual-channel operation (36-bit word transfer) is employed, the acknowledge timing will prevent another data-exchange operation of the same type on any channel of the entire drawer. Dual-channel operation is discussed in more detail later in this section. Maximum data-transfer rates are shown in table 2-96.

2-417. I/O SIGNALS. Input/Output signals consist of data-request, Real Time-Clock (RTC)-request and interrupts.

2-418. Data-Request Signal. Data-request signals are sent from a peripheral device

TABLE 2-95. I/O INTERFACE CHARACTERISTICS

Interface Type	Line Voltage Levels	Input Data or External Interrupt Acknowledge Signal	Output Data or External Function Acknowledge Signal
Slow	H=Ov (gnd); L=-15v	14.75 μs	12.75μs
Fast	H=Ov (gnd); L=-3v	2.725 μs	2.75 μs

TABLE 2-96. DATA TRANSFER RATES (MAXIMUM)*

Other Conditions	Mode (Word Size)	Fast Interface			Slow Interface		
		One Chassis	Two Chassis	Four Chassis	One Chassis	Two Chassis	Four Chassis
Alternate Inputs and Outputs on Alternate Channels	Single Channel	500	500	500	83.3	167	334
	Dual Channel	NA	250	250	NA	76.9	154
Inputs Only or Outputs Only on Alternate Channels	Single Channel	250	500	500	41.6	83.3	167
	Dual Channel	NA	167	250	NA	38.4	76.9
Inputs Only or Outputs Only on Individual Channel	Single Channel	167	NA	NA	41.6	NA	NA
	Dual Channel	NA	125	NA	NA	38.4	NA
Alternate Inputs and Outputs on Individual Channel	Single Channel	334	NA	NA	83.3	NA	NA
	Dual Channel	NA	250	NA	NA	76.9	NA

*All rates are in thousands of words per second.

to the computer, causing the transfer of one data word. The request-honoring operation requires the use of memory to output or input a word. Therefore, the request disables the instruction sequences to stop the program during its two-microsecond use of memory. Request operations use the I/01 and I/02 sequences. Data requests are of four types as follows:

1. Input-Data Request (IDR)

2. Output-Data Request (ODR)
3. External-Function Request (EFR)
4. External-Interrupt Request (EIR)

2-419. Real-Time-Clock Request (RTC Request) Signals. The RTC request is generated by an internal, accurate oscillator operating at a frequency of 1024 pulses per second. This oscillator is used to maintain a constantly updated 18-bit block count in the control memory address 00015_8. The RTC request causes the computer to increment by +1 the content of this address. The addition is performed in an open-ended adder such that the incrementing of the 777777_8 count results in the 000000_8 count. This clock count can be preset to any value by the program and periodically, is referenced so as to time certain program events. The honoring of the RTC request can be prevented only by the RTC DISCONNECT switch. Unlike data-request signals, the RTC request uses only the I/01 sequence to perform the updating operation.

2-420. Interrupt Signals. Interrupt signals causes an interruption to the program in the form of a jump. Each type of interrupt is assigned a special jump address. The jump is made without disturbing P, so that a return jump instruction can be used at the jump address to retain P for later return to the program at the point of interruption, as well as to execute a second jump to an interrupt routine. With the exception of the External Sync Interrupt, all interrupts are internally generated. Interrupt signal types are as follows:

1. Input-Data Monitor (ID Mon)
2. Output-Data Monitor (OD Mon)
3. External-Function Monitor (EF Mon)
4. External-Interrupt Monitor (EI Mon)
5. RTC Overflow Interrupt
6. RTC Monitor Interrupt
7. External-Sync Interrupt
8. Instruction-Fault Interrupt
9. Resume Fault Interrupt

2-421. BUFFERS AND ADDRESS CONTROL WORDS. Buffer refers to the number of words set up by the program to be transferred. ID, OD, and EF buffers are established by the instructions In (f = 50:11), Out (f = 50:12), and EXF (f = 50:13), respectively. The two addresses immediately following these instructions in the program are reserved as the buffer limits. Buffer limits are the initial and final addresses for data to be transferred.

2-422. Terminal Address Control Word (TACW). The TACW immediately follows the buffer instruction in the program. It contains the 16-bit final data address. For example, if an OD buffer is being established, TACW specifies the address from which the last data word of the buffer is to be obtained and outputted. TACW may also request the Continuous Data Mode (CDM) at buffer termination. CDM is discussed later in this section. Refer to figure 2-101 for TACW format.

2-423. Initial Address Control Word (IACW). The IACW immediately follows the TACW in the program. It contains the 16-bit initial data address. For example, if an ID buffer is being established, IACW specifies the address where the first input data word of the buffer is to be stored. IACW also may request the monitor interrupt to be generated a buffer termination. That is, after the last word of the buffer is transferred, the ID Mon, OD Mon, or EF Mon interrupt (depending upon the buffer type) is generated on the particular channel. IACW also specifies the direction of the buffer. A forward buffer is one which provides sequentially increasing addresses for the data. A backward buffer

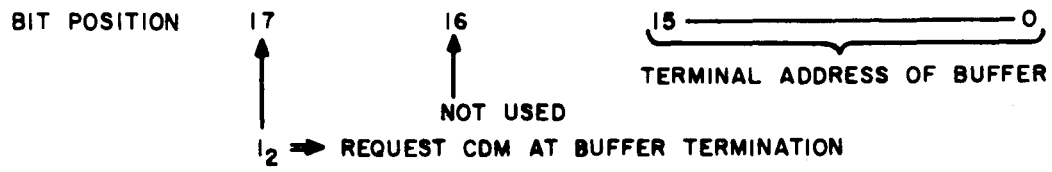


Figure 2-101. Terminal Address Control Word Format

provides sequentially decreasing data address In both cases, TACW and IACW specify the terminal and initial buffer addresses, respectively. Refer to figure 2-102 for the IACW format. Once the buffer is established (made active), the IACW is more appropriately referred to as the current address control word (CACW). The TACW and IACW are transferred into special control memory addresses (refer to table 2-97). In control memory, IACW is updated by ±1, depending upon the buffer direction, after each word transfer. Therefore, it always specifies the address for the next word transfer. Thus, IACW becomes CACW.

2-424. All of the I/O instructions are Format II type. The K-designator (6 least significant bits) specifies the channel number. Refer to figure 2-103 for programming instructions of buffer initiation.

2-425. SPECIAL I/O LOGIC. Each channel contains special I/O logic to allow the computer to detect and honor a request signal from a peripheral device, to establish priority of requests, and to establish an I/O translation used to develop control signals for execution of the proper I/O service.

2-426. Active Flip-Flop. Each channel has an active flip-flop for ID, EF, and OD. This flip-flop (in the set state) is the only requirement for the computer to detect and honor the associated request signal from a peripheral device. The active flip-flop is set by the buffer instruction (f = 50:11 through 50:13). At buffer termination, the active flip-flop is cleared to prevent honoring of more request than had been determined by the address control words.

2-427. Priority Logic. Since it is possible to have more than one I/O signal requesting I/O services at the same time, a priority system exists to select one of these signals to be honored. Priority is set up on a channel number and function basis. The channel priority can be altered by plug-in cards. The priority scheme is discussed in a later section.

2-428. I/O Translator. After a signal is selected by the priority logic, the information is placed in the I/O translator. This logic translates the signal type (function) and its channel number, and assists in developing the control signals necessary to execute the proper I/O service.

2-429. DATA-REQUEST OPERATIONS. The I/01 sequence is used to honor an IDR, ODR, EFR, or EIR. Each of these signals requires word transfer between the computer

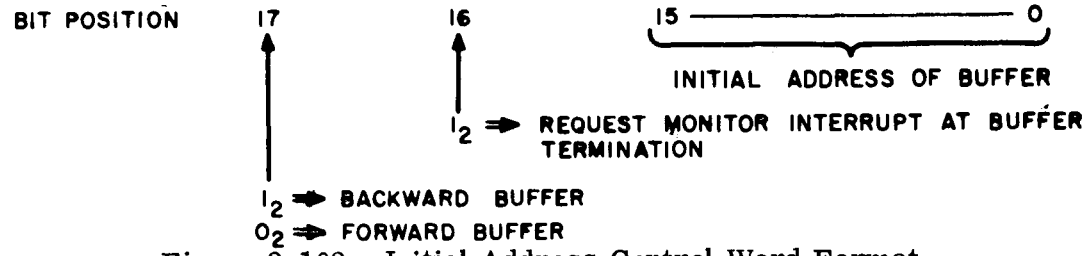


Figure 2-102. Initial Address Control Word Format

PROGRAM ADDRESS		
01000	501202	OUT INSTRUCTION, CHANNEL 2
01001	003000	TACW
01002	202000	IACW

NOTES: 1. IACW SPECIFIES:

a. FIRST OUTPUT DATA WORD WILL BE OBTAINED FROM ADDRESS 02000_8.
b. OD MON INTERRUPT ON CHANNEL 2 AT BUFFER TERMINATION.
c. FORWARD BUFFER.

2. TACW SPECIFIES:

a. LAST OUTPUT DATA WORD WILL BE OBTAINED FROM ADDRESS 03000_8.
b. NO CDM AT BUFFER TERMINATION.

3. IACW AND TACW TOGETHER ESTABLISH A 01001_8 -WORD BUFFER.

a. OD BUFFER EXAMPLE

PROGRAM ADDRESS		
04000	501104	IN INSTRUCTION, CHANNEL 4
04001	006000	TACW
04002	407500	IACW

NOTES: 1. IACW SPECIFIES:

a. FIRST INPUT DATA WORD WILL BE STORED AT ADDRESS 07500_8.
b. NO ID MON INTERRUPT AT BUFFER TERMINATION.
c. BACKWARD BUFFER.

2. TACW SPECIFIES:

a. LAST INPUT DATA WORD WILL BE STORED AT ADDRESS 06000_8.
b. NO CDM AT BUFFER TERMINATION.

3. IACW AND TACW TOGETHER ESTABLISH A 01501_8 -WORD BUFFER.

b. ID BUFFER EXAMPLE

Figure 2-103. Buffer Program Examples

TABLE 2-97. ASSIGNED ADDRESSES FOR ADDRESS CONTROL WORDS

Address Control Word	Special Addresses		
	ID	OD	EF/CDM
TACW	00060_8 + 2x chan	00040_8 + 2x chan	00020_8 + 2x chan
IACW (CACW)	00061_8 + 2x chan	00041_8 + 2x chan	00021_8 + 2x chan

and the requesting peripheral device. They may also generate a monitor interrupt, which is a separate signal to be considered as a priority before it can be honored to cause the program jump.

2-430. DUAL-CHANNEL OPERATION. Dual-channel operation employs an odd-numbered channel with the next lower even-numbered channel to effect a 36-bit word transfer. The request signal must occur on the odd channel and the CHANNEL FUNCTION switch (one for each odd channel) must be in the DUAL position. If the request occurs on the even channel, Single-Channel Mode is in effect regardless of the CHANNEL FUNCTION switch position. The 18-bit word for each channel is treated normally as a single channel. The only difference is that the acknowledge signal is not sent until both halves of the 36-bit word have been transferred.

2-431. CONTINUOUS-DATA MODE (CDM). CDM is initiated at buffer termination if requested by TACW ($TACW_{17} = 1$). It keeps the channel active by preventing normal clearing of the associated active flip-flop, and it establishes a new set of address control words. It obtains the new TACW from control memory address 00020_8 + 2x (x $\Rightarrow$ channel number), if the channel is 0 through 7, or address 00220_8 + 2x if the channel is 10 through 17_8. These words are transferred to normal special-control-memory addresses which hold the words during buffer operation. The new TACW is at 00020_8 + 2x + 1, or 00220_8 + 2x + 1. Prior to buffer termination, the program must have stored the reload TACW and CACW in preparation for CDM. If requested by CACW ($CACW_{16} = 1$), the normal monitor interrupt can occur at termination of the old buffer after the new buf has automatically been established. Each new set of address control words can determine its buffer direction, requirement for monitor interrupt, and requirement for CDM again. Usually, the monitor-interrupt technique is used with CDM. The interrup is the indication to the program that the address control words have been reloaded and, if CDM reload is to be performed again, the program must prepare the next TACW and CACW. For a particular channel, all three functions (ID, OD, and EF) use the same addresses to obtain the reload TACW and CACW. Therefore, if any two buffers are active at the same time on the same channel, the program probably would not request CDM for both.

2-432. EXTERNALLY SPECIFIED INDEX (ESI). ESI operation employs an odd-numbered channel with the next lower even-numbered channel. The normal special addresses used to obtain the address contr words are not used. The odd channel carries the control-memory address to obtain TACW. The CACW is extracted from the next consecutively high address. ESI allows the peripheral device to select its own addres

control words for each word transferred. These address control words must be set up by the program. The word transfer is effected over the even channel. To enable ESI operation, the request signal must occur on the odd channel and the CHANNEL FUNCTION switch must be in the ESI position. If the request occurs on the even channel, Single-Channel Mode is in effect regardless of the switch position. ESI mode with an EIR forces Dual-Channel Operation (36-bit status word).

2-433. EXTERNALLY SPECIFIED ADDRESS (ESA). ESA operation employs an odd-numbered channel with the next lower even-numbered channel. With this operation in effect, there are no address control words. The peripheral device directly specifies, over the odd channel, the address for the word transfer requested. The word transfer is effected over the even channel. To enable ESA operation, the request signal must occur on the odd channel and the CHANNEL FUNCTION switch must be in the ESA position. If the request occurs on the even channel, Single-Channel Mode is in effect regardless of switch position. ESA mode with an EIR forces Dual-Channel operation (36-bit status word).

2-434. EXECUTION OF SIN, SOUT, SEXF, INSTP, OUTSTP, EXFSTP INSTRUCTIONS (f = 50:01 through 50:03 and 50:15 through 50:17). These f = 50:01 through 50:03 and 50:15 through 50:17 instructions set one of the active flip-flops or terminate ID, OD, or EF operations on a single channel.

2-435. SIN (f = 50:01). The SIN instruction sets the ID Active flip-flop for the channel specified by the four least significant bits (k) of the instruction word. Input-data operations are enabled on this channel.

2-436. SOUT (f = 50:02). The SOUT instruction sets the EF/OD Active flip-flop and clears the EF Mode flip-flop for channel k. Output-data operations are enabled on this channel.

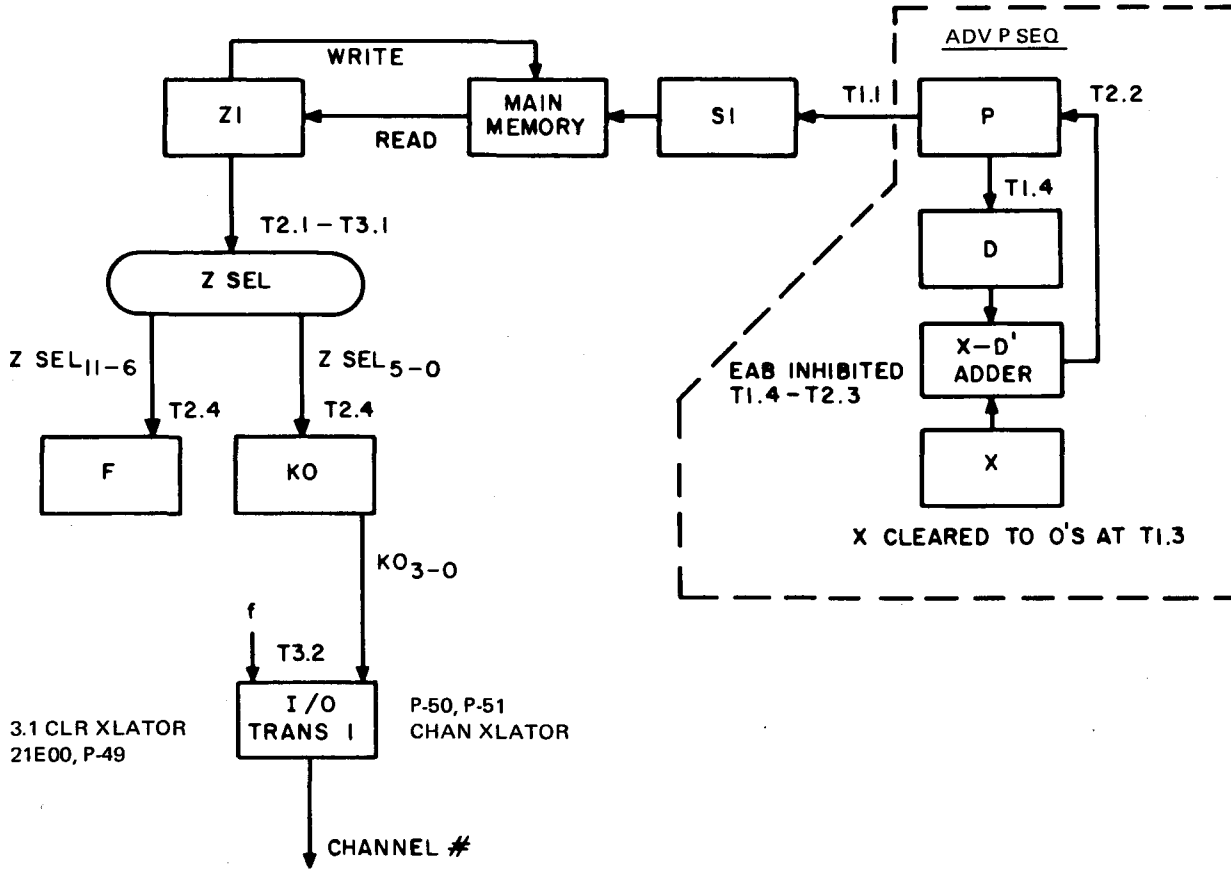
2-437. SEXF (f = 50:03). The SEXF instruction sets the EF/OD Active flip-flop and sets the EF Mode flip-flop for channel k. External-function operations are enabled on this channel.

2-438. INSTP (f = 50:15). The INSTP instruction terminates input-data operations on channel k by clearing its ID Active flip-flop. Its ID Monitor flip-flop is also cleared to prevent the occurrence of an ID monitor interrupt.

2-439. OUTSTP (f = 50:16). This instruction terminates output-data operation (OD) on channel k by clearing the associated OD Active flip-flop. The OD monitor flip-flop for the channel is also cleared to prevent the occurrence of an OD monitor interrupt.

2-440. EXFSTP (f = 50:17). This instruction terminates external-function operations (EF) on channel k by clearing the EF Active flip-flop for the channel is also cleared to prevent the occurrence of an EF monitor interrupt.

2-441. I-Sequence Data Flow for f = 50:01 through 50:03, and 50:15 through 50:17. All operations are performed within the I-sequence. Only one memory reference is necessary to obtain the instruction. Refer to figure 2-104 for a block diagram description of the execution of f = 50:01 through 50:03 and 50:15 through 50:17. KO is set to the six least significant bit positions of the instruction word from Z-Select. Only the



T4.3 IF f = 50:01; SET ID ACT FF
IF f = 50:02; SET OD ACT FF
IF f = 50:03; SET EF ACT FF
} SET ACTIVE OON44, P-53⇒P-57
ACT FF'S P. 9-61
IF f = 50:15; CLEAR ID ACT FF & CLEAR ID MON FF
IF f = 50:16; CLEAR OD ACT FF & CLEAR OD MON FF
IF f = 50:17; CLEAR EF ACT FF & CLEAR EF MON FF

Figure 2-104. I-Sequence Data Flow for f = 50:01 - 50:03, 50:15 - 50:17

four lower bits of this value are used. At time T3.2, the I/O translator register is set KO_{3-0} which is the channel number. This register is also affected by the function code which indicates the type of buffer (ID, OD, or EF). As noted earlier, the instruction would be obtained from bootstrap or control memory.

2-442. I-Sequence Essential Commands for f = 50:01 through f = 50:03 and 50:15 through 50:17. Refer to table 2-98 for a list of essential commands for f = 50:01 through 50:03 and 50:15 through 50:17. Those commands that have not been described in a preceding I-sequence are described in table 2-99 in detail. Refer to tables 2-100 through 2-102 for the setting of Channel translator, Group translator, and Function translator respectively.

TABLE 2-98. I-SEQUENCE ESSENTIAL COMMANDS FOR f=50:01 THROUGH 50:03, 50:15 THROUGH 50:17

Time Notation	Commands
T4.4	Clear S1
T1.1	P → S1, Init Memory, *set Incr P ff
T1.3	*Clear D, *clear X, clear Z1, clear F, *set OXL11 ff
T1.4	$*P_L \rightarrow D_L$, $*P_U \rightarrow D_U$, clear KO, *set Inhib EAB ff
T2.1	Z1 → Z − Sel, *clear Incr P ff
T2.2	*Clear P, *Adder → P
T2.3	*Clear OXL11 ff, *clear Inhib EAB ff
T2.4	$Z - Sel_{11-6} \rightarrow F$, set OXF06 ff, $Z - Sel_{5-0} \rightarrow KO$
T3.1	Clear I/O Trans 1, drop Z1 → Z − Sel
T3.2	f · KO → I/O Trans 1
T4.3	**Set Act ff, **clear Act and Mon ff's

*These events are concerned with or are controlled by the Advance-P subsequence.

**if f=50:01; set ID Act ff
if f=50:02; set OD Act ff
if f=50:03; set EF Act ff
if f=50:15; clear ID Act ff and clear ID Mon ff
if f=50:16; clear OD Act ff and clear OD Mon ff
if f=50:17; clear EF Act ff and clear EF Mon ff
I/O translator 1 selects the channel number.

TABLE 2-99. COMMANDS/DESCRIPTIONS FOR EXECUTION OF SIN, SOUT, SEXF, INSTP, OUTSP, EXFSTP INSTRUCTIONS (f = 50:01 THROUGH 50:03 AND 50:15 THROUGH 50:17) (Descriptions Limited to Commands Not Described in Table 2-19, I-Seq)

Time Notation	Command/Description
T3.1	Clear I/O Trans 1. Gate 21E00 (refer to plate P-49) outputs a high because of a L ⇒ I-Seq and a low from 03T32. The high is inverted by 01E00 and at Ø1, 02E00 outputs a H ⇒ Clear Translator via 04E00 to clear flip-flops OXG07, OXG08, OXG00, and OXG11 and OXG10 (refer to plate P-50). The high from 02E00 is also inverted by 03E00 which outputs a L ⇒ Clear Translator to clear Function Translator flip-flops OXG05 and OXG04 and Channel Translator flip-flops OXG02 and OXG01 (refer to plate P-51).
T3.2	f · KO → I/O Trans 1. Gate 22E00 (refer to plate P-49) will output a low because of a high from 21E00 which is enabled by a L ⇒ I-Seq and a low for 03T32. At Ø2, 23E00 will produce a L ⇒ f and KO → Translator to flip-flops OXG00, OXG07, and OXG08 (refer to plate P-50) and to flip-flops OXG01, OXG02, OXG04, and OXG05 (refer to plate P-51).
T4.3	Set Act ff if f = 50:01, 50:02, 50:03. To set the Active flip-flop (refer to plate P-61 through P-64) the following inputs are required: All flip-flops require a low input for channel number. The selected channel is determined by the status of the Channel Translator flip-flops OXG02, OXG01 (refer to plate P-51) which are conditioned by KO-1 and KO-2. For example, if channel 0, 1 is selected, neither flip-flop is set and 20G00 is enabled and a H ⇒ Channel 0, 1. (Refer to table 1-103 for condition of flip-flops for selected channel.) The high from 20G00 is inverted by 30Gg0 (refer to plate P-59A) and applied to 31Gg0. The other low input to 31Gg0 is from I/O Translator 1 - Group and Mode (refer to plate P-50) via 31Gg8 (refer to plate P-59). The status of flip-flops OXG07, OXG00 (refer to plate P-50) is determined by KO-0 and KO-3. (Refer to table 1-104 [Group Translator] for settings of flip-flops.) At Ø3, 31Gg0 (refer to plate P-59A) produces a L ⇒ Channel 0, 1 to the ID, OD and EF flip-flops (refer to plate P-61). Note that the low also goes to the clear side of the flip-flops. Other inputs required to complete the setting of the Active flip-flops are L ⇒ Set ID Active, L ⇒ Set OD Active, or L ⇒ Set EF Active. These inputs are initiated from the Set Active gate 00N44 (refer to plate P-53), which outputs a high when the inputs are L ⇒ T43 I-Seq Format II, L ⇒ f = 00-07 and L ⇒ f = XO-X3. (When f = 50:11-50:13, 00N44 is enabled by a L ⇒ B Seq.) The high output is inverted by 01N44 and is applied to 9gN44 (refer to plate P-57), and then to

TABLE 2-99. COMMANDS/DESCRIPTIONS FOR EXECUTION OF SIN, SOUT, SEXF, INSTP, OUTSP, EXFSTP INSTRUCTIONS (f = 50:01 THROUGH 50:03 AND 50:15 THROUGH 50:17) (Cont)

.me Notation	Command/Description
T4.3 (cont)	4gN44, 6gN44 and 5gN44. The L ⇒ Set OD Active is applied from 6gN44 and L ⇒ Set EF Active is applied from 5gN44 to the respective OD or EF flip-flop (refer to plates P-61 through P-64). To produce a L ⇒ Set ID Active, gate 4gN44 (refer to plate P-57) requires a H ⇒ ID input. This input is initiated at the Function Translator (refer to plate P-51) where the other inputs L ⇒ OD, L ⇒ EF to the OD and EF flip-flops are also initiated. (Refer to table 1-105 for setting of the Function Translator flip-flops OXG05, OXG04.) When 20G04 (refer to plate P-51) outputs a high it is inverted by 30Gg4 and 31Gg4 (refer to plate P-60) and applied to enable 4gN44 (refer to plate P-57) which produces a L ⇒ Set ID Active to set ID flip-flop (refer to plates P-61 through P-64). The output of 31Gg4 (refer to plate P-60) is also applied to 4gN42 (refer to plate P-56) when setting ID Mon flip-flop. When 20G06 (refer to plate P-51) is enabled, the H ⇒ OD is inverted by 30Gg6 (refer to plate P-60) and applied via 33Gg6 to the OD flip-flop (refer to plates P-61 through P-64). When 20G05 (refer to plate P-51) is enabled the H ⇒ EF is inverted by 30Gg5 (refer to plate P-60) and applied via 33Gg5 to the EF flip-flops (refer to plate P-61 through P-64). These, plus previously described inputs will cause the OD and EF Act flip-flops to be set. Clear ID Act ff and ID Mon ff if f = 50:15. To clear the ID Act and ID Mon flip-flops (refer to plates P-61 through P-64) two low inputs are needed, a low for the channel number (same as described for setting the ID Act flip-flop) and L ⇒ Clear ID Active or a L ⇒ Clear ID Monitor. The Clear Active gate 01N43 (refer to plate P-53) is enabled by a L ⇒ T43 I-Seq Format II, L ⇒ f = 14-17 and the absence of a H ⇒ f=14. The high output is inverted by 02N43 and 01N41. The L ⇒ Clear Active from 02N43 is inverted by 9gN43 (refer to plate P-56) and applied to 4gN43, 6gN43 and 5gN43. The other low input to 4gN43 is developed the same as described for setting the ID Act flip-flop. The L ⇒ Clear ID Active from 4gN43 will clear the ID Act flip-flop at this time. The L ⇒ Clear Monitor from 01N41 (refer to plate P-53) is inverted by 9gN41 (refer to plate P-56) and applied to 4gN41 and with the other input a H ⇒ ID, 4gN41 will output a L ⇒ Clear ID Monitor to clear the ID Mon flip-flop.

TABLE 2-99. COMMANDS/DESCRIPTIONS FOR EXECUTION OF SIN, SOUT, SEXF, INSTP, OUTSP, EXFSTP INSTRUCTIONS (f=50:01 THROUGH 50:03 AND 50:15 THROUGH 50:17) (Cont)

Time Notation	Command/Description
T4.3 (cont)	Clear OD Act ff and OD Mon ff if f = 50:16. This instruction uses Clear Active gate 01N43 (refer to plate P-53), Function Translator (refer to plate P-51) and the channel number as described previously. Clear EF Act ff and EF Mon ff if f = 50:17. This instruction uses Clear Active gate 01N43 (refer to plate P-53), Function Translator (refer to plate P-51) and channel number as described previously. The output of 9gN41 (refer to plate P-56) plus the H ⇒ EF Clear is applied to 5gN41, inverted and applied to the EF Mon flip-flops clear inputs. The EF Mon flip-flops are located on plate P-38.

TABLE 2-100. CHANNEL TRANSLATOR INDICATIONS

Translator Flip-Flops		Indications
OXGO2	OXGO1	
Clear	Clear	20G00 = H; channel 0, 1, 10, or 11 selected
Clear	Set	20G001 = H; channel 2, 3, 12, or 13 selected
Set	Clear	20G02 = H; channel 4, 5, 14, or 15 selected
Set	Set	20G03 = H; channel 6, 7, 16, or 17 selected

TABLE 2-101. GROUP TRANSLATOR INDICATIONS

Translator Flip-Flops		Indications
OXGO7	OXGOO	
Clear	Clear	20G10 = H; chassis 2 selected (lower-even group)
Clear	Set	20G11 = H; chassis 1 selected (lower-odd group)
Set	Clear	20G12 = H; chassis 10 selected (upper-even group)
Set	Set	20G13 = H; chassis 9 selected (upper-odd group)

TABLE 2-102. FUNCTION TRANSLATOR INDICATIONS

Translator Flip-Flops		Indications	Instruction
OXGO5	OXGO4		f =
Clear	Set	20G05 = H, EF	50:03, 50:17
Set	Clear	20G06 = H, OD	50:02, 50:16
Set	Set	20G04 = H, ID	50:01, 50:15

2-443. IB1 and IB2 SEQUENCE. The B1 and I-sequence run in parallel to obtain the TACW from memory. B2 and I-sequence run in parallel to store TACW in control memory, obtain IACW from memory and set the proper active flip-flop. See table 2-103 for a list of essential commands for a IB1 and IB2 sequence. Table 2-104 contains detail descriptions of commands not previously covered.

TABLE 2-103. IB1 AND IB2-SEQUENCE ESSENTIAL COMMANDS

Time	Command
	IB1 SEQUENCE
T4.2	Sequence Designator, Lower Rank → Upper Rank (Set I and B1 ff's)
T4.4	CLR S1
T1.1	P → S1 Enable Main Memory Set INCREMENT P flip-flop CLR Z0
T1.2	
T1.3	Inhibit CLR F CLR D, X, W, Z1
T1.4	Set INHIBIT EAB flip-flop PL → DL PU → DU Inhibit Clear KO
T2.1	CLR Sequence Designator, lower rank CLR Z0 Memory → Z1, Z1 → Z Select, Z Select → Arithmetic Selector
T2.2	Sequence Designator, Upper Rank → Lower Rank (Set I and B1 flip-flops) Adder → P Clear P
T2.3	CLR EAB flip-flops CLR D, X, W
T2.4	Arithmetic Selector → D Arithmetic Selector → X, $\overline{\text{Arithmetic Selector}}$ → X Arithmetic Selector → W Inhibit Z select Bits 00-05 → K0 Inhibit Z select → F CLR Main Memory Enable flip-flop Inhibit 0's → Arithmetic Selector Bits 12-17

TABLE 2-103. IB1 AND IB2-SEQUENCE ESSENTIAL COMMANDS (Cont)

Time	Command
	IB1-SEQUENCE
T3.1	ICR → SO Initiate Control Memory Clear I/O Translators (Xlator 1, Chan and Funct and Special Int Xlator) CLR ZO
T3.2	F and K → I/O Translator
T3.3	
T3.4	
T4.1	Inhibit Clear Run 1 flip-flop CLR ZO CLR Sequence Designator, Upper Rank CLR B±1 flip-flop
T4.2	Sequence Designator, Lower Rank → Upper Rank (Set I and B2 flip-flops)
T4.3	CLR Z1
	IB2-SEQUENCE
T4.4	Adder → Z1 CLR S1
T1.1	P → S1 I/O Translator → S0 Enable Main Memory Initiate Control Memory Set INCREMENT P flip-flop CLR Z0
T1.2	Z1 → Z0
T1.3	Inhibit CLR F CLR D, X, W, Z1 Inhibit Control Memory → Z0

TABLE 2-103. IB1 AND IB2-SEQUENCE ESSENTIAL COMMANDS (Cont)

Time	Command
	IB2-SEQUENCE
T1.4	Set INHIBIT EAB flip-flop PL → DL PU → DU Inhibit Clear K0
T2.1	CLR Sequence Designator, Lower Rank CLR Z0 Memory → Z1
T2.2	Sequence Designator, Upper Rank → Lower Rank (Set I flip-flop and B2 ff) Adder → P Clear P, Z1 → Z Select, Z Select → Arith Selector
T2.3	CLR EAB flip-flops CLR D, X, W
T2.4	Arithmetic Selector → D Arithmetic Selector → X, $\overline{\text{Arithmetic Selector}}$ → X Arithmetic Selector → W Inhibit Z select Bits 00-05 → K0 Inhibit Z select → F CLR Main Memory Enable flip-flop Inhibit 0's → Arithmetic Selector Bits 12-17
T3.1	ICR → S0 Initiate Control Memory Clear I/O Translators (Xlator 1, Chan and Funct and Special Int Xlator) CLR Z0
T3.2	f • K0 → I/O Translator
T3.3	
T3.4	

TABLE 2-103. IB1 AND IB2-SEQUENCE ESSENTIAL COMMANDS (Cont)

Time	Command
	IB2-SEQUENCE
T4.1	Inhibit Clear Run 1 flip-flop CLR Z0 CLR Sequence Designator, Upper Rank CLR B±1 flip-flop
T4.2	Sequence Designator, Lower Rank → Upper Rank (Set I flip-flop)
T4.3	CLR Z1 Set Active FF (as designated by F and K)
T4.4	Adder → Z1 At this point, computer enters I-Sequence of next instruction.
T1.1	P → S1 I/O Translator → S0 Set S0 2^0 Enable Main Memory Initiate Control Memory Set INCREMENT P flip-flop CLR P
T1.2	Z1 → Z0
T1.3	CLR F, D, W, Z1 Inhibit Control Memory → Z0

TABLE 2-104. IB1 AND IB2-SEQUENCE COMMANDS/DESCRIPTIONS

Time Notation	Command/Description
	IB1 SEQUENCE
T4.2	Set Bl_i ff. Gate 12E20 (refer to plate P-12) is enabled by a L $\Rightarrow \overline{I/O}$, a low from 03T42 and $\emptyset 2$ to produce a low to Bl_i flip-flop 4XG20. Gate 22G20 is enabled by a L $\Rightarrow$ Format II, L $\Rightarrow$ f=10 -13, L $\Rightarrow$ I-Seq and the absence of a H $\Rightarrow$ B1-Seq. The high output is inverted by 23G20 to set Bl_i ff at time T4.2.
T1.3	Inhibit Clear F. When B1 flip-flop 4XG20 (refer to plate P-12) was set at time T4.2 a H $\Rightarrow$ B1-Seq was applied to gate 43G20 (refer to plate P-14) and the low output was inverted by 44G20 and applied to 06N16 (refer to plate P-40) as a H $\Rightarrow$ Initiate Buffer to inhibit the clearing of the F-register.
T1.4	Inhibit Clear K0. The high from 44G20 (refer to plate P-14) is also applied to 09N14 (refer to plate P-37) to inhibit the clearing of the K0-register.
T2.1	Inhibit 0's $\rightarrow$ Arithmetic Selector Bits 12 through 17. The high output of 44G20 (refer to plate P-14) is also applied as a H $\Rightarrow$ Initiate Buffer to disable 70N01 (refer to plate P-17) and the low output is inverted by 78N01 and disables 79N01 to inhibit 0's $\rightarrow$ Arithmetic Selectors Bits 12 through 17.
T2.2	Set Bl_f ff. Gate 42E20 (refer to plate P-13) is enabled at $\emptyset 2$ by a L $\rightarrow$ Run$\cdot\overline{I/O}\cdot\overline{\text{Hold 1}}$ and a low from 03T22 and the low output is applied to Bl_f flip-flop 5XG20 which sets with a L $\Rightarrow$ B1-Seq from the Upper Rank Sequence Designator.
T2.4	Arithmetic Selector $\rightarrow$ X, $\overline{\text{Arithmetic Selector}} \rightarrow$ X. Gate 20N03 (refer to plate P-19) outputs a high due to a L $\Rightarrow$ T 24 I-Seq and a L $\Rightarrow$ B-Seq. The high output is inverted by gates 18N03 and 28N03 and at $\emptyset 4$, 19N03 produces a L $\Rightarrow$ Select $\rightarrow$ X and 29N03 produces a L $\Rightarrow \overline{\text{Select}} \rightarrow$ X. Inhibit Z Select Bits 00-05 $\rightarrow$ K0. The high that was applied to 09N14 (refer to plate P-37) to inhibit the clearing of K0 is also applied to 29N14 to inhibit Z Select Bits 00 through 05 $\rightarrow$ K0.

TABLE 2-104. IB1 AND IB2-SEQUENCE COMMANDS/DESCRIPTIONS (Cont)

Time Notation	Command/Description
	IB1 SEQUENCE
	Inhibit Z Select ⇒ F. Gate 10N16 (refer to plate P-40) is disabled by a H ⇒ Initiate Buffer and the low output is inverted by 11N16 to disable gates 29N16 and 19N16 to inhibit Z Select into the F-register.
T3.1	Clear I/O Translators (Xlator 1, Chan and Funct and Special Int Xlator). Gate 21E00 (refer to plate P-49) outputs a high due to a L ⇒ I-Seq and a low from 03T32. The high is inverted by 01E00 and at $\emptyset$1, 02E00 outputs a H ⇒ Clear Translator via 04E00 to clear flip-flops 0XG07, 0XG08, 0XG00, 0XG11 and 0XG10 (refer to plate P-50). The high from 04E00 is also applied to flip-flop 2XG15 through 2XG18 (refer to plate P-55) which clear at this time. Gate 03E00 (refer to plate P-49) inverts the high from 02E00 and produces a L ⇒ Clear Translator to clear flip-flops 0XG01, 0XG02, 0XG04 and 0XG05 (refer to plate P-51).
T3.2	f · K0 ⇒ Translator. Gate 22E00 (refer to plate P-49) will output a low due to a high from 21E00 which is enabled by a L ⇒ I-Seq and a low from 03T32. At $\emptyset$2, 23E00 will produce a L ⇒ f and K0 ⇒ Xlator to flip-flops 0XG00, 0XG07, 0XG08 (refer to plate P-50) and to flip-flops 0XG01, 0XG02, 0XG04, and 0XG05 (refer to plate P-51).
T4.1	Inhibit Clear Run 1 flip-flops. Gate 24J10 (refer to plate P-3) is disabled by H ⇒ B-Seq and the low output causes 25J10 to output a high to prevent 0XJ10 flip-flop from clearing.
T4.2	Set $B2_i$ ff. $B2_i$ flip-flop 6XG20 (refer to plate P-12) is set at time T4.2 by the low from 12E20 and a L ⇒ B1-Seq.
T4.3	Clear Z1. With B1 flip-flop 5XG20 set (refer to plate P-13) gate 53G20 produces a L ⇒ Initiate Buffer to 31N13 (refer to plate P-23) and with a L ⇒ I-Seq and a low from 03T44, the high output is inverted by 08N13 and at $\emptyset$3, 09N13 produces a L ⇒ Clear Z1.
T4.4	Adder → Z1. The high from 31N13 (refer to plate P-23) is inverted by 38N13 and at $\emptyset$4, 39N13 produces a L ⇒ Adder → Z1.
	Disable CM → Z0. Gate 10N13 (refer to plate P-23) outputs a high due to a L ⇒ I or I/O Seq, a low from 03T13 and the absence

TABLE 2-104. IB1 AND IB2-SEQUENCE COMMANDS/DESCRIPTIONS (Cont)

Time Notation	Command/Description
	IB1-SEQUENCE
T4.4 (cont)	of a H ⇒ Cont Data Seq. The H ⇒Z1 → Z0· $\overline{\text{Cont Mem}}$ → Z0 disables 11N11 (refer to plate P-25) which outputs a high to disable gates 17N11, 18N11 and 19N11.
	IB2 SEQUENCE
T1.1	I/O Translator → S0, Initiate Control Memory. The L ⇒ Initiate Buffer from 53G20 (refer to plate P-13) due to $B1_f$ flip-flop being set is applied to 10N10 (refer to plate P-24) and with a L ⇒ I-Seq and a low from 03T11, 10N10 outputs a high which is inverted by 18N10 and at $\emptyset 1$, 19N10 produces a L ⇒ I/O Translator → S0 to the S0-register (refer to plate P-114) and to Control Memory (refer to plate P-22).
T1.2	Z1 → Z0. Gate 10N13 (refer to plate P-23) outputs a H ⇒ Z1 → Z0 due a L ⇒ I or I/O-Seq, a low from 03T13 and the absence of a H ⇒ Cont Data Seq. The high is inverted by 28N11 (refer to plate P-25) and applied to 29N11. Other inputs to 29N11 are a low from 30N11 (disabled by the absence of an I/O-sequence and the absence of a f = 56, 76) and a low from Control Memory via 20N11. Gate 29N11 produces a L ⇒ Z1 → Z0.
T2.2	Set $B2_f$ ff. $B2_f$ flip-flop 7XG20 is set by the low from 42E20 (refer to plate P-13) and a L⇒B2-Seq from the Upper Rank Sequence Designator.
T4.3	Clear Z1. With B2 flip-flop 7XG20 set (refer to plate P-13) gate 53G20 outputs a low to clear Z1 as in a IB1-sequence.
	Set Active ff (as designated by f and K0). The setting of the Active flip-flops was described previously in 50:01 through 50:03 instructions with the only difference being that Set Active gate 00N44 (refer to plate P-53) is enabled by a L ⇒ B2-Seq rather than a L ⇒ f = 00 through 07 and a L ⇒ f = X0 through X3.

TABLE 2-104. IB1 AND IB2-SEQUENCE COMMANDS/DESCRIPTIONS (Cont)

Time Notation	Command/Description
	IB2 SEQUENCE
T4.4	Adder → Z1. Same as in IB1-sequence except $B2_f$ flip-flop 7XG20 is set (refer to plate P-13).
T1.1	I/O Translator → S0. Same as described in initial T1.1 except $B2_f$ flip-flop 7XG20 is set (refer to plate P-13).

NOTE: For those commands not described here is detail, refer to table 2-19, I-Sequence. For the setting and clearing of the Upper and Lower Rank ff's, refer to paragraphs 2-148 through 2-157.

2-444. EXECUTION OF IN, OUT, AND EXF INSTRUCTIONS (f = 50:11 through 50:13). The f = 50:11 through 50:13 instructions set up ID, OD, or EF buffer operations on a particular channel with specified buffer limits.

2-445. IN (f = 50:11). The IN instruction enables input-data operations on channel k by setting its ID Active flip-flop. The terminal address-control word (TACW) is taken from the next consecutive address after this instruction. The initial address-control word (IACW) is taken from the next consecutive address after TACW. These control words specify the area of memory involved in the data-exchange operation.

2-446. OUT (f = 50:12). The OUT instruction enables output data operations on channel k by setting its OD Active flip-flop. The origins of TACW and IACW are the same as for f = 50:11.

2-447. EXF (f = 50:13). The EXF instruction enables external functional operations on channel k by setting its EF Active flip-flop. The origins of TACW and IACW are the same as for f = 50:11.

2-448. Execution Sequences. Three execution sequences are required to execute IN, OUT AND EXF instructions: I-sequence, B1-sequence, and B2-sequence. In addition, the initial portion of the I-sequence for the next instruction is used to store IACW in control memory. During the I-sequence which obtains the instruction from memory, the channel number is placed in register KO. The IB1-sequence is created when the I and B1 sequences run in parallel to obtain TACW from memory. The IB2 sequence is created when the I and B2 sequences run in parallel to store TACW in control memory, obtain IACW from memory, and set the proper active flip-flop.

2-449. I-Sequence Data Flow For f = 50:11 through 50:13. I-sequence operations are as previously described. At the end of the I-sequence, KO contains the channel number. Only the four least significant bits of KO are interpreted.

2-450. IB1 Sequence. The IB1 sequence uses a memory reference to obtain TACW from the address contained in the P-register. The P-register is advanced to the next consecutive address after this instruction by the Advance-P subsequence, during the I-sequence which obtained the instruction. The Advance-P subsequence is used during the IB1 sequence to increment the P-register to the address of IACW. The TACW is applied to one side of the adder from the D-register. The X-register is set to all 1s (ones) and applies a -0 value to the other side of the adder. The adder then outputs an unmodified TACW which is placed in the Z1-register. The I/O translator register is set according to the channel number from KO, and the type of active flip-flop to be set from the function code translator.

2-451. IB2 Sequence. The IB2-sequence uses a control-memory reference to store TACW. The storage address is dependent upon the function code and the channel number supplied by the I/O translator register. The gating of control memory to the ZO-register is disabled during the read portion of the memory cycle, which destroys original memory content. Another memory cycle is used to obtain IACW from the address contained in the P-register IACW passes unmodified, through the X-D' adder and is placed in the Z1-register. The Advance-P subsequence is used to increment the P-register to the address of the next instruction in the program. The initial portion of the I-sequence for the next instruction initiates

a control memory reference to store IACW. The storage address for IACW is one greater than that for TACW. The IACW storage address is formulated by setting the SO-register from the I/O translator register, and by setting SO_{00} to 1_2.

2-452. IB1 and IB2 Sequence Data Flow for f = 50:11 through f = 50:13. Refer to figure 2-105 for a block diagram description of the executive of IB1 and IB2-Sequence for f = 50:11 through 50:13.

2-453. IB1 and IB2 Sequence Essential Commands for f = 50:11 through 50:13. Refer to table 2-105 for a list of essential commands for f = 50:11 through 50:13. Since IB1 and IB2, and I-sequences are run in parallel, most of the commands listed have been described in a preceding I-sequence. All commands have been previously covered. Refer to table 2-106 for setting of function translator Flip-Flops OXG04 and OXG05.

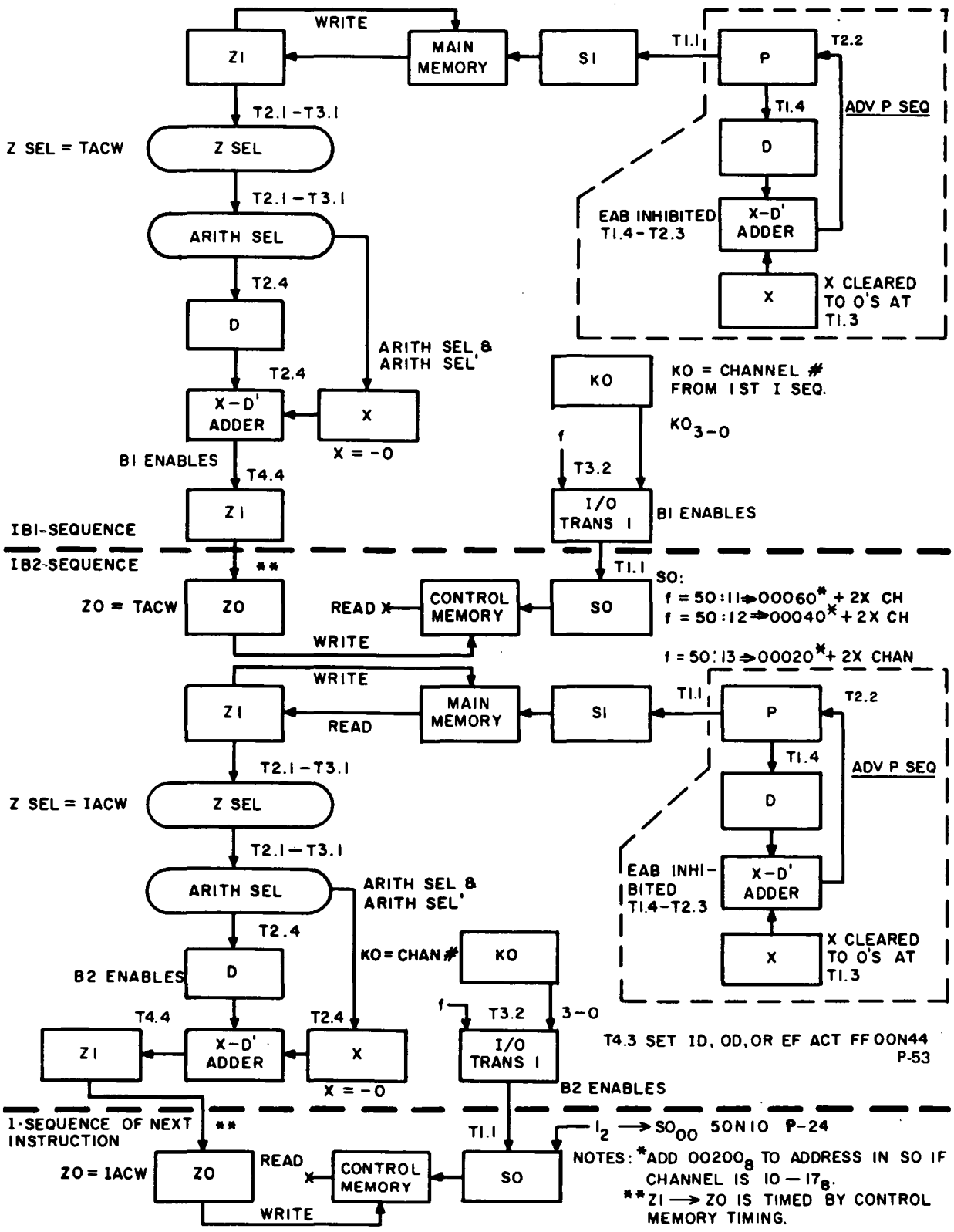


Figure 2-105. IB1 and IB2-Sequence Data Flow for f = 50:11 - 50:13

TABLE 2-105. IB1, IB2, AND NEXT I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:11 - 50:13

Time Notation	Commands
	IB1-SEQUENCE
T4.4	Clear S1
T1.1	P → S1, Init Memory, *set Incr P ff
T1.3	*Clear D, *clear X, clear Z1, *set OXL11 ff
T1.4	$*P_L \rightarrow D_L$, $*P_U \rightarrow D_U$, *set Inhib EAB ff
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel, *clear Incr P ff
T2.2	*Clear P, *adder → P
T2.3	*Clear OXL11 ff, *clear Inhib EAB ff, clear D, clear X
T2.4	Arith Sel → D, Arith Sel → X and Arith Sel' → X (set X = 1's)
T3.1	Clear I/O Trans 1, drop Z1 → Z-Sel, drop Z-Sel → Arith Sel
T3.2	f • K0 → I/O Trans 1
T4.3	Clear Z1
T4.4	Adder → Z1, ** disable CM → ZO, clear S1
	IB2-SEQUENCE
T1.1	I/O Trans 1 → SO, Init CM, P → S1, Init Memory, *set Incr P
T1.3	*Clear D, *clear X, clear Z1, *set OXL11 ff
T1.4	$*P_L \rightarrow D_L$, $*P_U \rightarrow D_U$ *set Inhib EAB ff, drop disable CM → ZO.
T2.1	Z1 → Z-Sel, Z-Sel → Arith Sel, *clear Incr P ff
T2.2	*Clear P, *Adder → P
T2.3	*Clear OXL11 ff, *clear Inhib EAB ff, clear D, clear X

TABLE 2-105. IB1, IB2, AND NEXT I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:11 - 50:13 (Cont)

Time Notation	Commands
T2.4	Arith Sel → D, Arith Sel → X and Arith Sel' → X (set X = 1's)
T3.1	Clear I/O Trans 1, drop Z1 → Z-Sel, drop Z-Sel → Arith Sel
T3.2	f · K0 → I/O Trans 1
T4.3	Clear Z1, ***set ID, OD, or EF Act ff
T4.4	Adder → Z1, **disable CM → ZO
	I-SEQUENCE OF NEXT INSTRUCTION
T1.1	I/O Trans 1 → SO, 1_2 → SO_{00}, Init CM
T1.4	Drop disable CM → ZO

*These events are concerned with or are controlled by the Advance-P subsequence.

**Z1 → ZO is timed by control memory timing.

***if f = 50:11, set ID Act ff
if f = 50:12, set OD Act ff
if f = 50:13 set EF Act ff

} Refer to Table 2-99 (T4.3 Set Act ff if f = 50:01 - 50:03) for a Detailed Description of command.

I/O translator 1 selects the channel number.

Refer to Tables 2-19 and 2-104 for a Detailed Description of all other commands.

TABLE 2-106. FUNCTION TRANSLATOR INDICATIONS

Translator Flip-Flops		Indications	Instruction f =
OXG05	OXG04		
Clear	Set	20G05 = H, EF	50:13
Set	Clear	20G06 = H, OD	50:12
Set	Set	20G04 = H, ID	50:11

2-454. PRIORITY SIGNAL DETECTION AND SELECTION. The I/O section of the computer handles signals generated by external sources as well as signals which originate internally. External signal types include External Interrupt Request (EIR), External Function Request (EFR), Output Data Request (ODR), Input Data Request (IDR), and External Sync. The first four of these signals (EIR, EFR, ODR, and IDR) are channel types, meaning that each of the 16 channels can generate these requests. All other signals are generated from within the computer and among them are monitor interrupts (EI Mon, EF Mon, OD Mon, and ID Mon). These interrupt signals are also channel types.

2-455. The computer can handle only one signal at a time; others must wait. To determine what signal is to be handled, a priority scheme is established by the logic to service signals in a specific sequence. The priority scheme has two levels; one for channel priority, as required by channel-type signals, and the other for function priority, to select the type of signal to be serviced. Refer to table 2-107 for the signal (types listed in order of function priority) presented in I/O logic.

TABLE 2-107. FUNCTION PRIORITY FOR SIGNALS PRESENTED TO INPUT/OUTPUT LOGIC

Channel Priority	Function Priority	
RTC Request (Non-Channel)		
Data Requests (Channel Type)	EIR EFR ODR* IDR*	disabled by External Interrupt Lockout flip-flop set
Special Interrupts (Non-Channel)	Instruction Fault Resume Fault RTC Mon Ext Sync RTC Overflow	disabled by All Interrupt Lockout flip-flop set
Monitor Interrupts (Channel Type)	EI Mon EF Mon OD Mon* ID Mon*	

Signals are listed in their order of function priority.

*Priority of these signals can reverse.

2-456. If a data-request or monitor-interrupt type signal is to be honored, channel priority is considered first. Then, the type of signal is selected from those of the function group which are present on the selected channel. The priority scheme is discussed in more detail later in this section. The priority logic for data requests and monitor interrupts is discussed elsewhere in this section. The priority logic for special interrupts is analyzed in a later section.

2-457. Data Request and Monitor Interrupt Priority. Priority logic is used to perform actual selection of one of the signals presented to it during one of the two scan periods. The scan periods are described later in this section. Channel logic is distributed among four chassis. Each chassis is connected to four channels. The channels are divided into two groups. Channels 0 to 7 are considered the lower group. Channels 10 to 17 are considered the upper group. For each group, channel priority selects the highest-numbered channel which has a signal present. Signals of the selected channel enter function priority, which determines the type of signal to be honored for the selected channel. Group priority then determines that a signal selected from the upper group will be honored before a signal from the lower group. The overall priority setup is such that the highest-numbered channel which carries a signal is selected, and one signal type on that channel is selected.

2-458. Data Request Scan. A certain period of the Main Timing cycle is used to scan computer input signals for data requests. Data requests include external interrupt, external function, input data, and output data requests. These are all signals from external devices which require some computer service. The priority scheme is altered if an IDR or another ODR occurs immediately after an ODR is honored. In this case, the IDR has higher priority than an ODR. The priority of ODR's and IDR's is alternated if they occur continuously. (Real Time Clock (RTC) operations are discussed in a later section.)

2-459. Interrupt Request Scan. If no data request was detected during the scan period just described, another scan occurs for monitor interrupts. The same priority logic is used to select a particular signal to be honored. Refer to table 2-107 for monitor interrupt signal types. If an ODR was previously honored and an IDR or EIR was not since honored, an ID Mon has higher priority than an OD Mon. (Special interru operations are discussed in a later section.

2-460. Request Signal Entrance Logic. The logic which detects the I/O signals is constructed in a one-shot manner. This arrangement prevents the same signal from being honored more than once. Any signal which is selected and honored must drop and reappear before it can again be detected.

2-461. IDR Signal Detection. IDR signal detection is typical of the method by which all input signals are handled. Refer to logic diagram plate P-61 for channel 0, 1 signals input logic. The logic is either 0, 10_8, 1, or 11_8, depending upon the particular chassis. In the case of IDR signal detection, the ID Request flip-flop OXRg0 provides the one-shot function. A low logic level from O5Yg0 represents an IDR being applied to the computer. If the IDR is absent, the ID Request flip-flop is set during $\emptyset 2$. A resulting low-level enable is applied to gate 05Rg0 on pi 8. The low-level enable on pin 7 is present only if the ID Active flip-flop is set. This flip-flop must have been set by previously executed f = 50:01 or 50:11 instruction. The low level on pin 6 of 05Rg0 is present if th

input circuitry is available. Actually, this line is at a low level if the input acknowledge signal is not currently being generated. The low-level enable on pin 5 of 05Rg0 is present during the signal scan period. 05Rg0 is fully enabled if the IDR is present which applies a low level on its pin 9. The resulting H⇒ ID REQ output of 05Rg0 applies this IDR to the priority logic on this particular chassis. The 80Rg0 output is applied to channel priority (refer to plate P-65 or P-65A) and the 16Rg4 output to function priority (refer to plate P-66).

2-462. If this IDR is selected by priority logic and honored, the ID Request flip-flop 0XRg0 is cleared to provide the signal one-shot function. Pin 10 of 0XRg0 becomes a low level during the honoring operation. Pin 9 is a low level, indicating that this channel is being serviced. The clearing of this flip-flop presents a high-level disable to gate 05Rg0, pin 8. Therefore, the recognition of the IDR signal on this channel is prevented until flip-flop 0XRg0 is set. The setting of 0XRg0 can only occur if the external equipment drops the IDR signal. The setting is timed by $\emptyset 2$. EIR, EFR, and ODR signals pass through similar entrance logic. EF and OD buffers can be active at the same time on the same channel.

2-463. I/O SIGNAL SELECTION. Reference should be made to figures 2-106 and 2-107 with the following discussion of I/O signal selection. Circuit elements are shown in figure 2-107.

2-464. Data-Request Scan. All signals are presented to the one-shot entrance gates. The Chan Req → Chan Prio (T4.2 through T1.2) command forces channel priority on each chassis to enable all channels. The resulting enables to the one-shot gates allow all input signals to be detected. However, only outputs of the gates for EIR, EFR, ODR, and IDR signals are allowed to enter channel priority at time T.1. These data-request signals are enabled by the Data Req → Chan Prio command. The monitor-interrupt signals do not enter channel priority at this time because of the absence of the Mon Req → Chan Prio command. As soon as the channel priority on each chassis receives the request signals, channel selection is made. The channel-priority register for channels 0, 2, 4, and 6 is connected to the channel-priority logic for channels 1, 3, 5, and 7 vice versa. This interconnection causes the channel-priority logic for these two chassis to act as one priority system. This system then selects the highest-numbered channel of 0 through 7 which carries an input signal. The channel priority logic for channels 10 through 17 is likewise interconnected. Channel priority logic then selects one channel of 10 through 17. The Chan Req → Prio command drops at time T1.2. By this time, channel priority has performed the selection. Since channel priority is no longer forced to indicate all channels, only the selected channel one-shot gates are enable (one gate of channels 0 through 7, if there is an input signal present and one gate of channels 10 through 17, if there is an input signal present). The Data Req Fct Prio commands enter the function-priority register with signals from the enable one-shot gates. Only signals on selected channels are affected. Function priority presents a high logic level to group priority if it contains a signal. Group priority selects the upper group (channels 10 through 17) if there is an associated input signal; otherwise, the lower group (channels 0 through 7) is selected. One particular channel is chosen after group priority, and function priority selects the type of signal on this channel. If a signal was processed during the events just discussed, the $I/01_i$ sequence flip-flop is set at time T2.2. At this time, the selected signal information is passed to the function,

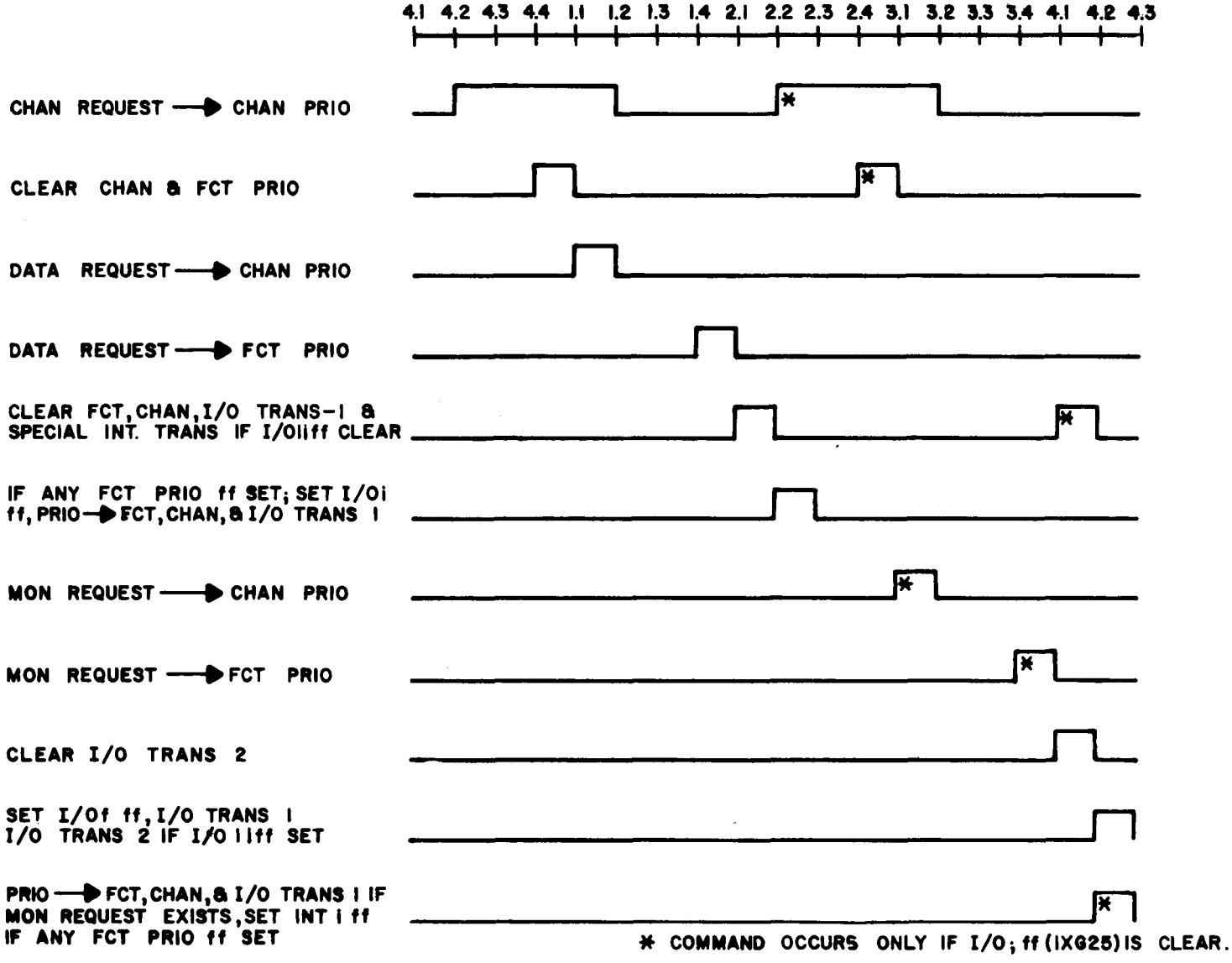
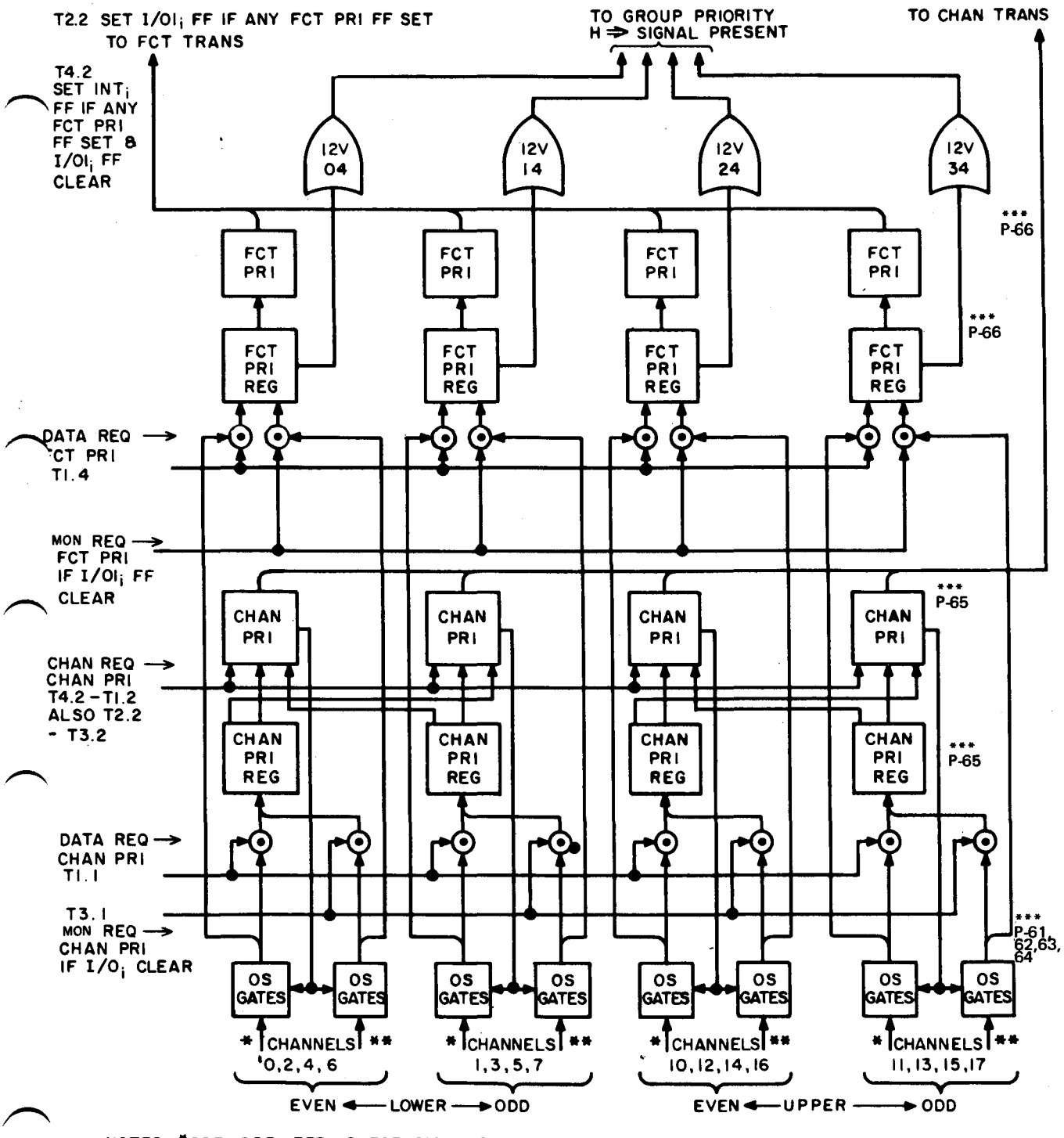


Figure 2-106. Scan Sequence Timing Cycle



NOTES: *ODR, IDR, EFR, & EIR SIGNALS.
**OD MON, ID MON, EF MON, & EI MON SIGNALS (INTERNALLY GENERATED).
***PLATES IN LOGIC DIAGRAMS.

Figure 2-107. Input/Output Signal Selection Flow (Sheet 1 of 2)

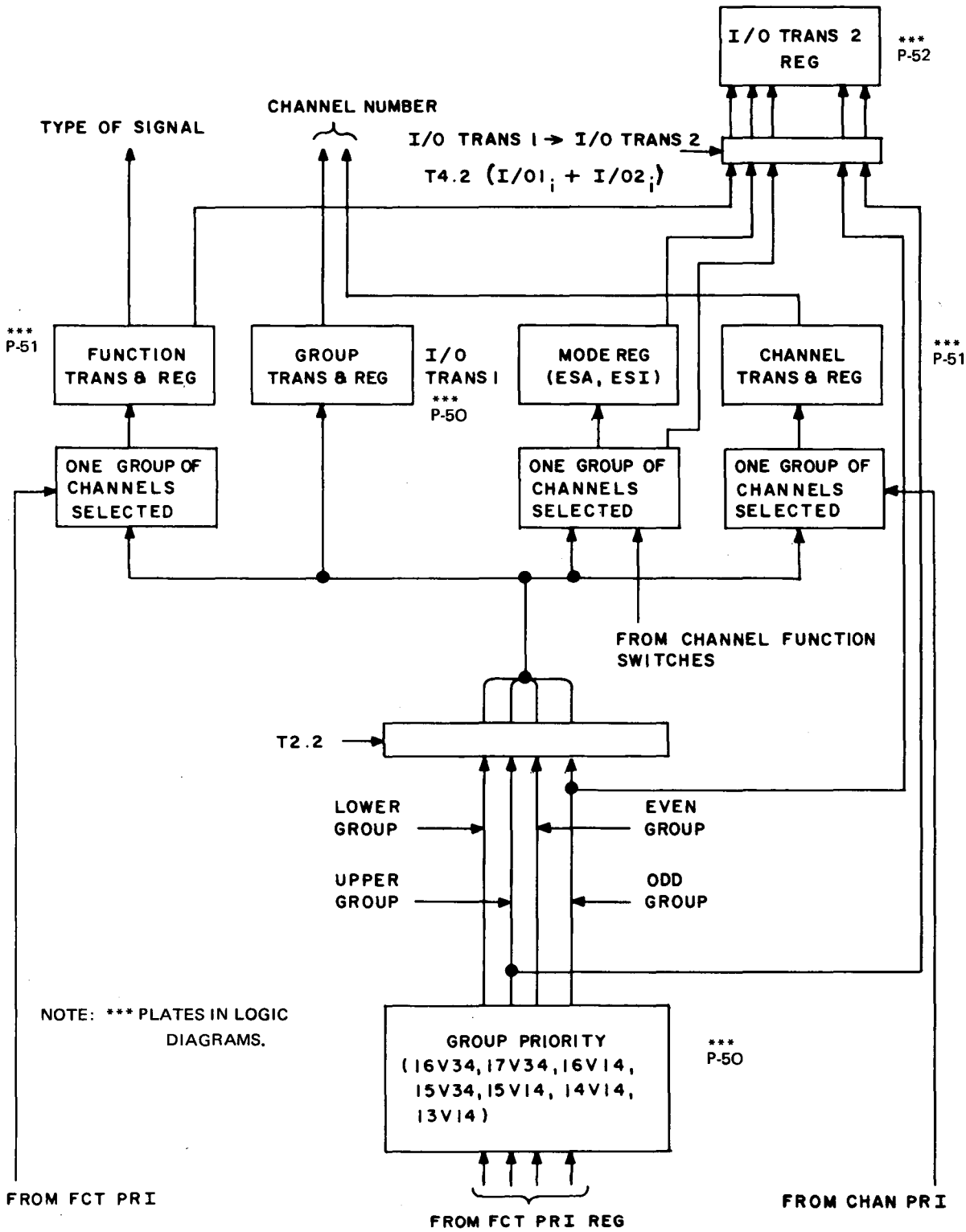


Figure 2-107. Input/Output Signal Selection Flow (Sheet 2 of 2)

channel, and I/O translators. Outputs from the translators are used to control the computer operations to effect honoring of the selected signal.

2-465. Interrupt Request Scan. If the $I/O1_i$ flip-flop was not set during the data request scan, no EIR, EFR, ODR nor IDR signal was detected. A scan is then made for monitor interrupts (refer to figure 2-106 and 2-107). The Chan Req → Chan Prio command occurs again at time T2.2 - T3.2 to allow channel priority to enable all one-shot entrance gates. During this scan period, however, only the entrance gates for monitor interrupts are allowed to pass signals to channel priority because of the presence of the Mon Req → Chan Prio command. Channel Priority functions in the same manner as described for data requests. During the Mon Req → Fct Prio command, signals of the selected channels enter function priority which selects one type of signal. Group priority selects on particular channel. At the next time T4.2, signal information is passed to the translators. If a monitor-interrupt signal was detected during the events just discussed, the Int Sequence flip-flop is set at time T4.2. Setting of this flip-flop is prevented by the $I/O1_i$ Sequence flip-flop if a data request signal is detected during the first scan.

2-466. I/O Signal Selection Essential Commands. Refer to table 2-108 for a list of essential commands concerning I/O signal selection. Events follow the commands shown that honor the selected signal. These events, and RTC operations, are discussed later in this section. Commands not previously described are described in table 2-109 in detail. Since many commands in I/O are executed exactly the same, most commands are only described in detail once. Table 2-110 is a list of commands, starting at T1.1, showing where the command was described previously.

TABLE 2-108. SCAN SEQUENCE ESSENTIAL COMMANDS

Time Notation	Commands
T4.2	Chan Req → Chan Prio
T4.4	Clear Chan and Fct Prio
T1.1	Data Req → Chan Prio
T1.2	Drop Chan Req → Chan Prio
T1.4	Data Req → Fct Prio
T2.1	Clear Fct, Chan, I/O Trans 1, and special interrupt translator
T2.2	Set $I/O1_i$ ff if any Fct Prio ff set
	Prio → Fct, Chan, and I/O Trans 1 if any Fct Prio ff set
	Chan Req → Chan Prio if $I/O1_i$ ff clear
T2.4	Clear Chan and Fct Prio if $I/O1_i$ ff clear
T3.1	Mon Req → Chan Prio if $I/O1_i$ ff clear
T3.2	Drop Chan Req → Chan Prio
T3.4	Mon Req → Fct Prio if $I/O1_i$ ff clear
T4.1	Clear I/O Trans 2, clear Fct, Chan, I/O Trans 1 and special interrupt Translator if $I/O1_i$ ff clear
T4.2	Set $I/O1_f$ ff if $I/O1_i$ ff set
	Set Int_i ff if any Fct Prio ff set · ($I/O1_i$ ff clear)
	I/O Trans 1 → I/O Trans 2 if $I/O1_i$ ff set
	Prio → Fct, Chan, and I/O Trans 1 if $I/O1_i$ ff clear and if Monitor Request exists

TABLE 2-109. I/O SIGNAL SELECTION COMMANDS/DESCRIPTIONS

Time Notation	Command/Description
T4.2	Chan Req → Chan Prio. Gate 00N61 (refer to plate P-49) outputs a high due to a low input from 01T11. The high is inverted by 00N65 and applied as a L ⇒ Clear Prio to 6gN65 (refer to plate P-58) where it is inverted and applied as a H ⇒ Chan Req → Chan Prio into gates 11Vg0 through 11Vg3 (refer to plate P-65 & 65A).
T4.4	Clear Chan and Fct Prio. The high out of 6gN65 (refer to plate P-58) is inverted by 7gN65 and at ∅4, 8gN65 produces a H ⇒ Clear Func Prio to clear flip-flops OXVg4 through OXVg7 (refer to plate P-66) and 9gN65 produces a H ⇒ Clear Chan Prio to clear flip-flops OXVg0 through OXVg3 (refer to plate P-65 & 65A).
T1.1	Data Req → Chan Prio. The high out of 00N61 (Refer to plate P-49) is applied as a H ⇒ Data Req → Chan Prio into 8gN61 (refer to plate P-58) where it is inverted, and at ∅1, 9gN61 produces a L ⇒ Data Req → Chan Prio into flip-flops OXVg0 through OXVg3 (refer to plate P-65 & 65A).
T1.2	Drop Chan Req → Chan Prio. Gate 00N61 (refer to plate P-49) now inputs a low due to the absence of low from 01T11 and drops Chan Req → Chan Prio as described at T4.2.
T1.4	Data Req → Fct Prio. Gate 00N62 (refer to plate P-49) outputs a high due to a low from 03T14 and the absence of H ⇒ I/O1 • (Dual + ESI Term) + Cont Data Req. The high is inverted by 8gN62 (refer to plate P-58) and at ∅4, 9gN62 produces a L ⇒ Data Req → Function Prio into OXVg4 through OXVg7 (refer to plate P-66).
T2.1	Clear Fct, Chan, I/O Trans I, and Special Interrupt Translator. Gate 00E00 (refer to plate P-49) outputs a high due to a low from 03T22 and the absence of $I/O1_i$ • (Dual +ESI Term) + Cont Data Req. The high is inverted by 01E00 and at ∅1, 02E00 outputs a H ⇒ Clear Translator via 04E00 to clear flip-flops OXG07, OXG08, OXG00, OXG11 and OXG10 (refer to plate P-50) and 20G15 through 20G18 (refer to plate P-55). The high from 02E00 is also inverted by 03E00 to output a L ⇒ Clear Translator to clear flip-flops OXG01, OXG02, OXG04 and OXG05.
T2.2	Set $I/O1_i$ ff if Any Fct Prio ff Set. If any of the Function Priority flip-flops OXVg4 through OXVg7 are set (refer to plate P-66) 12Vg4 will output a H ⇒ Request into 13V04 (refer to plate P-50) which produces a L ⇒ I/O Req to $I/O1_i$ flip-flop 1XG25 pin 13 (refer to plate P-13). Gate 11E25 is enabled by the low from 10E25 (which is disabled by the absence of an R1 Seq), and the low from 03T22 at ∅2, which causes 1Xg25 to set.

TABLE 2-109. I/O SIGNAL SELECTION COMMANDS/DESCRIPTIONS (Cont)

Time Notation	Command/Description
T2.2 (Cont)	Prio → Fct, Chan, and I/O Trans 1 if Any Fct Prio ff set. Refer to Function Priority (paragraph 2-470) for a description and setting of flip-flops (refer to plate P-66) and to Group Priority (paragraph 2-473) for a description of the group priority logic (refer to plate P-50). With any of the Function Priority flip-flops set (refer to plate P-66), 12Vg4 will produce a H ⇒ Request to one or more of the group priority gates (refer to plate P-50). A high is always applied to the 13V04 gate. Assuming a high is received from 12V34 (refer to plate P-66), gates 13V04, 15V34, 15V14, 14V14, and 17V34 will output lows (refer to Plate P-50). The low from 13V04 is applied to flip-flop OXG08 and the low from 15V34 is applied to flip-flop OXG07. The L ⇒ Request Gate into the flip-flops is developed from 15E00 (refer to plate P-49). To develop this enable, 10E00 is enabled by a low from 03T22, a low from 40E00 and 20E00 (disabled by the absence of a command or instruction) and the absence of a H ⇒ I/O1 · (Dual + ESI Term) + Cont Data Seq. The high output is inverted by 12E00 to enable 15E00 at Ø2. Gate 15V14 (refer to plate P-50) outputs a L ⇒ Odd Group to 13E00 (refer to plate P-49) which produces a L ⇒ Odd Channel Gate at Ø2 to flip-flops OXG00, OXG11 and OXG10 (refer to plate P-50). A L ⇒ Odd Channels is also applied to the Function translator flip-flops OXG05 and OXG04 and to the Channel Translator flip-flops OXG01 and OXG02 (refer to plate P-51). Chan Req → Chan Prio if $I/O1_i$ ff Clear. If $I/O1_i$ flip-flop 1XG25 is clear (refer to plate P-13) a L ⇒ $\overline{I/O1}$ Seq is applied to 13G25 (refer to plate P-14). The other input is also low because $I/O2_i$ flip-flop is not set. The high output is inverted by 14G25, due to the absence of a L ⇒ Cont Data Seq, and is applied to 00N63 (refer to plate P-49) as a L ⇒ I/O, which outputs a high to 00N65 due to the low from 01T31. The low from 00N65 now produces a Chan Req → Chan Prio as described at T4.2.
T2.4	Clear Chan and Fct Prio if $I/O1_i$ ff Clear. The low from 00N65 (refer to plate P-49) as developed at time T2.2 also clears Chan and Fct Prio as described at T4.4.
T3.1	Mon Req → Chan Prio if $I/O1_i$ ff Clear. The high that was developed at time T2.2 out of 00N63 (refer to plate P-49) is also applied as a H ⇒ Mon Req → Chan Prio into 8gN63 (refer to plate P-58) where it is inverted and at Ø1, 9gN63 outputs a L Mon Req → Chan Prio into flip-flop OXVg0 through OXVg3 (refer to plate P-65 & 65A).
T3.2	Drop Chan Req → Chan Prio. Gate 00N63 (refer to plate P-49) that developed a high as described at time T2.2 now produces a low due to the loss of a low from 01T31 and Chan Req → Chan Prio is dropped.

TABLE 2-109. I/O SIGNAL SELECTION COMMANDS/DESCRIPTIONS (Cont)

Time Notation	Command/Description
T3.4	Mon Req → Fct Prio if $I/O1_i$ ff Clear. Gate 00N64 (refer to plate P-49) outputs a H ⇒ Mon Req → Func Prio due to the absence of a H ⇒ I/O and H ⇒ f = I/O, and the low from 03T34. The high is applied to 8gN64 (refer to plate P-58) where it is inverted and at ∅4, 9gN64 produces a L ⇒ Mon Req → Func Prio into flip-flops OXVg4 through OXVg7 (refer to plate P-66).
T4.1	Clear I/O Trans 2. Gate 32E00 (refer to plate P-49) produces a H ⇒ Clear Translator 2 due to a low from 05T42 and at ∅1 the high is applied to flip-flops 4XG09, 4XG00, 4XG10, 4XG06, 4XG05 and 4XG07 (refer to plate P-52) which clear at this time. Clear Fct, Chan, I/O Trans 1 and Special Interrupt Translator if $I/O1_i$ ff Clear. Gate 00E00 (refer to plate P-49) outputs a high due to a low from 05T42 and the absence of a H ⇒ I/O and H ⇒ f = I/O and will now clear Fct, Chan 1, I/O Trans and Special Interrupt Translator as previously described at time T2.1.
T4.2	Set $I/O1_f$ ff if $I/O1_i$ ff Set. If $I/O1_i$ flip-flop 1XG25 (refer to plate P-13) is set, a L ⇒ I/O1 Seq is applied to I/O 1_f flip-flop OXG25 (refer to plate P-12) and at ∅2, the low from 22E20 sets $I/O1_f$ flip-flop. Set Int_i ff if Any Fct Prio ff Set ($I/O1_i$ ff Clear). The L ⇒ I/O Int from 13V04 (refer to plate P-50) is developed as described at time T2.2 and applied and inverted by 14V04 (refer to plate P-12) to 21G21. With the other inputs to 21G21 high, the low output at ∅2 sets Int_i flip-flop OXG21. I/O Trans 1 → I/O Trans 2 if $I/O1_i$ ff Set. The H ⇒ I/O1 Seq (refer to plate P-13) due to flip-flop 1XG25 being set, is applied to 13G25 (refer to plate P-14) and a L ⇒ I/O Seq via 15G25 is applied to 33E00 (refer to plate P-49) which is enabled by a low from 05T42 and at ∅2 the L ⇒ Translator 1 → Translator 2 from 33E00 is applied to flip-flops 4XG00, 4XG05 through 4XG07, 4XG09 and 4XG10. Prio → Fct, Chan, and I/O Trans 1 if $I/O1_i$ ff Clear and if Monitor request exists. This command is executed the same as described at time T2.2 except that gate 11E00 (refer to plate P-49) is enabled instead of 10E00 to enable 15E00 and 13E00 or 14E00. Gate 11E00 is enabled by a low from 05T42, the absence of a H ⇒ Spec Int Req and the absence of a H ⇒ I/O.

TABLE 2-110. LIST OF COMMANDS AND TABLES WHERE DETAILED DESCRIPTION OF COMMAND CAN BE FOUND

Time	Command	Table
T1.1	Clear ZO	2-19
	I/O Trans → SO, 1 → SO_{00}, B → S1	2-122
	Store Sel L1 → SO, Init CM	2-128
	I/O Trans → S1, 1 → $S1_{00}$, 1 → $S1_{06}$	2-128
T1.2	Input Data Amps → Store Select	2-122
	Drop ODD Chan Input Data Amps → Store Sel	2-128
	Even Chan Input Data Amps → Store Sel	2-128
	Clear All EF/OD Ack Reg ff	2-136
	Set 5XLg3 ff	2-140
T1.3	Clear Z1	2-19
	Clear ID Act ff if $\overline{(\text{Term ff clear})}$ · (Cont Data Req ff clear)	2-122
	Set ID Mon ff if $\overline{(\text{Term ff clear})}$· ($B_{16}$=1)	2-122
	Set EF/OD Ack Reg ff	2-136
	Clear OD Req ff if (1C· EF) + $\overline{(1C \cdot ODR)}$	2-136
	Set OD Mon ff if $\overline{(\text{Term ff clear})}$ · (B_{16}=1)	2-136
	Set Prio Alternator ff	2-136
	Set Term OD ff if $\overline{(\text{Term ff clear})}$	2-136
	Set or Clear EF/OD Mode ff	2-136
	Clear EF/OD Ack Gener ff, Drop EF/OD Ack Sig	2-139
	Clear EF Req ff	2-147

TABLE 2-110. LIST OF COMMANDS AND TABLES WHERE DETAILED DESCRIPTION OF COMMAND CAN BE FOUND (Cont)

Time	Command	Table
	Set EF Mon ff if ($\overline{\text{Term}}$ ff clear) · (B16=1)	2-147
	Clear Hold 2 ff	2-154
	Clear RTC Mon Req ff if RTC Mon ff Set	2-163
T1.4	Store Select → Z1	2-122
	Set 1D Ack Gen ff if Dual ff clear	2-122
	Set 1D Ack Gen ff, Send 1D Ack	2-125
	Set 6XLg3 ff	2-125
	Set Ack Delay ff if Dual ff clear	2-136
T2.1	Clear FCT, Chan & I/O Trans I & Special Interrupt Translator	2-109
	Drop Odd Chan Input Data Amps → Store Sel	2-128
	Z1 → Z Sel	2-19
	Clear I/O Trans I	2-154
	Drop Input Data Amps → Store Sel	2-122
	Clear RTC Seq ft	2-163
T2.2	Set $I/01_i$ ff	2-122
	Prior → FCT, Chan & I/O Trans I	2-109
	Set ESI ff	2-128
	Set ESA ff	2-130
	Set $I/02_i$ ff if $\overline{\text{Term}}$ ff clear	2-128
	Odd Chan Input Data Amps → Store Sel ff if $I/02_i$ ff set	2-128

TABLE 2-110. LIST OF COMMANDS AND TABLES WHERE DETAILED DESCRIPTION OF COMMAND CAN BE FOUND (Cont)

Time	Command	Table
	Clear 5XLg3	2-140
	Set Clear Hold ff, f·KO → I/O Trans I	2-154
	Set I/02 ff if Dual ft set	2-122
	Set RTC Seq ff	2-163
	Set Chan Trans ff's	2-163
	Chan Req → Chan Prio if $I/01_i$ ff clear	2-109
T2.3	Clear C	2-136
T2.4	Clear Chan Prior, Clear FCT Prior	2-109
	Drop Disable Mem → Z1, Drop Disable CM → Z0	2-12?
	Z Sel → C	2-136
T3.1	I/O Trans → S0, 1 → $S0_{00}$, Init CM	2-122
	Store Sel L1 → SO (ESI)	2-12
	Clear I/O Trans 1	2-152
	Mon Req → FCT Pri	2-109
	Clear ZO	2-19
T3.2	Clear B	2-19
	Clear ID Ack ff, Drop ID Ack Sig	2-125
	Clear 6XLg3	2-126
	Clear Ack Delay ff	2-139
	Drop Chan Req → Chan Priority	2-109

TABLE 2-110. LIST OF COMMANDS AND TABLES WHERE DETAILED DESCRIPTION OF COMMAND CAN BE FOUND (Cont)

Time	Command	Table
.3	Clear 6XLg0 ff	2-125
T3.4	ZO → B	2-19
	Store Select → B	2-130
	Set EF/OD Ack Gener ff, Send EF/OD Ack Sig (FAST INTERFACE)	2-139
	Set EF/OD Ack Gener ff, Send EF/OD Ack Sig (SLOW INTERFACE)	2-140
	Set Hold 1 ff	2-152
	Mon Req → FCT Pri	2-109
T4.1	I/O Trans → SO, Init CM	2-122
	Clear ZO, Clear B± ff	2-19
	Clear I/O Trans 2	2-109
	Drop Disable B Network 17, 16	2-122
	Set 1D Ack Gener ff	2-122
	Store Sel L1 → SO (ESI)	2-128
	Set Resume ff	2-139
	Clear OD Act ff if Term EF/OD ff Set	2-139
	Clear FCT, Chan, I/O Tran I	2-109
	Clear I/O Trans 2 & Special Interrupt if $I/O1_i$ ff clear	2-109
.4.2	Set $I/O1_f$ ff	2-122
	Set B± ff if B_{17}=1	2-122
	Disable B-Network 17, 16	2-122

TABLE 2-110. LIST OF COMMANDS AND TABLES WHERE DETAILED DESCRIPTION OF COMMAND CAN BE FOUND (Cont)

Time	Command	Table
	Clear All 1D Ack Reg ff	2-12
	I/O Trans 1 → I/O Trans 2	2-109
	Clear Hold 1 ff	2-154
	Prio → FCT, Chan, & I/O Trans 1 if $I/O1_i$ ff	2-109
	Clear & if Monitor Requests exists	
T4.3	Clear Pri Alternator ff	2-122
	Set 1D Ack Reg ff	2-122
	Set 6XLg0	2-122
	Clear $\overline{\text{Term}}$ ff	2-12
	Clear ID Req ff	2-122
	Clear Clear Hold ff	2-154
	Set Hold 2 ff	2-15
	Clear Resume ff	2-136
T4.4	Set $\overline{\text{Term}}$ ff if $(B_{15-0} \neq Z0_{15-0})$	2-122
	Set Cont Data Req ff if $(B_{15-0} = Z0_{15-0}) \cdot (ZO_{17}=1)$	2-122
	Disable CM → ZO	2-124
	Disable Mem → Z1	2-124
	Clear Term EF/OD ff	2-139
	Set $\overline{\text{Term}}$ ff if $B_{17-0} \neq Z0_{17-0}$	2-163

2-467. CHANNEL PRIORITY SELECTION. Each I/O chassis has channel priority logic (refer to plate P-65 and 65A). Request signals enter channel priority by way of the Rg-gates during data-request scan period. If no request is detected, monitor interrupt signals enter by way of the 80Mg-gates during the interrupt-request scan period. The channel flip-flops are set if any signal is present on their channels, regardless of signal type. The 11Vg-gates provide the channel enable which opens all of the signal-entrance gates to allow all data request or monitor interrupts to enter channel priority. The 11Vg-gates have a common Chan Req → Chan Prio input which forces these gates to enable all channels. When this signal drops, all signals have been inputted to channel priority and one channel has been selected. The 10Vg-inputs to the 11Vg-gates enable one channel. Thus, when the signals enter function priority, only data-requests or monitor-interrupts of the selected channel are involved.

2-468. The channel-priority logic on two chassis of the same drawer are interconnected. These are the dotted-line inputs and outputs shown on plates P-65 and P-65A. These lines connect a chassis with odd channels to a chassis with even channels and vice versa. This interconnection allows selection of the highest channel number with a signal present, from among the eight channels of the particular drawer. Thus, only one channel priority of each drawer can indicate a selected channel. The 10Vg-gates perform actual channel selection. Refer to table 2-111 for the conditions which enable the 10Vg-gates. These gates are set by the Channel Priority flip-flops on both chassis of the same drawer.

2-469. As an example, assume flip-flops 0XV03, 0XV13, and 0XV02 (refer to plates P-65 and P-65A) are set, indicating a request on channels 6, 7 and 4 respectively. The high from 00V13 (refer to plate P-65) disable gate 10V03 and enables gate 14V03 (refer to plate P-65A). The low from 14V03 enables 13V03 whose high output disables 10V00, 10V01, and 10V02. Since all the 10V0 gates are disabled no chan enable signal will originate from chassis 2. The high from 00V13 (refer to plate P-65) also enables gate 14V13 whose low output enables 13V13. The high from 13V13 disables 10V10, 10V11, and 10V12 resulting in no chan enable signal from channels 1, 3, or 5. And gate 10V13 is enabled by a low on pin 11 from 01V13, this results in a channel enable signal from 11V13. This results in the selection of the request on channel 7.

2-470. FUNCTION PRIORITY SELECTION. (Refer to plate P-66). Each I/O chassis has function priority logic. Only signals which are present on the channel selected by channel priority are enabled to enter function priority. There is a separate flip-flop for each function type. Request signals enter this logic by way of the 16Rg-gates during the data-request scan period. If no request is detected, monitor interrupt signals enter by way of the 16Mg-gates during the interrupt-request scan period. Even though each I/O chassis has function-priority logic like that shown, only one function priority on each drawer will be presented with signals. This is because the two channel priorities on each drawer are interconnected so that only one channel may be selected. Then only the function priority on the chassis of selected channel will receive signals.

2-471. Gate 11Vg4, outputting a low level, indicates that at least one of the functions priority flip-flops is set on its chassis. Outputs from 12Vg6 and 12Vg7 (refer to plate P-66) will specify the type of selected signal. If 0XVg7 is set the high output from the clear side enables 11Vg7 and disables 12Vg6, this

TABLE 2-111. CHANNEL PRIORITY GATES, CHASSIS 1 AND 2

Chassis	Gates	Conditions to Output High Level
1	10V13	Chan 7
	10V12	Chan 5 • $\overline{\text{chan 7}}$ • $\overline{\text{chan 6}}$
	10V11	Chan 3 • $\overline{\text{chan 7}}$ • $\overline{\text{chan 6}}$ • $\overline{\text{chan 5}}$ • $\overline{\text{chan 4}}$
	10V10	Chan 1 • $\overline{\text{chan 7}}$ • $\overline{\text{chan 6}}$ • $\overline{\text{chan 5}}$ • $\overline{\text{chan 4}}$ • $\overline{\text{chan 3}}$ • $\overline{\text{chan 2}}$
	12V11	Chan 5 or 7 selected
	12V10	Chan 3 or 7 selected
2	10V03	Chan 6 • $\overline{\text{chan 7}}$
	10V02	Chan 4 • $\overline{\text{chan 7}}$ • $\overline{\text{chan 6}}$ • $\overline{\text{chan 5}}$
	10V01	Chan 2 • $\overline{\text{chan 7}}$ • $\overline{\text{chan 6}}$ • $\overline{\text{chan 5}}$ • $\overline{\text{chan 4}}$ • $\overline{\text{chan 3}}$
	10V00	Chan 0 • $\overline{\text{chan 7}}$ • $\overline{\text{chan 6}}$ • $\overline{\text{chan 5}}$ • $\overline{\text{chan 4}}$ • $\overline{\text{chan 3}}$ • $\overline{\text{chan 2}}$ • $\overline{\text{chan 1}}$
	12V01	Chan 4 or 6 selected
	12V00	Chan 2 or 6 selected
Conditions for chassis 9 and 10 are similar. Add 10_8 to channel numbers above.		

TABLE 2-112. FUNCTION PRIORITY SELECTION INDICATIONS

Selected Signal Type	Priority Outputs	
	11Vg7	12Vg6
EI	L	L
EF	L	H
OD	H	L
ID	H	H

results in EI being selected since it has highest priority. If 0XVg5 is set the high from pin 8 enables 11Vg7 and disables 11Vg6 to partially enable 12Vg6. And gate 12Vg6 will output a high if there is no E1, the high from 12Vg and low from 11Vg7 indicate EF. If EF, EI and ID flip-flops are clear, and the OD flip-flop is set then 11Vg6 outputs a high causing 12Vg6 to output a low implying OD.

2-472. If the EI flip-flop is clear, 12Vg6 can output a high level if an ID type is selected. This selection is determined by 11Vg6. Gate 11Vg6 outputs a low level if EF or no OD type is present, or if an ID type is present and output was last. The last output indication into 10Vg4 means that the priority alternator is set (refer to plate P-68). This flip-flop is set whenever an OD or EF Acknowledge signal is set and is cleared whenever an ID acknowledge signal is sent. Both acknowledges are on this same chassis. This flip-flop then allows the function priority of OD and ID type signals to be reversed, depending upon the type of I/O operation performed last. The outputs of 11Vg7, 12Vg6, and 12Vg4 then indicate the presence and type of selected signal on this chassis. Refer to table 2-112 for the signal type translation indication from this logic.

2-473. GROUP PRIORITY SELECTION. Group priority logic is comprised of gates 16V34, 17V34, 16V14, 15V34, 15V14, 14V14, and 13V14 (refer to logic diagrams plate P-50). This logic receives indications from each of the four function priorities, by way of the 12Vg4 inputs (refer to plate P-66), as to whether or not these priorities contain a signal. A high level from a 12Vg4 gate indicates that a signal has been selected on this chassis. From this information, group priority selects the signal of the higher channel number, meaning I/O chassis that has the selected signal. The effect of chassis selection is to allow outputs of the channel priority and function priority on the selected chassis to enter the function and channel translators. Refer to table 2-113 for group priority selection indications. The group priority outputs are combined by I/O translator 1 to select one chassis.

2-474. GROUP TRANSLATOR (I/O TRANSLATOR 1). Group translator logic is comprised of flip-flops 0XG07, 0XG08, and 0XG00 and their output gates (refer to logic diagram, plate P-50). These flip-flops are set according to the selection made by group priority. Flip-flop 0XG00 is set by the odd-channel gate signal. Refer to logic diagrams (plate P-49) to confirm that this signal is dependent upon timing and the fact that group priority has selected an odd-channel chassis. Thus, 0XG00 being set indicates an odd-chassis signal has been selected. Its clear state indicates an even-chassis signal. The setting of flip-flop 0XG07 is dependent upon timing from the request-gate signal, and also upon the fact that group priority has selected an upper chassis (chassis 9 or 10). Thus, OXG07 being set indicates an upper-chassis signal has been selected. Its clear state indicates a lower-chassis signal. Flip-flop 0XG08 is set to indicate that a signal has been selected by priority and is present in the translator. The setting of this flip-flop is enabled by the same timing input as is 0XG07. The other enable to set 0XG08 is from 13V04. Notice that 13V04 is fed by the 12Vg4 gates of function priority. If any signal is present in any function priority, 13V04 outputs a low level. Thus, 0XG08 being set indicates a signal is present and enables the group-translator gates (20G10, 20G11, 20G12, and 20G13). Refer to table 2-114 for the group translator indications.

TABLE 2-113. GROUP PRIORITY SELECTION INDICATIONS

Group Priority Gates	Function Priority Gate Outputs upper group: chas 9 12V34			upper group: chas 10 12V24		lower group: chas 1 12V14	lower group: chas 2 12V04	Meaning of Group Priority Gate Outputs
13V14 = L						H		Chassis 1 signal
14V14 = H				(L	and	H)		(Chas 1 signal) • (no chas 10 signal)
15V14 = L	(H)	or		(L	and	H)		(Chas 9 signal) + (chas 1 signal • no chas 10 signal) - same as "odd group"
16V14 = L	(L)	and		(H	or	L)		(Chas 10 signal • no chas 9 signal) + (no chas 9 signal • no chas 1 signal). - same as (chas 10 signal) + (chas 2 signal)* - same as "even group"
15V34 = L & 17V34 = L	(H	or	H)					(Chas 9 signal) + (chas 10 signal) - same as "upper group"
16V34 = L	(L	and	L)					(No chas 9 signal) • (no chas 10 signal) - same as (Chas 1 signal) + (chas 2 signal)* - same as "lower group"

*This chart assumes a signal to be present on at least one chassis. Therefore, if a gate is conditioned by the absence of signals on a chassis, it indicates the presence of a signal on another chassis.

TABLE 2-114. GROUP TRANSLATOR INDICATIONS

Translator 0XG07	Flip-Flops 0XG00	Indications
Clear	Clear	20G10 = H, chassis 2 selected (lower-even group)
Clear	Set	20G11 = H, chassis 1 selected (lower-odd group)
Set	Clear	20G12 = H, chassis 10 selected (upper-even group)
Set	Set	20G13 = H, chassis 9 selected (upper-odd group)

A signal is assumed to be present (flip-flop 0XG08 set).

475. MODE TRANSLATOR (I/O TRANSLATOR 1). Refer to plate P-50. If the selected channel is odd, the position of the CHANNEL FUNCTION switch for the chassis is sensed. If the switch position is DUAL, ESI ESA, the associated flip-flop is set. The DUAL flip-flop is part of the I/O Translator 2, and its setting is performed at a later time if the output of 11G09 is a low level. If the selected channel is even, Single-Channel Mode is enforced.

2-476. CHANNEL TRANSLATOR. The channel translator is comprised of flip-flops 0XG02 and 0XG01 and their respective output gates (refer to logic diagrams plate P-51). These flip-flops are set according to the selection made by channel priority on the chassis selected by group priority. The channel-priority inputs are from the 12Vg-gates (refer to plate P-66). Two of these inputs enter the channel translator as selected by the 16V34, 17V34 (refer to plate P-50), 13E00, and 14E00 (refer to plate P-49) inputs from group priority. Refer to table 2-111 for translation of the 12Vg-inputs. The channel translator flip-flops indicate the selected channel of the selected chassis. When channel translation is combined with group translator outputs, the complete channel number is described. Channel translation is performed by gates 20G00, 20G01, 20G02, and 20G03. Refer to table 2-115 for channel-translator indications.

2-477. FUNCTION TRANSLATOR. The function translator (refer to logic diagrams, plate P-51) is comprised of flip-flops 0XG05 and 0XG04 and their output gates. These flip-flops are activated according to the selection made by function priority on the chassis selected by group priority. Function-priority inputs are from the 12Vg6 gates and the 11Vg7 gates. Two of these inputs enter the function translator as selected by the 16V34, 17V34, 13E00, and 14E00 inputs from group priority. Refer to table 2-112 for translation of the inputs from function priority. Function translation is performed by gates 20G04, 20G05, 20G06, and 20G07. Refer to table 2-116 for function translator indications. The function-translator outputs are used to dictate the basic operation to be performed by I/O logic in honoring the selected signal.

TABLE 2-115. CHANNEL TRANSLATOR INDICATIONS

Translator Flip-Flops		Indications
OXG02	OXG01	
Clear	Clear	20G00 = H; channel 0, 1, 10, or 11 selected
Clear	Set	20G01 = H; channel 2, 3, 12, or 13 selected
Set	Clear	20G02 = H; channel 4, 5, 14, or 15 selected
Set	Set	20G03 = H; channel 6, 7, 16, or 17 selected

TABLE 2-116. FUNCTION TRANSLATOR INDICATIONS

Translator Flip Flops		Indications
OXG05	OXG04	
Clear	Clear	20G07 = H, EI function selected
Clear	Set	20G05 = H, EF function selected
Set	Clear	20G06 = H, OD function selected
Set	Set	20G04 = H, ID function selected

2-478. DATA-REQUEST SEQUENCE TIMING. The honoring of a data-request involves the use of memory under control of the I/O sequence. Single-channel operation uses only I/01 sequence for one main memory cycle. ...this case, the program sequences are disabled for two microseconds. Dual-channel operation requires two main memory cycles to complete the transfers of both halves of a 36-bit word. The I/01 and I/02 sequences are used, and the program sequences are disabled for four microseconds.

2-479. The Scan sequence detects and selects the I/O signals to be honored. When a signal detected, it is gated into the priority logic ...d then into the I/O translator. If the scan sequence selects a data-request signal to be honored, a signal information is placed in the I/O translator, the $I/01_i$ flip-flop is set. The next two-microsecond period beginning with ...e T1.1 is then considered to be the I/01 sequence, and the requested I/O service is performed.

2-480. Since there is no scan flip-flop, every Main Timing cycle is considered to be a scan ...quence which is testing all signal-entrance logic for I/O requests. Therefore, during each sequence of each instruction, the possibility of I/O operation is being tested by the priority logic. Detection of a data request activates the I/01 sequence during which the program is effectively halted. The amount of delay to the program is determined by whether the I/O service involves single- or dual-channel operation. This delay period will be extended by consecutive I/O sequences if data requests are contin... ...lly selected by consecutive scan sequences. Since the scan sequence also runs during the I/01 sequence, it is conceivable that each Main Timing cycle could be under the control of an I/O sequence, and the program would ...t progress. In a practical situation, how... ...er, there would not be such an extended period of continual I/O activity because of the limited speed of peripheral devices, and the high speed of the computer in handling an I/O request.

2-481. The scan sequence can be effectively disabled under certain conditions by preventing the setting of the $I/01_i$ flip-flop. A direct disable to this flip-flop is present if f = 57 and the R1 sequence is in control, or if f = 20 - 23 and the first R1 sequence is in control (without the R2 sequence). These conditions also prevent the setting of the $I/02_i$ flip-flop. Refer to logic diagrams, plate P-13 for this disable.

2-482. An indirect disable to the setting of the $I/01_i$ flip-flop is the prevention of the Data Req → Function Priority command. This disable can exist if the $I/01_f$ flip-flop is set and the I/O service being performed involves dual-channel operation or ESI/ESA which has a terminate condition. Both conditions require the I/02 sequence to follow the I/01 sequence. This indirect disable (refer to plate P-49) to the I/01 sequence also exists if the CDM sequence is to be run.

2-483. The disable to the program sequences by I/O sequences occurs between the initial and final portions beginning at time T2.2. During the set time of either the $I/01_i$ or $I/02_i$ flip-flops, the outputs of the I_f, $R1_f$, $R2_f$, and W_f flip-flops, are disabled, and the setting of the I_i, $R1_i$, $R2_i$, W_i, $B1_i$, and $B2_i$ flip-flops is inhibited. During the set time of either the $I/01_f$ or $I/02_f$ flip-flops, Sequence Final flip-flops are prevented from advancing. That is, the setting and clearing of the I_f, $R1_f$, $R2_f$, W_f, $B1_f$, and $B2_f$ flip-flops is inhibited. The disable conditions are such that any sequence of any instruction may be effectively halted once the operations controlled by the Sequence Initial

flip-flop are executed. Further, those operations caused by the Sequence Final flip-flop will not be performed until after the interrupting I/O service is performed. The halting point can be considered to be at time T2.2, which is the setting time for the Program Sequence Final flip-flop, and the $I/01_i$ and $I/02_i$ flip-flops. Refer to figures 2-108 through 2-111 for timing diagrams which illustrate possible relationships of the program and I/O sequences. The diagrams show the I/O operations interrupting the I-sequence, however, a sequence, except those conditions already mentioned, can be disturbed in the same manner.

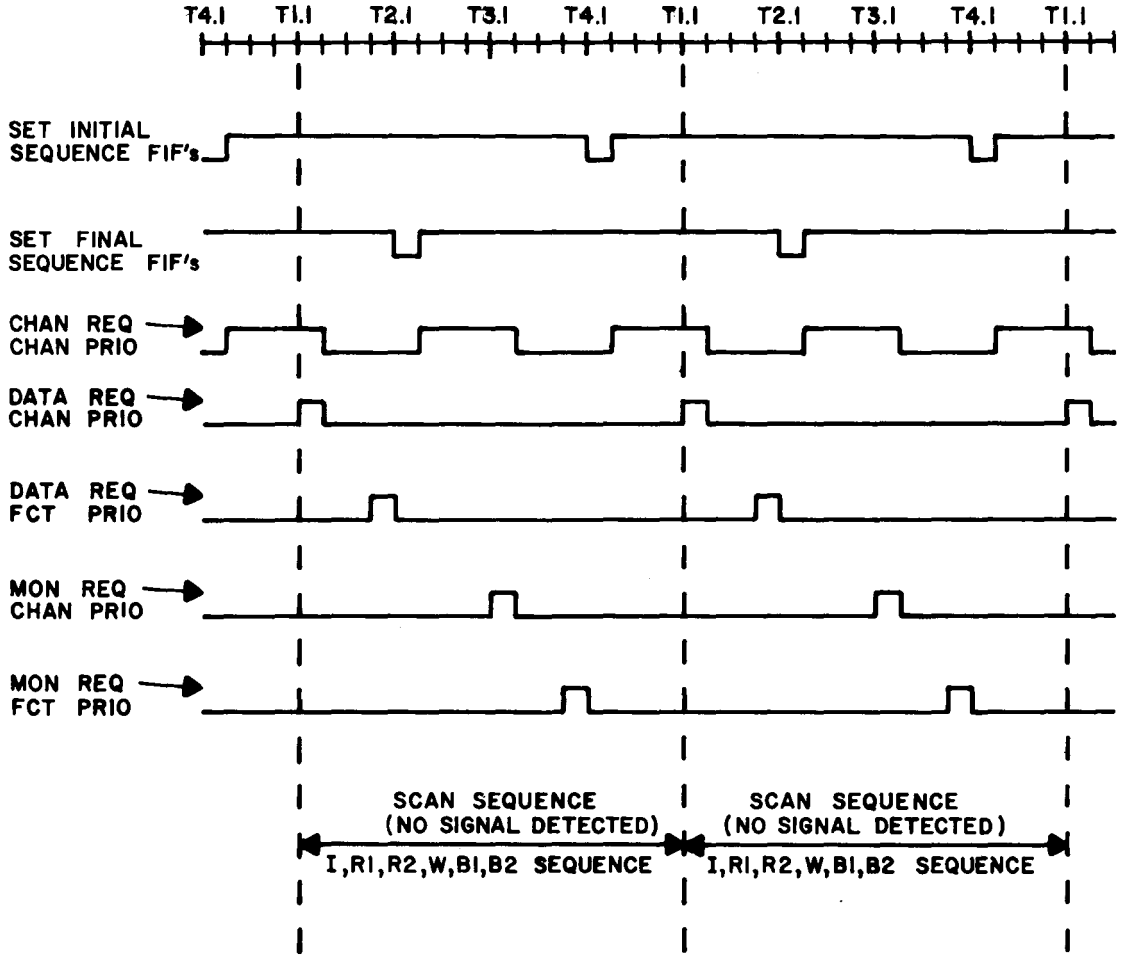


Figure 2-108. Scan and I, R1, R2, W, B1 or B2 Sequence Timing Example, (No Request Signal Detected)

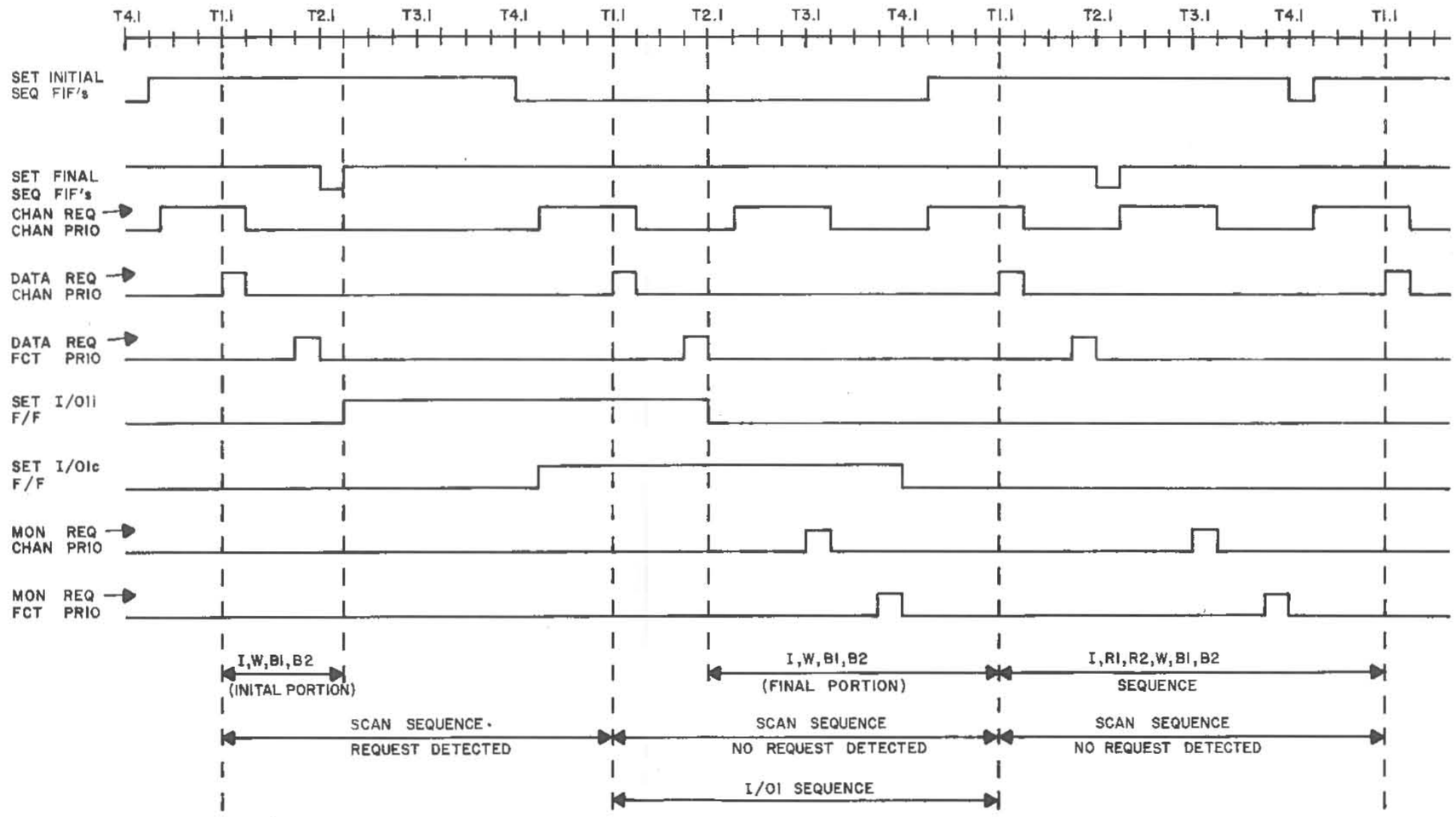


Figure 2-109. Single Request, Single Channel-Timing Example.

THIS PAGE INTENTIONALLY LEFT BLANK

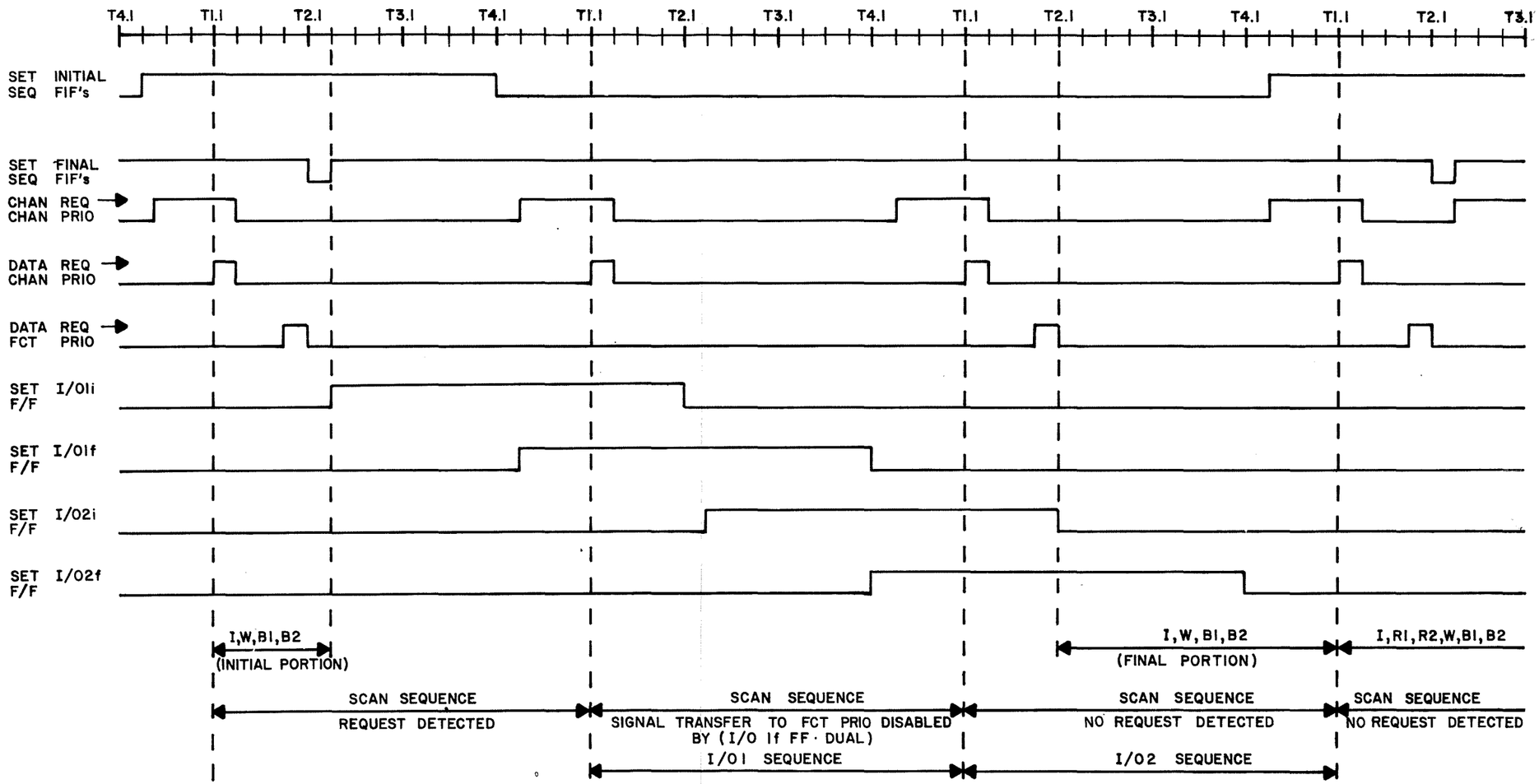


Figure 2-110. Single Data Request, Dual Channel - Timing Example.

THIS PAGE INTENTIONALLY LEFT BLANK

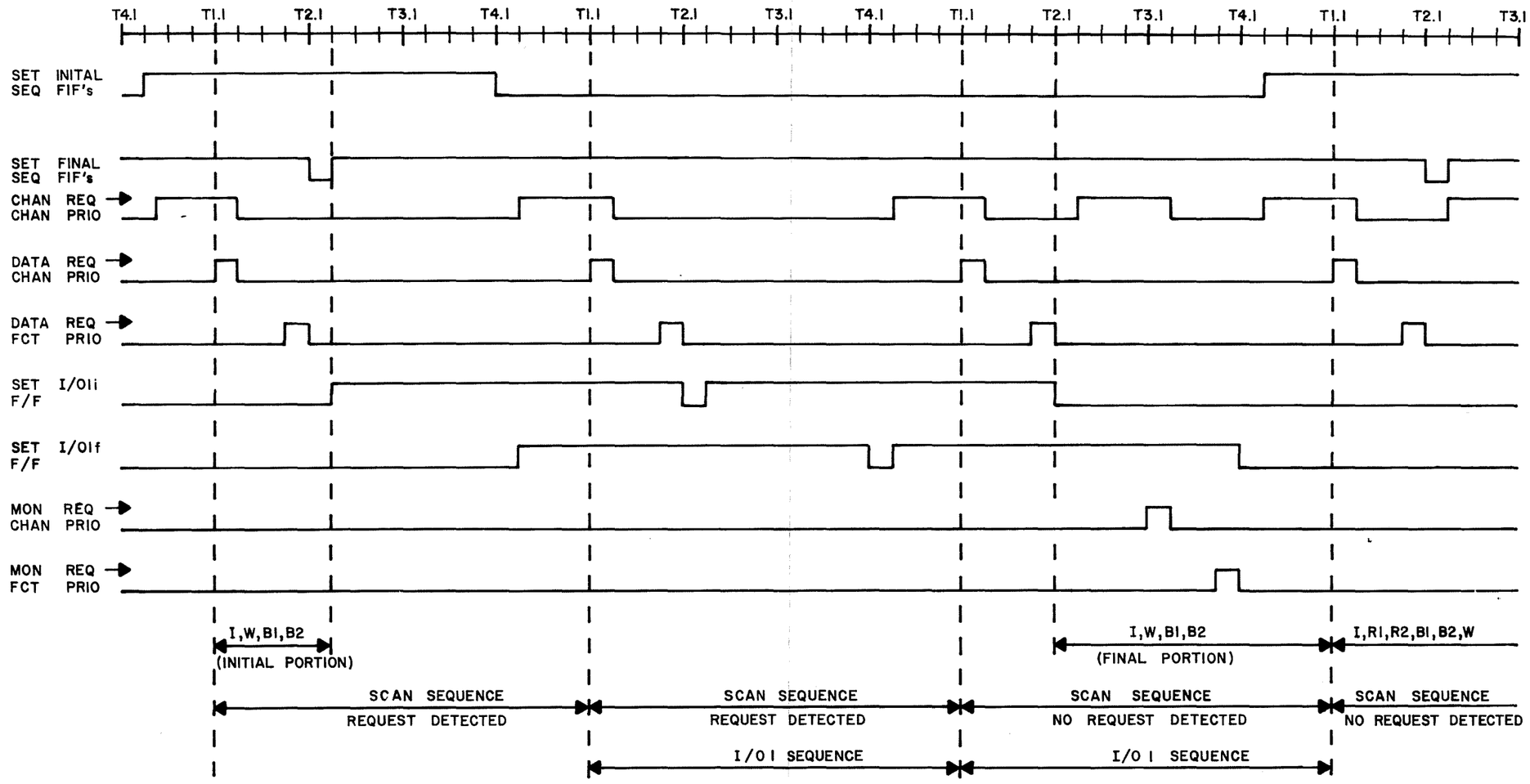


Figure 2-111. Multiple Data Request, Single Channel - Timing Example.

THIS PAGE INTENTIONALLY LEFT BLANK

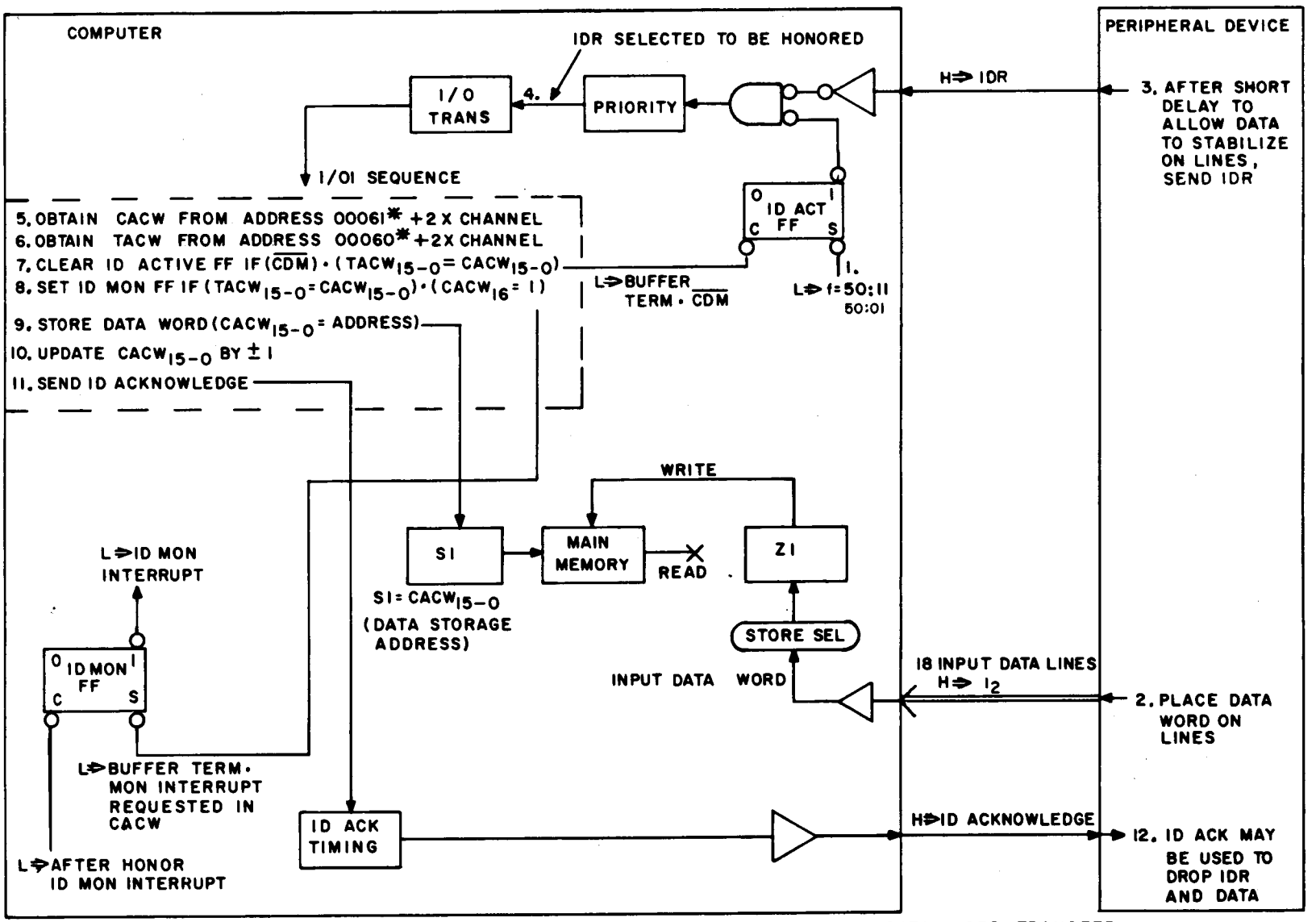
484. INPUT-DATA-REQUEST OPERATIONS. The input-data-request is a signal from an external device which requests the computer to accept an inputted data word. Input data requests may occur during either single- or dual-channel operation.

485. Single-Channel Operation. The input-data request (IDR) from an external device is honored by the computer only if the particular channel is input active, see figure 2-112. The input Active (ID) flip-flop is set during the execution of f = 50:01, 50:11. If the priority logic selects an IDR signal, the current address control word (CACW) is obtained from the control memory address 00061_8+2k (k=channel number) if the channel is 0-7, or from 00261_8+2k when the channel is $10-17_8$. The terminal-address-control-word (TACW) is obtained from the control-memory address 00060_8+2k if the channel is $0-7_8$, or from 00260_8+2k if the channel is $10-17_8$. The 18-bit input-data word is stored at the address specified by $CACW_{15-0}$. The $CACW_{15-0}$ is modified by one, as determined by the value in bit position 17 of CACW, and is stored in control memory. The $CACW_{17}$ determines whether the buffer is to be forward or backward. In a forward buffer ($CACW_{17}=0_2$), the data words are stored at consecutively incremented addresses as the IDR signals occur. In this case, $CACW_{15-0}$ is modified by plus one after each data word is stored. In a backward buffer ($CACW_{17}=1_2$), the data words are stored at consecutively decremented addresses as the IDR signals occur. In this case, $CACW_{15-0}$ is modified by negative one after each data word is stored. Refer to tables 2-117 and 2-118 for buffer examples.

486. The TACW is not modified but is compared with CACW. If the 16 least significant bits of TACW and CACW are equal before modification of CACW, the current input-data word being processed completes the buffer. To terminate the buffer, the ID Active flip-flop for the particular channel is cleared, disabling recognition of future IDR signals. At the buffer termination time, the value in bit position 16 of CACW is sensed. If $CACW_{16}=1_2$, an input-monitor-interrupt signal is generated to indicate termination to the program; thus, the ID Monitor flip-flop for the particular channel is set. The resulting ID Monitor signal is handled later by the computer as a separate request for I/O service and is discussed in a later section. The ID-Acknowledge signal is sent on the same channel which carried the request.

487. If $TACW_{17}=1_2$, Continuous Data Mode (CDM) is specified. This bit is sensed when the buffer is terminated, and allows the special CDM sequence to be activated. This sequence prevents clearing of the ID Active flip-flop and reloads TACW and CACW locations in control memory with new buffer control words. CDM operations are presented in a later section.

488. Dual-Channel Operation. The Dual-Channel Mode is initiated for a particular channel by positioning the CHANNEL FUNCTION switch to the DUAL position. Cables with 36 DATA-lines must be connected to the odd-numbered channel channel-pair (figure 2-113). If dual-channel has been selected, the IDR signal is actually requesting that two separate 18-bit data words be accepted, one on the IDR channel, which must be odd for dual-channel operation, and the other on the next lower channel which is even-numbered. The input operations for the second data word are identical to those for the first. The second word, therefore, is stored in memory at the address specified by $CACW_{15-0}$ which has been modified by one during storage of the first word. The type of buffer (forward or backward) determines which of the two words is stored first. The 18-bit data word from the odd channel (IDR channel) is stored first if the buffer is forward. The even-channel word is stored first if the buffer is backward. Refer to tables 2-119 and 2-120



NOTES: ONCE ID ACTIVE FF IS SET, STEPS 2 THROUGH 12 ARE REPEATED FOR EACH INPUT DATA WORD TRANSFER.
*ADD 00200_8 TO ADDRESSES OF CACW AND TACW IF CHANNEL IS 10-17$_8$.

Figure 2-112. Input-Data Operations Single Channel without ESI and ESA.

TABLE 2-117. SINGLE CHANNEL FORWARD ID BUFFER
IACW = 005000, TACW = 005003

Data Storage Address ($CACW_{15-0}$)	Word Input Sequence
05000	1
05001	2
05002	3
05003	4

TABLE 2-118. SINGLE CHANNEL BACKWARD ID BUFFER
IACW = 405003, TACW = 405000

Data Storage Address ($CACW_{15-0}$)	Word Input Sequence
05000	4
05001	3
05002	2
05003	1

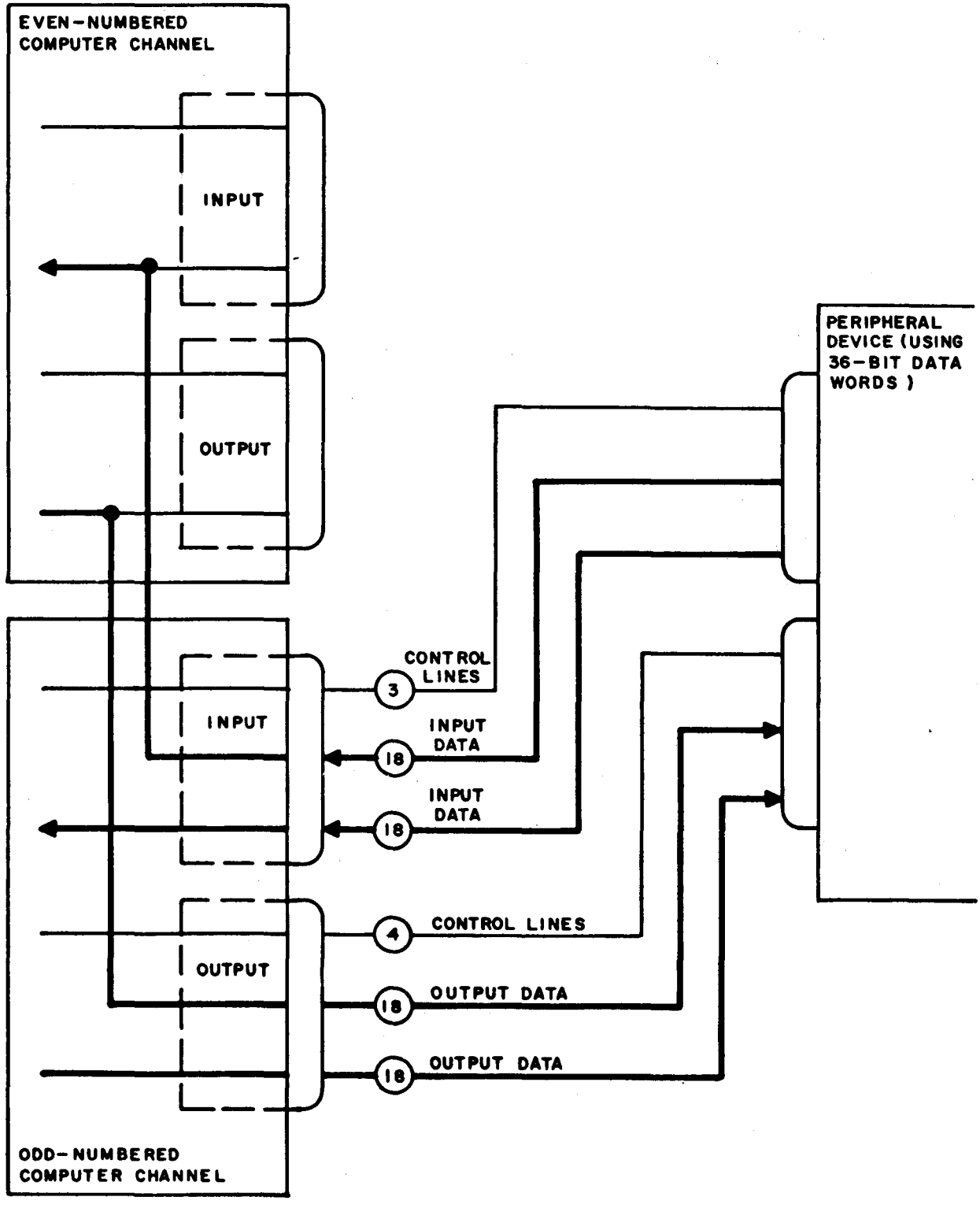


Figure 2-113. Dual-Channel I/O Interface.

TABLE 2-119. DUAL CHANNEL FORWARD ID BUFFER ON CHANNELS 2, 3
IACW = 005000, TACW = 005003

	Data Storage Address ($CACW_{15-0}$)	Inputting Channel	Word Input Sequence	IDR Sequence
storage of 1st 36-bit word	05000	3	1	1st IDR
	05001	2	2	
storage of 2nd 36-bit word	05002	3	3	2nd IDR
	05003	2	4	

TABLE 2-120. DUAL CHANNEL BACKWARD ID BUFFER ON CHANNELS 2, 3
IACW = 405003, TACW = 405000

	Data Storage Address ($CACW_{15-0}$)	Inputting Channel	Word Input Sequence	IDR Sequence
storage of 2nd 36-bit word	05000	3	4	2nd IDR
	05001	2	3	
storage of 1st 36-bit word	05002	3	2	1st IDR
	05003	2	1	

for buffer examples assuming the IDR signals occur on channel 3.

2-489. TACW and CACW are compared during each 18-bit word input operation. Therefore, clearing of the ID Active flip-flop and setting of the ID Monitor flip-flop can occur during the first or second word storage operation. TACW and IACW should be chosen to provide a buffer for an exact multiple of two 18-bit words, such that the buffer will always terminate as a result of the second word storage operation from a particular IDR. The ID-Acknowledge signal is sent on the odd-channel which carried the IDR. The acknowledgement occurs after the second word of the IDR operations has been accepted.

2-490. Scan and I/O Sequence Single-Channel ID Operation Data Flow. Refer to figure 2-114 for a block diagram description of the single-channel ID operation. The scan sequence is the Main Timing cycle during which the $I/O1_i$ flip-flop is set upon detection and selection of an IDR signal. During this sequence, CACW is obtained from control memory at the address of either the 00061_8 + 2k or the 00261_8 + 2k, depending upon the channel. The address of either the 00060_8 + 2k or the 00260_8 +2k is formulated in the I/O translator, which is set according to the priority selection. $S0_{00}$ is set to 1_2 to make the address odd. Another control-memory reference is used to obtain TACW. The TACW and CACW are compared from ZO and B. The result of the comparison is recorded by the Terminate flip-flop. If $TACW_{15-0} = CACW_{15-0}$, this IDR will be the last to be honored. The B-network is used to modify $CACW_{15-0}$ as determined by $CACW_{17}$. The value of this bit is recorded by the B±1 flip-flop. The I/O1 sequence is used to store the 18-bit input data word. The memory reference uses the address specified by the unmodified $CACW_{15-0}$ from B. Control-memory reference is used at time T1.1 to store the mofified CACW value for use during the next input operation. The remaining I/O1 sequence operations are used only if the operation is dual channel.

2-491. Scan, I/O1, and I/O2 Sequence Dual-Channel ID Operation Data Flow. Refer to figure 2-114 for a block diagram description of the Scan and I/O1 sequence for Single or Dual-Channel ID Operations, and to figure 2-115 for a description of the I/O2 Sequence for Dual-Channel ID operations. The Scan and I/O1 sequences are used in the same manner as for single-channel operation. The last portion of the I/O1, along with the I/O2 sequence, is used to perform the same operations executed during the last portion of the Scan sequence and the first portion of the I/O sequence. These operations input the second word and store it at the first word storage address ± 1.

2-492. Scan and I/O Sequence Essential Commands for Single-or Dual-Channel ID Operations. Refer to table 2-121 for a list of essential Scan, I/O1 and I/O2 sequence commands for single-or dual-channel ID operations. Events concerning priority selection have been previously discussed and are not shown. Events concerning the Continuous Data Mode Sequence, which is initiated by the setting of the Continuous Data Request flip-flop, are analyzed in a later section. The ID-Acknowledge timing is also discussed later in this section. Commands that have not been described in a preceding Scan or I-Sequence are described in detail in table 2-122. (For a more detailed description of the upper and lower-rank flip-flops, refer to the description of upper and lower-rank sequence designators.)

2-493. Some of the I/O2 sequence commands are executed exactly as described in the Scan and I/O1 sequences, except that some gates are enabled by the I/O2 flip-flops rather than the I/O1 flip-flops.

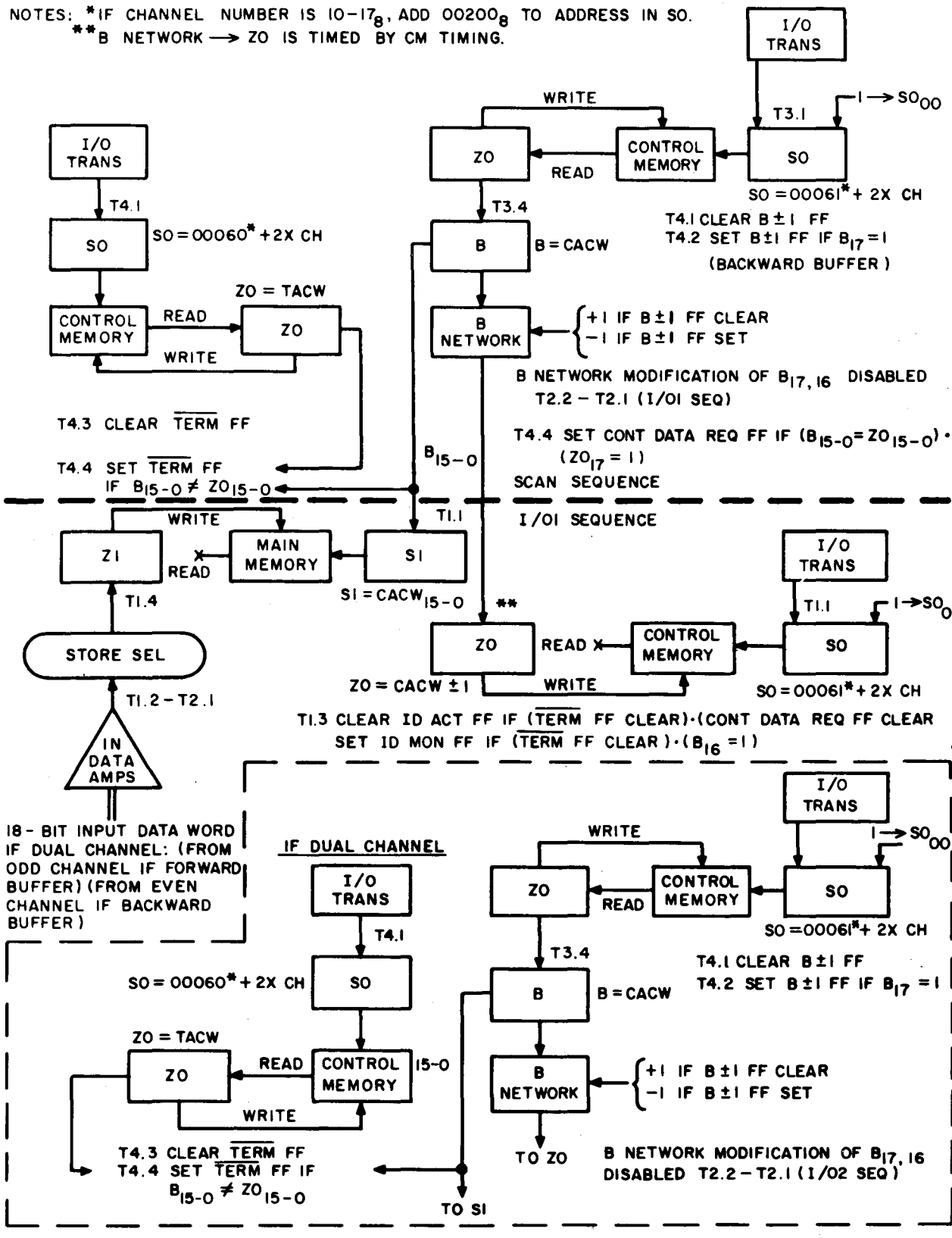


Figure 2-114. Scan and I/O1-Sequence Data Flow for Single- or Dual-Channel ID Operations

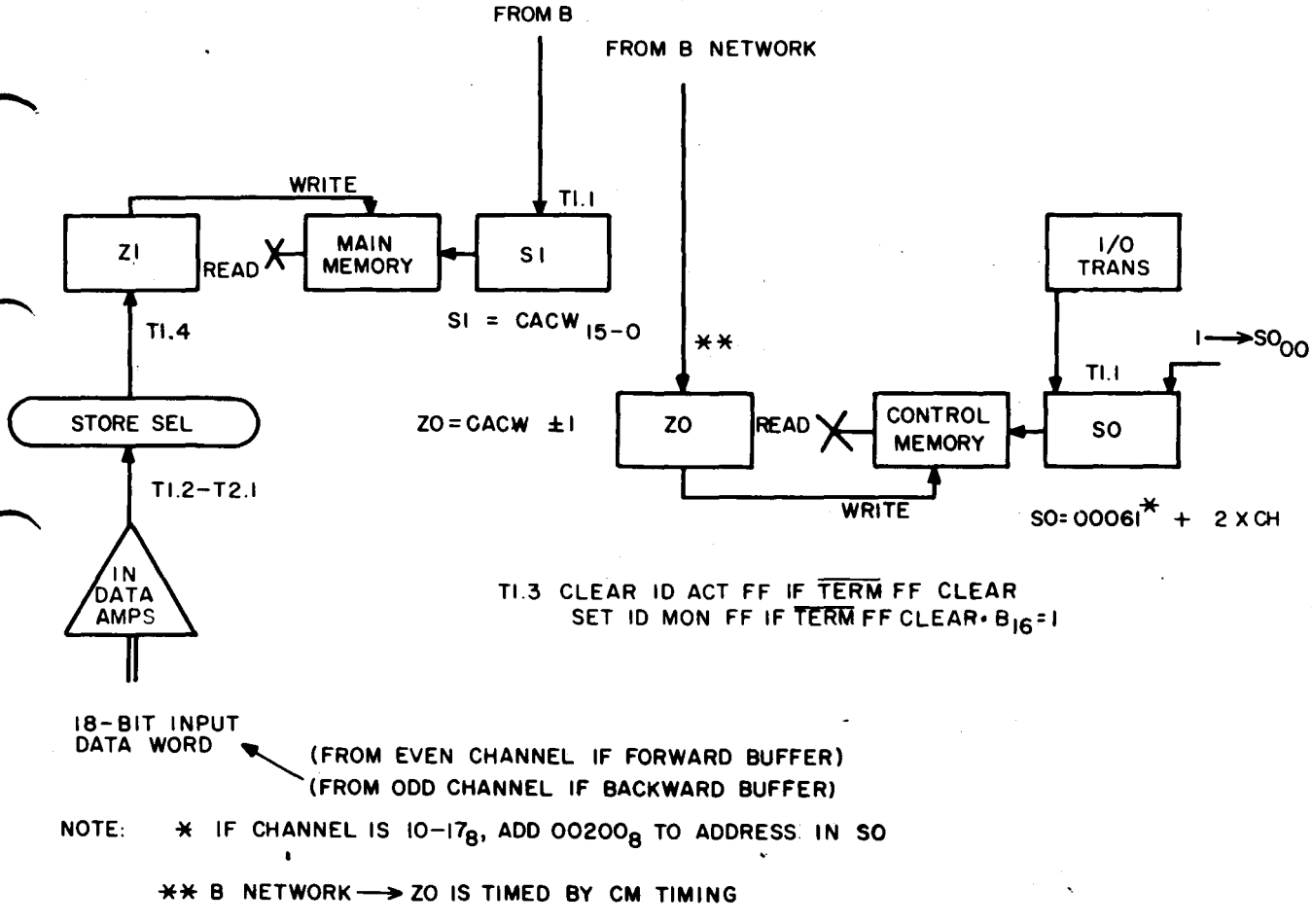


Figure 2-115. I/O2-Sequence Data Flow for Dual-Channel ID Operations

TABLE 2-121. SCAN, I/01, AND I/02 SEQUENCE ESSENTIAL COMMANDS FOR SINGLE-OR DUAL-CHANNEL ID OPERATIONS

Time Notation	Commands
	SCAN SEQUENCE
T2.1	Clear Fct, Chan, and I/0 Trans 1
T2.2	Set $I/01_i$ ff, Prio→Fct, Chan, and I/O Trans 1
T3.1	I/O Trans → SO, 1 → SO_{00}, Init CM, clear ZO
T3.2	Clear B
T3.4	ZO → B*
T4.1	I/O Trans → SO, Init CM, clear ZO, clear B ± 1 ff (+1 → B-network) clear I/O Trans 2
T4.2	Set $I/01_f$ ff, set B ± 1 ff (-1 → B-network) if $B_{17} = 1$
	I/O Trans 1 → I/O Trans 2 (set Dual ff if Dual-Chan selected) disable B-network $_{17,16}$, clear all ID Ack Reg ff's
T4.3	Clear Pri Alternator ff, set ID Ack Reg ff, set 6XLg0 ff, clear $\overline{\text{Term}}$ ff, clear ID Req ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{15-0} \neq ZO_{15-0}$, disable CM → ZO
	Clear S1, disable Mem → Z1, set Cont Data Req ff if $(B_{15-0} = ZO_{15-0}) \cdot (ZO_{17} = 1)$
	I/01 SEQUENCE
T1.1	I/O Trans → SO, 1 → SO_{00}, Init CM, B → S1, Init Memory
T1.2	Input Data Amps → Store Sel**
T1.3	Clear Z1, clear ID Act ff if ($\overline{\text{Term}}$ ff clear) (Cont Data Req ff clear)
	Set ID Mon ff if ($\overline{\text{Term}}$ ff clear) ($B_{16} = 1$)
T1.4	Store Sel → Z1, set ID Ack Gener ff if Dual ff clear

TABLE 2-121. SCAN, I/01, AND I/02 SEQUENCE ESSENTIAL COMMANDS FOR SINGLE-OR DUAL-CHANNEL ID OPERATIONS (Cont)

Time Notation	Commands
T2.1	Clear I/01, ff, drop Input Data Amps → Store Sel
T2.2	Set $I/02_i$ ff if Dual ff set
T2.4	Drop disable Mem → Z1, drop disable CM → ZO
The following occurs if $I/02_i$ ff set (dual channel) excepting, clear $I/01_f$ ff	
T3.1	I/O Trans → SO, 1 → SO_{00}, Init CM, clear ZO
T3.2	Clear B
T3.4	
T4.1	ZO → B*
	Clear $I/01_f$ ff, I/O Trans → SO, Init CM, clear Zo, drop disable B-$network_{17,16}$
	Clear B ± 1 ff (+1 → B-network), clear I/O Trans 2
T4.2	Set $I/02_f$ ff, set B ± 1 if (-1 → B-network) if B_{17} = 1, disable B $Network_{17,16}$
	I/O Trans 1 → I/O Trans 2 (set Dual ff), clear all ID Ack Reg ff's
T4.3	Set ID Ack Reg ff, set 6XLgO ff, clear Term ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{15-0} \neq ZO_{15-01}$ disable CM → ZO
	Clear S1, disable Mem → Z1, set Cont Data Req ff if (B_{15-0} = ZO_{15-0})-(ZO_{17} = 1)
	I/02 SEQUENCE (if dual-channel)
T1.1	I/O Trans → SO, 1 → SO_{00}, Init CM, Clear ZO
	B → S1, Init Memory
T1.2	Input Data Amps → Store Sel**
T1.3	Clear Z1, clear ID Act ff if ($\overline{\text{Term}}$ ff clear) (Cont Data Req ff clear)
	Set ID Mon ff if $\overline{\text{Term}}$ ff clear • B_{16} = 1
T1.4	Store Sel → Z1, set ID Ack Gener ff
T2.1	Clear $I/02_i$ ff, drop Input Data Amps → Store Sel

TABLE 2-121. SCAN, I/O1, AND I/O2-SEQUENCE ESSENTIAL COMMANDS FOR SINGLE-OR DUAL-CHANNEL ID OPERATIONS (Cont)

Time Notation	Commands
T2.4	Drop disable Mem → Z1, drop disable CM → ZO
T4.1	Clear $I/O2_f$, ff, drop disable $B\text{-network}_{17,16}$

*B-network → ZO is timed by CM timing.

**If operation is in Dual-Channel Mode, the channel inputted to Store Sel is as follows:

	I/O1 Seq.	I/O2 Seq.
B ± ff clear (forward buffer)	odd	even
B ± 1 ff set (backward buffer)	even	odd

TABLE 2-122. SCAN I/O1 AND I/O2 SEQUENCE COMMANDS/DESCRIPTIONS FOR SINGLE-OR DUAL-CHANNEL ID OPERATIONS
(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
T2.2	Set I/O 1_i ff. Gate 10E25 (refer to plate P-13) is disabled by the absence of a L ⇒ R1 Seq or the absence of a L ⇒ f = 57 and the absence of a L ⇒ f = 20-23. The low output enables 11E25, at Ø2, with a low from 03T22. The low from 11E25 and a L ⇒ I/O Req sets I/O 1_i flip-flop 1NG25 at time T2.2.
T3.1	I/O Trans → SO, Init CM. Gate 13G25 (refer to plate P-14) outputs a H ⇒ I/O Seq. The low is applied as a L ⇒ I/0 Seq, via 15G25, to 10N10 (refer to plate P-24). The other input to 10N10 is a low from 31T31, which has a high input from 00T31. The high out of 10N10 is inverted by 18N10 and at Ø1 19N10 outputs a L ⇒ I/O Translator → SO to SO-register (refer to plate P-114) and to Control Memory (refer to plate P-122). 1 → SO_{00}. Gate 50N10 (refer to plate P-24) outputs a high due to inputs of L ⇒ I/O Seq, a low from 00T41 and a low from 31T31. The high is inverted by 58N10 and applied as a L ⇒ Set SO Bit 2^0 to SO-register OXS00 (refer to plate P-114) which sets with the L ⇒ I/O Translator → SO.
T4.1	I/O Trans → SO, Init CM. This command is executed the same as at time T3.1 except the high from 00T41 into 31T31 is used (refer to plate P-24). Output is from 19N10 at Ø1 with a L ⇒ Clear B ± 1 ff (+1 → B-network) from 18N10. When B ± 1 flip-flop OXG37 (refer to plate P-27) is cleared by the low input from 01T42 and Ø1, the low out of the clear side is applied as a L ⇒ +1 → B via 02G37 into ± 1 network (refer to plates P-118 through P-120).
T4.2	Set I/O 1_f ff. I/O 1_f flip-flop OXG25 (refer to plate P-12) is set at time T4.2 by the low from 22E20 (enabled by the low from 03T42 and Ø2) and by a L ⇒ I/O 1 Seq from the Lower Rank Sequence Designator. Set B ± ff (-1 → B-network) if B_{17} = 1. If OXB17 (refer to plate P-120) is set, the L ⇒ Backward Buffer from the set side is applied to B ± flip-flop OXG37 (refer to plate P-27). Gate 10E37 is enabled by a L ⇒ I/O Seq, a low from 01T42 and the absence of H ⇒ RTC. The high output is inverted by 11E37 and at Ø2, OXG37 sets. The low from the set side is applied as a L ⇒ -1 → B via 03G37 into the ± 1 Network (refer to plate P-118 through P-120). Disable B-network 17, 16. The H ⇒ I/O Seq is inverted by 03G25 and applied to 05G25 (refer to plate P-14) and due to the absence of H ⇒ Cont Data Seq & H ⇒ RTC Seq, 05G25 outputs a low which is inverted

TABLE 2-122. SCAN I/O1 AND I/O2 SEQUENCE COMMANDS/DESCRIPTIONS FOR SINGLE-OR DUAL-CHANNEL ID OPERATIONS (Cont)

Time Notation	Command/Description
T4.2 (Cont)	by 06G25 and applies a H ⇒ I/O Seq → RTC to disable 11B16 (refer to plate P-120). The low out of 11B16 disables 10B17, thus bits 16 and 17 cannot be modified. I/O Trans 1 → I/O Trans 2 (Set Dual ff if Dual-Chan Selected). I/O Trans 1 → I/O Trans 2 as previously described in Scan Seq. If DUAL-CHANNEL is selected Dual flip-flop 4XG09 (refer to plate P-52) is set by a L ⇒ Dual and L ⇒ Translator 1 → Translator 2. Clear All ID Ack Reg ff. Set In Ack Reg gate 00N49 (refer to plate P-53) outputs a high due to inputs of L ⇒ I/O Seqs, a low from 03T43 and a low from 00N40 which is disabled by the absence of L ⇒ Out + EF. The H ⇒ Set ID Ack is inverted by 1gN49 (refer to plate P-58) and applied to 2gN49. The other input to 2gN49 is from the I/O Translator 1 - Group and Mode (refer to plate P-59). The output of 21G13, 21G12, 21G11, or 21G10 (refer to plate P-50) depending on Lower, Upper-Even or Odd, is applied as a low to 31Gg8 (refer to plate P-59) and the low output is applied as a L ⇒ Group g to 2gN49. Gate 2gN49 will produce a L ⇒ Chan → Input Ack to 3gN49 (refer to plate P-68) and at Ø2, the high output via 3gN51 clears Input Acknowledge flip-flops 5XVg0 through 5XVg3.
T4.3	Clear Prio Alternator ff. The L ⇒ Chan → Input Ack used to clear the Input Acknowledge flip-flops also clears Priority Alternator flip-flop 5XVg4 (refer to plate P-68). Set ID Ack Reg ff. Two inputs are required to set the Input Acknowledge flip-flops (refer to plate P-68). One input is the L ⇒ Chan → Input Ack, which was used to clear the flip-flops at Ø2, and the other input is the channel number developed by the Channel Translator (refer to plate P-51). Depending on channel number one of the 20G0– gates outputs a high. If 20G00 is enabled, the H ⇒ Channel 0, 1 is inverted by 30Gg0 (refer to plate P-59A) and at Ø3, 31Gg0, which has another low input from 31Gg8 (developed at time T4.2), will produce a L ⇒ Chan 0,1 to set 5XVg0 (refer to plate P-68). Set 6XLg0 ff. The L ⇒ Chan → Input Ack developed at time T4.2 out of 2gN49 (refer to plate P-58) will set 6XLg0 (refer to plate P-70) at Ø3. Clear $\overline{\text{TERM}}$ ff. $\overline{\text{TERMINATE}}$ flip-flop OXG19 (refer to plate P-106) is cleared by the low from 03T44 at Ø3.

TABLE 2-122. SCAN I/O1 AND I/O2 SEQUENCE COMMANDS/DESCRIPTIONS FOR SINGLE-OR DUAL-CHANNEL ID OPERATIONS (Cont)

Time Notation	Command/Description
T4.3 (Cont)	Clear ID Req ff. Two inputs are required to clear the ID Req flip-flops (refer to plates P-61 through P-64). One input is the channel number, which was developed earlier to set the ID Ack Req flip-flop, and the other input is a L ⇒ Clear ID Req, which is produced by gate 4gN40 (refer to plate P-57). Gate 4gN40 is enabled by the high from 9gN49, developed by the L ⇒ Set ID Ack and the H ⇒ ID. Refer to the detailed description of the f = 50:01 through 50:03 instructions for the development of the H ⇒ ID.
T4.4	Set $\overline{\text{TERM}}$ ff if $B_{15-0} \neq Z_{15-0}$. The 20B00 through 20B15 gates (refer to plate P-106) compare the clear side of the B-register with the set sides of the ZO-register and also the set side of the B-register with the clear side of the ZO-register. If any of the gates are enabled due to any bits in B ≠ ZO, 21B00 outputs a L ⇒ B ≠ ZO. At ∅4, 10E19 produces a low to set $\overline{\text{Terminate}}$ flip-flop OXG19. Disable CM → ZO. This command is executed the same as previously described in a IB1 Sequence. Clear S1. Gate 30N12 (refer to plate P-22) is disabled by the absence of L ⇒ Int Seq, L ⇒ Cont Data Seq and L ⇒ Ext Int + ESI • Pass 2. The low output along with a L ⇒ I/O Seq and a low from 03T11 enables 40N12 and the high is inverted by 08N12 and at ∅4, 09N12 produces L ⇒ Clear S1. Disable Mem → Z1. Gate 10N13 (refer to plate P-23) outputs a high due an input of L ⇒ I or I/O Seq, a low from 03T13 and the absence of a H ⇒ Cont Data Seq. The high disables 11N13, which disables 17N13, 18N13 and 19N13 thus disabling Mem → Z1. Set Cont Data Req ff if $(B_{15-0} = ZO_{15-0}) \cdot (ZO_{17} = 1)$. If $B_{15-0} = ZO_{15-0}$, gate 21B00 (refer to plate P-106) outputs a high to 03G28 (refer to plate P-34). If $ZO_{17} = 1$, the clear side of OXZ17 (refer to plate P-117) also outputs a high to 03G28. Gate 00E28 is enabled by a L ⇒ I/O Seq and a low from 01T44, and the high output will cause 03G28 to output a low and at ∅4, Cont Data Request flip-flop OXG28 will set.

TABLE 2-122. SCAN I/O1 AND I/O2 SEQUENCE COMMANDS/DESCRIPTIONS FOR SINGLE-OR DUAL-CHANNEL ID OPERATION (cont)

Time Notation	Command/Description
T1.1	I/0 Trans → S0, Initiate CM. This command is executed the same as described at time T3.1 except that the high input to 31T31 (refer to plate P-24) is from 04T11. $1 \rightarrow S0_{00}$. This command is executed the same as described at time T3.1 except that the high input to 31T31 (refer to plate P-24) is from 04T11. B → S1. Gate 40N12 (refer to plate P-22) is enabled by a L ⇒ I/0 Seq, a low from 03T11 and a low from gate 30N12, which is disabled by lack of proper command or sequence. The high out of 40N12 is inverted by 48N12 and at Ø1, 49N12 produces a L ⇒B → S1.
T1.2	Input Data Amps → Store Sel. Store Select Control is shown on plate P-26. Gates 09N09, 19N09, 29N09 and 39N09 all have a common input of L ⇒ I/0 Seqs. The other inputs depend on the output of I/0 Translator I - Group and Mode gates 03G07 and 02G07 (refer to plate P-50), I/0 Translator 2, gates 44G00 and 43G00, via 10N09 and 30N09. The other input is a L ⇒ T21 I/0 Seq via 18N09, 10N09, or 30N09. If the input is Odd and Upper gate, 10N09 is enabled by the L ⇒ Odd and L ⇒ T21 I/0 Seqs, and the high is inverted by 18N09 and applied to 39N09. With the other input L ⇒ Upper, 39N09 produces a L ⇒ I/0 Odd Upper to Store Select (refer to plate P-108 through P-110).
T1.3	Clear ID Act ff if $\overline{\text{(Term ff Clear)}}$ · (Cont Data ff Clear). With $\overline{\text{Terminate}}$ flip-flop OXG19 clear (refer to plate P-106), the high from the set side is inverted by 02G19 and applied as a L ⇒ Terminate to 00N43 (refer to plate P-53). With Cont Data Request flip-flop OXG28 clear (refer to plate P-34) the low from the clear side is applied to 00N43 (refer to plate P-53). Other inputs to 00N43 are a L ⇒ T13 I/0 Seq and a low from 00E12 which is disabled by the absence of a L ⇒ Out + EF. The high from 00N43 is inverted by 02N43 and the output is applied to the ID Act flip-flop in the same manner as described in a 50:15 instruction. Set ID Mon ff if $\overline{\text{(Term ff Clear)}}$ · $(B_{16} = 1)$. The L ⇒ Terminate developed in the proceeding is also applied to Set Monitors gate 00N42 and with $B_{16} = 1$ a L ⇒ Monitor from the B-register,

TABLE 2-122. SCAN I/O1 AND I/O2 SEQUENCE COMMANDS/DESCRIPTIONS FOR SINGLE-OR DUAL-CHANNEL ID OPERATION (Cont)

Time Notation	Command/Description
T1.3 (cont)	OXB16 (refer to plate P-120) is also applied to 00N42. With the L ⇒ T13 I/0 Seq, 00N42 outputs a high which is inverted by 01N42 to produce a L ⇒ Set Monitor to 9gN42 (refer to plate P-56), which is inverted and applied to 4gN42. Refer to the IB2 Sequence for development of the H ⇒ ID into 4gN42. With 4gN42 enabled, a L ⇒ Set ID Monitor is applied to ID Mon flip-flop (refer to plates P-61 through P-64), and the flip-flop will set when a low is applied from the Channel Translator.
T1.4	Store Sel → Z1. When flip-flop 4XG06 (refer to plate P-52) clears at time T4.1, the high from the set side is applied to 43G06 and with the absence of a L ⇒ NDRO Bootstrap Address, 43G06 outputs a L ⇒ In + Int + ESI 2 · $\overline{\text{NDRO Bootstrap Add}}$ to 20N13 (refer to plate P-23). The other inputs to 20N13 are a L ⇒ I/0 Seq and a low from 03T14 to produce a high which is inverted by 28N13 and applied to 29N13. Gate 30N13 is disabled by the lack of proper time, instruction or sequence, and outputs a low at Ø4; 29N13 produces a L ⇒ Store Select → Z1. Set ID Ack Gener ff if Dual ff Clear. If Dual flip-flop 4XG09 (refer to plate P-52) is clear, the high from the set side disables 42G09. Gate 42G10 is disabled by not having ESA ESI flip-flop 4XG10 set. Lows are applied to 30G09 and the high output disables 31G09, producing a low to 00E04 (refer to plate P-53). The other inputs are a L ⇒ ID/EI from the clear side of flip-flop 4XG06 (refer to plate P-52) and a L ⇒ T21 I/0 Seq. The high output is inverted by 01E04 and applied as a L ⇒ Set ID Ack to ID Ack Generator flip-flop 6XLg1 (refer to plate P-70) which will set with the low from flip-flop 6XLg0 and a Ø4.
T2.1	Drop Input Data Amps → Store Sel. With I/0 1_i flip-flop cleared, gate 13G25 (refer to plate P-14) now outputs a high which is applied via 15G25 as a H ⇒ I/0 Seqs to disable gates 09N09, 19N09, 29N09 and 39N09 (refer to plate P-26). These gates prevent the Input Data Amps → Store Sel.
T2.2	Set I/0 2_i ff if Dual ff Set. With Dual flip-flop 4Xg09 (refer to plate P-52) set, the low from the set side, with the absence of a H ⇒ OD + EF, enables 42G09, the high output of 42G09 is inverted by 30G09 and applied as a L ⇒ Dual to I/0 2_i flip-flop 1XG26, which sets due to a L ⇒ I/0 1 Seq and the low from 11E25.

TABLE 2-122. SCAN I/O1 AND I/O2 SEQUENCE COMMANDS/DESCRIPTIONS FOR SINGLE-OR DUAL-CHANNEL ID OPERATION (Cont)

Time Notation	Command/Description
T2.4	Drop Disable Mem → Z1, Drop Disable CM → ZO. Gate 10N13 (refer to plate P-23) is disabled by the loss of the low from 13T23, which kept 10N13 enabled after the 03T13 went to a high. The low output is now applied to 11N13 and to 11N11 (refer to plate P-25). NOTE: Some of the commands after T2.4 are executed exactly as described previously except some gates are enabled by the I/02 flip-flops rather than the I/01 flip-flops. Only commands that are executed differently are described here.
T4.1	Drop Disable B-network 17, 16. Gate 03G25 (refer to plate P-14) produced a low output at time T4.2 of the Scan Sequence by the H ⇒ I/01 Seq on the input, but now produces a high output due to the absence of the H ⇒ I/01 Seq which drops the disable B-network 17, 16 as described at time T4.2. Set ID Ack Gener ff. ID Ack Generator flip-flop 6XLg1 (refer to plate P-70) is set in the same manner as described at T1.4 of the I/01 Seq, except gate 31G09 (refer to plate P-52) is now disabled by the absence of a L ⇒ I/01 Seq, rather than a high from 30G09.

494. ID Acknowledge Timing. The ID Acknowledge signal is sent after the data word storage operation. In the case of dual-channel operation, the acknowledge is sent after both words have been inputted. The purpose of this signal is to acknowledge receipt of data. Usually the communicating external device uses the acknowledge to time dropping of its IDR signal. The ID Acknowledge signal is sent for a duration of 2.25 microseconds (fast interface) and 14.75 microseconds (slow interface). The ID Acknowledge Generation flip-flop determines acknowledge duration by its set time. Refer to tables 2-123 and 2-124 for the timing analysis of the acknowledge for fast and slow interface operations.

495. Scan, I/O1, and I/O2 Sequence Essential Commands (Fast Interface). Commands at time T4.2, T4.3, and T1.4 are executed in the manner described in the Scan, I/O1 and I/O2 sequences. Fast interface commands not previously described are discussed in table 2-125 in detail.

2-496. Scan, I/O1, and I/O2 Sequence Essential Commands (Slow Interface). Commands at times T4.2, T4.3, and T1.4 are executed in the manner described in the Scan, I/O1, and I/O2 sequences. Slow interface commands not previously described are described in table 2-126 in detail.

2-497. INPUT DATA OPERATIONS WITHESI. The Externally Specified Index Mode (ESI) is instituted for a particular channel by manually positioning the CHANNEL FUNCTION switch on the channel drawer to the ESI position. When this mode is in effect, it alters operations which honor an IDR on this channel. Two channels are used together. One, on which the IDR is sent, must be odd-numbered, and the other is the next lower even-numbered channel. Instead of TACW and CACW being obtained from their normal addresses in control memory, their control memory origins are specified by the requesting external device. Refer to figure 2-116 for the input word format. The IDR signal must appear on an odd-channel. The input word on that channel is accepted as the control memory address of TACW. The CACW is obtained from the next consecutively higher address. $CACW_{15-0}$ is modified by ± 1 depending upon its bit position 17. The data word on the even channel is stored at the address specified by the unmodified $CACW_{15-0}$. When $TACW_{15-0} = CACW_{15-0}$ (unmodified), the buffer is terminated and the input monitor interrupt signal is generated if $CACW_{16} = 1$. The ID Acknowledge signal is sent on the odd channel. The ESI Mode of operation is useful if more than one piece of equipment is connected to a common channel by means of a multiplexing device. Each piece of equipment can indirectly specify its data storage locations in memory by means of the ESI, which could be generated in the multiplexer as it handled each data word. Each equipment would have a unique index value. The multiplexer must occupy an odd and the next lower even channels. Upon buffer termination, the ESI is also stored in main memory at the address $00161_8 + 2k$ (k = channel number) if the channel is 0-7, or at the $00361_8 + 2k$, if the channel is $10-17_8$. By inspecting the content of this address, the program would determine which equipment on that channel made the last input, and made the channel inactive.

2-498. Scan and I/O1 Sequence Data Flow For ID Operations with ESI. Refer to figure 2-117 for a block diagram description of the OD operations with ESI. Data flow is similar to that for normal input operations. Exceptions are that control memory addresses for CACW and TACW are from the odd-numbered channel. The ESI is used in SO at time T4.1

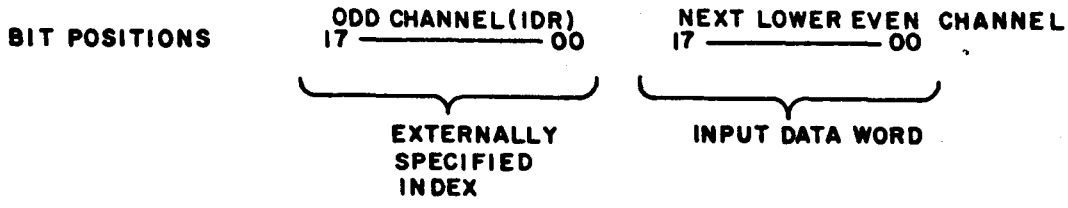


Figure 2-116. ESI Input Word Format For IDR Operations

to obtain TACW. The transfer into SO (refer to figure 2-118) involves a single place left-shift. Also, SO has no bit position 06. The inputted bits 05 and 06 are therefore placed in SO_{07} and SO_{08}, respectively. Nothing is placed in SO_{00} so that the address of TACW is always even. The address of CACW is formulated by setting SO_{00} to 1_2 along with the left-shift input. CACW is thus obtained from the next consecutive higher address. All other I/O1 sequence operations are the same as those for normal input operations. CACW is modified by ± 1 and is restored by another control memory reference. The input data word on the even channel is stored at the address specified by the unmodified $CACW_{15-0}$. Termination and monitor interrupt capabilities are the same as for normal input operations. The Continuous Data Mode sequence is initiated upon buffer completion if $TACW_{17} = 12$. However, reloading the address control words is unnecessary since this sequence always reloads the normal control memory address of CACW and TACW. That is, the sequence will place new control words in control memory address 00060_8 + 2k and 00061_8 + 2k or 00260_8+ 2k and 00261_8 + 2k depending upon the channel number. These addresses are not used since the external device specifies the origins of CACW and TACW. However, if the Continuous Data Request flip-flop is set, it will prevent deactivation of the buffer (the ID Active flip-flop will not be cleared). Refer to figure 2-119 for I/O2 sequence operations executed upon buffer termination. The I/O2 sequence is initiated if TACW equals the unmodified CACW. It uses a memory reference to store the ESI in memory for use by the program.

2-499. Scan, I/O1, and I/O2 Sequence Essential Commands For ID Operation with ESI. Refer to table 2-127 for a list of Scan, I/O1, and I/O2 sequence events. Events concerning priority selection have been discussed earlier in this section and are not shown. Events concerning the Continuous Data Mode sequence are analyzed later in this section. ID Acknowledge timing is the same as for normal input operations. Commands not previously described are described in table 2-128 in detail.

TABLE 2-123. ID-ACKNOWLEDGE TIMING FOR FAST INTERFACE

Time Notation	Action
T4.2	Clear all ID Ack Reg ff's
T4.3	Set ID Ack Reg ff, set 6XLg0 ff
T1.4	Set ID Ack Gener ff, send ID Acknowledge signal
0.85 μs / 2.725 μs	
T3.3	Clear 6XLg0 ff
1.875 μs	
T3.2	Clear ID Ack Gener ff, drop ID Acknowledge signal

TABLE 2-124. ID-ACKNOWLEDGE TIMING FOR SLOW INTERFACE

Time Notation	Action
T4.2	Clear all ID Ack Reg ff's
T4.3	Set ID Ack Reg ff, set 6XLg0 ff
T1.4	Set ID Ack Gener ff, send ID Acknowledge signal
T1.4	
T1.4	
12 μs delay of 61 Lg2 / 14.75 μs: T1.4	
T1.4	
T1.4	
T1.4	
T1.4	Set 6XLg3 ff
0.875 μs: T3.3	Clear 6XLg0 ff
1.875 μs	
T3.2	Clear ID Ack Gener ff, drop ID-Acknowledge signal
4 μs delay of 60 Lg2: T3.2	
T3.2	
T3.2	Clear 6XLg3 ff

TABLE 2-125. SCAN, I/O1, AND I/O2-SEQUENCE COMMANDS/DESCRIPTIONS FOR ID-ACKNOWLEDGE TIMING (FAST INTERFACE)
(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
T1.4	Set ID Ack Gener ff and send ID Ack Signal. When the ID Ack Generat. flip-flop 6XLg1 (refer to plate P-70) is set, the ID-Acknowledge signal is sent from the clear side of the flip-flop to 63Lg1 and the low from the set side is applied to pin 13 of 6XLg0 via gJ23B which will clear at T3.3
T3.3	Clear 6XLg0 ff. When the 04T33 input to 8gT33 goes high, the low is applied to 6XLg0, which will clear at Ø3.
T3.2	Clear ID Ack Gener ff, Drop ID-Acknowledge Signal. The low from the clear side of 6XLg0 is applied to 6XLg1 and when 04T33 again goes high, the low from 8gT33 will clear 6XLg1 at Ø2 and the ID-Acknowledge signal will be dropped.

TABLE 2-126. SCAN, I/01 AND I/02-SEQUENCE COMMANDS/DESCRIPTIONS FOR ID-ACKNOWLEDGE (SLOW INTERFACE)

Time Notation	Command/Description
T1.4	Send ID Acknowledge. When the ID ACK Generator flip-flop 6XLg1 (Refer to plate P-70) is set, the high from the clear side is applied to 63Lg1. The low output from 63Lg1 fully enables the 50 Yg gates (Refer to plate P-68). This starts the ID Acknowledge. Set 6XLg3. When the ID Ack Generator flip-flop 6XLg1 (refer to plate P-70) is set, the high from the clear side is applied to delay line 61Lg2. 12 μsec later a low from pin 6 set Flip-Flop 6XLg3.
T3.3	Clear 6XLg0 ff. This command is executed the same as in Fast Interface.
T3.2	Clear ID Ack Gener ff, Drop ID-Acknowledge Signal. This command is executed the same as in the Fast Interface. Clear 6XLg3 ff. When ID Ack Generator clears, the high from the set side is applied to Delay Line 60Lg2 and after 4 μsec, 6XLg3 will clear, at Ø2.

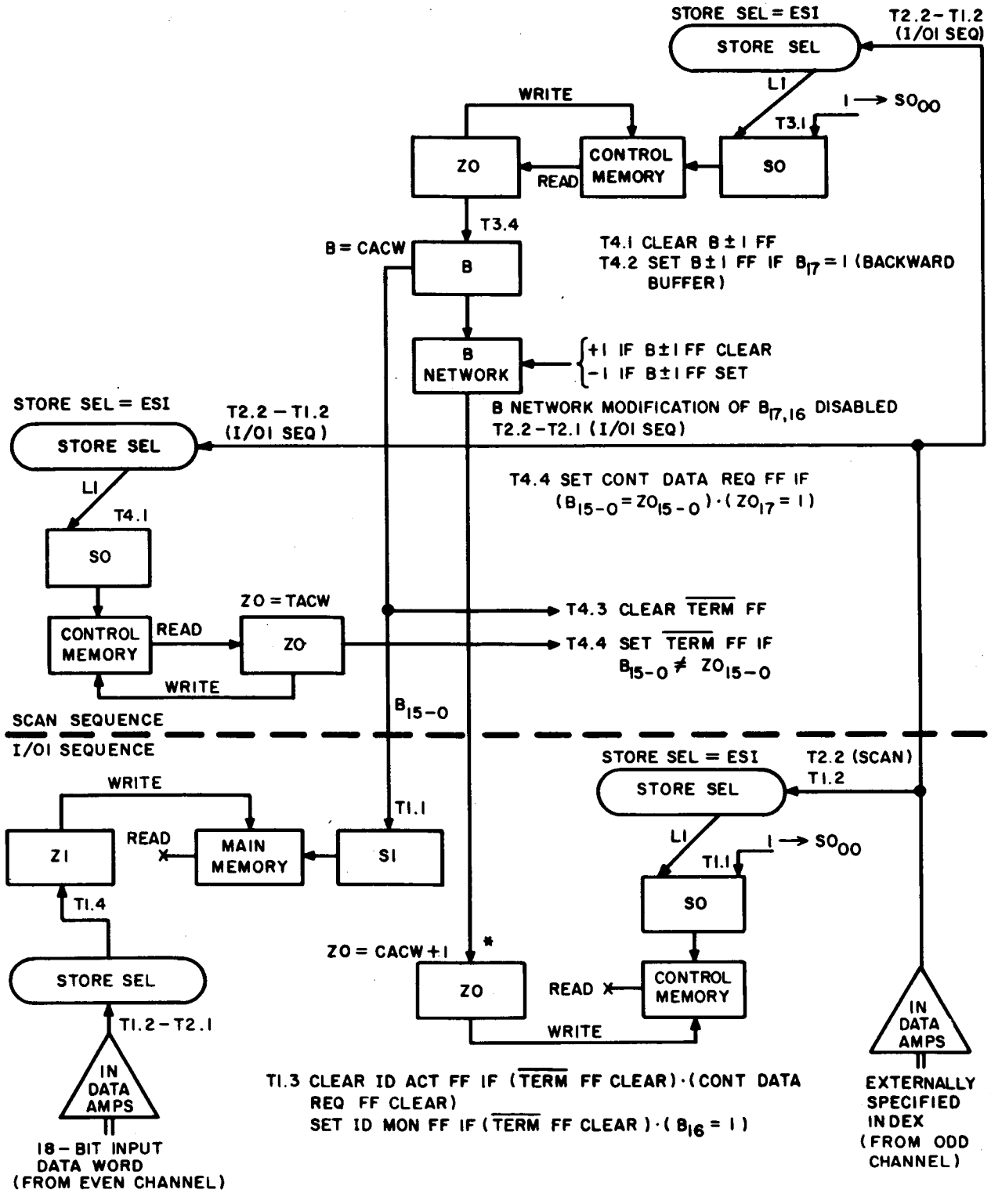


Figure 2-117. Scan and I/O1-Sequence Data Flow for ID Operations with ESI

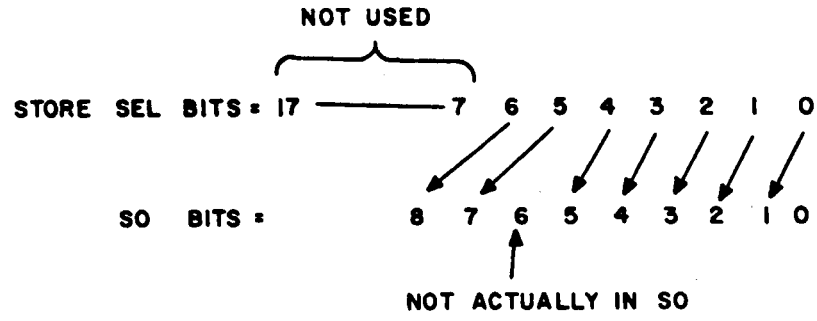
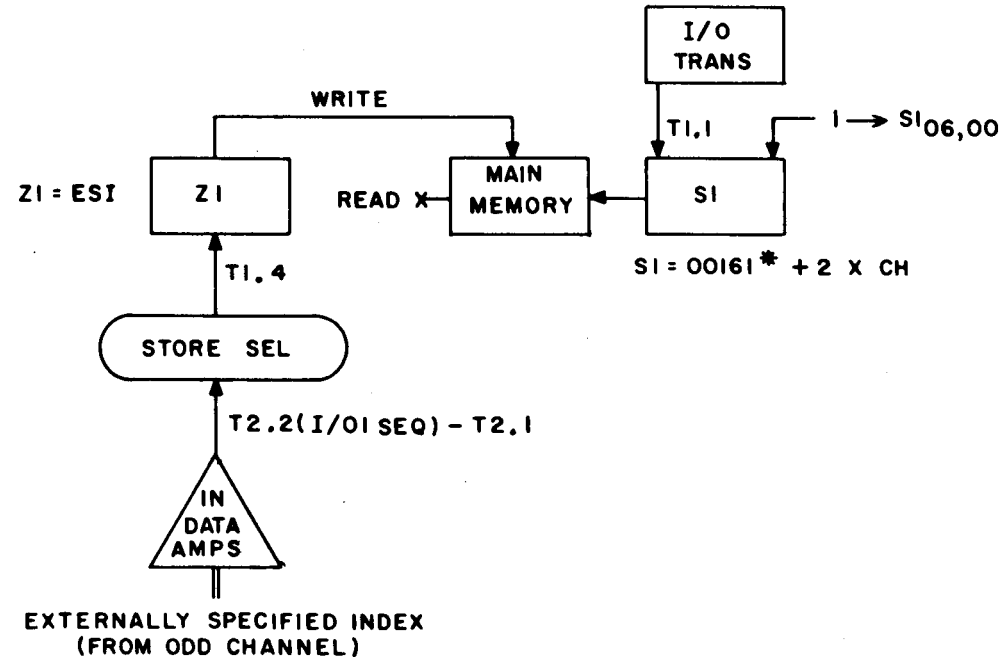


Figure 2-118. Store Select L1 → SO Transfer for ESI



NOTE: * IF CHANNEL IS $10-17_8$, ADD 00200_8 TO ADDRESS IN SI.

Figure 2-119. I/O2-Sequence Data Flow for Terminate ID Operations With ESI

TABLE 2-127. SCAN, I/01, AND I/02-SEQUENCE ESSENTIAL COMMANDS FOR ID OPERATIONS WITH ESI

Time Notation	Commands
	SCAN SEQUENCE
T2.1	Clear Fct, Chan, and I/O Trans 1
T2.2	Set $I/01_i$ ff, Pri → Fct, Chan, and I/O Trans 1 (set ESI ff), Odd chan Input Data Amps → Store Sel
T3.1	Store Sel L1 → SO (ESI), 1 → SO_{00}, Init CM, clear ZO
T3.2	Clear B
T3.4	ZO → B*
T4.1	Store Sel L1 → SO (ESI), Init CM, clear ZO, clear I/O Trans 2 Clear B ± 1 ff (+1 → B-Network)
T4.2	Set $I/01_f$ ff, set B ±1 ff (-1 → B-network) if B_{17} = 1 disable B-network 17,16 I/O Trans 1 → I/O Trans 2, clear all ID Ack Reg ff's
T4.3	Clear ID Req ff, set ID Ack Req ff, set 6XLg0 ff, clear Prio Alternator ff, clear $\overline{\text{Term}}$ ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{15-0} \neq ZO_{15-0}$, disable CM → ZO, disable Mem → Z1, clear S1 Set Cont Data Req ff if $(B_{15-0} = ZO_{15-0}) \cdot (ZO_{17} = 1)$
	I/01 SEQUENCE
T1.1	Store Sel L1 → SO (ESI), 1 → SO_{00}, Init CM, clear ZO B → S1, Init Memory
T1.2	Drop odd chan Input Data Amps → Store Sel, even chan Input Data Amps → Store Sel
T1.3	Clear Z1, clear ID Act ff if ($\overline{\text{Term}}$ ff clear) • (Cont Data Req ff clear) Set ID Mon ff if ($\overline{\text{Term}}$ ff clear) • $(B_{16} = 1)$

TABLE 2-127. SCAN, I/01, AND I/02-SEQUENCE ESSENTIAL COMMANDS FOR ID OPERATIONS WITH ESI (Cont)

Time Notation	Commands
T1.4	Store Sel → Z1, set ID Ack Gener ff if $\overline{\text{Term}}$ ff set
T2.1	Clear $I/01_i$ ff, drop even chan input Data Amps → Store Sel
T2.2	Set $I/02_i$ ff if $\overline{\text{Term}}$ ff clear, odd chan Input Data Amps → Store Sel if $I/02_i$ ff set
T2.4	Drop disable Mem → Z1, drop disable CM → ZO
T4.1	Clear $I/01_f$ ff drop-disable B-network$_{17,16}$
The following occurs if $I/02_i$ ff set (terminate)	
T4.2	Set $I/02_f$ ff, clear all ID Ack Req ff's
T4.3	Set ID Ack Req ff, set 6XLg0 ff
T4.4	Disable Mem → Z1, Clear S1
	I/02 SEQUENCE (if terminate)
T1.1	I/0 Trans → S1, 1 → $S1_{00}$, 1 → $S1_{06}$, Init Memory
T1.3	Clear Z1
T1.4	Store Sel → Z1, set ID Ack Gener ff
T2.1	Clear $I/02_i$ ff, drop odd chan Input Data Amps → Store Sel
T2.4	Drop disable Mem → Z1
T4.1	Clear $I/02_f$ ff

*B-network → ZO is timed by CM timing.

TABLE 2-128. SCAN, I/O1, AND I/O2 COMMANDS/DESCRIPTIONS FOR ID OPERATIONS WITH ESI

(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
	SCAN SEQUENCE
T2.2	Set ESI ff. The high from the Function Priority gate 12Vg4 (refer to plate P-66) is applied to 15V34 (refer to plate P-50), inverted and applied via 17V34, to 11G10. The other input to 11G10 is a H ⇒ ESI Mode Odd, developed at gate 91Vg1 and 90Vg1 (refer to plate P-67) when CHANNEL FUNCTION switch, S10A, is placed to the ESI position. The high that was applied to 15V34 (refer to plate P-50) was also applied to and inverted by 15V14 and applied as a L ⇒ Odd Group to 13E00 (refer to plate P-49). Gate 10E00 outputs a high due to a low from 03T22, a low from 40E00, a low from 20E00, and the absence of a H ⇒ I/O1 • (Dual + ESI Term) + Cont Data Req. The high is inverted by 12E00 and at $\phi 2$, 13E00 outputs a L ⇒ Odd Channel Gate to ESI flip-flop OXG10 (refer to plate P-50) and with the low from 11G10, OXG10 Sets.
	ODD Chan Input Data Amps → Store Sel. Due to the absence of a L ⇒ T21 I/O Seqs, 18N09 outputs a low to 19N09 (refer to plate P-26) and 39N09. The other inputs are a L ⇒ I/O Seq and either a L ⇒ Upper to 39N09 or a L ⇒ Lower to 19N09 from I/O translator 1 - Group and Mode (refer to plate P-50). The lows from either of these gates are applied to Store Select (refer to plate P-108 through P-110).
T3.1	Store Sel L1 ⇒ SO (ESI), Init CM. Gate 20N10 (refer to plate P-24) is enabled by a L ⇒ ESI, L ⇒ $\overline{\text{I/O2 Seqs}}$, and a low from 31T31 (due to a high from 00T31), a L ⇒ I/O Seq, and the absence of a H ⇒ (ESI + ESA) • I/O2 + Ext. The high is inverted by 28N10 and at $\phi 1$, 29N10 produces a L ⇒ ESI → SO to SO-register (refer to plate P-114) and to Control Memory (refer to plate P-122).
	1 → SO_{00}. With the L ⇒ ESI → SO and a L ⇒ Set SO Bit 2^0 (same as previously described in the Scan, I/O1 and I/O2 Sequence) OXS00 is set (refer to plate P-114).
T4.1	Store Sel L1 → SO (ESI), Init CM. This command is executed the same as described at T3.1, except the input to 31T31 (refer to Plate P-24) is a high from 00T41.
	I/O1 SEQUENCE
T1.1	Store Sel L1 → SO (ESI), 1 → SO_{00}, Init CM. This command is executed the same as described at T3.1, except the input to 31T31 (refer to plate P-24) is from 04T11.

TABLE 2-128. SCAN, I/O1, AND I/O2 COMMANDS/DESCRIPTIONS FOR ID OPERATIONS WITH ESI (Cont)

Time Notation	Command/Description
T1.2	Drop Odd Chan Input Data Amps → Store Sel. Gate 18N09 (refer to plate P-26) now outputs a high due to a L ⇒ T21 I/O Seq, and gates 19N09 and 39N09 are disabled. Even Chan Input Data Amps → Store Sel. Gate 44G00 (refer to plate P-52) is disabled by the high input from the Dual and Odd Group flip-flops not being set, and outputs a L ⇒ Even into 30N09 (refer to plate P-26). With the L ⇒ T21 I/O Seqs, 30N09 outputs a high which is inverted by 08N09 and applied to 09N09 and 29N09 to produce low outputs, as described at T2.2 for the Odd gates.
T1.4	Set ID Ack Gener ff if $\overline{\text{Term}}$ ff Set. If $\overline{\text{Terminate}}$ flip-flop OXG19 (refer to plate P-106) is set, the low is inverted by 02G19 and applied as a H ⇒ $\overline{\text{Terminate}}$ to 42G10 (refer to plate P-52). The low output is inverted by 30G09 to disable 31G09. The low is then applied to 00E04 (refer to plate P-53) and with the other inputs of a L ⇒ ID/EI and a L ⇒ T21 I/O Seq a high is outputted and inverted by 01E04. The output of 01E04 is applied as a L ⇒ Set ID Ack to ID Ack Generator flip-flop LXLg1 (refer to plate P-70) which will set with the low from flip-flop 6XLg0 (set at time T4.3) and a ϕ4.
T2.1	Drop Even Chan Input Data Amps → Store Sel. Gate 31N09 (refer to plate P-26) is now disabled by the absence of a L ⇒ T21 I/O Seqs.
T2.2	Set $I/O2_i$ ff if $\overline{\text{Term}}$ ff Clear. If $\overline{\text{Terminate}}$ flip-flop OXG19 (refer to plate P-106) is clear, the high is inverted by 02G19 and applied as a L ⇒ Terminate to 42G10 (refer to plate P-52). With the low from the set side of ESA ESI flip-flop 4XG10, the high is inverted by 30G09 and applied as a L ⇒ Dual + ESI Terminate to set $I/O2_i$ flip-flop 1XG26 (refer to plate P-13). Odd Chan Input Data Amps → Store Sel if $I/O2_i$ ff Set. This command is executed the same as described at time T2.2 of the Scan Sequence except that the output of $I/O2_i$ flip-flop is used to produce the L ⇒ I/O Seqs (refer to plate P-26).
	I/O 2 SEQUENCE (IF TERMINATE)
T1.1	I/O Trans → S1, 1 → $S1_{00}$, 1 → $S1_{06}$. With 44G10 enabled (refer to plate P-52) by the low from 4XG10 and a L ⇒ I/O 2 Seq, the high output is inverted by 42G05 and applied as a L ⇒ Ext Int + ESI • Pass 2 to 30N12 (refer to plate P-22). With a L ⇒ I/O Seq and a low from 03T11, 30N12 outputs a high which is inverted by 38N12 and at ϕ1, 39N12 produces a L ⇒ I/O Translator → S1 (refer to plate P-1 The high from 30N12 is also applied to 58N12 and 35N12. With the absence of a L ⇒ Cont Data Seq, 35N12 produces a L ⇒ Set S1 = 100 into 11S06. The low from 42G05 (Refer to plate P-52) is inverted by 45G05 and applied to 58N12 (refer to plate P-22), which produces a L ⇒ Set S1 Bit 2^0 into 11S00.
T2.1	Drop Odd Chan Input Data Amps → Stor Sel. Gate 18N09 (refer to pl P-26) outputs a high to disable 19N09 and 39N09 because of a L ⇒ T2 I/O Seq.

2-500. INPUT DATA OPERATIONS WITH ESA. The externally specified address mode (ESA) is instated for a particular channel by positioning the CHANNEL FUNCTION switch on its drawer to the ESA position. When in effect, alters the operations which honor an IDR on its channel. Two channels are used together; the one on which the IDR is sent which must be odd-numbered and the other is the next lower even-numbered channel. Instead of the storage address for input data being determined by the CACW, the external device directly specifies the address. Refer to figure 2-120 for the input word format.

2-501. The IDR signal must come from an d-channel. The input word on that channel is accepted as the data storage address. The word on the even channel is stored at this address. There are no buffer limits (address control words). The external device can rminate the input buffer only if the 16 least gnificant bits inputted as the address are all zeros. Upon termination, the input monitor interrupt signal is generated if bit 16 = 1_2 for the odd channel. Also, at termination, the externally specified address is stored at the dress 00161_8 + 2k (k channel number) if the channel is 0-7, or 00361_8 +2k if the channel is $10\text{-}17_8$.

2-502. Scan, I/O1, and I/O2 Sequence Data Flow Block Diagram For ID Operations with ESA. Refer to figure 2-121 for a block diagram description of the ID operations with ESA. During the last portion of the Scan sequence, the externally specified address is inputted and placed in B. The circuitry which is normally used to compare TACW with CACW is still effective in examining ZO_{15-0} and B_{15-0}. If $B_{15-0} = 0$, the input operations will terminate after this current data storage. The I/O1 sequence performs the data storage from the even channel using the externally specified address in B. The input monitor-interrupt signal can be generated upon termination if selected by bit 16 of this address word. The I/O2 sequence is used only if the input operation is terminated. It performs the storage of the ESA.

2-503. Scan, I/O1, and I/O2 Essential Commands For ID Operations with ESA. Refer to table 2-129 for a list of Scan, I/O1, and I/O2 sequence events. The events concerning the Continuous Data Mode sequence are analyzed in a later section. The ID-Acknowledge timing is the same as for normal input operations. Commands that have not been previously described are described in table 2-130 in detail.

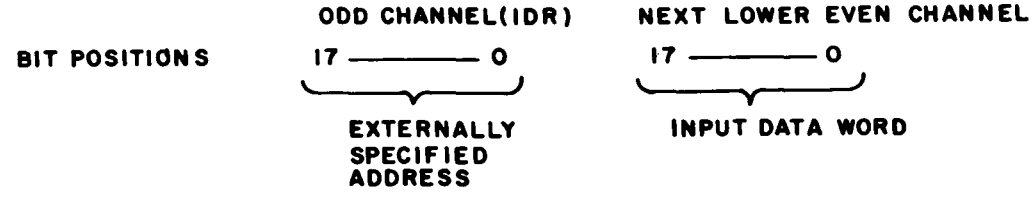


Figure 2-120. ESA Input Word Format for IDR Operations

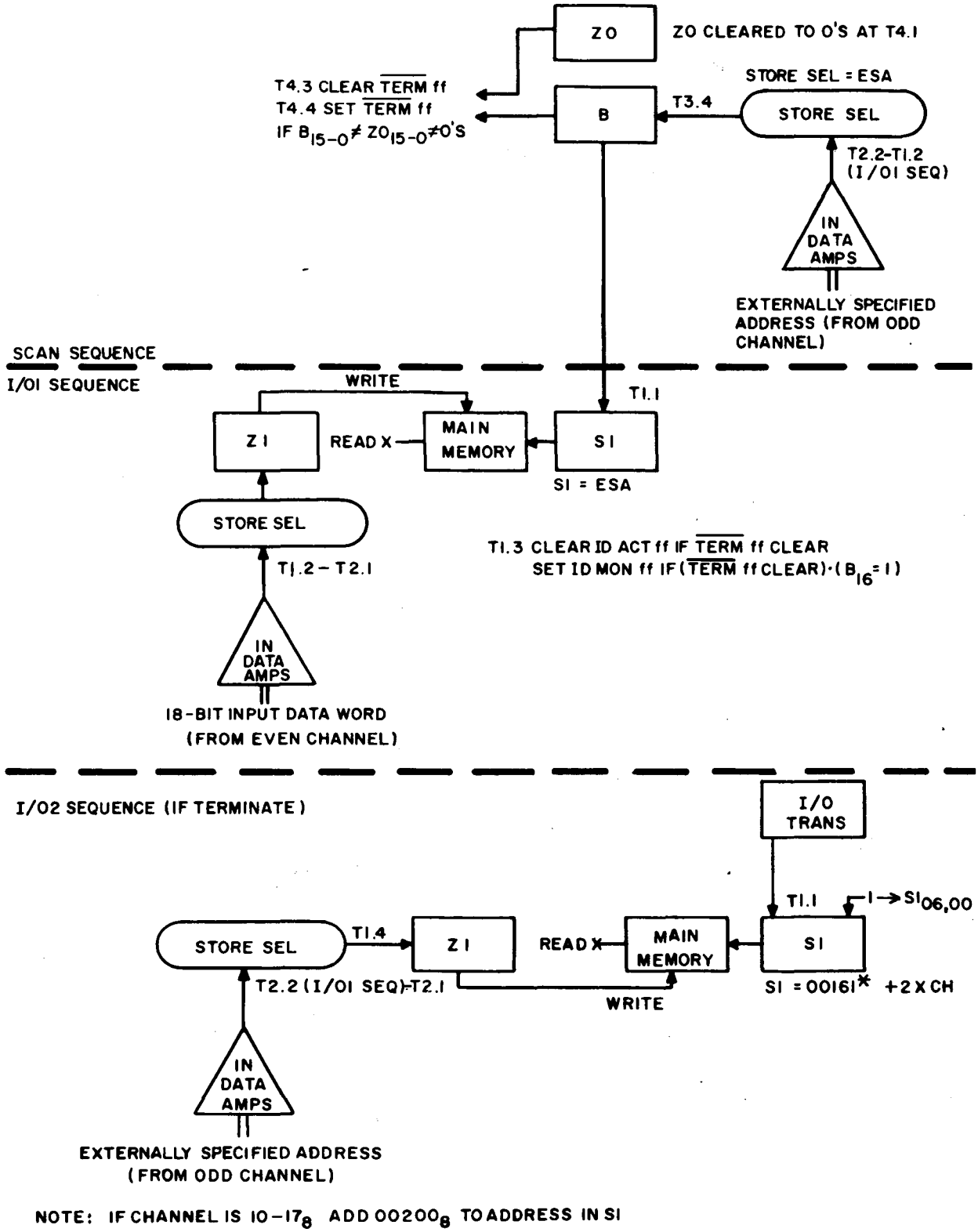


Figure 2-121. Scan, I/O1, and I/O2-Sequence Data Flow for ID Operations with ESA

TABLE 2-129. SCAN, I/01, AND I/02-SEQUENCE ESSENTIAL COMMANDS FOR ID OPERATIONS WITH ESA

Time Notation	Commands
	SCAN SEQUENCE
T2.1	Clear Fct, Chan, & I/O Trans 1
T2.2	Set $I/01_i$ ff, Prio→Fct, Chan, & I/O Trans 1 (set ESA ff)
	Odd chan Input Data Amps → Store Sel
T3.2	Clear B
T3.4	Store Sel → B
T4.1	Clear I/O Trans 2, clear ZO
T4.2	Set $I/01_f$ ff, I/O Trans 1 → I/O Trans 2, clear all ID Ack Reg ff's
T4.3	Clear ID Req ff, set ID Ack Reg ff, set 6XLg0 ff
	Clear $\overline{\text{Term}}$ ff, clear Prio Alternator ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{15-0} \neq ZO_{15-0}$, clear S1, disable Mem → Z1
	I/01-SEQUENCE
T1.1	B→S1, Init Memory
T1.2	Drop odd chan Input Data Amps → Store Sel
	Even chan Input Data Amps → Store Sel
T1.3	Clear Z1, clear ID Act ff if $\overline{\text{Term}}$ ff clear
	Set ID Monitor ff if ($\overline{\text{Term}}$ ff clear)· ($B_{\overline{16}}$ = 1)
T1.4	Store Sel→ Z1, set ID Ack Gener ff if $\overline{\text{Term}}$ ff set
T2.1	Clear $I/01_i$ ff, drop even chan Input Data Amps → Store Sel
T2.2	Set $I/02_i$ ff if $\overline{\text{Term}}$ ff clear, odd chan Input Data Amps → Store Sel if $I/02_i$ ff set
T2.4	Drop disable Mem → Z1
T4.1	Clear $I/01_f$ ff
ıe following occurs if $I/02_i$ ff set (terminate)	
T4.2	Set $I/02_f$ ff, clear all ID Ack Reg ff's
T4.3	Set ID Ack Reg ff, set 6XLg0 ff
T4.4	Disable Mem → Z1, clear S1

TABLE 2-129. SCAN, I/01, AND I/02-SEQUENCE ESSENTIAL COMMANDS FOR ID OPERATIONS WITH ESA (Cont)

Time Notation	Commands
	SCAN SEQUENCE
	I/02 SEQUENCE (if terminate)
T1.1	I/O Trans → S1, 1 → $S1_{00}$, 1 → $S1_{06}$, Init Memory
T1.3	Clear Z1
T1.4	Store Sel → Z1, set ID Ack Gener ff
T2.1	Clear $I/02_i$ ff, drop odd chan Input Data Amps → Store Sel
T2.4	Drop disable Mem → Z1
T4.1	Clear $I/02_f$ ff

TABLE 2-130. SCAN, I/O1, AND I/O2-SEQUENCE COMMANDS/DESCRIPTIONS FOR ID OPERATIONS WITH ESA
(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
T2.2	Set ESA ff. ESA flip-flop OXG11 (refer to plate P-50) is set as described for ESI flip-flop in ID Operations with ESI, except that switch S10A (refer to plate P-97) is placed to the ESA position and 91Vg2 produces a H ⇒ Mode Odd to 11G11 (refer to plate P-50).
T3.4	Store Sel → B. Gate 10N08 (refer to plate P-21) outputs a high because of a L ⇒ I/0 Seq, L ⇒ ESA Mode, and a low from 03T34. The high is inverted by 18N08 and, at Ø4, 19N08 produces a L ⇒ Input → B (refer to plates P-118 through P-120) to allow Store Sel → B.

2-504. OUTPUT DATA REQUEST OPERATIONS WITHOUT ESI OR ESA. The output data request is a signal from an external device which request the computer to output a data word.

2-505. Single-Channel Operation. The output data request (ODR) from an external device is honored by the computer only if the particular channel is output active (refer to figure 2-122). The Output Data (OD) Active flip-flop is set during the execution of f = 50:02 or 50:12. If the priority logic selects an ODR, the current address control word (CACW) is obtained from the control memory address 00041_8 + 2K (K = channel number) if the channel is 0-7 or 000241_8 + 2K if the channel is $10\text{-}17_8$. The terminal address control word (TACW) is obtained from the control memory address 00040_8 + 2K if the channel is 0-7 or 00240_8 + 2K if the channel is $10\text{-}17_8$. The 18-bit word which is outputted is obtained from the address specified by the $CACW_{15-0}$. The $CACW_{15-0}$ is modified by ±1 as determined by the value in bit position 17 of the CACW and is restored in control memory. $CACW_{17}$ determines whether the buffer is to be forward or backward. In a forward buffer ($CACW_{17} = 0_2$), the outputted words are obtained from consecutively incremented addresses as the ODR signals occur. In this case, $CACW_{15-0}$ is modified by +1 after each word is outputted. In a backward buffer ($CACW_{17} = 1_2$), the outputted words are obtained from consecutively decremented addresses as the ODR signals occur. In this case, the $CACW_{15-0}$ is modified by -1 after each word is outputted. Refer to tables 2-131 and 2-132 for buffer examples.

2-506. TACW is not modified but is compared with CACW. If their 16 least significant bits are equal before the modification of CACW, and if the plug-in jumper card selects 1219B or 1218 NTDS compatible I/O modes, the current output word being processed completes the buffer. If 1218 (normal) mode is in effect, the current output word is not transmitted. Thus, the buffer limits in the program must be one greater than the number of words to be sent. To terminate the buffer the OD Active flip-flop for the particular channel is cleared, disabling recognition of future ODR signals. At this buffer termination time, the value in bit position 16 of CACW is sensed. If $CACW_{16} = 1_2$, an output data monitor signal is to be generated to indicate termination to the program; thus, the OD Monitor flip-flop for the particular channel is set. The resulting monitor signal is handled later by the computer as a separate request for I/O service and is discussed in a later section. The OD Acknowledge signal is sent on the same channel which carried the request.

2-507. If $TACW_{17} = 1_2$, and the jumper card selects 1219B mode, Continuous Data Mode (CDM) is specified. This bit is sensed when the buffer is terminated and allows the special CDM sequence to be activated. This sequence prevents clearing of the OD Active flip-flop and reloads TACW and CACW locations in control memory with new buffer control words. CDM operations are presented in a later section.

2-508. Dual-Channel Operation. The dual-channel mode is initiated for a particular channel by positioning the drawer mounted CHANNEL FUNCTION switch to the DUAL position. If dual-channel has been selected, the ODR signal is requesting the computer to output two separate 18-bit words, one on the request channel, which must be odd for dual-channel operation, and the other word on the next lower channel which is even-numbered. Output operations for the second word are identical to those for the first. The second word is obtained from memory at the address specified by $CACW_{15-0}$, which

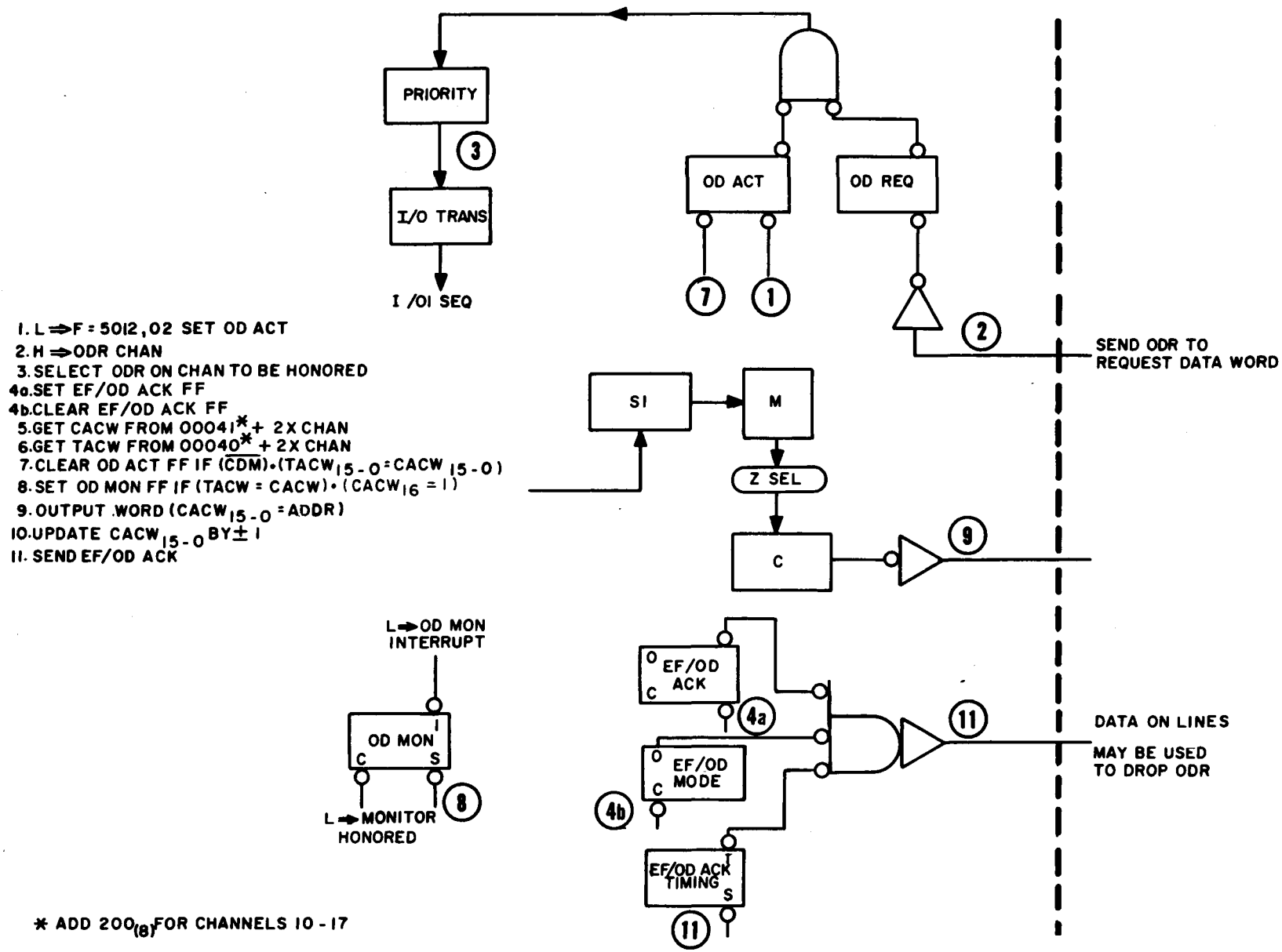


Figure 2-122. Output Data Operations, Single Channel without ESI and ESA

TABLE 2-131. SINGLE-CHANNEL FORWARD OD OR EF BUFFER
IACW = 005000, TACW = 005003

Address of Outputted Word ($CACW_{15-0}$)	Word Output Sequence
05000	1
05001	2
05002	3
05003	4

TABLE 2-132. SINGLE-CHANNEL BACKWARD OD OR EF BUFFER
IACW = 405003, TACW = 405000

Address of Outputted Word ($CACW_{15-0}$)	Word Output Sequence
05000	4
05001	3
05002	2
05003	1

has been modified by 1 during outputting of the first word. The type of buffer (forward or backward) determines which of the two channels is serviced first. The odd channel, ch must carry the ODR, receives the first 18-bit word if the buffer is forward. The even channel is serviced first if the buffer is backward. Refer to tables 2-133 and 2-134 for buffer examples, assuming the ORD signals occur on channel 3.

2-509. The 16 least significant bits of TACW and CACW are compared during each 18-bit word output operation. Therefore clearing the OD Active flip-flop can occur during either first or second word output operation. TACW and IACW should be chosen to provide a buffer for an exact multiple of two 18-bit words so that the buffer will always terminate as a result of the second word output operation from a particular ODR or EFR. The OD Acknowledge signal is sent on the odd channel which carried the ODR. The acknowledge occurs after the second word has been outputted.

2-510. Scan and I/O Sequence Data Flow for Single-Channel Operation. Refer to figure 2-123 for a block diagram description of the single channel OD operation. The Scan sequence is the main timing cycle during which the $I/01_i$ flip-flop is set upon detection and selection of an ODR signal. During this sequence CACW is obtained from control memory at the address 00041_8 + 2K (K = channel number) or at 000241_8 + 2K depending upon the channel. The address of either 00040_8 + 2K or 000240 + 2K is formulated in the I/O translator which is set according to the priority selection. $S0_{00}$ is set to 1_2 to make the address odd. Another control memory reference is used to obtained TACW. The TACW and CACW are compared from Z0 and B.

TABLE 2-133. DUAL-CHANNEL FORWARD OD BUFFER ON CHANNELS 2 AND 3
IACW = 005000, TACW = 005003

	Address of Outputted Word ($CACW_{15-0}$)	Receiving Channel	Word Output Sequence	ODR Sequence
1st 36-bit	05000	3	1	1st ODR
output word	05001	2	2	
2nd 36-bit	05002	3	3	2nd ODR
output word	05003	2	4	

TABLE 2-134. DUAL-CHANNEL BACKWARD OD BUFFER ON CHANNELS 2 AND 3
IACW = 405003, TACW = 405000

	Address of Outputted Word ($CACW_{15-0}$)	Receiving Channel	Word Output Sequence	ODR Sequence
2nd 36-bit	05000	3	4	2nd ODR/E
output word	05001	2	3	
1st 36-bit	05002	3	2	1st ODR/EFR
output word	05003	2	1	

The result of this comparison is recorded by the Terminate flip-flop. If $TACW_{15-0}$ = CACW 15-0, this ODR will be the last to be honored. In 1218 (normal) mode, this ODR will not be honored. The B-network is used to modify the $CACW_{15-0}$ as determined by $CACW_{17}$. The value of this bit is recorded by the B $\pm$ 1 flip-flop. The I/O-sequence is used to obtain and output the 18-bit word. The memory reference used the address specified by the unmodified $CACW_{15-0}$ from B. A control memory reference is used at time T1.1 to store the modified CACW value for use during the next output operations are used only if the operation is dual-channel.

2-511. Scan and I/O-Sequence Data Flow For Dual-Channel Operation. Refer to figures 2-123 and 2-124. The Scan and I/O1 sequences are used for single-channel operations. The last portion of the I/O1 sequence, along with the I/O2 sequence is used to perform the same operations executed during the last portion of the Scan Sequence and the first portion of the I/O1 sequence. These operations obtain t second word from the first word address $\pm$1.

2-512. Scan, I/O1, I/O, and I/O2 Sequence Essential Commands for Dual-Channel OD Operations. Refer to table 2-135 for a list of essential Scan, I/O1, I/O2, and I/O-sequence events. Events concerning priority selection have been previously described and are not shown. Events concerning the CDM Mode sequence, which is initiated as a result of the setting of the CDR flip-flop, are an yzed in a later section. The OD Acknowledge timing is also discussed later in this section. Commands which have not been described previously are described in table 2-136 in detail.

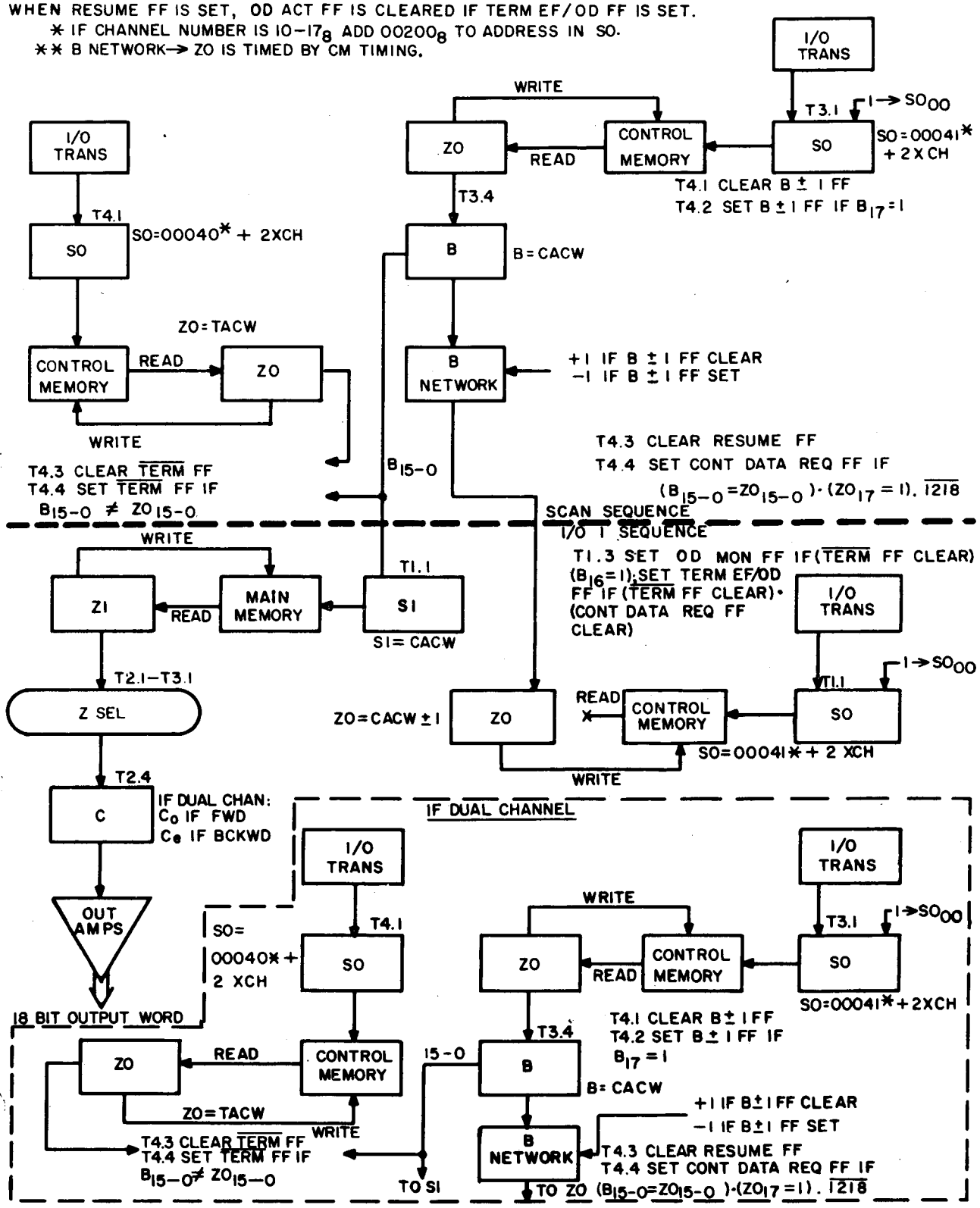
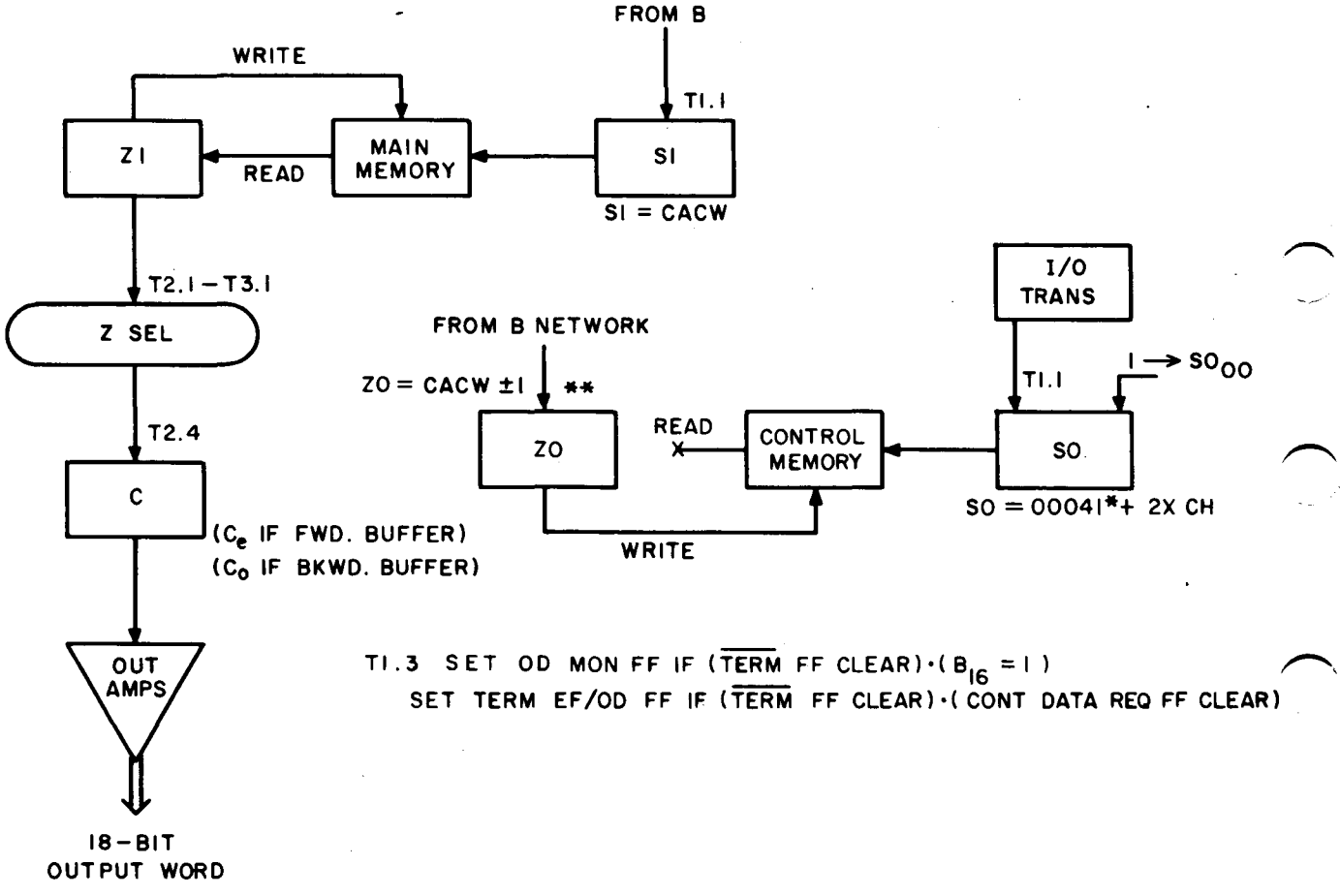


Figure 2-123. Scan and I/01-Sequence Data Flow for Single or Dual-Channel OD Operations.



NOTES: WHEN RESUME FF IS SET, OD ACT FF IS CLEARED IF TERM EF/OD FF IS SET.

*IF CHANNEL NUMBER IS $10-17_8$, ADD 00200_8 TO ADDRESS IN SO.

**B NETWORK ⟶ Z0 IS TIMED BY CM TIMING.

Figure 2-124. I/02 Sequence Data Flow for Dual-Channel OD Operations.

TABLE 2-135. SCAN, I/01, AND I/02 SEQUENCE ESSENTIAL COMMANDS FOR SINGLE-OR DUAL-CHANNEL OD OPERATIONS

Time Notation	Commands
	SCAN SEQUENCE
T2.1	Clear Fct, Chan, and I/0 Trans 1
T2.2	Set $I/01_i$ ff, Prio → Fct, Chan, and I/0 Trans 1
T3.1	I/0 Trans → S0, 1 → $S0_{00}$, Init CM, clear ZO
T3.2	Clear B
T3.4	ZO → B*
T4.1	I/0 Trans → SO, Init Cm, clear Z0, clear B±1 ff (+1 → B-network), clear I/0 Trans 2
T4.2	Set $I/01_f$ ff, set B±1 ff (-1→B-network) if $B_{17} = 1$, I/0 Trans 1 → I/0 Trans 2 (set Dual ff if dual chan selected), disable $B\text{-network}_{17,16}$
T4.3	Clear $\overline{\text{Term}}$ ff, clear Resume ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{15-0} \neq ZO_{15-0}$, disable CM → ZO Clear S1, set Cont Data Req ff if $(B_{15-0} = ZO_{15-0}) \cdot (Z0_{17} = 1)$
	I/01 SEQUENCE
T1.1	I/0 Trans → S0, 1 → $S0_{00}$, Init CM, B → S1, Init Memory
T1.2	Clear all EF/OD Ack Reg ff's
T1.3	Clear Z1, clear EF/OD mode ff, set EF/OD Ack Reg ff, clear OD Req ff if (inter-cmptr and EF)+($\overline{\text{inter-comptr}}$ · ODR) set OD Mon ff if ($\overline{\text{Term}}$ ff clear) · ($B_{16} = 1$), set Prio Alternator ff set Term OD ff if ($\overline{\text{Term}}$ ff clear) · (Cont Data Req ff clear)
T1.4	set Ack Delay ff if Dual ff clear
T2.1	Clear $I/01_i$ ff, Z1 → Z-Sel
T2.2	Set $I/02_i$ ff if Dual ff set

TABLE 2-135. SCAN, I/01, AND I/02 SEQUENCE ESSENTIAL COMMANDS FOR SINGLE-OR DUAL-CHANNEL OD OPERATIONS (Cont)

Time Notation	Commands
T2.3	Clear C**
T2.4	Z-Sel → C**, drop disable CM → ZO
The following occurs if $I/02_i$ ff set (dual-channel) excepting, clear $I/01_f$ ff, and drop Z1 → Z-Sel	
T3.1	I/0 Trans → S0, 1 → $S0_{00}$, Init CM, Clear ZO, drop Z1 → Z-Sel
T3.2	Clear B
T3.4	Z0 → B*
T4.1	Clear $I/01_f$ ff, I/O Trans → SO, Init CM, clear ZO clear B±1 ff (+1 → B-network), clear I/O Trans 2, drop disable $B\text{-network}_{17,16}$
T4.2	Set $I/02_f$ ff, set B±1 ff (-1 → B-network) if $B_{17}=1$, I/O Trans 1 → I/O Trans 2 (set Dual ff), disable $B\text{-network}_{17,16}$ if $I/02_f$ ff set
T4.3	Clear $\overline{\text{Term}}$ ff, clear Resume ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{15-0} \neq ZO_{15-0}$, disable CM → ZO Clear S1, set Cont Data Req ff if $(B_{15-0} = ZO_{15-0}) \cdot (Z0_{17} = 1)$
	I/02 SEQUENCE
T1.1	I/0 Trans → S0, 1 → $S0_{00}$, Init CM, clear Z0 B → S1, Init Memory
T1.2	Clear all EF/OD Ack Reg ff's
T1.3	Clear Z1, set EF/OD Ack Reg ff set OD Mon ff if ($\overline{\text{Term}}$ ff clear) • (B_{16} = 1) set Term OD ff if ($\overline{\text{Term}}$ ff clear) • (Cont Data Req ff clear)
T1.4	Set Ack Delay ff
T2.1	Clear $I/02_i$ ff, Z1 → Z-Sel
T2.3	Clear C**

TABLE 2-135. SCAN, I/01, AND I/02 SEQUENCE ESSENTIAL COMMANDS FOR SINGLE-OR DUAL-CHANNEL OD OPERATIONS (Cont)

Time Notation	Commands
T2.4	Z-Sel → C**, drop disable CM → Z0
T3.1	Drop Z1 → Z-Sel
T4.1	Clear $I/02_f$ ff drop disable $B\text{-network}_{17,16}$

*B-network → Z0 is timed by CM timing.

**If operation is in dual channel mode, the C-register which is cleared and receives word from Z-select is as follows:

	I/01 Seq	I/02 Seq
B±1 ff clear (forward buffer)	Co	Ce
B±1 ff set (backward buffer)	Ce	Co

TABLE 2-136. SCAN, I/01, I/0, AND I/02 SEQUENCE COMMANDS/DESCRIPTIONS FOR DUAL-CHANNEL OD OPERATIONS

(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
	SCAN SEQUENCE
T4.3	Clear Resume ff. Being in an OD operation, gate 20G06 (refer to plate P-51) is enabled by the Function Translator and outputs a high which is inverted by 21G06 and applied to 00N40 (refer to plate P-53) as a L ⇒ Out + EF. The other inputs into 00N40 are L ⇒ I/O Seqs and a low from 03T43. The high is inverted by 01N40 and applied to Resume flip-flop 5XLg0 (refer to plates P-69 and 69A). The other inputs are L ⇒ Group g and a Ø3, which clears the flip-flop.
	I/01-SEQUENCE
T1.2	Clear all EF/OD Ack Reg ff's. With card 10G19 (refer to plate P-35) inserted for 1219 Mode Only, gate 11G19 is disabled and outputs a low to 53T13 (refer to plate P-14). The other input to 53T13 is a low from 51T13, enabled by a low from 01T13 and a low from 05G25. The low from 53T13 is applied to pin 11 of 00N48 (refer to plate P-53) and with a L ⇒ Out + EF, 00N48 outputs a high which is inverted by 01N48 and applied as a L ⇒ Set EF/OD Ack to 2gN48 (refer to plate P-58). The other low input is a L ⇒ Group g from the I/O Translator 1 - Group and Mode (refer to plate P-50). The output of 2gN48 is applied as a L ⇒ EF/OD to 3gN48 (refer to plate P-68) and at Ø2, the high output is inverted by 3gN50 and clears the EF/OD Acknowled flip-flops 4XVg0 through 4XVg3.
T1.3	Set EF/OD Ack Reg ff's. The L ⇒ EF/OD that cleared the flip-flops is also used to set the flip-flops. The other low input is the channel number which is outputted at Ø3 by either gate 31Gg0, 31Gg1, 31Gg2, or 31Gg3 (refer to plates P-59 and P-59A). Clear EF/OD MODE ff. The L ⇒ EF/OD that cleared the EF/OD Ack flip-flops is also applied to EF/OD Mode flip-flop 4XVg4 (refer to plate P-68) and with a L ⇒ OD applied, the flip-flop sets at Ø3. (Note that when in EF Mode the flip-flop will set.) Clear OD Req ff if (Inter-Cmptr and EF) + ($\overline{\text{Inter-Cmptr}}$ · ODR). If in Inter-Cmptr and EF, a L ⇒ EF from the Function Translator is applied through INTERCOMPUTER NORMAL switch pins 5 and 6 (refer to plates P-61 through P-64) to 93Rg0 (refer to plate P-60) and with the other low input from the selected channel, 93Rg0 outputs a H ⇒ Clear OD Oneshot to 6gN40 (refer to plate P-57). The other input is from

TABLE 2-136. SCAN, I/01, I/0, AND I/02 SEQUENCE COMMANDS/DESCRIPTIONS FOR DUAL-CHANNEL OD OPERATIONS (Cont)

Time Notation	Command/Description
T1.3 (Cont)	the Start Out/EF Time gate 00E06 (refer to plate P-53) which is enabled by a L ⇒ Out + EF, L ⇒ T21 I/0 Seqs and the absence of a H ⇒ (ESI Term + Dual) • I/01 Seq. The H ⇒ START EF/OD Ack is applied to 1gE06 (refer to plate P-57), inverted by 9gN48 and applied to 6gN40, which produces a L ⇒ Clear OD Req to OD Req flip-flop (refer to plates P-61 through P-64) which will clear with the selected channel input from the Channel Translator. If in $\overline{\text{Inter-Cmptr}}$ • ODR, gate 6gN40 (refer to plate P-57) is now enabled by a H ⇒ OD from the Function Translator and a high from 9gN48 to clear OD Req flip-flop as previously described. Set OD Mon ff if ($\overline{\text{Term}}$ ff Clear) • (B_{16} = 1). Gate 6gN42 (refer to plate P-56) is enabled by a H ⇒ EF/OD and a high from 9gN42 (Refer to ID operation table 1-124), T1.3 for the development of the L ⇒ Set Monitor into 9gN42). The L ⇒ Set OD Monitor from 6gN42 is applied to OD Mon flip-flops (refer to plates P-61 through P-64). The other low inputs to the flip-flops are the channel numbers. Set Term ODff if ($\overline{\text{Term}}$ ff Clear) • (Cont Data Req ff Clear). Set Term Out + EF gate 00E12 (refer to plate P-53) is enabled by a L ⇒ Terminate ($\overline{\text{Term}}$ flip-flop Clear), a low from 53T13 (developed previously to clear the OD Ack flip-flops), a L ⇒ Out + EF and the absence of a H ⇒ Cont Data Mode. The high output is inverted by 01E12 and applied as a L ⇒ Terminate Output to Terminate EF/OD flip-flop 7XLg0 (refer to plate P-69 and P-69A). The other inputs are a L ⇒ Group and a ∅3 to set the flip-flop. Set Prior Alternator ff. The L ⇒ EF/OD (refer to plate P-68) sets Priority Alternator ff 5XVg4 at ∅3.
T1.4	Set Ack Delay ff if Dual ff Clear. Gate 00E06 (refer to plate P-53) is enabled by a L ⇒Out + EF, a L ⇒ T21 I/0 Seq and the absence of a H ⇒ (ESI Term + Dual) • I/01 Seq. The output is applied as a H Start EF/OD Ack to 1gE06 and inverted and applied as a low to Ack Delay flip-flop 5XLg1 (refer to plate P-69 and P-69A). The other inputs are L ⇒ Group g and a ∅4, which sets the flip-flop.
T2.3	Clear C. Gate 44G10 (refer to plate P-52) is disabled by the absence of a L I/02 Seq, or ESA ESI flip-flop 4XG10 not being set, and the output is applied as a L ⇒ $\overline{\text{Disable Z}}$ ⇒C into gates 00N60, 10N60, 20N60 and 30N60 (refer to plate P-53). The other common input is a L ⇒I/0 Seq Out + EF. Depending upon Odd or Even and Upper or Lower

TABLE 2-136. SCAN, I/01, I/0, AND I/02 SEQUENCE COMMANDS/DESCRIPTIONS FOR DUAL-CHANNEL OD OPERATIONS (Cont)

Time Notation	Command/Description
T2.3 (Cont)	one of the gates will be enabled and outputs a H ⇒ Z → C into g1N60 (refer to plate P-58) and at ø3, g2N60 produces a H ⇒ Clear C via g3N60 to the C-register (refer to plates P-71 and P-72).
T2.4	Z — Sel → C. At ø4, g4N60 (refer to plate P-58) produces a L ⇒ Z → C into the C-register.

2-513. OD Acknowledge Timing. The OD acknowledge signal is sent after the word output operation. In the case of dual-channel operation, the acknowledge is sent after both words have been outputted. This signal indicates that a word is currently being outputted and is ready for receipt by the receiving device. The receiving device usually uses the acknowledge to time dropping of its ODR signal. The acknowledge signal is sent for a duration of 2.75 microseconds (fast interface) and 14.87 microseconds (slow interface). The EF/OD Mode flip-flop is clear to gate the OD Acknowledge. When operating in the inter-computer mode, the acknowledge timing assists in processing the next ODR from the receiving computer. When not in the inter-computer mode (refer to figure 2-125), the EF/OD Acknowledge Generator flip-flop determines the acknowledge duration by its set time. The The EF/OD Mode flip-flop clear determines that an OD Acknowledge is sent. The Resume flip-flop is set from the acknowledge timing via 52Lg1 and the INTER-COMPUTER/NORMAL switch. When set, the Resume flip-flop clears the OD Active flip-flop via 72Lg0 if the Terminate EF/OD flip-flop is set.

2-514. When in the Inter-Computer mode (refer to figure 2-125), most of the timing is the same as for Normal Mode. The major difference involves dropping of the acknowledge. This signal is applied to the other computer until that computer sends an ID Acknowledge indicating that it has received the word just outputted. The ID Acknowledge is detected by this computer as an ODR. Also, until this acknowledge is received, the Resume flip-flop is not cleared until the acknowledge, which appears as an ODR, is received. The delayed clearing of the active flip-flop allows recognition of one more ODR than is necessary to complete the buffer and set the Resume flip-flop which in turn can clear the active flip-flop. Until the Resume flip-flop is set, output is not available on the particular chassis. Therefore, no other word can be outputted via the same C-register, and the current output word will be applied to the receiving computer until it responds with the ID Acknowledge (ODR). As described above, this ODR signal sets the Resume flip-flop. If the buffer is not to terminate (OD Active flip-flop is not cleared when the Resume flip-flop is set), the Resume flip-flop completely satisfies 25Rg0, which generates the next ODR signal to request the next word to be outputted. The status of the Resume flip-flop provides an indication of whether or not the receiving computer acknowledged receipt of the last outputted word. If it is not set within a specific time period, the Resume Fault special interrupt is generated to notify the program. This signal is discussed in a later section. Refer to tables 2-137 and 2-138 for the timing analysis of acknowledges for fast and slow interface, respectively, for non-intercomputer modes. Commands not previously described are described in detail in tables 2-139 and 2-140. Refer to tables 2-141 and 2-142 for the timing analysis of a acknowledge for fast and slow interface, respectively, for inter-computer modes. Commands not previously described for fast interface are described in detail in table 2-143.

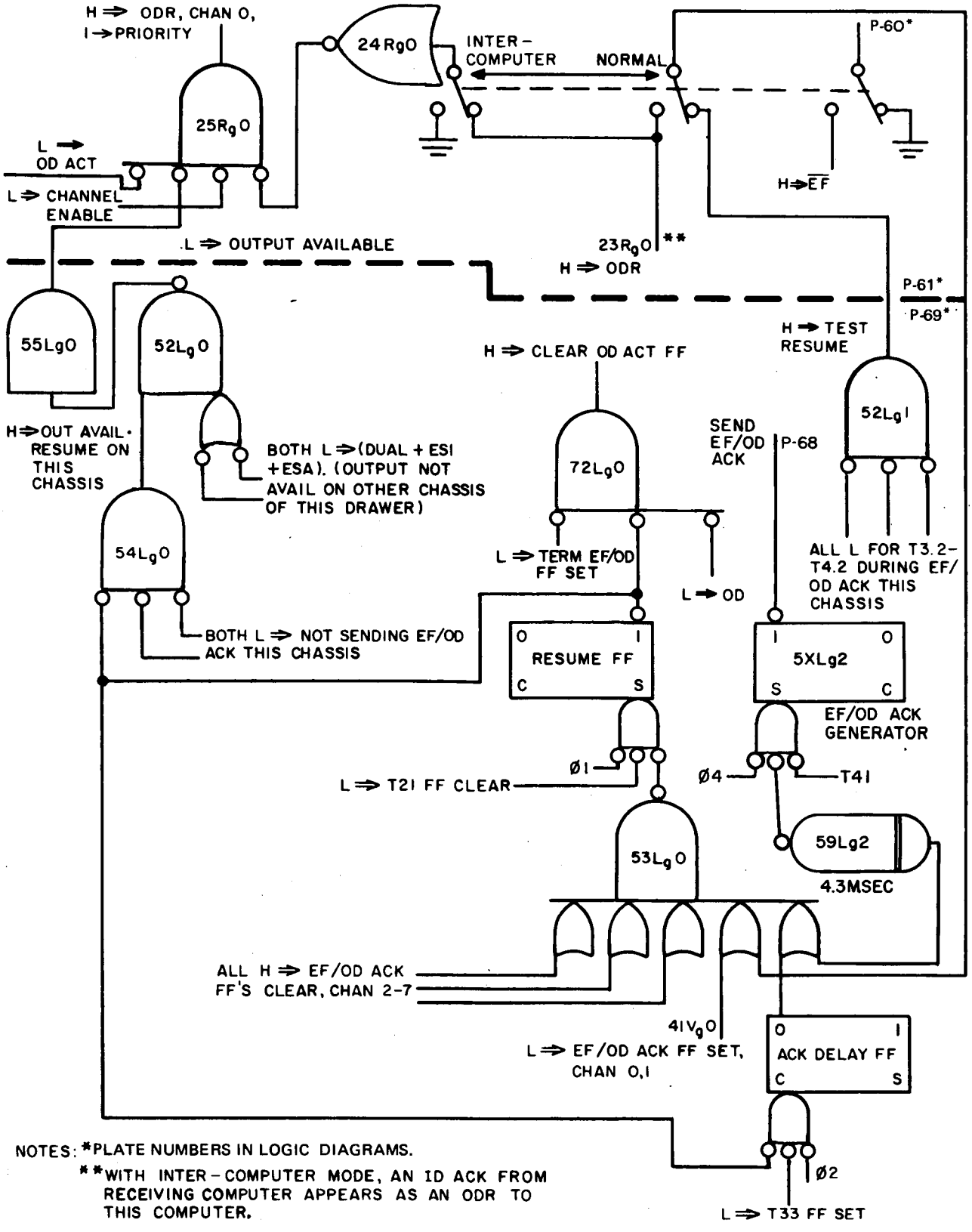


Figure 2-125. OD Acknowledge Simplified Logic

TABLE 2-137. OD ACKNOWLEDGE TIMING FOR FAST INTERFACE, NOT INTER-COMPUTER

Time Notation	Action
T1.2	Clear all EF/OD Ack Reg ff's
T1.3	Set EF/OD Ack Reg ff
	Set EF/OD Mode ff if EF Mode, clear EF/OD Mode ff if OD Mode
T1.4	Set Ack Delay ff
T3.4	Set EF/OD Ack Gener ff, send EF/OD Acknowledge signal
T4.1	Set Resume ff, clear OD Act ff if Term EF/OD ff set
T4.4	Clear Term EF/OD ff
75 μs T3.2	Clear Ack Delay ff
T1.3	Clear EF/OD Ack Gener ff, drop EF/OD Acknowledge signal

TABLE 2-138. OD ACKNOWLEDGE TIMING FOR SLOW INTERFACE, NOT INTER-COMPUTER

Time Notation		Action
	T1.2	Clear all EF/OD Ack Reg ff's
	T1.3	Set EF/OD Ack Reg ff
		Set EF/OD Control ff if EF Mode, clear EF/OD Control ff if OD Mode
4.3 μs delay of 59 Lg 2	T1.4	Set Ack Delay ff
	T1.4	
	T1.4	
	T2.2	
.75	T3.4	Set EF/OD Ack Gener ff, send EF/OD Acknowledge signal
	T3.4	
10.7 μsec delay of 59Lg3	T3.4	
	T3.4	
	T3.4	
14.875 μs	T3.4	
	T1.2	Set 5XLg3 ff
	T4.1	Set Resume ff, clear EF/OD Act ff if Term EF/OD ff set
	T4.4	Clear Term EF/OD ff
	T3.2	Clear Ack Delay ff
	T1.3	Clear EF/OD Ack Gener ff, drop EF/OD Acknowledge signal
	T2.2	Clear 5XLg3 ff.

TABLE 2-139. OD ACKNOWLEDGE TIMING COMMANDS/DESCRIPTIONS FOR FAST INTERFACE NOT-INTER-COMPUTER

(For Detailed Description of all Commands Previously Describe Refer to Table 2-110)

Time Notation	Command/Description
T3.4	Set EF/OD Ack Gener ff, Send EF/OD Acknowledge Signal. The low from the set side of Ack Delay flip-flop 5XLg1 (refer to plates P-69 & 69A) is applied to EF/OD Ack Generator flip-flop 5XLg2. The high from 04T41 is inverted by 8gT41 and at Ø4, flip-flop 5XLg2 sets and sends an EF/OD-Acknowledge signal.
T4.1	Set Resume ff. Only one of the EF/OD Ack Reg flip-flops (refer to plate P-68) was set at time T1.3 and the others were left clear. If flip-flop 4XVg0 (Channels 0, 1) was set, the low from the set side of the flip-flop is applied to 74Lg0 pin 12 and 53Lg0 pin 7 (refer to plate P-69 and 69A) as a L ⇒ Chan 0, 1 EF/OD Ack. All other inputs to 53Lg0 are a H ⇒ $\overline{\text{Chan EF/OD Ack}}$, from the set side of the EF/OD Ack Reg flip-flops. The input to pin 5 on 53Lg0 is a high from the clear side of Ack Delay flip-flop 5XLg1. Since pin 7 is a low, a high must be brought in on pin 8 to enable 53Lg0. With Ack Delay flip-flop set the low is applied to 52Lg1 (refer to plate P-69 and 69A) and, with the low from 8gT41 and pin 7 open, 52Lg1 outputs a H ⇒ Test Resume to 53Lg0 pin 8 via INTERCOMPUTER-NORMAL switch (refer to plate P-61). With a low from 04T21 Resume, the flip-flop sets at Ø1. Clear OD Act ff if Term EF/OD ff Set. If Terminate EF/OD flip-flop 7XLg0 (refer to plate P-69 and 69A) is set, a low is applies to 72 Lg0 and 74Lg0 through 74Lg3. The other inputs to 72Lg0 are a L ⇒ OD and a low from resume flip-flop 5XLG0. The high from 72Lg0 is inverted by 73Lg0 and applied as a L ⇒ Clear OD Act to OD Act flip-flop (refer to plates P-61 through P-64). Since only one of the EF/OD Ack Reg flip-flops are set (Channels 0, 1 for this example), only gate 74Lg0 will be enabled to produce a H ⇒ Terminate Chan 0, 1 to 75Lg0 (refer to plate P-61) and the low output will clear OD Act flip-flop 6XRg0.
T4.4	Clear Term EF/OD ff. The low output from the Resume flip-flop (refer to plate P-69 and 69A) is applied to Terminate EF/OD flip-flop which clears at Ø4.
T3.2	Clear Ack Delay ff. The low output of the Resume flip-flop is also applied to Ack Delay flip-flop 5XLg1, which clears with a L ⇒ T33 and a Ø2.
T1.3	Clear EF/OD Ack Gener ff, Drop EF/OD Acknowledge Signal. The low output from the clear side of the Ack Delay flip-flop is applied to EF/OD Ack Generator and, with the low from 05T13, clears at Ø3 and the EF/OD Acknowledge signal is dropped.

TABLE 2-140. OD ACKNOWLEDGE TIMING COMMANDS/DESCRIPTIONS FOR SLOW INTERFACE, NOT-INTER-COMPUTER
(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
T3.4	Set EF/OD Gener ff, Send EF/OD Acknowledge Signal. In slow interface the set side of the Ack Delay ff 5XLg1 (refer to plate P-69 and P-69A) is not used to set EF/OD Generator ff 5XLg2. At T1.4 a high was applied to Delay line 59Lg2 from the clear side of 5XLg1. Note that T1.4 appears two more times (4μs elapsed) but causes no action. At T2.2 approximately 4.3μs after Ack Delay ff was set, 59Lg2 outputs a low to 5XLg2 which will set on the next Ø4 due to the low input from 8gT41 and will now send the EF/OD acknowledge signal.
T1.2	Set 5XLg3 ff. The high from the clear side of ff 5XLg2 was applied to 5XLg3 at T3.4. Note that T3.4 appears five more times (10μs elapsed) but causes no action. At T.2 approximately 10.7μs after 5XLg2 was set, 59Lg3 outputs a low to set 5XLg3. NOTE: After flip-flop 5XLg3 is set, the commands are executed the same as for Fast Interface except at T2.2, flip-flop 5XLg3 is cleared.
T2.2	Clear 5XLg3 ff. When flip-flop 5XLg2 clears at T1.3, the low from the clear side is applied to 5XLg3, which clears at the next Ø2.

TABLE 2-141. OD-ACKNOWLEDGE TIMING FOR FAST INTERFACE, INTER-COMPUTER

Time Notation	Action
T1.2	Clear all EF/OD Ack Req ff's
T1.3	Set EF/OD Ack Req ff, Clear EF/OD Mode ff
T1.4	Set Ack Delay ff
T3.4	Set EF/OD Ack Gener ff, send OD Acknowledge signal
wait for next ODR	
receive ODR from other computer (ID Acknowledge)	
Next T1.1, T3.1, or T4.1	Set Resume ff, internally generate next ODR, clear OD Act ff if Term EF/OD ff set
Next Ø4	Clear Term EF/OD ff
T3.2	Clear Ack Delay ff
T1.3	Clear EF/OD Ack Gener ff, drop OD Acknowledge signal

TABLE 2-142. OD ACKNOWLEDGE TIMING FOR SLOW INTERFACE, INTER-COMPUTER

Time Notation		Action
	T1.2	Clear all EF/OD Ack Reg ff's
	T1.3	Set EF/OD Ack Req ff Clear EF/OD Mode ff
4.3 μs delay of 59 Lg 2	T1.4	Set Ack Delay ff
	T1.4	
	T1.4	
.75 μs	T2.2	
	T3.4	Set OD Ack Gener ff, send OD Acknowledge signal
	T3.4	
	T3.4	
10.7 μs delay of 9 Lg3	T3.4	
	T3.4	
	T3.4	
	T3.4	
	T1.2	Set 5XLg3 ff
wait for next ODR		
receive ODR from other computer (ID Acknowledge)		
	next T1.1, T3.1, or T4.1	Set Resume ff, internally generate next ODR, clear OD Act ff if Term OD ff set
	next Ø4	Clear Term OD ff
	T3.2	Clear Ack Delay ff
	T1.3	Clear OD Ack Gener ff, drop OD Acknowledge signal
	T2.2	Clear 5XLg3 ff

TABLE 2-143. OD ACKNOWLEDGE TIMING COMMANDS/DESCRIPTIONS FOR FAST INTERFACE, INTER-COMPUTER
(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
	RECEIVE ODR FROM OTHER COMPUTER (ID-ACKNOWLEDGE)
Next T1.1, T3.1 or T4.1	Set Resume ff, Internally Generate Next ODR, Clear OD Act ff if Term EF/OD ff Set. OD Req flip-flop 2XRg0 (refer to plate P-61, assuming Channel 0, 1 used), is set by either a L ⇒ Clear I/O or a low from 22Rg0 and a ∅2. When an ODR is received by inverter 25Yg0, the low is applied to 23Rg0, and with the low from the OD Req flip-flop, 23Rg0 outputs a high via INTERCOMPUTER NORMAL switch to 53Lg0 (refer to plate P-69 and P-69A) to set Resume flip-flop as previously described. Gate 24Rg0 (refer to plate P-61) now outputs a low to 25Rg0 which has other low inputs from the OD Act flip-flop, a L ⇒ Chan Enable and a L ⇒ Output Available which cause 25Rg0 to output a H ⇒ OD Req 0, 1. Note that Resume ff cannot set at T2.1 due to the high input from 04T21.

2-515. OUTPUT DATA REQUEST OPERATIONS WITH ESI. The Externally Specified Index (ESI) mode is instated for a particular channel by positioning the drawer mounted CHANNEL FUNCTION switch to ESI position. When in effect, it alters the operations which honor an ODR on its channel. Two channels are used together, one on which the ODR is sent, which must be odd-numbered, and the other the next lower even-numbered channel. Instead of TACW and CACW being obtained from their normal addresses in control memory, their control memory origins are specified by the requesting external device. The ODR signal must appear on an odd channel. There must be an ESI address word accompanying the request on this same channel. This word is the control memory address of TACW. CACW is obtained from the next consecutively higher address. The content of the address specified by $CACW_{15-0}$ is the requested output word. This word is outputted to both the requesting odd channel and the next lower even channel. Refer to figure 2-126 for the input and output word formats.

2-516. $CACW_{15-0}$ is modified by ± 1 depending upon its bit position 17. The outputted word is obtained from the address specified by the unmodified $CACW_{15-0}$. If $TACW_{15-0} = CACW_{15-0}$ (unmodified), the buffer is terminated and if $CACW_{16} = 1$, the output data/external-function monitor-interrupt signal is generated. The OD or EF Acknowledge signal is sent on the odd channel. The ESI Mode of operation is useful if more than one piece of equipment is connected to a common channel by means of a multiplexing device. Each piece of equipment can indirectly specify the address of the outputted word which it requests by means of the ESI. This index could be generated in the multiplexer as it requested each word. Each equipment would have a unique index value. The multiplexer must occupy an odd and the next lower even channels. Upon buffer termination, the ESI is also stored in main memory at the address $00141_8 + 2k$ ($k \Rightarrow$ channel number) if the channel is 0-7, or $00341_8 + 2k$ if the channel is $10\text{-}17_8$. By inspecting the content of this address, the program could determine which of the equipment on that channel received the last output word and made the channel inactive.

2-517. Scan and I/O-Sequence Data Flow For OD Operations with ESI (Not Buffer Termination). Refer to figure 2-127 for a block diagram description of the OD operations with ESI. Data flow is similar to that for normal OD operations. Exceptions are that the control memory addresses for CACW and TACW are from the odd-numbered channel. The ESI is used in SO at time T4.1 to obtain TACW. The transfer into SO involves a one place left-shift. SO has no bit position 06. The inputted bits 05 and 06 are therefore placed in SO_{07} and SO_{08} respectively. Nothing is placed in SO_{00}, the address of the TACW is always even. Refer to figure 2-128 for a description of the ESI transfer to SO.

2-518. The address of CACW is formulated by setting SO_{00} to 1_2 in addition to the left shift input. CACW is therefore obtained from the next consecutively higher address. All other I/O1-sequence operations are the same as those for normal output operations. CACW is modified by ± 1 and is restored by another control memory reference. The output word is obtained from the address specified by the unmodified $CACW_{15-0}$. The termination and monitor interrupt capabilities are the same as for normal output operations. The Continuous Data Mode sequence (CDM) can be initiated upon buffer completion if $TACW_{17} = 1$, but is of no use in reloading the address control words, since this sequence always reloads the normal

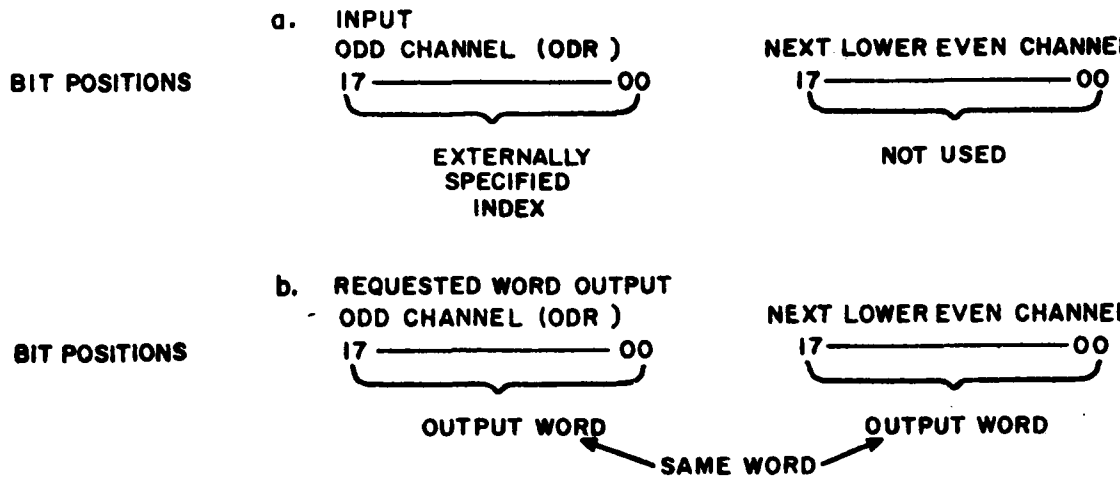


Figure 2-126. ESI Word Formats For ODR Operations

control memory addresses of CACW and TACW. That is, it places new control words into control memory addresses 00040_8 + 2k and 00041_8 + 2k or into 00240_8 + 2k and 00241_8 + 2k, depending upon the channel number. These addresses are not used since the external device specifies the origins of CACW and TACW. However, if the Continuous Data Request (CDM) flip-flop is set, it will prevent deactivating the buffer (the OD Active flip-flop will not be cleared).

2-519. I/02 Sequence Data Flow (Buffer Termination). Refer to figure 2-129 for a block diagram description of the I/02 sequence operations executed upon buffer termination. The I/02 sequence is initiated if $TACW_{15-0}$ equals the unmodified $CACW_{15-0}$. It uses a memory reference to store the ESI in memory for use by the program.

2-520. I/01 and I/02 Sequence Essential Commands for OD Operations with ESI. (Refer to table 2-144 for a list of Scan, I/01, and I/02 sequence events. The events concerning priority selection are as previously discus and are not shown. Events concerning CDM sequence are analyzed in a later section. The OD Acknowledge timing is the same as for normal output operations. Commands listed in table 2-144 have been previously described and a detailed description is not provided.

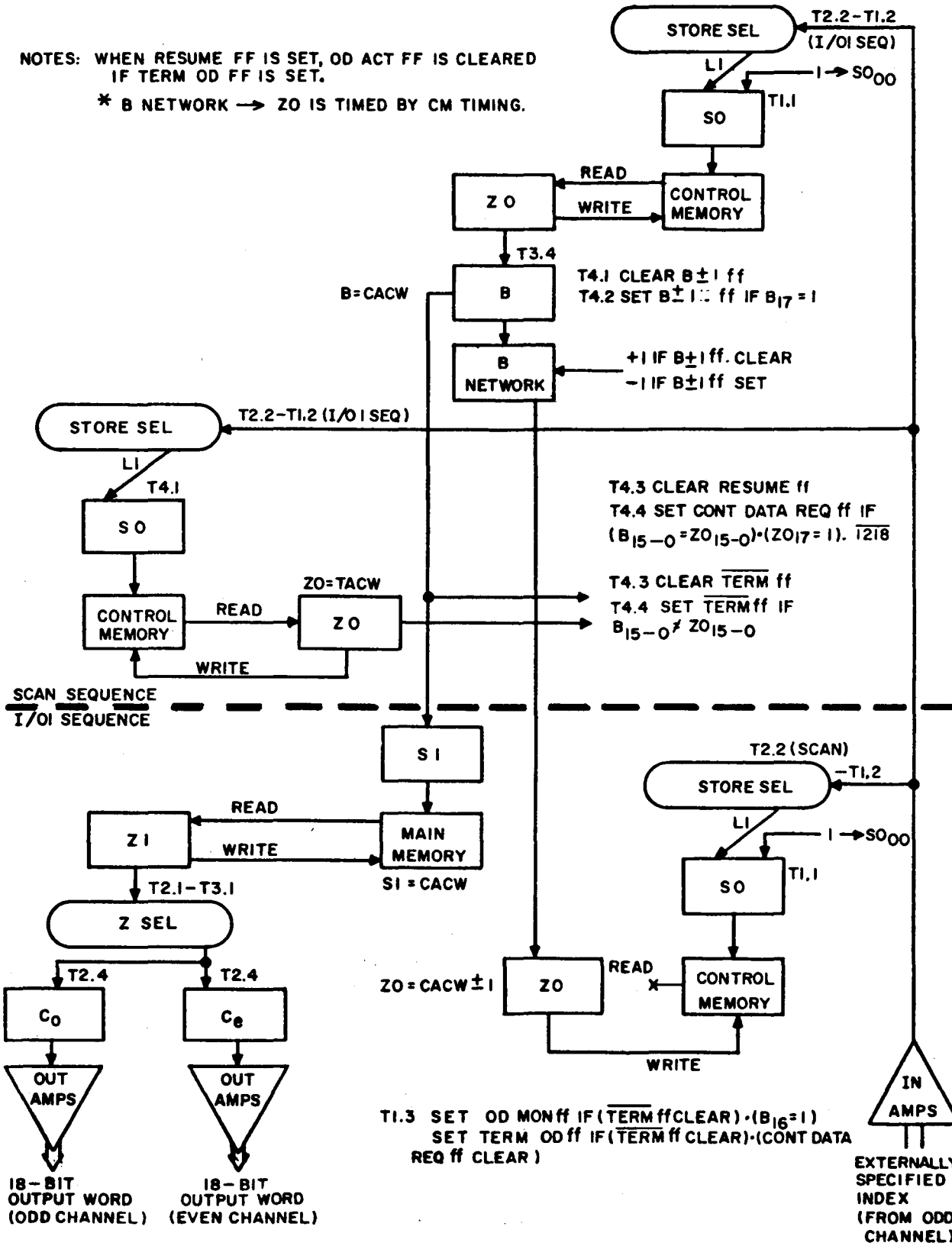


Figure 2-127. Scan and I/O1-Sequence Data Flow for OD Operations with ESI

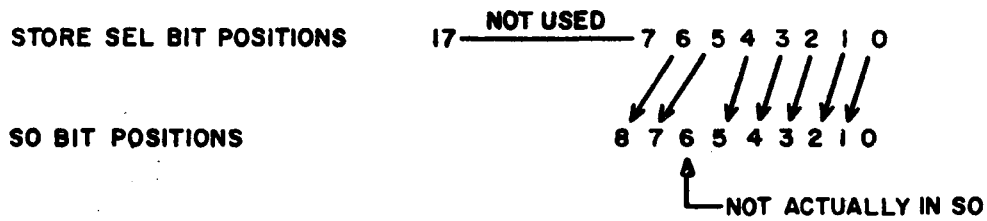
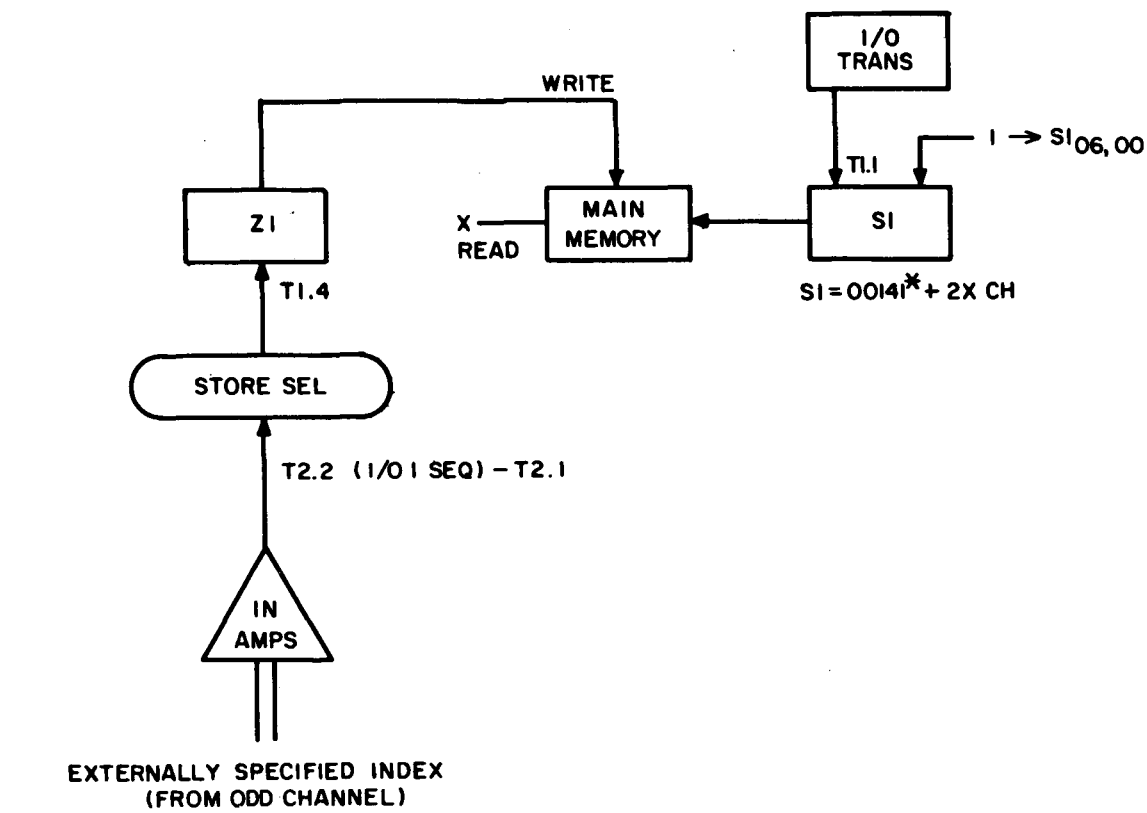


Figure 2-128. Store Select L1 → SO Transfer for ESI



NOTE: *IF CHANNEL IS $10-17_8$, ADD 00200_8 TO ADDRESS IN SI.

Figure 2-129. I/O2-Sequence Data Flow for Terminate OD Operations with ESI

TABLE 2-144. SCAN, I/O1, AND I/O2 SEQUENCE ESSENTIAL COMMANDS FOR OD OPERATIONS WITH ESI

(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Commands
	SCAN SEQUENCE
T2.1	Clear Fct, Chan, and I/0 Trans 1
T2.2	Set $I/01_i$ ff; Pri → Fct, Chan, and I/0 Trans 1 (set ESI ff) odd chan Input Amps → Store Sel
T3.1	Store Sel L1 → S0 (ESI), 1 → $S0_{00}$, Init CM, clear ZO
T3.2	Clear B
T3.4	ZO → B*
T4.1	Store Sel L1 → S0 (ESI), Init CM, clear ZO, clear I/0 Trans 2 clear B±1 ff (+1 → B-network)
T4.2	Set $I/01_f$ ff, set B±1 ff (-1 → B-network) if $B_{17} = 1$ I/0 Trans 1 → I/0 Trans 2, disable $B\text{-network}_{17, 16}$
T4.3	Clear $\overline{\text{Term}}$ ff, clear Resume ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{15-0} \neq Z0_{15-0}$, disable CM→Z0, clear S1 Set Cont Data Req ff if $(B_{15-0}=Z0_{15-0}) \cdot (Z0_{17}= 1)$
	I/01-SEQUENCE
T1.1	Store Sel L1 → S0 (ESI), 1 → $S0_{00}$, Init CM, clear Z0 B → S1, Init Memory
T1.2	Drop odd chan Input Amps → Store Sel, clear all EF/OD Ack Reg ff's
T1.3	Clear Z1 Clear OD Req ff if (inter-cmptr and EF) + ($\overline{\text{inter-comptr}}$ • ODR) Clear EF/OD Mode ff Set Pri Alternator ff. set EF/OD Ack Reg ff Set Term OD ff if (Term ff clear) (Cont Data Req ff clear) Set OD Mon ff if (Term ff clear) ($B_{16} = 1$)
T1.4	Set Ack Delay ff if Term ff Set
T2.1	Clear $I/01_i$ ff, Z1 Z-Sel

TABLE 2-144. SCAN, I/01, AND I/02 SEQUENCE ESSENTIAL COMMANDS FOR OD OPERATIONS WITH ESI (Cont)

Time Notation	Commands
T2.2	Set $I/02_i$ ff if Term ff clear, odd chan Input Amps → Store Sel if $I/02_i$ ff set
T2.3	Clear C_o and C_e
T2.4	Z-Sel → C_o and C_e, drop disable CM → Z0
T3.1	Drop Z1 → Z-Sel
T4.1	Clear $I/01_f$ ff drop disable $B\text{-network}_{17,16}$
T4.2	Set $I/02_f$ ff
T4.4	Clear S1, disable Mem → Z1
	I/02 SEQUENCE (if terminate)
T1.1	I/0 Trans → S1, 1 → $S1_{00}$, 1 → $S1_{06}$, Init Memory
T1.2	Clear all EF/OD Ack Reg ff's
T1.3	Clear Z1, set EF/OD Ack Reg ff, set Ack Delay ff
T1.4	Store Sel → Z1
T2.1	Clear $I/02_i$ ff, drop odd chan Input Amps → Store Sel
T2.4	Drop disable Mem → Z1
T4.1	Clear $I/02_f$ ff

*B-network → Z0 is timed by CM timing.

521. OUTPUT DATA REQUEST OPERATIONS WITH ESA. The Externally Specified Address (ESA) mode is instated for a particular channel by positioning the drawer-mounted CHANNEL FUNCTION switch to the A position. When in effect, it alters the operations which honor an ODR on its channel. Two channels are used together, one on which the ODR is sent, which must be odd-numbered, and the other is the next lower even-numbered channel. Instead of the output word being obtained from the address determined by CACW, the external device directly specifies the address. The ODR signal must appear on an odd channel. There must be an ESA address word accompanying the request on the same channel. This word is the address the requested output word. There are no buffer limits (address control words). The requested word is outputted on both the requesting odd channel, and the next lower even channel. Refer to figure 2-130 for the input output word formats. The external device can terminate the OD buffer only if the 16 least significant bits inputted as the address are all zeros. Upon termination, the output data/external-function monitor-interrupt signal is generated if bit 16 = 1_2 of the odd channel. Also, at termination, the ESI is stored at the address $00141_8 + 2k$ (k ⇒channel number), if the channel is 0-7, or at $00341_8 + 2k$, if the channel is $10\text{-}17_8$.

2-522. Scan, I/O1, and I/O2 Sequence Data Flow For OD Operations with ESA. Refer to figure 2-131 for a block diagram description of the OD operations with ESA. During the last poriton of the Scan sequence, the ESA is inputted and placed in B. The circuitry which is normally used to compare TACW with CACW is still effective in examining ZO_{15-0} and B_{15-0}. If B_{15-0}=0's, the output operations will terminate after this current request has been handled. The I/O1 sequence obtains the requested word using the ESA in B. This word is outputted on the requesting odd channel and also on the next lower even channel. As normal, the output data/external-function monitor-interrupt signal can be generated upon termination of selected by bit-16 of this address word in B. The Continuous Data Mode (CDM) sequence cannot

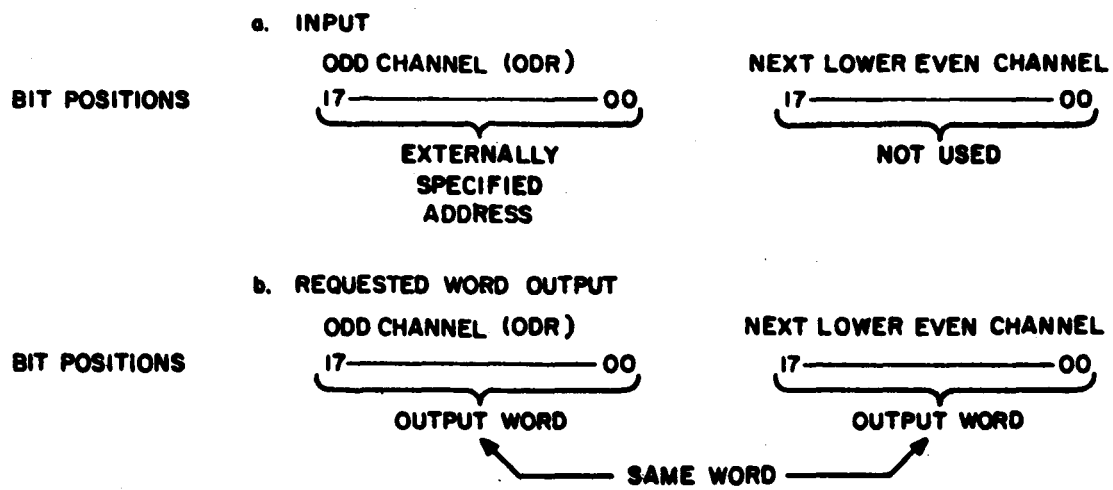


Figure 2-130. ESA Word Formats for ODR Operations

be initiated. That is, the Continuous Data Request (CDR) flip-flop is not set since it requires that $ZO_{17} = 1$. The I/O2 sequence is used only if the output operation is terminated. It performs the storage of the ESA.

2-523. Scan, I/O1, and I/O2 Sequence Essential Commands for OD Operations with ESA. Refer to table 2-145 for a list of Scan, I/O1, and I/O2 sequence events. Events concerning priority selection have been previously described and are not shown. The OD Acknowledge timing is the same as for normal output operations. Commands listed in table 2-145 have been described previously and are not provided with a detailed description.

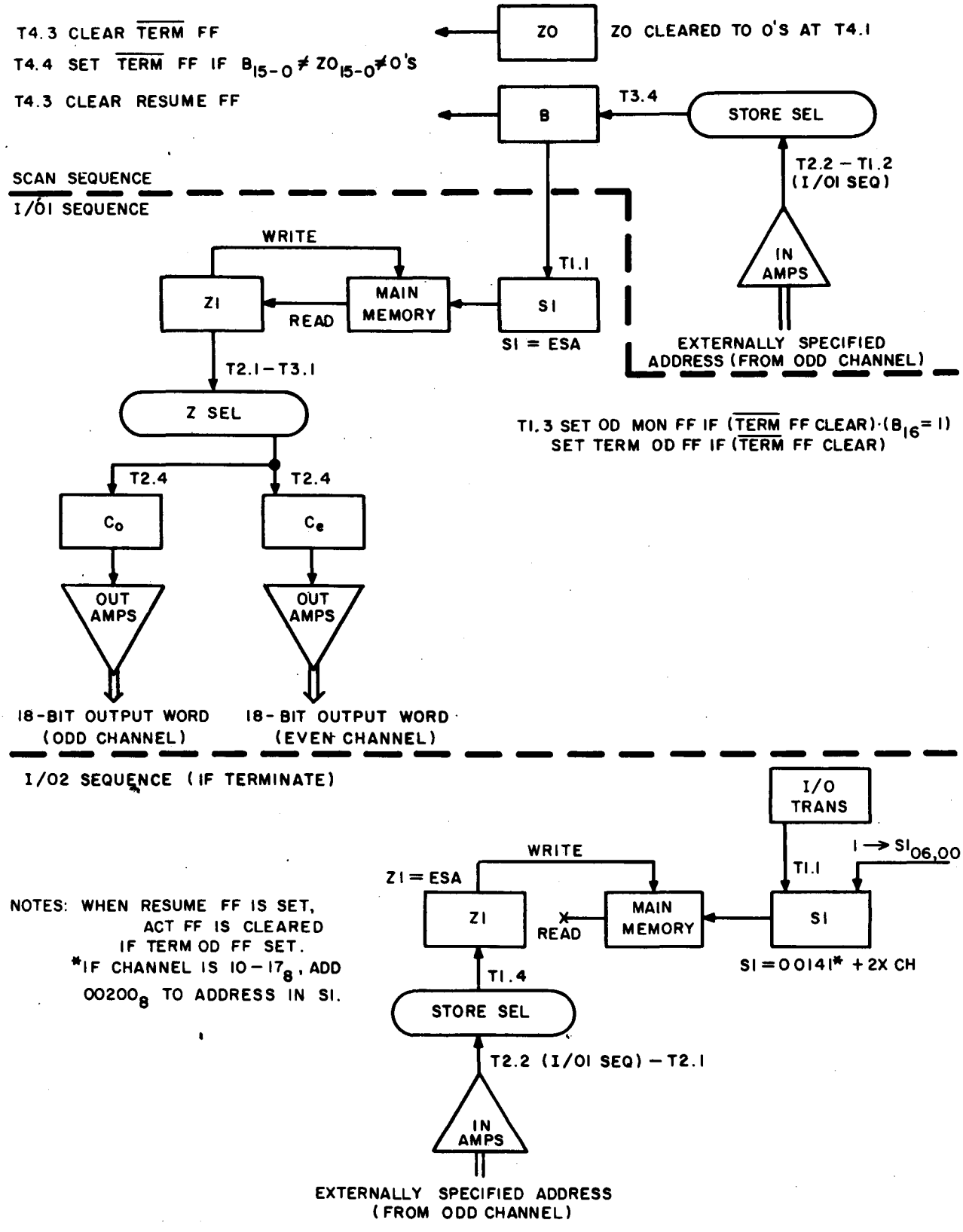


Figure 2-131. Scan, I/O1, and I/O2-Sequence Data Flow for OD Operations with ESA

TABLE 2-145. SCAN, I/O1, AND I/O2 SEQUENCE ESSENTIAL COMMANDS FOR OD OPERATIONS WITH ESA

(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Commands
	SCAN SEQUENCE
T2.1	Clear Fct, Chan, and I/0 Trans 1
T2.2	Set $I/01_i$ ff; Pri → Fct, Chan, and I/0 Trans 1 (set ESA ff) Odd chan Input Amps → Store Sel
T3.2	Clear B
T3.4	Store Sel → B
T4.1	Clear I/0 Trans 2, clear Z0
T4.2	Set $I/01_f$ ff, I/0 Trans 1 → I/0 Trans 2
T4.3	Clear $\overline{\text{Term}}$ ff, clear Resume ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{15-0} \neq Z0_{15-0}$, clear S1
	I/01 SEQUENCE
T1.1	B → S1, Init Memory
T1.2	Drop odd chan Input Amps → Store Sel, clear all EF/OD Ack Reg ff's
T1.3	Clear Z1 Clear OD Req ff if (inter-cmptr and EF) + ($\overline{\text{inter-cmptr}}$ · ODR) Clear EF/OD Mode ff Set Pri Alternator ff, set EF/OD Ack Reg ff Set Term OD ff if Term ff clear Set OD Mon ff if (Term ff clear) · ($B_{16} = 1$)
T1.4	Set Ack Delay ff if Term ff set
T2.1	Clear $I/01_i$ ff, Z1 → Z Sel
T2.2	Set $I/02_i$ ff if Term ff clear, odd chan Input Amps → Store Sel if $I/02_i$ ff set
T2.3	Clear C_o and C_e
T2.4	Z-Sel → C_o and C_e

TABLE 2-145. SCAN, I/01, AND I/02 SEQUENCE ESSENTIAL COMMANDS FOR OD OPERATIONS WITH ESA (Cont)

Time Notation	Commands
T3.1	Drop Z1 → Z-Sel
T4.1	Clear $I/01_f$ ff
The follow occurs if $I/02_i$ ff set (terminate)	
T4.2	Set $I/02_f$ ff
T4.4	Clear S1, disable Mem → Z1
	I/02 SEQUENCE (if terminate)
T1.1	I/0 Trans → S1, 1 → $S1_{00}$, 1 → $S1_{06}$, Init Memory
T1.2	Clear all EF/OD Ack Reg ff's
T1.3	Clear Z1, set EF/OD Ack Reg ff, set Ack Delay ff
T1.4	Store Sel → Z1
T2.1	Clear $I/02_i$ ff, drop odd chan Input Amps → Store Sel
T2.4	Drop disable Mem → Z1
T4.1	Clear $I/02_f$ ff

2-524. EXTERNAL FUNCTION REQUEST OPERATIONS. The External Function Request (EFR) is a signal from an external device which requests that the computer output a function control word. Since the EFR is a request for an output, this operation is very similar to the ODR already described. This section will present only the differences between ODR operations and EFR operations.

2-525. Single-Channel EFR Operation Without ESI or ESA. The EFR from an external device is honored by the computer only if the computer is external-function active. (Refer to figure 2-132.) The External Function Active (EF Act) flip-flop is set during the execution of f = 50:03 or 50:13. If the priority logic selects an EFR signal, CACW is obtained from control memory address 00021_8 + 2k (k ⇒ channel number) if channels 0-7, or from 00221_8 + 2k if channels 10-17. Similarly, TACW is obtained from addresses 00020_8 + 2k or 00220_8 + 2k. $CACW_{17}$ determines a forward or backward buffer as is the case for ODR operations. (Refer to tables 2-131 and 2-132 for review.) Buffer termination is the same as OD buffer termination. If $CACW_{16}$ = when the lower 16-bits of the control words are equal, an EF Monitor interrupt is requested, Thus, the EF Monitor flip-flop for the channel is set. The handling of the monitor is discussed in a later section.

2-526. Dual-Channel EFR Operation Without ESI or ESA. Review dual-channel OD buffer operation. Except for the acknowledge signal generated, EFR operations are similar to ODR operations.

2-527. Scan and I/O1 Sequence Data Flow For Single-Channel Operation. Refer to figure 2-133 for a block-diagram description of single-channel EF operation. The Scan sequence is the main timing cycle during which the $I/O1_i$ flip-flop is set upon detection and selection of an EFR signal. During this sequence, the CACW is obtained from control memory at address 00021_8 + 2k or 00221_8 + 2k, depending upon the channel. The address is formulated in the I/O translator which is set according to priority selection. SO_{00} is set to 1_2 to make the address odd. Another control-memory reference obtains TACW ($SO_{00} = 0_z$). The TACW and CACW are compared from ZO and B. The result of this comparison is recorded by the Terminate flip-flop. If $TACW_{15-0} = CACW_{15-0}$, this EFR will be the last to be honored. In 1218 (Normal) mode, this EFR will not be honored. The B-network is used to modify CACW as determined by the B ± 1 flip-flop. The I/O1 sequence is used to obtain and to output the 18-bit word. The memory reference uses the unmodified $CACW_{15-0}$ from B. A control-memory reference is initiated at time T1.1 to store the modified CACW for use during the next external function operation. The remaining I/O1 sequence operations are used only if the operation is dual-channel.

2-528. Scan, I/O1 and I/O2 Sequence Data Flow For Dual-Channel Operation. Refer to figures 2-133 and 2-134. The Scan and I/O1 sequences are the same as for single-channel operations. The last portion of the I/O1 and I/O2 sequences obtain the second word from the first word addresses ± 1.

2-529. Scan, I/O1, and I/O2 Sequence Essential Commands For Single or Dual-Channel EF Operations. Refer to table 2-146 for a list of essential Scan, I/O1 and I/O2 sequence events. Develop the commands shown by referring to the proper enable pages in the logic diagrams. Commands that have not been described previously are described in table 2-147 in detail.

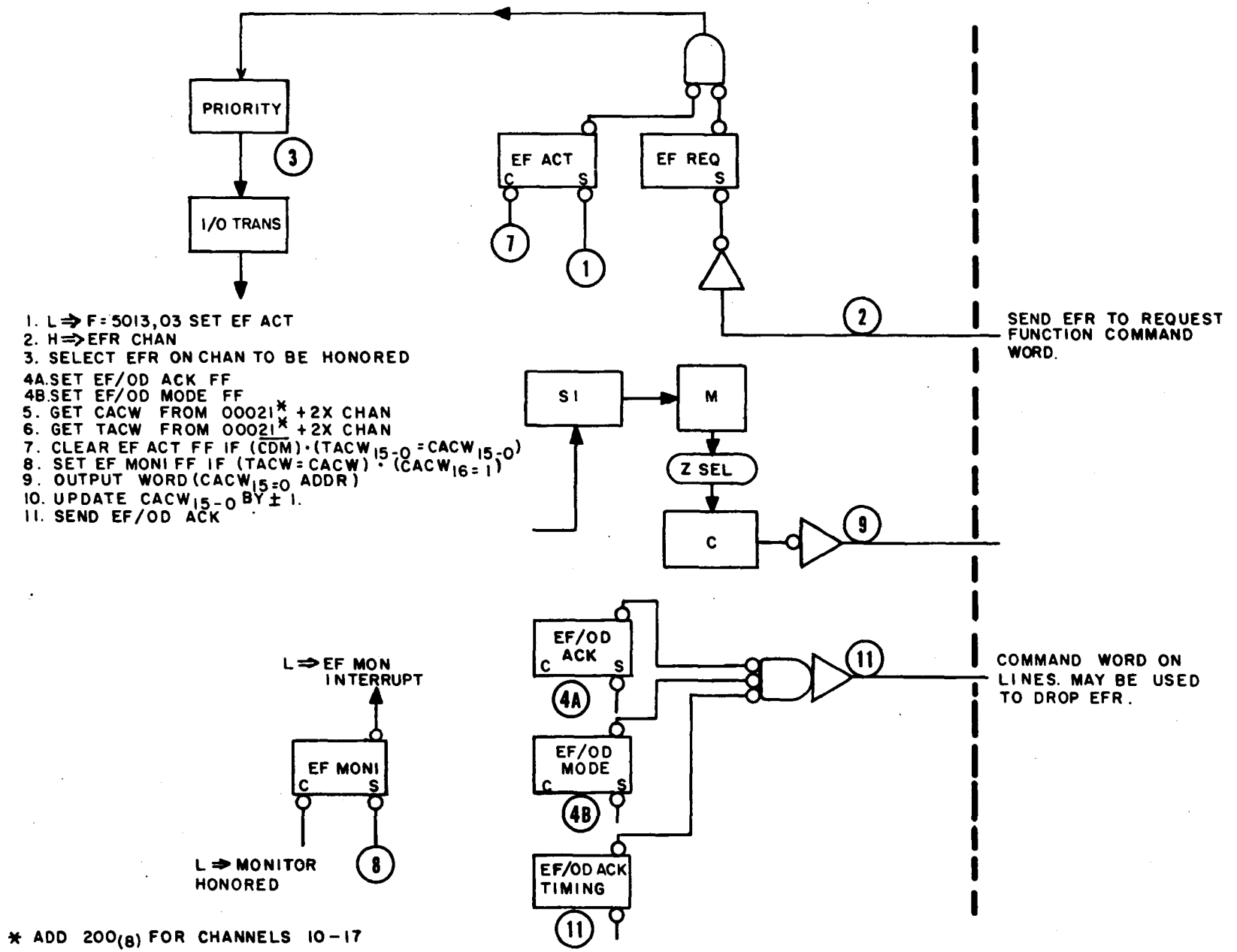


Figure 2-132. External Function Operations, Single Channel Without ESI and ESA

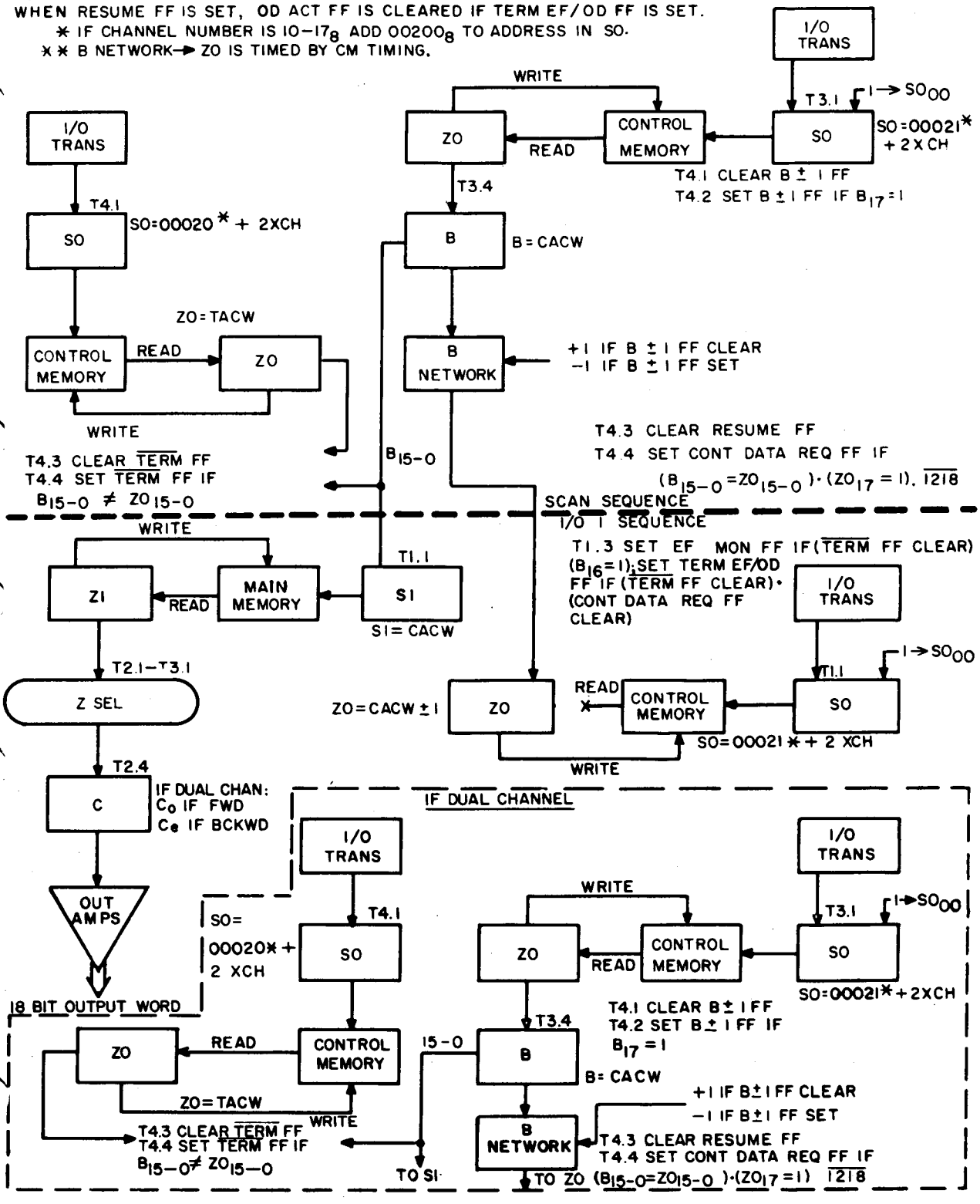
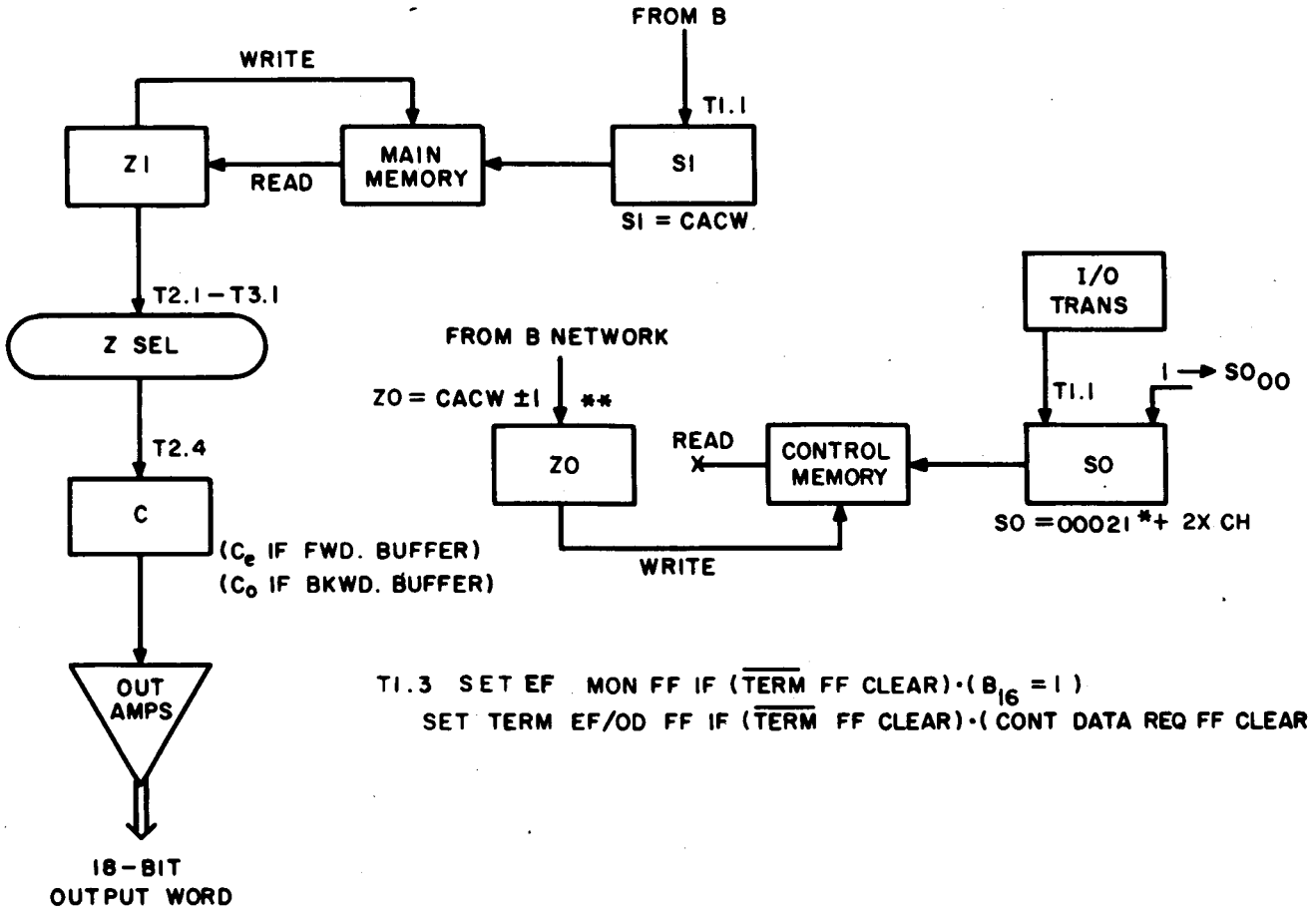


Figure 2-133. Scan and I/O1-Sequence Data Flow for Single or Dual Channel EF Operations



NOTES: WHEN RESUME FF IS SET, EF ACT FF IS CLEARED IF TERM EF/OD FF IS SET.

*IF CHANNEL NUMBER IS $10-17_8$, ADD 00200_8 TO ADDRESS IN SO.

**B NETWORK ⟶ ZO IS TIMED BY CM TIMING.

Figure 2-134. I/O2-Sequence Data Flow for Dual-Channel EF Operations

TABLE 2-146. SCAN, I/01, AND I/02-SEQUENCE ESSENTIAL COMMANDS FOR SINGLE-OR DUAL-CHANNEL EF OPERATIONS

Time Notation	Commands
	SCAN SEQUENCE
T2.1	Clear Fct, Chan, and I/0 Trans 1
T2.2	Set $I/01_i$ ff, Pri → Fct, Chan, and I/0 Trans 1
T3.1	I/0 Trans → S0, 1 → $S0_{00}$, Init CM, clear Z0
T3.2	Clear B
T3.4	Z0 → B*
T4.1	I/0 Trans → S0, Init CM, clear Z0, clear B±1 ff (+1 → B-network), clear I/0 Trans 2
T4.2	Set $I/01_f$ ff, set B±1ff (-1 → B-network) if B_{17}=1 I/O Trans 1 → I/O Trans 2 (set Dual ff if dual chan selected) disable $B\text{-network}_{17,16}$
T4.3	Clear $\overline{\text{Term}}$ ff, clear Resume ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{15-0} \neq Z0_{15-0}$, disable CM → Z0 Clear S1, set Cont Data Req ff if $(B_{15-0}=Z0_{15-0}) \cdot (Z0_{17}=1)$
	I/01 SEQUENCE
T1.1	I/0 Trans → S0, 1 → $S0_{00}$, Init CM, B → S1, Init Memory
T1.2	Clear all EF/OD Ack Reg ff's
T1.3	Clear EF Req, clear Z1, set EF/OD Mode ff Set EF Mon ff if ($\overline{\text{Term}}$ ff clear) • (B_{16}=1), set Pri Alternator ff, Set Term EF/OD ff if ($\overline{\text{Term}}$ ff clear) • (Cont Data Req ff clear) set EF/OD Ack Reg ff
T1.4	Set Ack Delay ff if Dual ff clear
T2.1	Clear $I/01_i$ ff, Z1 → Z Sel
T2.2	Set $I/02_i$ ff if Dual ff set
T2.3	Clear C**
T2.4	Z Sel → C**, drop disable CM → ZO

TABLE 2-146. SCAN, I/01, AND I/02-SEQUENCE ESSENTIAL COMMANDS FOR SINGLE-OR DUAL-CHANNEL EF OPERATIONS (Cont)

Time Notation	Commands
The following occurs if $I/02_i$ ff set (dual channel) excepting, clear $I/01_f$ ff, and drop Z1 → Z-Sel.	
T3.1	I/0 Trans → S0, 1 → $S0_{00}$, Init CM, Clear Z0, drop Z1 → Z-Sel
T3.2	Clear B
T3.4	Z0 → B*
T4.1	Clear $I/01_f$ ff, I/O Trans → SO, Init CM, Clear ZO clear B±1 ff (+1→ B-network), clear I/O Trans 2 drop disable $B\text{-network}_{17,16}$
	I/01 SEQUENCE
T4.2	Set $I/02_f$ ff, set B±1 ff (-1 → B Network) if B_{17} = 1 I/O Trans 1 → I/O Trans 2 (set Dual ff), disable $B\text{-network}_{17,16}$
T4.3	Clear $\overline{\text{Term}}$ ff, clear Resume ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{15-0} \neq Z0_{15-0}$, disable Cm→ Z0 Clear S1, set Cont Data Req ff if $(B_{15-0}=Z0_{15-0}) \cdot (Z0_{17}=1)$
	I/02 SEQUENCE
T1.1	I/0 Trans →S0, 1 → $S0_{00}$, Init CM, clear Z0 B → S1, Init Memory
T1.2	Clear all EF/OD Ack Reg ff's
T1.3	Clear Z1, set EF/OD Ack Reg ff set EF Mon ff if ($\overline{\text{Term}}$ ff clear) • (B_{16}=1) set Term EF/OD ff if ($\overline{\text{Term}}$ ff clear)•(Cont Data Req ff clear)
T1.4	Set Ack Delay ff
T2.1	Clear $I/02_i$ ff, Z1 → Z Sel
T2.3	Clear C**
T2.4	Z-Sel → C**, drop disable CM → Z0

TABLE 2-146. SCAN, I/01, AND I/02-SEQUENCE ESSENTIAL COMMANDS FOR SINGLE-OR DUAL-CHANNEL EF OPERATIONS (Cont)

Time Notation	Commands
T3.1	Drop Z1 → Z-Sel
T4.1	Clear $I/02_f$ ff, drop disable B-network$_{17,16}$

*B-network → Z0 is timed by CM timing.

**If operation is in Dual Channel Mode, the C-register which is cleared and receives word from Z-select is as follows:

	I/01 Seq.	I/02 Seq.
B±1 ff clear (forward buffer)	Co	Ce
B±1 ff set (backward buffer)	Ce	Co

TABLE 2-147. SCAN, I/O1, AND I/O2-SEQUENCE COMMANDS/DESCRIPTIONS FOR SINGLE-OR DUAL-CHANNEL EF OPERATIONS

(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
T1.3	Clear EF Req ff. Gate 00E06 (refer to plate P-53) is enabled by a L ⇒ Out + EF, L ⇒ T21 I/O Seq and the absence of a H ⇒ (ESI Term + Dual) · I/01 Seq. The output is applied as a H ⇒ Start EF OD Ack to 1GE06 (refer to plate P-57) and applied to 5gN40 via 9gN48. With a H ⇒ EF, 5gN40 produces a L ⇒ Clear EF Req to EF Req flip-flops (refer to plates P-61 through P-64). Set EF Mon ff if ($\overline{\text{Term}}$ ff Clear) · (B_{16} = 1). Set Monitor gate 00N42 (refer to plate P-53) is enabled by a L ⇒ T13 I/O Seq, L ⇒ Terminate ($\overline{\text{Term}}$ ff Clear) and a L ⇒ Monitor (B_{16} = 1). The high is inverted by 01N42 and applied as a L ⇒ Set Monitor to 9gN42 (refer to plate P-56), inverted and applied as a H ⇒ Set Monitors to 5gN42 (refer to plate P-57). The other input is a H ⇒ EF + I/O Clear. The output of 5gN42 is applied as a L ⇒ Set EF Monitor to EF Mon flip-flops (refer to plate P-38). Depending upon the channel number, one of the flip-flops will set.

2-530. EF Acknowledge Timing. The EF Acknowledge signal is sent after the output operation places the external function word on the data lines. For dual-channel operation, the acknowledge is sent after both words have been outputted. The EF/OD Control flip-flop is set to gate the EF Acknowledge generated by the EF/OD acknowledge timing circuits. The external-function acknowledge can appear as an EIR to the receiving computer, enabling control of a receiving computer by the sending computer.

2-531. Normal Mode EF Acknowledge Timing. The EF/OD Acknowledge generation flip-flop determines the acknowledge duration by its time as for OD operations. The EF/OD Control flip-flop being set determines that an EF Acknowledge is sent. The Resume flip-flop clears the EF Act flip-flop via 72 Lg1 if the Terminate EF/OD flip-flop is set. (Refer to figure 2-135.)

2-532. Intercomputer Mode EF Acknowledge Timing. The IA from the interrupted computer appears as an ODR and sets the Resume flip-flop. (Review intercomputer OD operations.)

2-533. EF Operations with ESI. Except for the type of acknowledge sent and the index storage address (refer to figure 2-136) EF operations with ESI are the same as OD operations. At buffer termination, the ESI is stored in main memory at address 00121_8-2k (k $\Rightarrow$ channel number), if channel 0-7, or at 00321_8 - 2k, if channel 10-17.

2-534. I/O2 Sequence Data Flow for Terminate EF Operations with ESI. Refer to figure 2-137 for a block diagram description of terminate EF operations with ESI, and to table 2-148 for a list of I/O2 sequence essential commands. Commands listed in table 2-148 have been described previously and are not provided with a detailed description.

2-535. Scan, I/O1 and I/O2 Sequence Data Flow For EF Operations with ESA. Except for the type of acknowledgement sent and the storage address in control memory, EF operations with ESA are similar to OD operations. At termination, the ESA is stored in main memory at address 00121_8 + 2k, or 00321_8 + 2k. Refer to figure 2-138 for a block diagram description of the execution of EF operations with ESA.

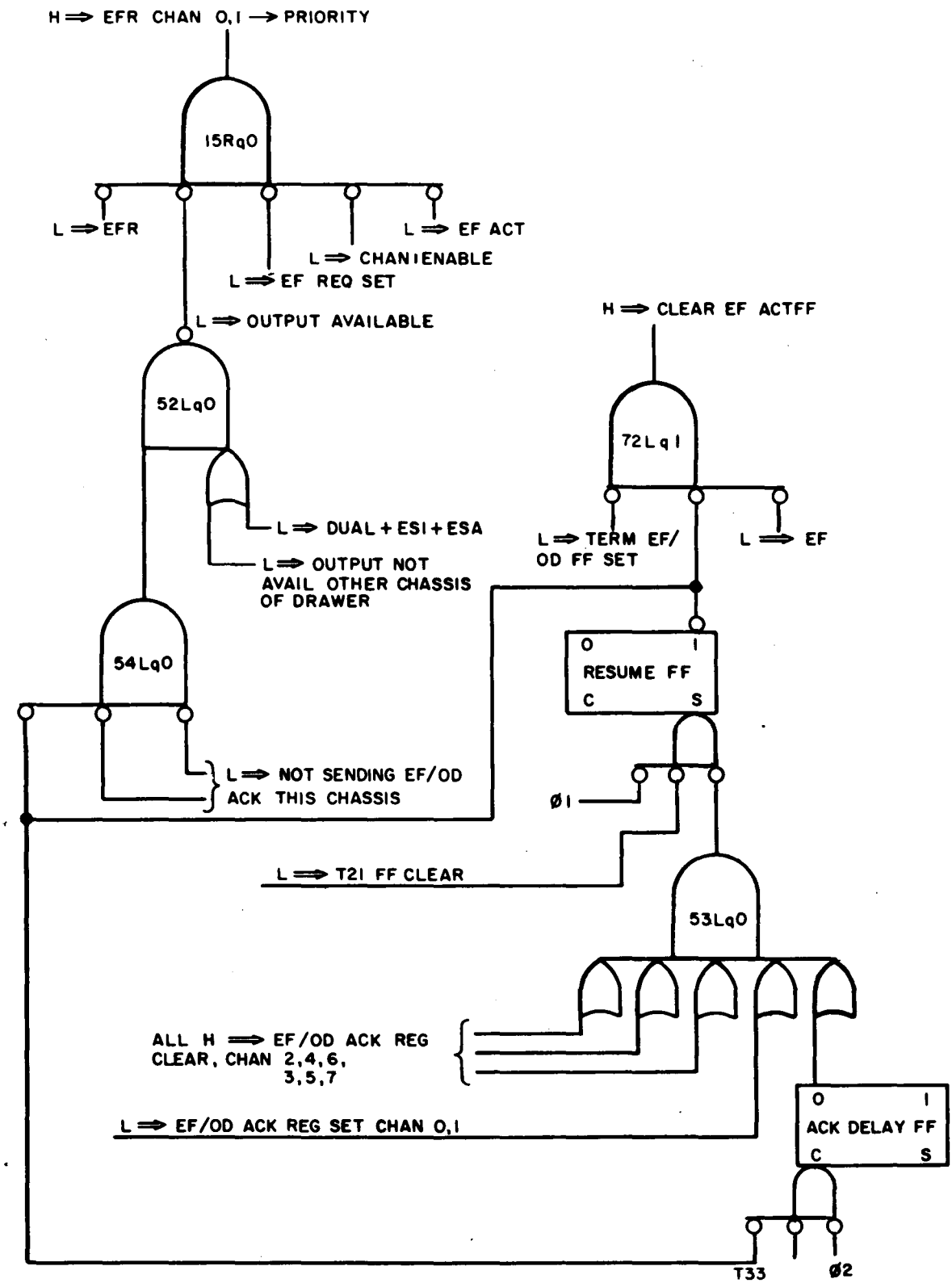


Figure 2-135. EF-Acknowledge Simplified Logic

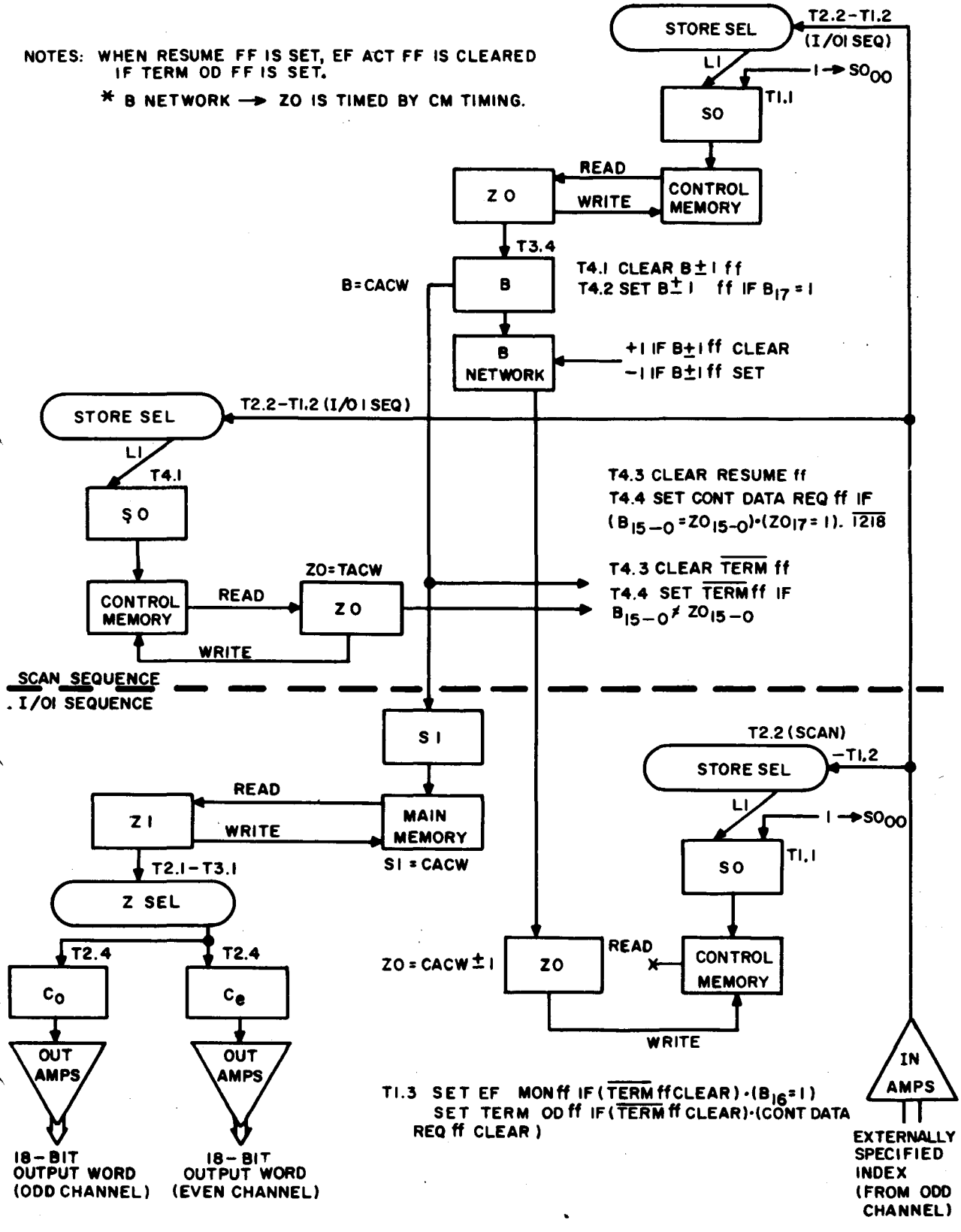
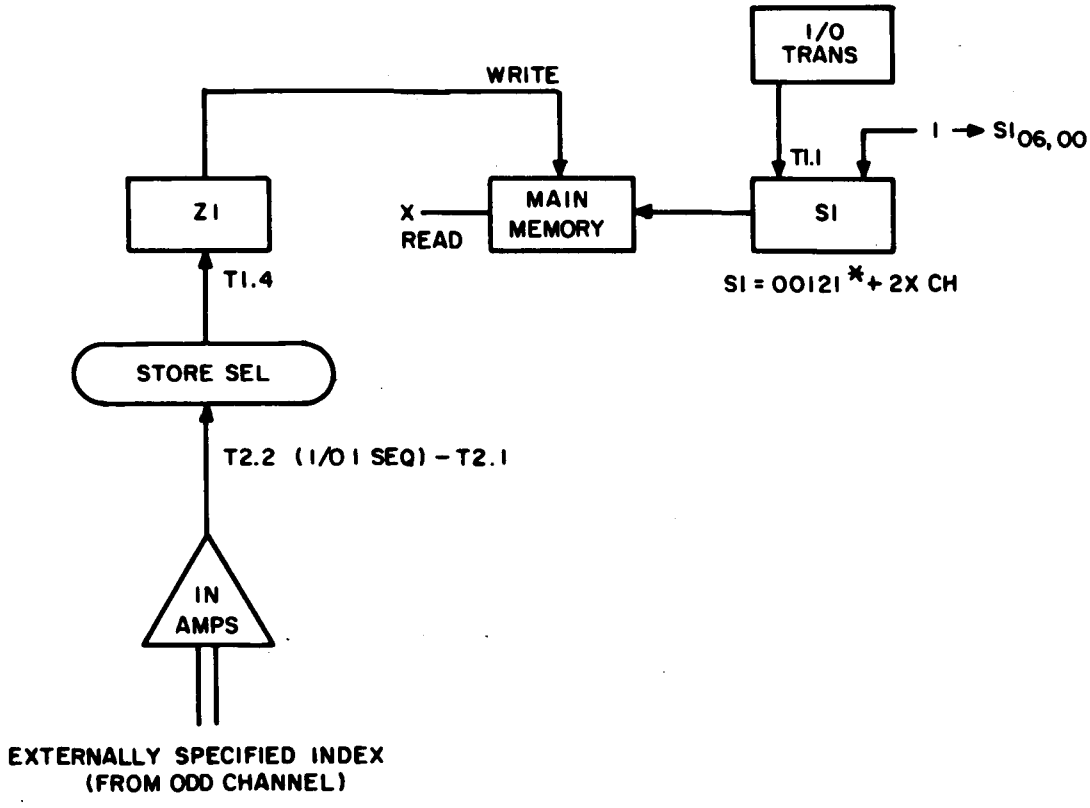


Figure 2-136. Scan and I/O1-Sequence Data Flow for EF Operations with ESI



NOTE: *IF CHANNEL IS $10\text{-}17_8$, ADD 00200_8 TO ADDRESS IN SI.

Figure 2-137. I/O2-Sequence Data Flow For Terminate EF Operations with ESI

TABLE 2-148. SCAN, I/O1, AND I/O2-SEQUENCE ESSENTIAL COMMANDS FOR EF OPERATIONS WITH ESI
(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Commands
	SCAN SEQUENCE
T2.1	Clear Fct, Chan, and I/0 Trans 1
T2.2	Set $I/01_i$ ff; Pri → Fct, Chan, and I/0 Trans 1 (set ESI ff) disable B-network$_{17, 16}$, odd chan Input Amps → Store Sel
T3.1	Store Sel L1 → S0 (ESI), 1 → $S0_{00}$, Init CM, clear Z0
T3.2	Clear B
T3.4	Z0 → B*
T4.1	Store Sel L1 → S0 (ESI), Init CM, clear Z0, clear I/0 Trans 2, clear B±1 ff (+1 → B-network)
T4.2	Set $I/01_i$ ff, set B±1 ff (-1 → B-network) if B_{17}=1 I/0 Trans 1 → I/0 Trans 2
T4.3	Clear $\overline{\text{Term}}$ ff, clear Resume ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{15-0} \neq Z0_{15-0}$, disable Cm → Z0, clear S1 Set Cont Data Req ff if $(B_{15-0}=Z0_{15-0}) \cdot (Z0_{17}=1)$
	I/01-SEQUENCE
T1.1	Store Sel L1 → S0 (ESI), 1 → $S0_{00}$, Init CM, clear Z0 B → S1, Init Memory
T1.2	Drop odd chan Input Amps → Store Sel, clear all EF/OD Ack Reg ff's
T1.3	Clear Z1, clear EF Req ff set EF/OD Mode ff set Pri Alternator ff, set EF/OD Ack Reg ff set Term EF/OD ff if ($\overline{\text{Term}}$ ff clear) • (Cont Data Req ff clear) set EF Mon ff if ($\overline{\text{Term}}$ ff clear) • (B_{16}=1)
T1.4	set Ack Delay ff if $\overline{\text{Term}}$ ff set
T2.1	Clear $I/01_i$ ff, Z1 → Z-Sel,
T2.2	Set $I/02_i$ ff if $\overline{\text{Term}}$ ff clear, odd chan Input Amps → Store Sel if $I/02_i$ ff set

TABLE 2-148. SCAN, I/01, AND I/02-SEQUENCE ESSENTIAL COMMANDS FOR EF OPERATIONS WITH ESI (Cont)

Time Notation	Commands
T2.3	Clear C_o and C_e
T2.4	Z-Sel → C_o and C_e, drop disable CM → Z0
T3.1	Drop Z1 → Z-Sel
T4.1	Clear $I/01_f$ ff drop disable $B\text{-network}_{17,16}$
The following occurs if $I/02_i$ ff set (terminate)	
T4.2	Set $I/02_f$ ff
T4.4	Clear S1, disable Mem → Z1
	SCAN SEQUENCE
	I/02-SEQUENCE (if terminate)
T1.1	I/0 Trans → S1, 1 → $S1_{00}$, 1 → $S1_{06}$, Init Memory
T1.2	Clear all EF/OD Ack Reg ff's
T1.3	Clear Z1, set EF/OD Ack Reg ff, set Ack Delay ff
T1.4	Store Sel → Z1
T2.1	Clear $I/02_i$ff, drop odd chan Input Amps → Store Sel
T2.4	Drop disable Mem → Z1
T4.1	Clear $I/02_f$ff

*B-network → Z0 is timed by CM timing.

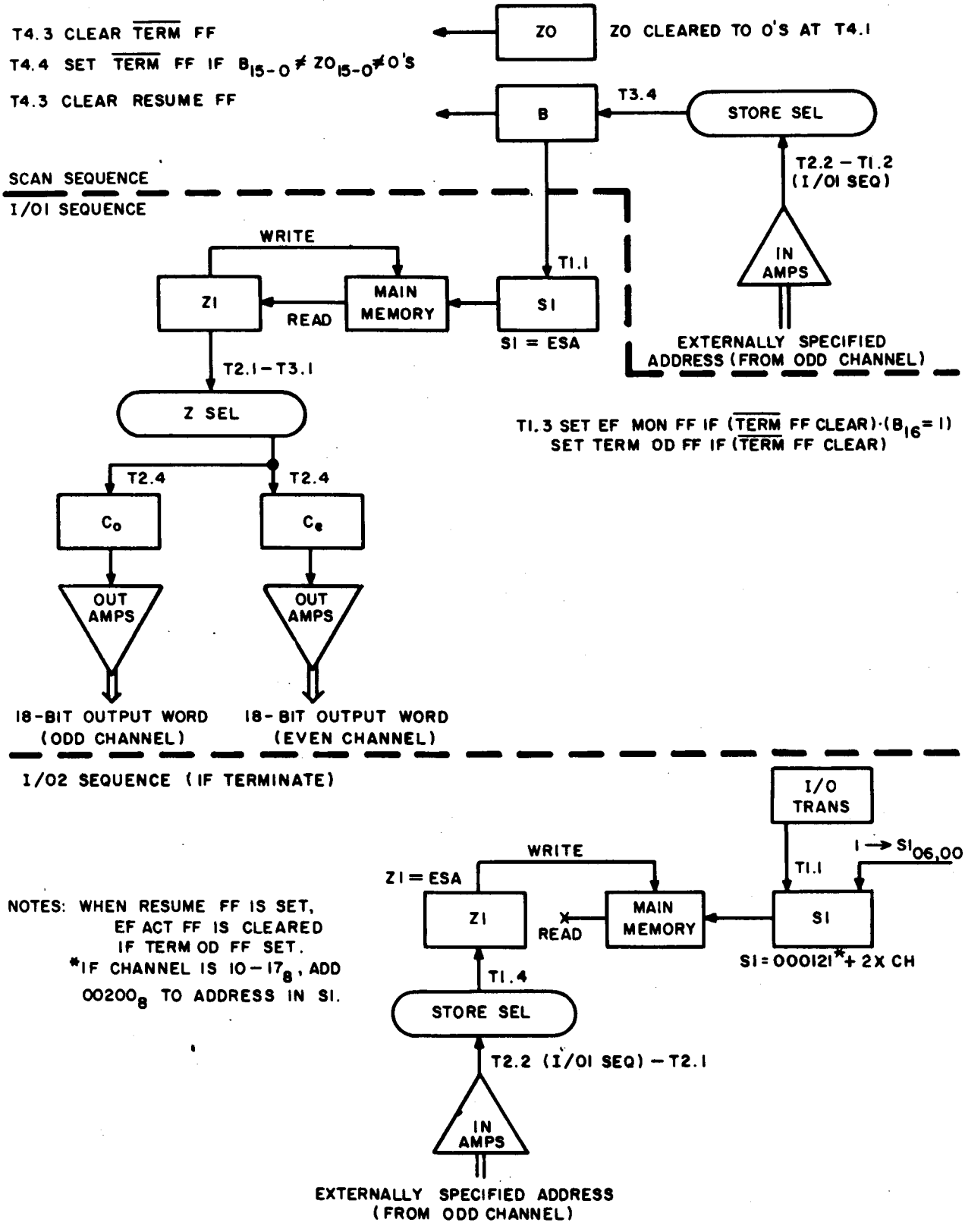


Figure 2-138. Scan, I/O1, and I/O2-Sequence Data Flow for EF (OD) Operations with ESA

2-536. CONTINUOUS DATA MODE OPERATIONS. The Continuous Data Mode (CDM) sequence is initiated upon any buffer termination if requested by the $TACW_{17} = 1$. This sequence automatically reloads the two control memory addresses involved with a new TACW and CACW. The particular channel remains active so that operations can continue with a new buffer. CDM is program selectable. If $TACW_{17} = 1$, CDM operation is initiated upon buffer termination. The purpose of CDM is to automatically, without a required program operation, provide a new buffer when one has terminated without deactivating the channel. That is, new predetermined TACW and CACW replace those which have terminated. The CDM operation effectively provides a continuous ID or OD buffer as seen by the communicating external device. The new reload TACW can request another CDM operation upon its termination by $TACW_{17} = 1$. Even though the channel is not made inactive, the monitor interrupt can occur as normal upon each buffer completion, if requested by $CACW_{16} = 1$. This interrupt notifies the program that a buffer has been completed, and that the new reload TACW and CACW are currently being used. If the new TACW requests another CDM operation, the program probably will set up the next reload TACW and CACW. The size of the current buffer determines the amount of time that the program has available in which to set up the next reload address control words. Also, each new reload TACW and CACW can specify the buffer direction (forward or backward) with $CACW_{17}$. On a particular channel, CDM operation should be used with only one type of buffer at a time. This restriction exists because there is only one pair of control memory addresses to hold the reload TACW and CACW for each channel. These reload words are used for the ID or OD operations on that channel, depending upon which of these operations requests the CDM reload function.

2-537. The reload TACW is held in control memory at the address $00020_8 + 2k$ k $\Rightarrow$ channel number if the channel is 0-7, or $00220_8 + 2k$ if the channel is $10\text{-}17_8$. The reload CACW is held in control memory at the address $00021_8 + 2k$ if the channel is 0-7, or $00221_8 + 2k$ if the channel is $10\text{-}17_8$. During the CDM sequence, these reload address words are placed in the normal control-memory address reserved for the ID or OD operation on the particular channel which terminated. Since CDM and EF buffer control words share the same control memory addresses, it is unneccessary to request CDM for EF because there is no way to reload TACW and CACW before they would be needed.

2-538. Continuous Data Mode Sequence Data Flow. Refer to figure 2-139 for a block diagram description of the CDM sequence operations. The CDM sequence is initiated in parallel with the last portion of either the I/01 or I/02 sequence when control memory is available for use. The first control memory reference at time T3.1 obtains the reload CACW which is placed in Z0. A second memory reference stores this word from Z0. The storage address is the address which is reserved for the CACW as used by the ID or OD operation. The address used to obtain the reload CACW is dependent upon the channel number which initiated the CDM sequence. The channel number and type of I/O operation information is still contained in the I/Otranslator. Thus, the I/Otranslator is used to formulate the proper control memory address in S0. $S0_{00}$ is set to 1_2 so that reload CACW origin and destination addresses are odd. The remainder of the CDM sequence involves two more control memory references

which obtain and store the reload TACW. The major difference between this operation and the operation for handling the reload CACW, is the addresses placed in S0. $S0_{00}$ is not set, thus the reload TACW origin and destination addresses are even and in each case, one address lower than that for the reload CACW. The address placed in S0 at time T2.1 is transferred by way of S1 because there is no direct transfer into S0 from the I/O translator at this time.

2-539. CDM Sequence Essential Command. Refer to table 2-149 for a list of essential CDM sequence events. Commands that have not been previously described are described in table 2-150 in detail.

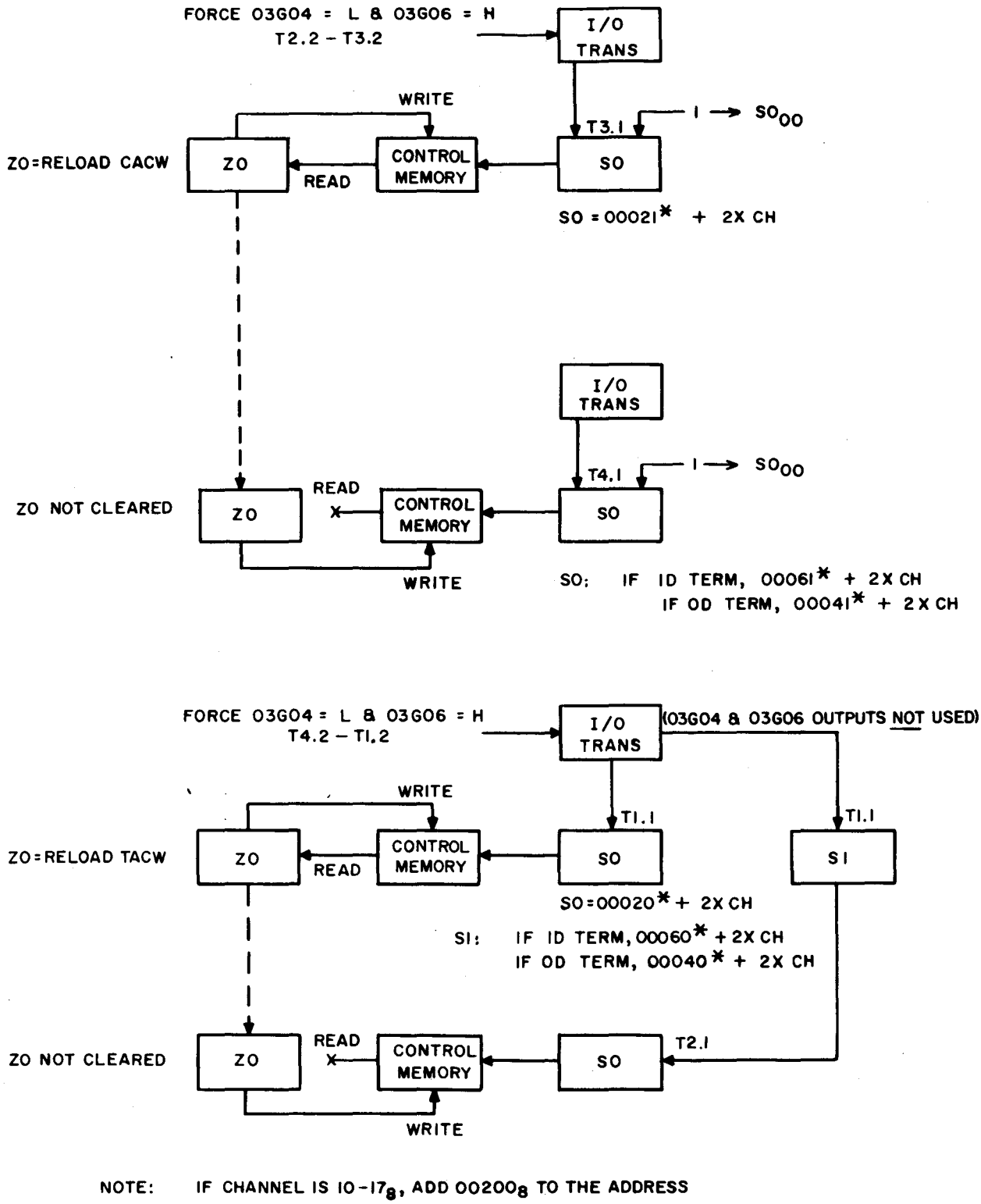


Figure 2-139. Continuous Data Mode Sequence Data Flow

TABLE 2-149. CONTINUOUS DATA MODE SEQUENCE ESSENTIAL COMMANDS

Time Notation	Commands
T4.2	Set $I/0_f$ ff (1 or 2)
T4.4	Set Cont Data Req ff if ($B_{15-0} = Z0_{15-0}$) · ($Z0_{17} = 1$)
(wait until can set Cont Data Seq ff)	
T2.2	Set Cont Data Seq ff if ($I/01_f$ ff set) + (Dual ff set + ESI+ $\overline{\text{Term}}$ ff clear) Force 03G04 = L and 03G06 = H*
Following in parallel with I/01-sequence if not dual-channel and not ESI terminate or I/02-sequence.	
T3.1	Set 2XG28 ff, I/0 Trans → S0, 1 → $S0_{00}$, Init CM, clear Z0 Clear Cont Data Reg ff
T3.2	Drop force 03G04 and 03G06*
T3.3	Disable CM → Z0
T4.1	I/0 Trans → S0, 1 → $S0_{00}$, Init CM
T4.2	Force 03G04 = L and 03G06 = H*
T4.3	Drop disable CM → Z0
T4.4	Clear S1
T1.1	I/0 Trans → S0, Init CM, clear Z0, I/0 Trans → S1
T1.2	Drop force 03G04 and 03G06*
T1.3	Disable CM → Z0
T2.1	Clear Cont Data Seq ff, S1 → S0 Init CM
T2.3	Clear 2XG28 ff, drop disable CM → Z0

*03G04 and 03G06 (function translator) are gates involved in the I/0 Trans → S0 transfer. These gates do not affect the I/0 Trans → S1 transfer.

TABLE 2-150. COMMANDS/DESCRIPTIONS FOR CDM SEQUENCE
(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
T4.4	Set Cont Data Req ff if $(B_{15-0} = Z0_{15-0}) \cdot (Z0_{17} = 1)$. When gate 00E28 (refer to plate P-34) is enabled by a L ⇒ I/O Seq and a low from 01T44, it outputs a high to 03G28. The other inputs into 03G28 are a high from the clear side of 0XZ17 and the absence of a L ⇒ B ≠ Z0. The low outputs sets Cont Data Request flip-flop 0XG28 at ø4.
T2.2	Set Cont Data Seq ff if: (Cont Data Req ff set). [(I/O1 ff $\overline{\text{set}}$) + (Dual ff $\overline{\text{set}}$ + H⇒OD+ EF·TERM·1218 MODE) + (ESA/ESI ff $\overline{\text{set}}$ + Term ff set)]. Gate 30G09 (ref plate P-52) will produce a high output if both 42G09 and 42G10 are disabled. Gate 42G09 is disabled by the DUAL ff 4XG09 being cleared or the H⇒OD+ EF·TERM·1218 Mode. Gate 42G10 is disabled by the ESA/ESI ff 4XG10 being cleared or the absence of L⇒Terminate. The lows out of both 42G09 and 42G10 are inverted by 30G09 and applied as a high to 31G09, which is disabled and outputs a low to 10E28 (ref plate P-34) which will produce a high output because of a low from Cont Data Req ff being set and a low from 03T22. This high is inverted by 11E28 and, at ø2, Cont Data Seq ff 1XG28 is set. Gate 31G09 can also be disabled by the absence of L⇒I/O1 seq from I/O1 seq from I/O1 ff $\overline{\text{set}}$ producing the same results. Force 03G04 = L and 03G06 = H. The low from the set Cont Data Sequence flip-flop is applied to 12G28 (refer to plate P-34). With lows from 31T31 (due to a high from 00T31 [refer to plate P-24] and 00T41), a high output is inverted by 13G28 to disable 03G06 (refer to plate P-51) which outputs a high. The high from 12G28 is applied to 03G04 (refer to plate P-51) to output a low.
T3.1	I/O Trans → So, I $S0_{00}$, Init CM. This command is executed the same as described previously except 10N10 and 50N10 (refer to plate P-24) are enabled by a L ⇒ Cont Data Mode. Clear Cont Data Reg ff. The low from the set side of 1XG28 (refer to plate P-34) will clear Cont Data Reg ff 0XG28 at ø1. Clear Z0. Gate 31T42 (refer to plate P-25) outputs a high because of both inputs being low from 00T42 and 00T22, and at ø1, the low from 00N11 causes 09N11 to produce a L ⇒ Clear Z0. Set 2XG28. Flip-flop 2XG28 (refer to plate P-34) is set at ø1, by the low from Cont Data Sequence flip-flop.
T3.2	Drop Force 03G04 and 03G06. Gate 12G28 (refer to plate P-34) is disabled by 31T31 going high and the force into 03G04 and 03G06 is dropped.

TABLE 2-150. COMMANDS/DESCRIPTIONS FOR CDM SEQUENCE (Cont)

Time Notation	Command/Description
T3.3	Disable CM → Z0. The input from 00T42 into 31T42 (refer to plate P-25) is now high and the low output into 00N11 with a L ⇒ Cont Data Seq causes 00N11 to output a high to 11N11. The high output disables 17N11, 18N11 and 19N11. CM → Z0 will again be disabled at time T1.3 by the high from 00T22 into 31T42.
T4.2	Force 03G04 = L and 03G06 = H. This command is executed the same as described at T2.2, except 31T31, due to a high from 04T11, (refer to plate P-24) now outputs a low to 12G28.
T4.3	Drop disable CM → Z0. Gate 31T42 again outputs a high due to both inputs being high, dropping the disable CM → Z0.
T4.4	Clear S1. Gate 30N12 (refer to plate P-22) is enabled by a low from 03T11 and a L ⇒ Cont Data Seq and the high is inverted by 08N12 and at ∅4; 09N12 produces a L ⇒ Clear S1.
T1.1	I/O Trans → S1. Gate 30N12 (refer to plate P-22) is enabled by a L ⇒ Cont Data Seq and a low from 03T11. The high output is inverted by 38N12 and at ∅1, 39N12 produces a L ⇒ I/O Translator → S1.
T1.2	Drop Force 03G04 and 03G06. Gate 12G28 (refer to plate P-34) is again disabled by 31T31 going high.
T1.3	Disable CM → Z0. See T3.3.
T2.1	Clear Cont Data Seq ff. Cont Data Sequence flip-flop (refer to plate P-34) is cleared by the low from 03T22 at ∅1. S1 → S0, Init CM. Gate 30N10 (refer to plate P-24) is qualified by a L ⇒ I, R1, W or I/O, L ⇒ S1 = CONTROL or BOOTSTRAP MEMORY ADDRESS and low from 03T21. The high output is inverted by 38N10 and at ∅1, 39N10 will produce a L ⇒ S1 → S0 which will also Init CM.
T2.3	Clear 2XG28. The low from Cont Data Seq flip-flop clears 2XG28 at ∅3. Drop Disable CM Z0. See T4.3.

2-540. EXECUTION OF OUTOV AND EXFOV INSTRUCTIONS (f = 50:26, 50:27). OUTOV and EXFOV force out either an OD or EF word by simulating a request signal. Whether the word is OD or EF is indicated by the type of acknowledge signal sent with the word. The particular buffer does not need to be active. The output operation is accomplished by simulating an ODR or EFR. The origin of the output word is determined by the CACW. Also, $CACW_{15-0}$ is modified by ± 1 as specified by the $CACW_{17}$. Normal EF or OD monitor-interrupt operations and CDM operations can occur. As for normal ODR/EFR operations, Channel Mode can be selected by the CHANNEL FUNCTION switch if the channel is odd. The channel is specified by k of the instruction word. Single-or Dual-Channel Mode can be used. However, if the CHANNEL FUNCTION switch is in the ESI or ESA position, Single Mode is used. During the I-sequence, which obtains the instruction from memory, channel number and function information are placed in the I/O translator, as would normally be done by an ODR or EFR. If output is not available on the particular chassis, the sequences are disabled by the Hold flip-flops. The hold condition remains until output is available, after which the word is outputted with the use of the I/O1 and I/O2 sequences. I/O2 sequences issued for dual channel only. If a Resume Fault interrupt is present, the no-resume condition on the particular chassis makes output unavailable. In this case, the Hold flip-flops are cleared to allow the sequences to continue, and the Resume Fault interrupt will be honored. The word is not outputted; however, P is decremented to the address of the same f = 50:26, 50:27 instruction. Thus, when the program resumes after the interrupt is honored, the same instruction will again attempt to force out the same word.

2-541. I-Sequence Data Flow. Refer to figure 2-140 for a block diagram description of the I-sequence. Most of the I-sequence operations are as previously described. KO receives the channel number. This value together with the EF or OD function information, is placed in I/O translator 1 at time T3.2. As discussed in a preceding section, the instruction may be obtained from bootstrap or control memory.

2-542. First I-Sequence Essential Commands. Refer to table 2-151 for a list of first I-sequence events for f = 50:26 and 50:27. Commands not previously described are described in table 2-152 in detail.

2-543. Remaining Sequences With Output Available and No Resume-Fault. During this condition, the word is outputted with the use of I/O sequences. Prior to the I/O-sequence, instruction sequences are held disabled for a main timing cycle. This allows time for setting of the $I/O1_i$ flip-flop and performance of I/O1 operations that would normally occur during the Scan sequence.

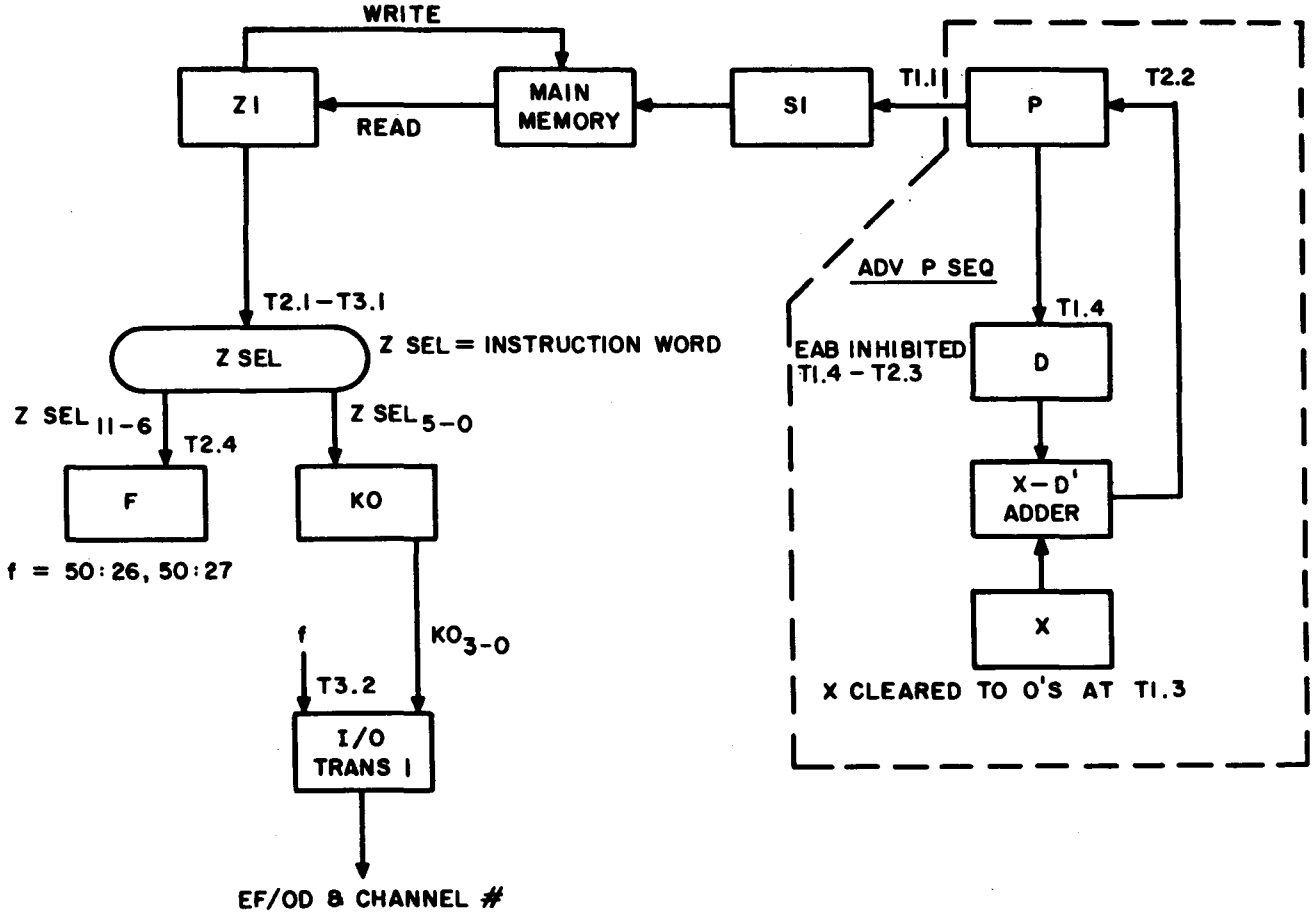
2-544. Sequences After First I-Sequence Essential Commands With Output Available and No Resume-Fault. Refer to table 2-153 for a list of essential events for f = 50:26 and 50:27. Commands that have not been described previously are described in table 2-154 in detail.

2-545. Remaining Sequences With Output Not Available and No Resume-Fault. During this condition, the instruction sequences must be held disabled until output becomes available. When output is available, the Hold flip-flops are cleared and normal I/O sequence operations occur to output the word.

2-546. Sequence After First I-Sequence Essential Commands With Output Not Available and No Resume-Fault. See Table 2-155 for list of essential events. Commands for these sequences have been previously described.

2-547. Remaining Sequences with Resume-Fault. During this condition, the sequence Hold function terminates and the next I-sequence is allowed to run. If no higher priority I/O request is present, the interrupt sequence will also run next to honor the interrupt. The next I-Sequence which runs will be under control of the interrupt sequence to cause the interrupt program jump. P is also decremented to the address of this instruction to allow a program return to this same instruction. Thus, a later attempt can be made to force out the same word. Decrement operations are discussed later in this section.

2-548. Sequences After First I-Sequence Essential Commands With Resume-Fault. Refer to table 2-156 for a list of sequences after first I-sequence events. Commands not previously described are described in table 2-157 in detail.



T3.4 SET HOLD 1 FF
T4.2 SET INT$_1$ FF IF RESUME FAULT INTERRUPT TO BE HONORED
T4.3 SET HOLD 2 FF

Figure 2-140. I-Sequence Data Flow for f = 50:26, 50:27

TABLE 2-151. I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:26, 50:27

Time Notation	Commands
T4.4	Clear S1
T1.1	P → S1, Init Memory, *set Incr P ff
T1.3	*Clear D, *clear X, clear Z1, *set OXL11 ff, clear f
T1.4	$*P_L \rightarrow D_L$, $*P_U \rightarrow D_U$, *set Inhib EAB ff, clear K0
T2.1	Z1 → Z Sel, *clear Incr P ff
T2.2	*Clear P, *Adder → P
T2.3	*Clear OXL11 ff, *clear Inhib EAB ff
T2.4	$Z\text{-}Sel_{11-6} \rightarrow F$, set OXF06 ff, $Z\text{-}Sel_{5-0} \rightarrow K0$
T3.1	Clear I/0 Trans 1, drop Z1 → Z-Sel
T3.2	f · K0 → I/0 Trans 1
T3.4	Set Hold 1 ff
T4.1	Clear I_i ff
T4.2	Set I_i ff
T4.3	Set Hold 2 ff

NOTE * These events are concerned with or are controlled by the Advance-P subsequence.

TABLE 2-152. COMMANDS/DESCRIPTIONS FOR FIRST I-SEQUENCE (f = 50:26 AND 50:27)

(For Detailed Description of all Commands Previously Described Refer to Table 2-19)

Time Notation	Command/Description
T3.1	Clear I/O Trans 1 Gate 21E00 (refer to Plate P-49) output a high due to a low from 03T32 & L ⇒ I SEQ. The high is inverted by 01E00 & at ∅1, 02E00 outputs a H ⇒ Clear Translator via 04E00 to clear ff's 0XG07, 0XG08, 0XG00, 0XG11 & 0XG10 (Plate P-50).
T3.4	Set Hold 1 ff. Hold 1 flip-flop 1XG38 (refer to plate P-28) is set by a L ⇒ f = 50:26, 27, 40-47, L ⇒ T34 I Seq Format II and a ∅4.
T4.3	Set Inti ff if Resume - Fault to be Honored. Gate 21G21 (refer to plate P-12) is enabled by a H ⇒ Spec Int, high outputs from gates 20G20 and 23G20, the absence of a L ⇒ B1 Seq, and a high from 14G18 (refer to plate P-54) generated by INSERT EAB and HOLD 2 flip-flops being set. The H ⇒ Spec int is the output of 16G14 (refer to plate P-54) which outputs a high due to Resume Fault flip-flop 1XG18 being set. Set Hold 2 ff. Hold 2 flip-flop 2XG38 (refer to plate P-28) is set by the low from Hold 1 flip-flop at ∅3.

TABLE 2-153. SEQUENCES AFTER FIRST I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:26, 50:27 WITH OUTPUT AVAILABLE AND NO RESUME FAULT

Time Notation	Commands
	HELD SEQUENCE FOLLOWING NORMAL I-SEQUENCE OF INSTRUCTION
T2.1	Clear I/O Trans 1
T2.2	Set $I/O1_i$ ff, set clear Hold ff, f KO → I/O Trans 1
T4.2	Set $I/O1_f$ ff, clear Hold 1 ff
T4.3	Clear Clear Hold ff
	I-SEQUENCE OF NEXT INSTRUCTION
T1.3	Clear Hold 2 ff

NOTES: First sequence shown is not under instruction sequence control. Sequences are disabled by the Hold 1 and 2 flip-flops set during the first I-sequence. The I/O operations performed during this time simulate those which occur during the Scan sequence that precedes I/O1.

The I/O1-sequence performs normal EF or OD operations. Normal events which occur after the setting of $I/O\ 1_i$ flip-flop are not shown. Normal I/O 2-sequence follows if dual-channel operations is in effect.

TABLE 2-154. COMMANDS/DESCRIPTIONS FOR SEQUENCES AFTER FIRST I-SEQUENCE WITH OUTPUT AVAILABLE AND NO RESUME-FAULT
(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
T2.2	Set Clear Hold ff. Depending upon the channel, one of four gates, 10G57, 11G57, 12G57 or 13G57 (refer to plate P-30) will be enabled. Assuming channel 0 is used, gate 10G57 will be enabled by a L ⇒ Resume, Chan 0, 2, 4, 6 a low from 02K00, and a low from 02K03, which had a high input from 01K03. The high output is inverted by 14G57 and applied as a L ⇒ Resume to 20E00 (refer to plate P-49). Other inputs to 20E00 are a low from 03T22, the absence of H ⇒ I/01 · (Dual + ESI Term + Cont Data Mode, a low from 40E00 (which is disabled by the absence of a L ⇒ RTC Seq) and a L ⇒ Force. The L ⇒ Force is developed due to Hold 1 flip-flop (refer to plate P-28) set and a L ⇒ f = 50:26, 27, 40-47 and the absence of a H ⇒ f = 50:40-47 to 12G38. The high from 20E00 is applied as a H ⇒ Force into 25E00 (refer to plate P-28) and the low output sets Clear Hold flip-flop OXG38 at ∅2. f · K0 → I/O Trans 1. The high from 20E00 (refer to plate P-49) is inverted by 22E00 and at ∅2, 23E00 produces a L ⇒ f and K0 → I/O Trans 1.
T4.2	Clear Hold 1 ff. The low from Clear Hold flip-flop (refer to plate P-28) and a low from 05T42 clears Hold 1 flip-flop ∅2.
T4.3	Clear Clear Hold ff. The low from the clear side of Hold 1 flip-flop clears Clear Hold flip-flop. I-SEQUENCE OF NEXT INSTRUCTION
T1.3	Clear Hold 2 ff. Hold 2 flip-flop (refer to plate P-28) is cleared by the low from Hold 1 flip-flop and a low from 13T14 at ∅3.

TABLE 2-155. SEQUENCES AFTER FIRST I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:26, 50:27 WITH OUTPUT NOT AVAILABLE AND NO RESUME-FAULT
(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Commands
	Held Sequences Following Normal I-Sequence of Instruction
	(wait for Output Available)
	Output Available
next T2.1	Clear I/O Trans 1
T2.2	Set I/O_i ff, set Clear Hold ff, f • K0 → I/O Trans 1
T4.2	Clear Hold 1 ff, set I/O_f ff
T4.3	Clear Clear Hold ff
	I-SEQUENCE OF NEXT INSTRUCTION
T1.3	Clear Hold 2 ff

NOTES: Until output is available, no instruction sequence is in control. Sequences are disabled by the Hold 1 and 2 flip-flops set during the first I-sequence. The I/O operations performed after the setting of $I/O1_i$ flip-flop and prior to the I/O1 sequence simulate those which occur during the Scan sequence that precedes I/O1.

The I/O1 sequence performs normal EF/OD operations. Normal events which occur after the setting of $I/O1_i$ flip-flop are not shown. Normal I/O2 sequence follow if dual-channel operation is in effect.

TABLE 2-156. SEQUENCES AFTER FIRST I-SEQUENCE ESSENTIAL COMMANDS FOR f = 50:26, 50:27 WITH RESUME-FAULT

Time Notation	Commands
	HELD SEQUENCE FOLLOWING NORMAL I-SEQUENCE OF INSTRUCTION
T1.4	*Set Decr P ff, *set Insert EAB ff
	*Arith Sel and Arith Sel' → X (set X = all 1's)
T2.2	Set Clear Hold ff, * set OXL11 ff
T2.3	*clear Insert EAB ff, *clear D
T2.4	*Set Insert EAB ff, $*P_L \rightarrow D_L$, $*P_U \rightarrow D_U$
T3.1	*Clear Decr P ff
T3.2	*Clear P, *Adder → P
T3.3	*Clear OXL11 ff
T4.2	Clear Hold 1 ff
T4.3	*Clear Insert EAB ff, clear Clear-Hold ff
	INTERRUPT AND I-SEQUENCE IN PARALLEL
T1.3	Clear Hold 2 ff
T2.1	Clear I_f ff
T2.2	Set I_f ff, set Int_f ff

NOTES: * These events are concerned with or are controlled by the decrementing function of the Advance-P subsequence.

First sequence shown is not under sequence control. Sequences are disabled by the Hold 1 and 2 flip-flops set during the first I-sequence.

Second sequence is a normal interrupt sequence to honor the resume-fault if no higher priority signal is present. All commands for this sequence are not shown.

TABLE 2-157. COMMANDS/DESCRIPTIONS FOR SEQUENCES AFTER FIRST I-SEQUENCE WITH RESUME-FAULT

(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
T1.4	Set Decr P ff. Gate 13G38 (refer to plate P-28) has a L ⇒ Force output, due to Hold 1 flip-flop being set at time T3.4 of the first I-sequence. The L ⇒ Force is applied to 30L10 (refer to plate P-35). With Resume-Fault flip-flop 1XG18 (refer to plate P-54) set, a L ⇒ Resume-Fault is also applied to 30L10, and with the low from 01T21, 30L10 outputs a high which is inverted by 31L10 to set Decrement P flip-flop 4XL10 at ∅4. Set Insert EAB ff. A H ⇒ Decrement P (refer to plate P-35) from 42L10 is applied to 21E32 (refer to plate P-27) where it is inverted to set Insert EAB flip-flop OXG32 at ∅4. Arith Sel and Arith Sel' → X(set X = All 1's). A H ⇒ Set X = -0 from 42L10 (refer to plate P-35) is applied to gates 18N03 and 28N03 (refer to plate P-19) where it is inverted. At ∅4, 19N03 outputs a L ⇒ Select → X and 29N03 outputs a L ⇒ Select → X into the X-register (refer to plates P-87 and P-88) where the input from the Arthmetic Selector sets the X-register to 1's.
T2.2	Set OXL11 ff. Flip-flop OXL11 (refer to plate P-35) is set at ∅2 by a L ⇒ $\overline{\text{Force}}$, and a low from Decrement P flip-flop 4XL10.
T2.3	Clear Insert EAB ff. Gate 02E32 (refer to plate P-27) is enabled by the low from 01E32 (which is enabled by the high from 00T23) and ∅3 to clear Insert EAB flip-flop OXG32. Clear D. The high from Decrement P flip-flop 4XL10 (clear side) (refer to plate P-35) is inverted by 03L10 to produce a L ⇒ Advance P to 30N02 (refer to plate P-18). The high output is inverted by 08N02 and with a low from disabled gate 01N02, 09N02 outputs a L ⇒ Clear D at ∅3.
T2.4	Set Insert EAB ff. The EAB flip-flop is set the same as T1.4. $P_L \rightarrow D_L$, $P_U \rightarrow D_U$. The high from 30N02 (refer to plate P-18) is inverted by 36N02 and 37N02, and at ∅4, 38N02 produces a L ⇒ PL → DL and 39N02 produces a L ⇒ PU → DU.
T3.1	Clear Decr P ff. The high from 30L10 (refer to plate P-35) that set Decrement P flip-flop 4XL10 also clears the flip-flop at ∅1.
T3.2	Clear P, Adder P. This command is executed the same as described at T2.2 of an I-sequence.
T3.3	Clear OXL11 ff. Flip-flop OXL11 (refer to plate P-35) is cleared at ∅3 by the lows from the clear sides of the Increment P and Decrement P flip-flops.
T4.3	Clear Insert EAB ff. This command is executed the same as T2.3 except that the high into 01E32 (refer to plate P-27) is from 00T43.

2-549. Decrement P Data Flow. Refer to figure 2-141 for a block diagram description of the decrementing function for the Advance P subsequence. Prior to this operation, P has been advanced to the next address. Since the output function has not been accomplished and a Resume-Fault interrupt will be honored, it is necessary to decrement to the address of this same instruction to execute it again after interrupt operations. Decrementation operations are similar to incrementation operations. The P value to be decremented is placed in D. X is set to all ones (-0). Thus, the subtraction of X - D' will not require an End-Around-Borrow (EAB) and the adder output should be (-0)-D' or D. The borrow is inserted to cause the adder output to be D-1 which is placed in P

2-550. EXECUTION OF SKPIIN, SKPOIN, SKPFIN, SKPNR, SRSM INSTRUCTIONS. These instructions provide means for checking the activeness of I/O buffers on any channel, checking the Resume conditon on any chassis, and forcing the Resume condition on any chassis.

2-551. SKPIIN (f = 50:21). The SKPIIN instruction senses the status of ID Active flip-flop for channel k. If this flip-flop is clear, the next sequential instruction is skipped.

2-552. SKPOIN (f = 50:22). The SKPOIN instruction senses the status of OD buffer for channel k. If the OD Active flip-flop is clear, the next sequential instruction is skipped.

2-553. SKPFIN (f = 50:23). The SKPFIN instruction senses the status of EF buffer for channel k. If the EF Active flip-flop is clear, the next sequential instruction is skipped.

2-554. SKPNR (f = 50:57). The SKPNR instruction senses the status of output-available condition on I/O chassis group, which includes channel k. The next sequential instructions is skipped if output is not available. On any I/O group, output is available if an OD or EF Acknowledge signal is not currently being sent on a channel of that group, and the Resume flip-flop on that chassis is set. For non-intercomputer

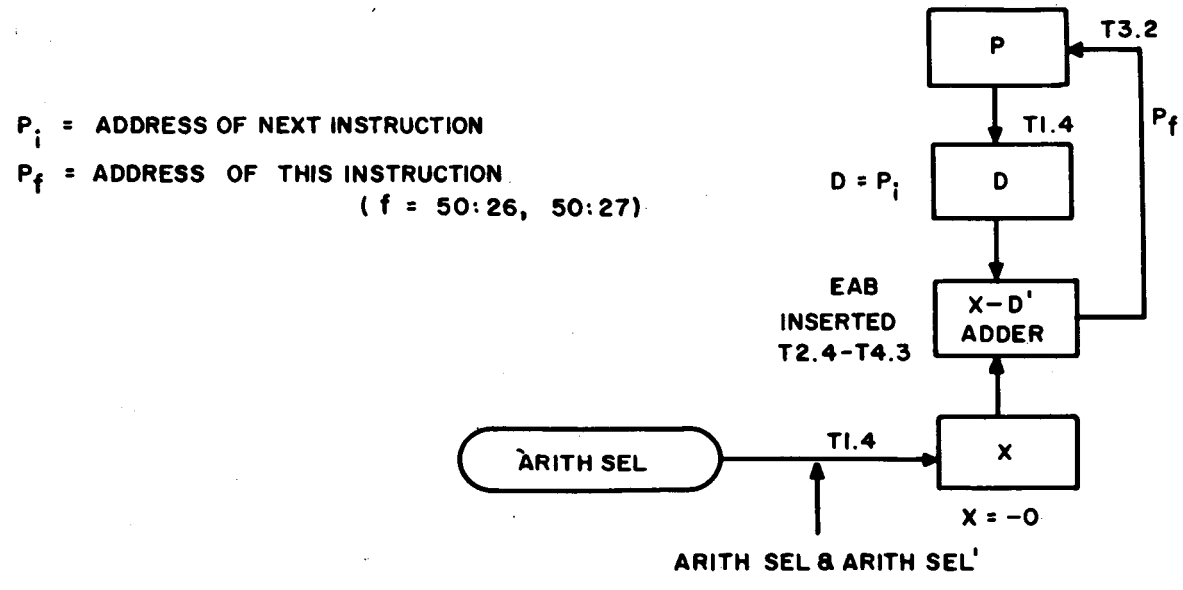


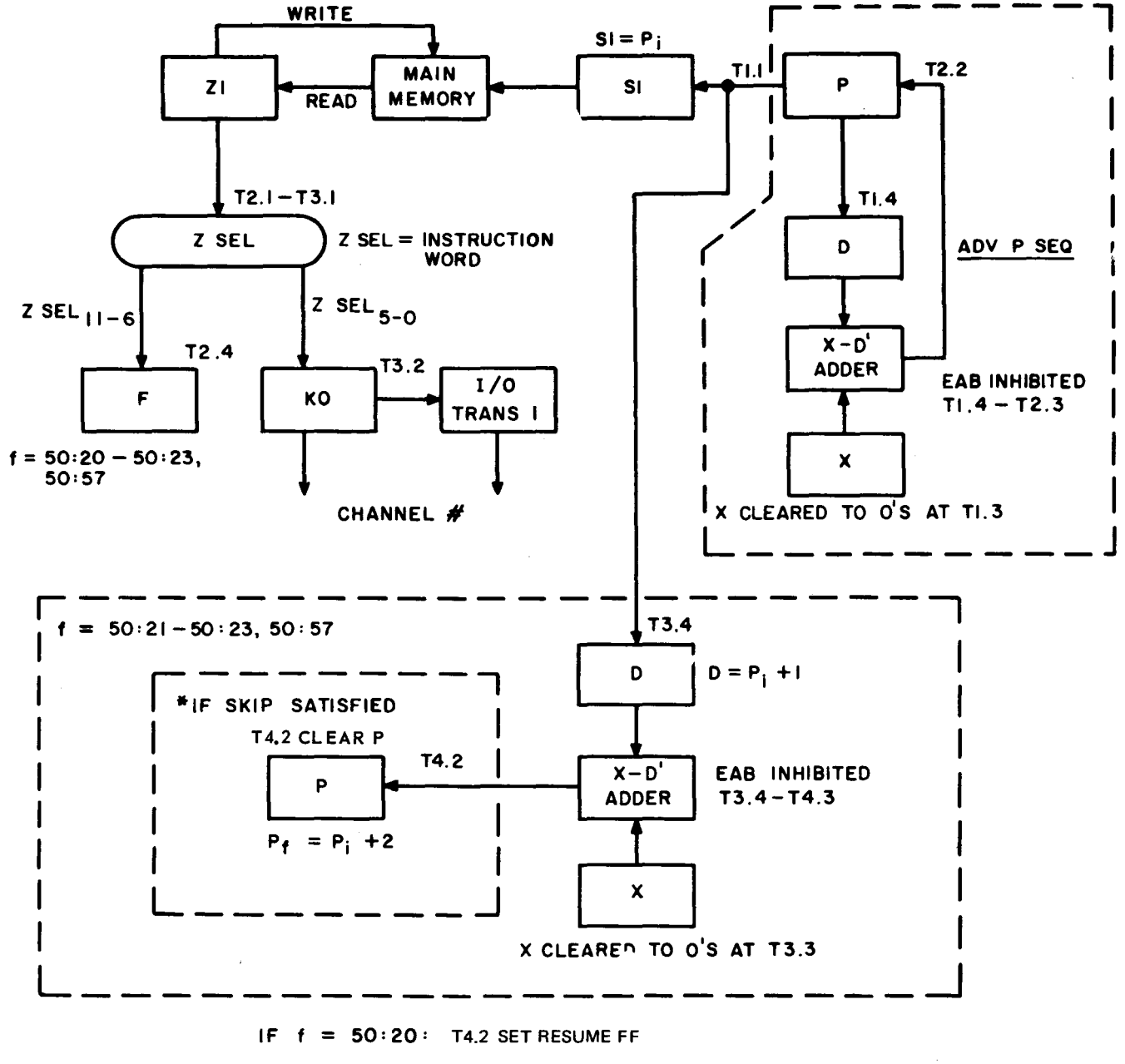
Figure 2-141. P Decrement for f = 50:26, 50:27 with Resume-Fault

operation, the condition of output-not-available only exists for the duration of the acknowledge since the Resume flip-flop is set by the acknowledge timing. For intercomputer operations, the Resume flip-flop is initially set; however, as each word is outputted, it is cleared. The ID Acknowledge from the receiving computer, which appears to the sending computer as an ODR, sets the Resume flip-flop. The output-not-available or no-resume condition, then, is an indication that the last word sent to the receiving computer has not been acknowledged.

2-555. SRSM, f = 50:20. The SRSM instruction sets the Resume flip-flop on the I/O chassis group which includes channel k. Output then becomes available on this chassis if an OD or EF Acknowledge is not currently being sent. The SRSM instruction would probably be executed after occurrence of the Resume-Fault special interrupt. The Resume-Fault interrupt occurs if the Resume flip-flop remains clear for a specific period of time. The interrupt occurs in the Inter-computer mode, and indicates that the receiving computer did not accept the last outputted word within the allowed time period. The Resume flip-flop, in the clear state, causes output not to be available on its chassis, and prevents further OD and EF operations on that chassis. The SRSM instruction re-establishes the output-available and Resume condition. All operations are performed within the I-sequence. Only the memory reference needed to obtain the instruction is necessary.

2-556. I-Sequence Data Flow For f = 50:20 through 50:23 and 50:57. Refer to figure 2-. for a block diagram description of the execution of f = 50:20 through 50:23, 50:57. Most I-sequence operations are as previously described. K0 and I/O translation 1 receive the channel and I/O chassis selection. This information is specified by the four least significant bits of k. For f = 50:21 - 50:23, 50:57, the X – D' adder is used to increment a second time, P by +1, just as is done by the Advance-P subsequence. The second clearing of P and transfer into P from the adder is conditioned by the buffer activeness or output available condition as determined by function code. If P receives the result of this second incrementation, the next sequential instruction will be skipped. For f = 50:20, the Resume flip-flop on the chassis specified by k is set at time T4.2. As previously discussed, the instruction could be obtained from bootstrap or control memory.

2-557. I-Sequence Essential Commands Fc f = 50:20 through 50:23 and 50:57. Refer to table 2-158 for a list of essential I-Sequence events for f = 50:20 through 50:23 and 50:57. Commands not previously described are described in table 2-159 in detail.



NOTE: *SKIP IF: (f = 50:21)·(ID ACT FF CLEAR)
(f = 50:22)·(OD ACT FF CLEAR)
(f = 50:23)·(EF ACT FF CLEAR)
(f = 50:57)·(OUTPUT NOT AVAILABLE)

Figure 2-142. I-Sequence Data Flow for f = 50:20 - 50:23, 50:57

TABLE 2-158. I-SEQUENCE DATA FLOW FOR f = 50:20-50:23, 50:57

Time Notation	Commands	f = 50:21 50:22 50:23 50:57	50:20
T4.4	Clear S1	X	X
T1.1	P → S1, Init Memory, *set Incr P ff	X	X
T1.3	*Clear D, *clear X, clear Z1, clear F, *set OXL11 ff	X	X
T1.4	*P_L → D_L, *P_U → D_U, *set Inhib EAB ff	X	X
T2.1	Z1 → Z-Sel, *clear Incr P ff	X	X
T2.2	*Clear P, *Adder → P	X	X
T2.3	*Clear OXL11 ff, *clear Inhib EAB ff	X	X
T2.4	$Z\text{-}Sel_{11-6}$ → F, set OXF06 ff	X	X
T3.1	Drop Z1 → Z-Sel, clear I/0 Trans 1	X	X
T3.2	f • K0 → I/0 Trans 1	X	X
T3.3	Clear D, clear X	X	
T3.4	P_L → D_L, P_U → D_U, set Inhib EAB ff	X	
T4.2	Set Resume ff		X
	Clear P if skip satisfied **	X	
	Adder → P if skip satisfied **	X	
T4.3	Clear Inhib EAB ff	X	

*These events are concerned with or are controlled by the Advance-P subsequence.

**Skip condition is satisfied if:

(f = 50:21)	(ID Act ff clear)
(f = 50:22)	(OD Act ff clear)
(f = 50:23)	(EF Act ff clear)
(f = 50:57)	(output not available)

TABLE 2-159. I-SEQUENCE COMMANDS/DESCRIPTIONS
FOR f = 50:20 THROUGH 50:23 AND 50:57
(For Detailed Description of all Commands Previously Described Refer to Table 2-19)

Time Notation	Command/Description
T3.3	Clear D. Gate 30N02 (refer to plate P-18) is enabled by a L ⇒ f = 50:21-23, 50-55, 57 and a L ⇒ T34 I Seq Format II to clear D as described in a previous I-sequence. Clear X. Gate 30N02 (refer to plate P-18) also outputs a H ⇒ Advance P into 08N03 (refer to plate P-19) which is inverted and at ∅3, 09N03 produces a L ⇒ Clear X.
T3.4	$P_L \rightarrow D_L$ $P_U \rightarrow D_U$. Set Inhibit EAB ff. The high from 30N02 (refer to plate P-18) is also inverted by 36N02 and 37N02. At ∅4, 39N02 produces L ⇒ $P_U \rightarrow D_U$ and 38N02 produces L ⇒ $P_L \rightarrow D_L$. The low from 36N02 produces a L ⇒ Set Inhib EAB.
T4.2	Set Resume ff. Set Resume gate 00E18 (refer to plate P-53) is enabled by a L ⇒ f = Even (from the F-register), L ⇒ f = 20, 21 and a L ⇒ T43 I Seq Format II. The high output is inverted by 01E18 and applies a L ⇒ Set Resume to Resume flip-flop 5XLg0 (refer to plate P-69 & 69A) which is set with a L ⇒ Group g at ∅2. Clear P if Skip Satisfied, Adder → P if Skip Satisfied. For a 50:21 instruction and if ID Act flip-flop (refer to plates P-61 through P-64) is clear, the high from the flip-flop is applied to pin 5 of 90Rg0 (refer to plate P-61, using channel 0, 1 as an example). Pins 10 and 13 are high due to the absence of a L ⇒ f = 50:22 and 50:23. For a 50:22 instruction and OD Act flip-flop clear (refer to plates P-61 through P-64) the high from the flip-flop is applied to pin 8 of 90Rg0. Pins 6 and 13 are high due to the absence of a L ⇒ f = 50:21 and 50:23. For a 50:23 instruction and EF Act flip-flop clear (refer to plates P-61 through P-64) the high from the flip-flop is applied to pin 12 of 90Rg0. Pins 6 and 10 are high due to the absence of a L ⇒ f = 50:21 and 50:22. Since 90Rg0 is disabled it outputs a L ⇒ $\overline{\text{Active}}$ to 91Rg0 (refer to plate P-60) and, with a selected channel, 91Rg0 outputs a H ⇒ $\overline{\text{Active}}$ to disable 92R00 (refer to plate P-30). The low output is inverted by 92R10 to disable 20G50 and output a L ⇒ Skip to 10N07 (refer to plate P-21). The other inputs are L ⇒ f = 50:21-23, 50-55, 57 and a L ⇒ T42 I Seq. The high output now clears P and sends Adder → P as described in a previous I-Sequence.

TABLE 2-159. I-SEQUENCE COMMANDS/DESCRIPTIONS
FOR f = 50:20 THROUGH 50:23 AND 50:57 (Cont)

Time Notation	Command/Description
T4.2 (Cont)	For a f = 50:57 instruction and output not available, gate 52Lg0 (refer to plate P-69 & 69A) outputs a H ⇒ $\overline{\text{Output Available}}$ to disable 10G57, 11G57, 12G57 and 13G57 (refer to plate P-30) via plate P-34. The low output is inverted by 14G57 and applied to pin 8 of 20G50 to produce a L ⇒ Skip, which clears P and sends Adder → P as previously described.

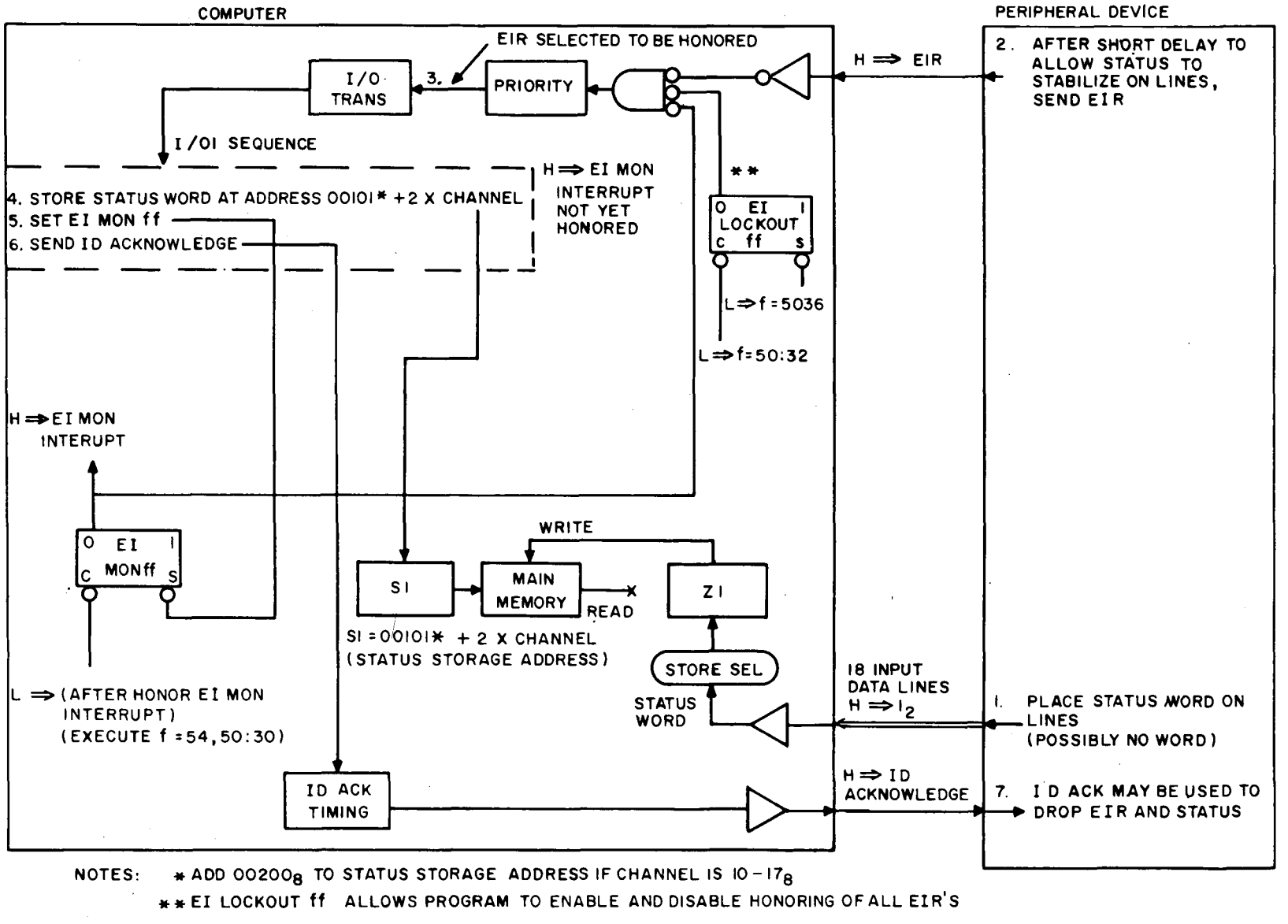
2-558. EXTERNAL INTERRUPT REQUEST OPERATIONS. The external interrupt is a signal from an external device which requests a program jump to a special address as well as acceptance of an inputted code word.

2-559. Single-Channel Operation. The External Interrupt signal (EIR) from an external device may be considered as a request. (Refer to figure 2-143). It requests that the computer accept the inputted 18-bit word from the input data lines which usually accompanies the EIR signal. The code word is sometimes referred to as a status word. Its bit configuration may indicate data handling errors, or affect control over a certain program. However, the EIR signal may itself be the desired indication without a code word. For example, it may signify completion of a specific time period to synchronize computer program operations with real time. In addition to storage of the code word, the EIR signal sets the EI Monitor flip-flop for the particular channel. This flip-flop generates the internal EI Mon signal, which is handled as a separate request for I/O services. This signal is detected during the interrupt scan period. When honored, the EI Mon signal causes a program jump to the address 00100_8 + 2K (K ⇒ channel number) if the channel is 0-7, or 00300_8 + 2K if the channel is $10\text{-}17_8$. Setting the External Interrupt Lockout flip-flop by f = 50:36 and 50:37 (discussed in a later section) prevents detection of both EIR and EI Mon on any channel. Therefore, honoring of these signals can be delayed for a long as desired, after which this lockout flip-flop can be cleared by f = 50:32 and 50:33. During EIR-honoring operations, the code word is stored and the IE Mon flip-flop is set. Once the EI Mon flip-flop is set, it prevents detection of another EIR signal on the same channel. The EI Mon flip-flop is not cleared until its signal is honored. Thus, the EIR signal occurrence is recorded by this flip-flop until the program jump is effected by honoring the EI Mon signal. During the honoring of the EIR signal, the code word is stored in memory at address 00101_8 + 2K if the channel is 0-7, or at 00301_8 +2K if the channel is $10\text{-}17_8$.

2-560. Dual-Channel, ESI, or ESA Operation. If the CHANNEL FUNCTION switch is positioned to select Dual-Channel, ESI, or ESA modes, dual-channel operation is initiated. The difference between resulting EIR-signal-honoring operations, and those with single-channel selection is storage of a 36-bit inputted code word instead of an 18-bit word. First, the EIR carrying the odd-channeled half of the code word is stored in memory at address 00101_8 + 2K of the channel is 0-7, or at 00301_8 + 2K if the channel is $10\text{-}17_8$. Then, the next lower even-channeled half of the code word is stored in memory at address 00101_8 + 2K even-channel if the channel is 0-7, or at 00301_8 + 2K even-channel if the channel is $10\text{-}17_8$ The storage address for the second half of the code word is dependent upon the even channel number.

2-561. I/O1 and I/O2 Sequence Data Flow For EIR Operations. Other than those events concerned with selection of the EIR signal and setting translators accordingly, no essential data flow occurs during the Scan sequence. Refer to figure 2-144 for a block diagram description of EIR operations. If single-channel operation is in effect, only the I/O1 sequence is used. A memory cycle is initiated to store the 18-bit code word from the data lines. If the dual flip-flop is set (dual-channel, ESI, or ESA), half of the 36-bit code word from the odd channel is stored and the I/O2 sequence is used to store the second half of the 36-bit code word in memory.

2-562. Scan and I/O2 Sequence Essential Commands for EIR Operations. Refer to table 2-160 for a list of essential scan and I/O2 sequence events. Events concerning priority selection were previously discussed and are not shown. The ID Acknowledge timing is the same as presented for ID honoring operations. Commands not described previously are described in table 2-161 in detail.



NOTES: * ADD 00200_8 TO STATUS STORAGE ADDRESS IF CHANNEL IS $10-17_8$

** EI LOCKOUT ff ALLOWS PROGRAM TO ENABLE AND DISABLE HONORING OF ALL EIR'S

Figure 2-143. External Interrupt Operations, Single Channel

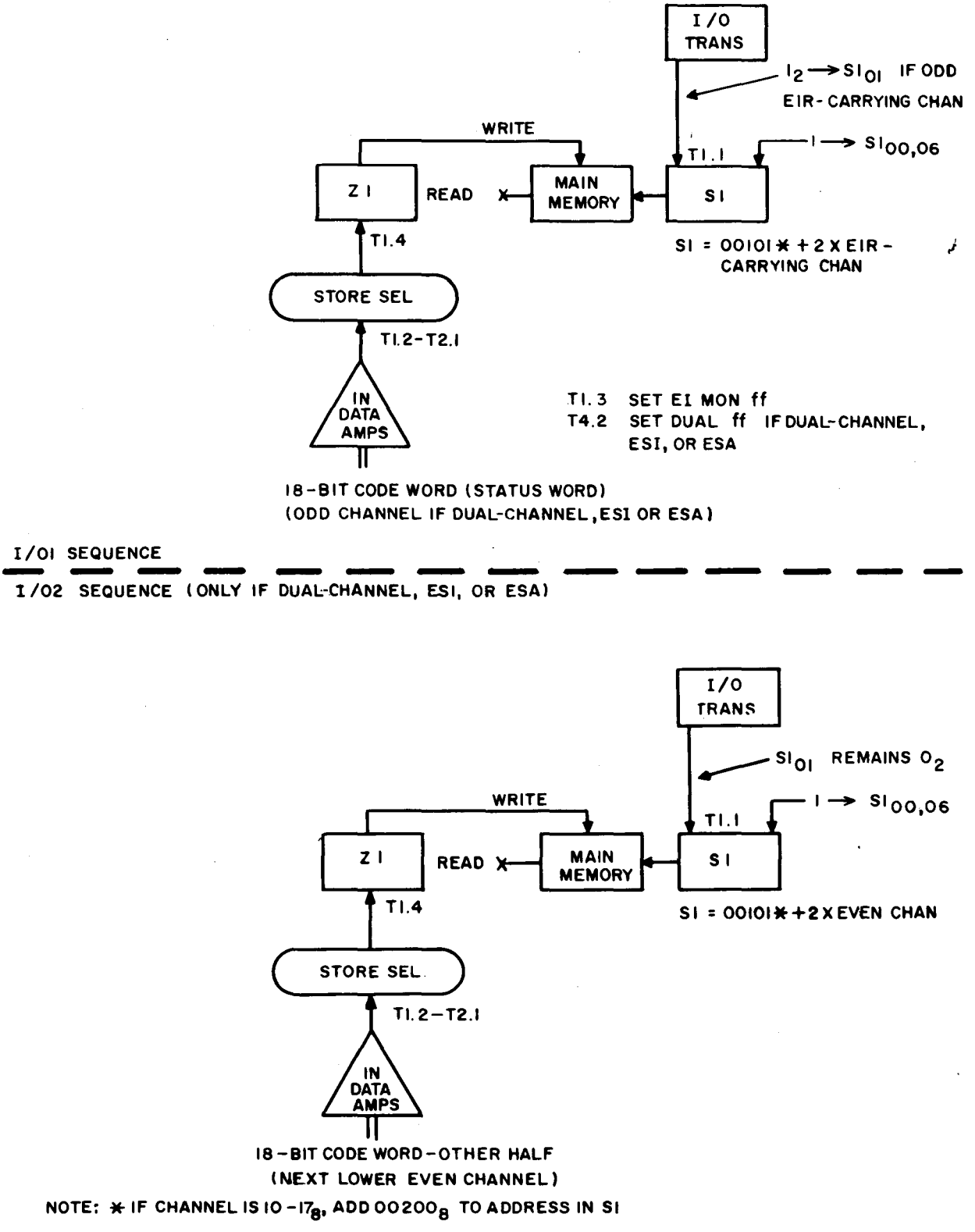


Figure 2-144. I/01 and I/02-Sequence Data Flow for EIR Operations

TABLE 2-160. SCAN I/01, AND I/02 SEQUENCE ESSENTIAL COMMANDS FOR EI OPERATIONS

Time Notation	Commands
	SCAN SEQUENCE
T2.1	Clear Fct, Chan, and I/O Trans 1
T2.2	Set I/O_i ff, Pri → Fct, Chan, and I/O Trans 1
T3.1	Clear ZO
T3.2	Clear B
T3.4	ZO → B
T4.1	Clear I/O Trans 2, clear B ± 1 ff
T4.2	Set $I/01_f$ ff, clear all ID Ack Reg ff's, I/O Trans 1 → I/O Trans 2 Set Dual ff if dual, ESI, or ESA
T4.3	Clear EI Req ff, set ID Ack Reg ff, set 6XLg0 ff Clear Pri Alternator ff
T4.4	Clear S1, disable Mem → Z1
	I/01 SEQUENCE
T1.1	I/O Trans → S1**, 1 → $S1_{00}$, 1 → $S1_{06}$, Init Memory
T1.2	Input Data Amps → Store Sel*
T1.3	Set EI Mon ff, clear Z1
T1.4	Store Sel → Z1, set ID Ack Gener ff if Dual ff clear
T2.1	Clear $I/01_i$ ff, drop Input Data Amps → Store Sel
T2.2	Set $I/02_i$ ff if Dual ff set
T2.4	Drop disable Mem → Z1
The following occurs if $I/02_i$ ff set (dual-channel) except clear $I/O1_f$ ff	
T3.1	Clear ZO
T3.2	Clear B
T3.4	ZO → B
T4.1	Clear $I/01_f$ ff, clear I/O Trans 2, clear B ± 1 ff

TABLE 2-160. SCAN I/01, AND I/02 SEQUENCE ESSENTIAL COMMANDS FOR EI OPERATIONS (Cont)

Time Notation	Commands
T4.2	Set $I/02_f$ ff, I/O Trans 1 → I/O Trans 2, clear all ID Ack Reg ff's Set Dual ff if dual, ESI or ESA
T4.3	Set ID Ack Reg ff, set 6XLg0 ff
T4.4	Clear S1, disable Mem → Z1
	I/02 SEQUENCE (if dual channel)
T1.1	I/O Trans → S1**, 1 → $S1_{00}$, 1 → $S1_{06}$, Init Memory
T1.2	Input Data Amps → Store Sel*
T1.3	Clear Z1
T1.4	Store Sel → Z1, set ID Ack Gener ff
T2.1	Clear $I/02_i$ ff, drop Input Data Amps → Store Sel
T2.4	Drop disable Mem → Z1
T4.1	Clear $I/02_i$ ff

*If operation is in Dual-Channel, ESI or ESA Mode, the channel inputted to Store Sel is as follows:

I/01 Seq.	I/02 Seq.
odd	even

**$S1_{01}$ remains 0_2 during I/02 sequence but is set to 1_2 during I/01 sequence if EIR-carrying channel is odd.

TABLE 2-161. SCAN AND I/02 SEQUENCE COMMANDS/DESCRIPTIONS FOR EIR OPERATIONS

(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
T4.2	Set Dual ff if Dual, ESI, or ESA. If either ESA flip-flop OXG11 or ESI flip-flop OXG10 (refer to plate P-50) is set, the high from the clear side is inverted by 05G10 and applied to 06G10. The other input to 06G10 is a L ⇒ Ext Int from the Function Translator. The high output is inverted by 07G10 and applied as a L ⇒ Int + ESI into Dual flip-flop 4XG09 (refer to plate P-52). If in Dual mode a L ⇒ Dual is applied to the Dual flip-flop , a L ⇒ Translator 1 ⇒ Translator 2 allows the flip-flop to be set at time T4.2.
T4.3	Clear EI Req ff. Set In Ack Reg gate 00N49 (refer to plate P-53) is enabled by a L ⇒ I/0 Seq, a low from 03T43 and a low from 00N40, which is disabled by the absence of a L ⇒ Out + EF. The H ⇒ Set ID Ack output is inverted by 1gN49 (refer to plate P-58) and applied as a L ⇒ Set ID Ack to 9gN49 (refer to plate P-57), inverted and applied to 7gN40. The other input to 7gN40 is a H ⇒ Ext Int from the Function Translator. A L ⇒ Clear EI Req from 7gN40 is applied to the EI Req flip-flop (refer to plate P-61 through P-64) and will clear with a low from the selected channel.
T1.3	Set EI Mon ff. Set Monitors gate 00N42 (refer to plate P-53) is enabled by a L ⇒ Ext Int from the Function Translator and a L ⇒ T13 I/0 Seq and the high output is inverted by 01N42 and applied as a L ⇒ Set Monitor to 9gN42 (refer to plate P-56), inverted and applied to 7gN42. The other input is a H ⇒ Ext Int and 7gN42 outputs a L ⇒ Set EI Monitor to EI Mon flip-flops (refer to plate P-61 through P-64) which will set with the selected channel.

2-563. REAL-TIME-CLOCK (RTC) REQUEST OPERATIONS, GENERATION OF RTC MONITOR INTERRUPTS, AND EXECUTION OF RTC INSTRUCTIONS. The real-time-clock (RTC) is an oscillator running at 1024 pulses per second. It is used to increment by +1 the content of control-memory address 00015_8 at a constant rate. One complete cycle of the RTC takes place each four and one quarter minutes. By programmed instructions of memory address 00015_8, The RTC can be used to time program operations. This address can be entered by a Store instruction to preset it to a specific count which later can be used as a real-time reference. Each cycle of the RTC generates a request signal which is honored by the I/O logic. The RTC honoring operations increment by +1 the content of address 00015_8. Each bit position of the memory location represents twice the time value of the next less significant bit position. Refer to figure 2-145 for bit position time weights. If the RTC DISCONNECT switch is up, the RTC signal is not presented to the I/O logic.

BIT POSITIONS	09	08	07	06	05	04	03	02	01	00
	1	1	1	1	1	1	1	1	1	1
	2	4	8	16	32	64	128	256	512	1024
BIT POSITIONS	17	16	15	14	13	12	11	10		
	128	64	32	16	8	4	2	1		

Figure 2-145. RTC Address 00015_8 Bit Weights in Seconds

2-564. RTC (f = 50:14) Instruction and RTC Monitor Interrupt. The RTC Monitor Request flip-flop is set by the execution of f = 50:14. When this flip-flop is set, the clock count in control memory address 00015_8 is automatically monitored. The monitor function is effected by comparing the clock count each time the content of control memory address 00015_8 is incremented. A comparison count must be set in address 00014_8 by a Store instruction. When the clock count is incremented to equal the comparison count, the RTC Monitor flip-flop is set to generate the RTC monitor interrupt. This interrupt signal is handled by a separate I/O operation and causes a program jump to control memory address 00012_8.

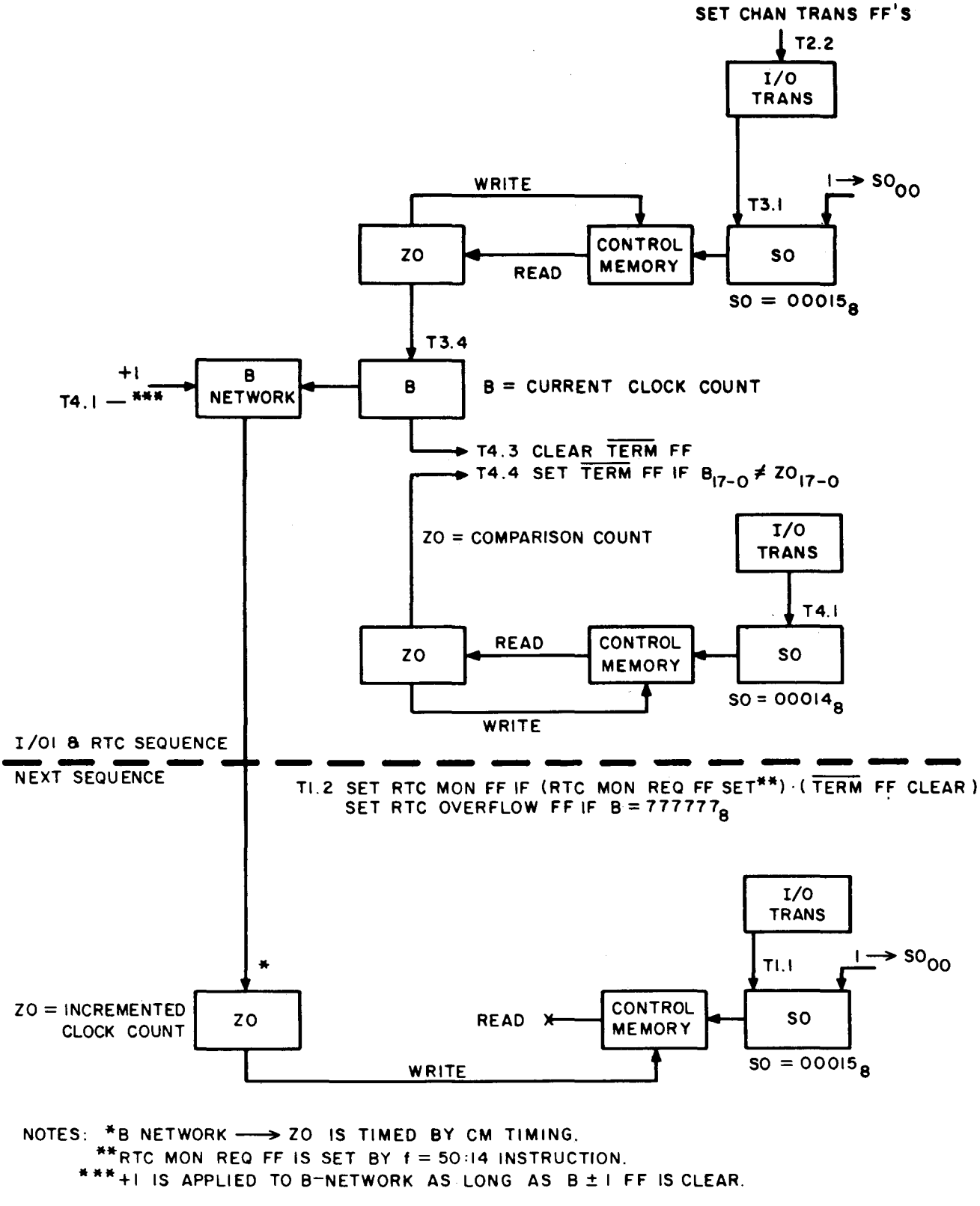
2-565. RTC Overflow Interrupt. When the clock count is 777777_8, the next RTC request will cause it to be incremented to 000000. This condition is referred to as RTC overflow, and results in setting the RTC Overflow flip-flop, which generates the RTC Overflow interrupt. The interrupt signal is handled by a separate I/O operation, and causes a program jump to control memory address 00013_8.

2-566. RTC Request Operations Data Flow. Refer to figure 2-146 for a block diagram description of RTC request operations. RTC request operations are performed during the I/01 sequence with the RTC sequence flip-flop set. At T3.1, a control memory reference is initiated to obtain the current clock count from address 00015_8. The address in SO is formulated by setting $SO_{03, 02}$ from the two-Channel Translator flip-flops. Both

flip-flops are set at time T2.2. SO_{00} is also set to complete address 00015_8. The extracted current clock-count is placed in B and a second control memory reference is used to obtain the comparison count from address 00014_8. This address is formulated in SO in the same way as 00015_8, except that SO_{00} remains clear. From ZO and B, the current clock-count is compared with the comparison count. If these two counts are equal, the Terminate flip-flop remains clear. If the RTC Monitor Request flip-flop has not been set, the result of the comparison will have no effect. The current clock-count in B is incremented by +1 in the B-network, and is restored in address 00015_8 by a third memory cycle. This memory cycle occurs during the first portion of the next sequence. The incremented clock-count is stored from ZO. The gating of control memory is disabled during the read portion of the control-memory cycle which destroys the previous clock count.

2-567. RTC (f = 50:14) Instruction. The RTC instruction uses only the I-sequence. The RTC Monitor Request flip-flop 0XG15 (refer to plate P-54) is set at time T3.3 by a L ⇒ f = 50:14 instruction. This flip-flop partially enables the setting of the RTC Monitor flip-flop. Note that if the RTC Monitor flip-flop is set, the RTC Monitor Request flip-flop is cleared. Thus, for each RTC Monitor interrupt signal desired, a separate f = 50:14 instruction must be executed.

2-568. I/01, RTC and Next Sequence Essential Commands For RTC Operations. Refer to table 2-162 for a list of I/01, RTC, and next sequence events for RTC request operations. The time notation is determined by the RTC oscillator pulse. Commands not previously described are described in detail in table 2-



NOTES: *B NETWORK ⟶ ZO IS TIMED BY CM TIMING.
**RTC MON REQ FF IS SET BY f = 50:14 INSTRUCTION.
***+1 IS APPLIED TO B-NETWORK AS LONG AS B ± 1 FF IS CLEAR.

Figure 2-146. I/01 and RTC and Next Sequence Data Flow for RTC Operations

TABLE 2-162. I/01, RTC, AND NEXT SEQUENCE ESSENTIAL COMMANDS FOR RTG OPERATIONS

Time Notation	Commands
RTC Osc Pulse	
The following occurs only if RTC Disconnect switch is down	
next ø1	Set OXG29 ff
RTC Osc pulse	
next ø3	Set RTC Req ff
ø4	
ø1	Clear OXG29 ff
	I/01 AND RTC SEQUENCE
T2.2	Set $I/01_i$ ff, set RTC Seq ff, set Chan Trans ff's
T3.1	I/O Trans → SO, 1 → SO_{00}, Init CM, set 3XG29 ff
	Clear RTC Req ff, clear ZO
T3.2	Clear B
T3.4	ZO → B
T4.1	I/O Trans → SO, Init CM, clear B ± 1 ff (+1 → B-network*), clear ZO
T4.2	Set $I/01_f$ ff
T4.3	Clear $\overline{\text{Term}}$ ff
T4.4	Set $\overline{\text{Term}}$ ff if $B_{17-0} \neq ZO_{17-0}$, disable CM→ ZO
	NEXT SEQUENCE
T1.1	I/O Trans → SO, 1 → SO_{00}, Init CM, clear ZO
	Set Resume-Fault ff if (carry** → $B\text{-network}_{10}$)·(any 8XL00-8XL30 ff clear)
T1.2	Set RTC Mon ff if (RTC Mon Req ff set)·($\overline{\text{Term}}$ ff clear)
	Set RTC Overflow ff if B = 777777
	Clear all 8XL00-8XL30 ff's if carry → $B\text{-network}_{10}$
T1.3	Clear RTC Mon Req ff if RTC Mon ff set
T2.1	Clear $I/01_i$ ff, clear RTC Seq ff
T2.3	Clear 3XG29 ff
T4.1	Clear $I/01_f$ ff

*B-network → ZO is timed by CM timing.

**Delete if ORDALT 8331 installed and change to (0XB00 Clear)

TABLE 2-163. I/01, RTC AND NEXT SEQUENCE COMMANDS/DESCRIPTIONS FOR RTC OPERATIONS
(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
Next Ø1	Set OXG29 ff. With RTC DISC switch S14 (refer to plate P-34) in the down position, a low is applied to 02G29. When the output of the RTC Osc is high, 02G29 is disabled and sets 0XG29 at Ø1.
Next Ø3	Set RTC Req ff. When RTC Osc outputs a low, 02G29 is enabled. The high is inverted by 03G29 and RTC REQUEST flip-flop lXG29 is set due to the low from 0XG29 at Ø3.
Ø1	Clear OXG29 ff. The low from RTC REQUEST flip-flop clears OXG29 at Ø1.
	I/01 AND RTC SEQUENCE
T2.2	Set RTC Seq ff. Gate 40E00 (refer to plate P-49) is enabled by a low from 03T22, a L⇒RTC REQ, the absence of a H⇒I/O1 · (Dual + ESI Term) and the absence of a H⇒R1 Seq · (f=20-23 or 57). The high output is inverted by 41E00 and applies a L⇒Set RTC to set RTC Sequence flip-flop 2XG29 (refer to plate P-34) at Ø2. Set Chan Trans ff's. The output from 41E00 (refer to plate P-49) is also applied as a L⇒RTC Advance to set Channel Translator flip-flops OXG01 and OXG02 (refer to plate P-51) at Ø2.
T3.1	Set 3XG29, Clear RTC Request ff. The low from RTC Seq flip-flop (refer to plate P-34) sets flip-flop 3XG29 and clears RTC Req flip-flop at Ø1.
T4.4	Set $\overline{\text{Term}}$ ff if $B_{17-0} \neq ZO_{17-0}$. If $B_{17-0} \neq ZO_{17-0}$, 21B16 (refer to plate P-106) will output a low and with a L⇒RTC Seq, $\overline{\text{Terminate}}$ flip-flop OXG19 will set at Ø4 with the low from 10E19.
	NEXT SEQUENCE
T1.1	Set Resume-Fault ff if (Carry* ⇒B Network$_{10}$)·(Any 8XL00-8XL30 ff clear). The Resume-Fault ff can be set in two different ways. If ORDALT 8331 is not installed, a H⇒Set Resume Fault FF Signal from 11B10 (plate 119)(indicating a resume signal was not received in one second B=1024) is applied to 11E18 (plate P-54) If ORDALT 8331 is installed, a H⇒SET RESUME FAULT FF from OXB00 (plate P-118) (indicating a resume signal was not received in one millisecond is applied to 11E18) Gate 10E15 outputs a high, due to a L⇒RTC

*Delete if ORDALT 8331 installed and change to (OXB00 Clear)

TABLE 2-163. I/01, RTC, AND NEXT SEQUENCE COMMANDS/DESCRIPTIONS FOR RTC OPERATIONS (Continued)

Time Notation	Command/Description
	Seq and a low from 01T12. If any of the flip-flops 8XL00 through 8XL30 (refer to plate P-34) are clear, the high output of the flip-flop will be inverted by 82L00 and apply a L ⇒ Resume Not Available to the Resume-Fault flip-flop 1XG18 (refer to plate P-54), which will set at ø1.
T1.2	Set RTC Mon ff if (RTC Mon Req ff Set) • ($\overline{\text{Term}}$ ff Clear). The high from 10E15 (refer to plate P-54) is also applied to the 11E15. If RTC Mon Req flip-flop QXG15 is set, a high will be applied to 11E15. The other input is a H ⇒ Terminate, due to $\overline{\text{Term}}$ flip-flop being clear. The low output of 11E15 will set RTC Monitor flip-flop 1XG15 at ø2.
	Set RTC Overflow ff if B = 777777. The setting of the RTC Overflow flip-flop is conditioned by the B-network. Gate 10B17 (refer to plate P-120), at a low level, indicates a carry to B-network$_{17}$. Flip-flop OXB17 at a low level, indicates $B_{17}=1_2$. These low inputs into 10E17 (refer to plate P-54), which outputs a high to 11E17, imply $B= 777777_8$. The other input to 11E17 is the high from 10E15 and at ø2, RTC Overflow flip-flop 1XG17 is set.
	Clear all 8XL00-8XL30 ff if (*Carry → B-network$_{10}$.) When 11E18 (refer to plate P-54) was enabled a L ⇒ Overtime Fault was applied to 85L00 (refer to plate P-34), which outputted a low to clear flip-flops 8XL00 through 8XL30 at ø2.
T1.3	Clear RTC Mon Req ff if RTC Mon ff Set. The low from RTC Monitor flip-flop (refer to plate P-54) clears RTC Mon Req flip-flop at ø3.
T2.1	Clear RTC Seq ff. RTC Sequence flip-flop 2XG29 (refer to plate P-34) is cleared at ø1 and by a low from 03T22.
T2.3	Clear 3XG29 ff. The low from RTC Sequence flip-flop (refer to plate P-34) clears 3XG29 at ø3.

*Delete if ORDALT 8331 installed and change to (0XB00 Clear)

2-569. EXECUTION OF RIL, EXL, SIL, SXL, WTFI, AND SPECIAL AND MONITOR INTERRUPT OPERATIONS INSTRUCTIONS. Monitor interrupts indicate OD, EF, and ID buffer termination as well as occurrence of an EI-request to the program. Special interrupts indicate fault function codes, intercomputer error, special RTC conditions, and occurrence of the external sync signal. These instructions allow program control over honoring of interrupts and a wait for an interrupt.

2-570. RIL (f = 50:30). The RIL instruction clears the All interrupt Lockout (AIL) flip-flop to enable honoring of any special or monitor interrupt. EI monitor interrupts may still be locked out by the External Interrupt Lockout (EIL) flip-flop. The k designator is not used. If an EI monitor interrupt was the last interrupt to be honored and the All Interrupt Lockout flip-flop has not since been cleared, the particular EI Monitor flip-flop is cleared.

2-571. EXL (f = 50:32). The EXL instruction clears the External Interrupt Lockout (EIL) flip-flop to enable honoring of any EI request signal or EI monitor interrupt. EI monitor interrupts may still be locked out by the All-Interrupt Lockout flip-flop. The k designator is not used.

2-572. SIL (f = 50:34). The SIL instruction sets the All Interrupt Lockout flip-flop thereby preventing honoring of all special and monitor interrupts. The k designator is not used. When honored, interrupts cause program jumps to special addresses. The SIL instruction could be used to prevent any undesirable interruptions.

2-573. SXL (f = 50:36). The SXL instruction sets the External Interrupt Lockout flip-flop thereby preventing honoring of all EI request signals and EI monitor interrupts. The k designator is not used. The SXL instruction could be used to inhibit program interruptions from EI monitor interrupts without disabling the effect of other interrupts.

2-574. WTFI (f = 50:24). The WTFI instruction initiates the Wait sequence. The Wait sequence continues to reinitiate itself, thus stopping the program. This sequence is only terminated by a special or monitor interrupt as it initiates the interrupt sequence. This instruction causes the program to wait for an interrupt.

2-575. Execution Sequences. During the I-Sequence, which obtains instruction from memory, the lockout flip-flops are controlled by RIL, EXL, SIL, and SXL instructions. The Wait sequence is initiated by the WTFI instruction and remains in control until occurrence of an interrupt.

2-576. RIL, EXL, SIL, SXL Operations. Refer to logic diagram, plate P-28. The All-Interrupt Lockout flip-flop, 0XG70, is set and cleared by f = 50:34 and f = 50:30 respectively. This flip-flop can also be cleared by the f = 54 instruction, which performs an indirect program jump. The lockout function is accomplished by preventing setting of the Int_i Sequence flip-flop 0XG21 (refer to plate P-12) via 03G70 by a L $\Rightarrow$ Interrupt Lockout. The External Interrupt Lockout flip-flop is set and cleared by f = 50:36 and f = 5-:32, respectively. The lockout function is accomplished by disabling all EI request and EI monitor interrupt gates, which present these signals to the priority logic. The disable signal, L $\Rightarrow$ Lockout Ext Int, is transferred through 3gG71 (refer to plate P-58).

2-577. WTFI Instruction. Refer to logic diagram, plate P-12. The Wait flip-flop, 0XG27, is set by f = 50:24 at time T4.2.

2-578. SPECIAL AND MONITOR INTERRUPT OPERATIONS. The following subparagraphs discuss various interrupts and the special conditions under which they can occur.

2-579. Instruction Fault. This interrupt occurs any time the program attempts to execute an illegal code (f = 00, 01, 77). The program Fault indicator is illuminated and remains lighted until the Master Clear switch has been depressed. This switch can be manipulated during the Run Mode without effecting its register-clearing function.

2-580. Resume-Fault (Intercomputer Time-Out Fault). This interrupt can only occur if a channel is in the intercomputer mode. If the receiving computer does not accept an outputted word, and reply with the ID Acknowledge signal within the allotted time period, this fault condition exists.

2-581. RTC Monitor. This interrupt occurs if enabled by the previously executed f = 50:14 instruction. During each RTC updating operation, the current clock-count is compared with the constant contained in control memory address 00014_8. If they are equal, this interrupt is generated.

2-582. External Sync. This non-channel interrupt is sent from an external device. This signal could be used for a timing input to synchronize the RTC or as a control function to cause a program switch to perform some particular operation.

2-583. RTC Overflow. This interrupt occurs whenever the RTC updating operation causes the clock-count to advance from 777777_8 to 000000.

2-584. External Interrupt Monitor (EI Mon). This interrupt occurs whenever an EI request occurs. As is the EI request, the EI Mon is locked out if the External Interrupt Lockout flip-flop is set. This flip-flop is program set and cleared. Each of the 16 channels can generate an EI Mon interrupt.

2-585. External Function Monitor (EF Mon). This interrupt occurs if requested when an EF buffer terminates, that is, if all words have been outputted as specified by the TACW and CACW. The buffer monitor function and interrupt is enabled by $CACW_{16}$= 1. Each of the 16 channels can generate an EF Mon interrupt.

2-586. Output Data Monitor (OD Mon). This interrupt occurs if requested when an OD buffer terminates; i.e., if all words have been outputted as specified by TACW and CACW. The buffer monitor function and interrupt is enabled by $CACW_{16}$ = 1. Each of the 16 channels can generate an OD Mon interrupt.

2-587. Input Data Monitor (ID Mon). This interrupt occurs if an ID buffer terminates. The interrupt must be requested by $CACW_{16}$= 1. Each of the 16 channels can generate an ID Mon interrupt.

2-588. Honoring Operations. Honoring an interrupt involves execution of a program jump to a special address which is determined by the interrupt type and in the case of a monitor interrupt, by the channel number. Refer to table 2-164 for jump addresses. The jump address is placed in S1 at the beginning of the next I-sequence. The normal transfer of P → S1 is disabled. Also, the normal program incrementing of P by +1 is prevented. Therefore, the instruction from the jump address is executed rather than the next sequential instruction. When execut

TABLE 2-164. SPECIAL AND MONITOR INTERRUPT JUMP ADDRESSES

	Interrupt Type	Jump Address
Special Int. (Non-channel type)	Instruction Fault	00000 (00500_8 if Auto Recovery)
	Resume-Fault	00011_8
	RTC Monitor	00012_8
	RTC Overflow	00013_8
	External Sync	00016_8
Monitor Interrupts (Channel type)	EI Monitor	00100_8* + 2x chan
	EF Monitor	00120_8* + 2x chan
	OD/Monitor**	00140_8* + 2x chan
	ID Monitor**	00160_8* + 2x chan

* Add 00200_8 to jump address if channel is $10\text{-}17_8$. Interrupts are listed according to priority.

**Priority of OD and ID-monitor interrupts may be reversed according to the Priority Alternator flip-flop.

due to an interrupt, the instruction at the interrupt address can only use the U portion of its word for an address. That is, it cannot formulate U_p or U_{SR}. However, XU can be formulated. Usually this instruction performs a jump to an interrupt routine.

2-589. If the instruction at the jump address does not alter P (not a jump instruction), the computer will return to the normal program sequence and continue from the interrupted point. These interrupt operations are controlled by the interrupt sequence which is initiated so it will run in parallel with the I-sequence.

2-590. Except in the case of the instruction fault, the interrupt will be ignored (interrupt sequence will not be initiated) if the All Interrupt Lockout flip-flop is set. This flip-flop can be program set and cleared to allow the program to determine when to be interrupted. This flip-flop is also set for an interrupt during honoring operations. Thus, after each interrupt is honored, the Lockout flip-flop must be program cleared if another interrupt is to be detected.

2-591. Except in the case of EI Mon interrupts during the interrupt/I-sequence, which honors the interrupt, the flip-flop which was set to generate the interrupt is cleared. The EI Mon flip-flop has a dual function. It generates the EI monitor interrupt signal and until it is cleared, it also disables the recognition of any future EI request on its channel. This prevents another possible status word being brought into memory and

destroying the current status word. The program is not aware of the existence of the status word in memory until the EI monitor interrupt occurs. Until the EI Monitor flip-flop is cleared, the status word will be retained, and the program can operate on this word at any time. This flip-flop is cleared at the same time that the program clears the All Interrupt Lockout flip-flop.

2-592. Special Interrupt Translator. Refer to logic diagrams, plate P-55. This translator selects the special interrupt to be honored according to a hard-wired priority, and also formulates the jump address to be placed in S1. If the Instruction-Fault interrupt is being honored, nothing is placed in the translator register. Thus, the jump address is 00000. A Set Translator flip-flop inserts a 1_2 in the jump address. Interrupts are listed according to priority. Refer to table 2-165 for translator conditions affected by special interrupts. The gates listed under other conditions in table 2-165 enable the correct interrupt Generation flip-flop to be cleared when it is honored.

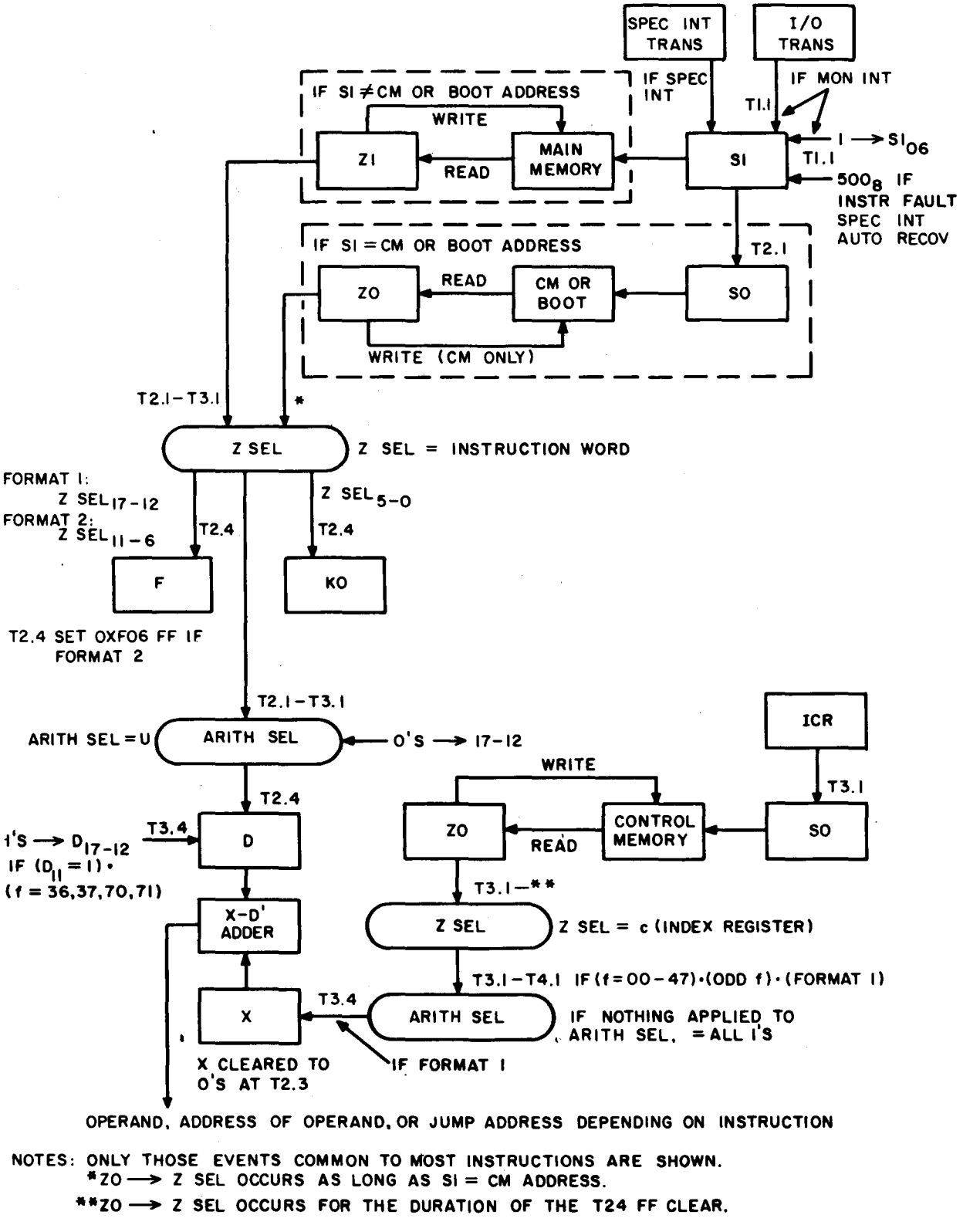
2-593. Interrupt/I-Sequence Data Flow. Refer to figure 2-147 for a block diagram description of the honoring operations for special and monitor interrupts. The translators are set according to the type of signal and its channel number (monitor interrupts only) during the Scan sequence. The instruction for the I-sequence is obtained from the address placed in S1. P is not advanced and therefore retains the address of the instruction which would normally have been executed if the interrupt had not occurred. Execution of the instruction is normal except that the values U_p and U_{SR} cannot be formulated. If this instruction does not alter P (program jump), the interrupted program will continue after completion of this instruction. Usually, the instruction at the interrupt address performs a return jump to an interrupt routine and retains P, so that a subsequent return to the interrupted point in the program can be affected.

2-594. Scan and Interrupt/I-Sequence Essential Commands. Refer to table 2-166 for a list of essential Scan and Interrupt/I-Sequence events. Only those commands which differ from normal I-Sequence operations are listed. Commands not described previously are described in table 2-167 in detail.

TABLE 2-165. SPECIAL INTERRUPT TRANSLATOR CONDITIONS

Special Int. Type (Plate P-54)	Translator Register ff's				Jump Address	Other Conditions
	2XG18	2XG17	2XG16	2XG15		
Instr Fault	clear	clear	clear	clear	00000*	
Resume-Fault	set	clear	clear	set	00011_8	23G18 = L
RTC Monitor	set	clear	set	clear	00012_8	23G15 = L
RTC Overflow	set	clear	set	set	00013_8	23G17 = L
Ext Sync	set	set	set	clear	00016_8	23G16 = L
	$S1_{03}$	$S1_{02}$	$S1_{01}$	$S1_{00}$		
	Corresponding S1 Bits					

*Instruction fault jump address is 00500_8 if AUTOMATIC RECOVERY switch is on. A separate source supplies S1 with this address.



NOTES: ONLY THOSE EVENTS COMMON TO MOST INSTRUCTIONS ARE SHOWN.
*ZO → Z SEL OCCURS AS LONG AS SI = CM ADDRESS.
**ZO → Z SEL OCCURS FOR THE DURATION OF THE T24 FF CLEAR.

Figure 2-147. Interrupt/I-Sequence Data Flow

TABLE 2-166. SCAN AND INTERRUPT IN PARALLEL WITH I-SEQUENCE ESSENTIAL COMMANDS FOR SPECIAL AND MONITOR INTERRUPTS

Time Notation	Commands	Spec Int	Mon Int
	SCAN SEQUENCE		
T2.1	Clear Spec Int Trans Req	X	
T2.2	Chan Req → Chan Pri		X
T2.4	Clear Chan Pri, Clear Fct Pri		X
T3.1	Mon Req → Chan Pri		X
T3.2	Drop Chan Req → Chan Pri		X
T3.4	Mon Req → Fct Pri		X
	Set Instr Fault ff and Prog Fault ff if f = 00, 01, 77	X	
	Init Load Mode and disable Run Mode if (auto recov on)•(Instr Fault ff set)	X	
T4.1	Clear Fct, Chan, and I/O Trans 1		X
	Clear Spec Int Trans Req	X	
T4.2	Pri → Fct, Chan, and I/O Trans 1*		X
	Set Int_i ff if (all Int Lockout ff clear + Instr Fault spec int)•(I Seq next)	X	X
	Set I_i ff if I Seq next	X	X
T4.3	Spec Int Req → Spec Int Trans Req if $\overline{\text{Instr Fault spec int}}$	X	
T4.4	Clear S1	X	X
The following occurs if Int_i ff set; otherwise, normal I-Seq follows			
	INTERRUPT AND I-SEQUENCE IN PARALLEL		
T1.1	I/O Trans → S1*, 1 → $S1_{06}$*, Init Memory		X
	Spec Int Trans → S1, Init Memory	X	
	00500_8 → S1 if Load Mode (Instr Fault• auto recov)	X	
	Disable P → S1 and Adv P Subsequence	X	X
T1.2	S1 → P if Load Mode (Instr Fault · Auto Recov)	X	

TABLE 2-166. SCAN AND INTERRUPT IN PARALLEL WITH I-SEQUENCE ESSENTIAL COMMANDS FOR SPECIAL AND MONITOR INTERRUPTS (Cont)

Time Notation	Commands	Spec Int	Mon Int
T1.3	Clear Instr Fault ff, clear Spec Int ff just honored	X	
	Clear EF, OD or ID Mon ff just honored		X
	Set 7XMg ff if EI Mon** just honored		X
T1.4	Set All Int Lockout ff	X	X
T2.1	S1 → SO and Init CM/boot if S1 = CM or boot address	X	X
	Set Bootstrap ff if S1 = $00500\text{-}00537_8$		X
T2.2	Set Int_f ff, set I_f ff	X	X
T3.4	Disable SR → D_{15-12} and P_{15-12} → D_{15-12}	X	X
T4.1	Clear Int_i ff, clear I_i ff	X	X
T2.1	Clear Int_f ff, clear I_f ff	X	X

*These commands occur if there is no special interrupt; thus, special interrupts have higher priority than monitor interrupts.

**If honored, EI Mon ff not cleared until program clears All Interrupt Lockout ff. EI Mon ff to be cleared is selected by 7XMg ff's. EI Mon is not detected unless the EI Lockout ff is clear. Except Advance-P subsequence, P → S1, SR → D_{15-12}, and P_{15-12} → D_{15-12}, all normal I seq events occur (not shown). Refer to table 2-168 for a detailed description for EI Mon operation.

TABLE 2-167. SCAN AND INTERRUPT/I-SEQUENCE COMMANDS/DESCRIPTIONS
(For Detailed Description of all Commands Previously Described Refer to Table 2-110)

Time Notation	Command/Description
T3.4	Set Instr Fault ff and Prog Fault ff if f = 00, 01, 77. Instr Fault flip-flop 1XG14 (refer to plate P-54) is set at ∅4 by a L ⇒ f = 00, 01, 77 and a L ⇒ T34 I Seq. A L ⇒ Program Fault from 1XG14 will set Program Fault flip-flop 9XG14. Init Load Mode and Disable Run Mode if (Auto Recov on) • (Insts Fault ff Set). A L ⇒ Fault from Inst Fault flip-flop (refer to plate P-54) is applied to 00J03 (refer to plate P-3). With AUTO RECOVERY switch S13 to the ON (UP) position, a H ⇒ $\overline{\text{Remote}}$ is applied to 90Y03. The low output will cause 00J03 to output a high to 00J00 which disables Run Mode, and initiates a L ⇒ Load Mode via 01J03.
T4.2	Set Int_i ff if (All Int Lockout ff Clear + Instr Fault Spec Int) • (I Seq Next). If All Interrupt Lockout flip-flop OXG70 (refer to plate P-28) is clear, the low output is inverted by 03G70 and applies a H ⇒ $\overline{\text{Interrupt Lockout}}$ to 21G21 (refer to plate P-12). If Instr Fault flip-flop (refer to plate P-54) is set, a H ⇒ Inst Fault is applied to 21G21. Other inputs to 21G21 are a H ⇒ Spec Int, the absence of a L ⇒ B1 Seq and high outputs from disabled gates 20G20 and 23G20. The low from 21G21 sets Int_i flip-flop OXG21 at time T4.2, with the low from 12E20.
T4.3	Spec Int Req → Spec Int Trans Req if $\overline{\text{Instr Fault Spec Int.}}$ If Inst Fault flip-flop (refer to plate P-54) is cleared, a L ⇒ $\overline{\text{Inst Fault}}$ from the clear side is applied to 21E15 (refer to plate P-55), which outputs a low to 2XG15 through 2XG18 due to a low from 03T43 at ∅3. The high from either Resume Fault, RTC Mon, Ext, Syn, or RTC Overflow flip-flop is inverted by 15G14 and applies a L ⇒ Spec Int Req to 2XG18. Clear S1. Gate 70N12 (refer to plate P-22) is enabled by a L ⇒ Int Seq, L ⇒ Spec Int Seq and a low from 03T11. The high output is inverted by 08N12 and at ∅4, 09N12 produces a L ⇒ Clear S1. INTERRUPT AND I-SEQUENCE IN PARALLEL
T1.1	I/O Trans → S1. Gate 30N12 (refer to plate P-22) is enabled by a L ⇒ Int Seq, a low from 70N12 (which is disabled by the absence of a L ⇒ Spec Int Req) and a low from 03T11. The high output is inverted by 38N12 and at ∅1, 39N12 produces a L ⇒ I/O Translator → S1 (11S04, 11S05 and 11S07). Spec Int Trans Reg → S1. Gate 70N12 (refer to plate P-22) is qualified by a L ⇒ INT Seq, L ⇒ Spec Int Req and a low from 03T11. The high is inverted by 36N12 and applied to 37N12 which at ∅1 produces a L ⇒ Translator → S1 + Special Interrupt Reg → S1 to 1XS01 through

TABLE 2-167. SCAN AND INTERRUPT/I-SEQUENCE COMMANDS/DESCRIPTIONS (Cont)

Time Notation	Command/Description
T1.1 (Cont)	1XS03 (plate P-104). The high from 70N12 is also inverted by 78N12 and applies a L ⇒ Special Interrupt Req → S1 to 25G16 through 25G18 to insure the gates will not be qualified unless the appropriate ff is set (plate P-55). The low from 78N12 is also inverted by 79N12 and applies a high to gates 25G16 through 25G18 which will be enabled if the appropriate ff 2XG16 through 2XG18 is set. 1 → $S1_{06}$. The high from 30N12 (refer to plate P-22) is also applied to 35N12 and with the absence of L ⇒ Cont Data Seq, 35N12 outputs a L ⇒ Set S1 = 100 to set 1XS06 (refer to plate P-104) at ∅1. 00500_8 → S1 if Load Mode (Instr Fault. Auto Recov). Gate 60N12 (refer to plate P-22) is enabled by a L ⇒ Load Mode and a low from 03T11. The high output is inverted by 68N12 to produce a L ⇒ Set S1 = 400 to 1XS08 (refer to plate P-105). Disable P ⇒ S1. The high from 60N12 (refer to plate P-22) is also applied to disable 29N12, which disables P → S1. Disable Adv P Subsequence. Gate 10L10 (refer to plate P-35) is disabled by the absence of a L ⇒ Int Seq and the low output is inverted by 11L10 to prevent Increment P flip-flop OXL10 from setting, which prevents OXL11 from setting, disabling the advance P subsequence.
T1.2	S1 → P if Load Mode (Instr Fault · Auto Recov). Gate 20N07 (refer to plate P-21) is enabled by a L ⇒ I Seq, a L ⇒ Load Mode and a low from 01T12. The high output is inverted by 28N07 and applied to 29N07. The high from 20N07 is also inverted by 07N07 to enable 08N07 at ∅2. The high output is inverted by 11N07, via 09N07 and applied to 29N07 to produce a L ⇒ S1 → P.
T1.3	Clear Instr Fault ff. Clear Monitors gate 00N41 (refer to plate P-53) is enabled by a L ⇒ Int Seq and a low from 01T13. The high output is inverted by 02N41 to produce a L ⇒ Clear Fault to 03N41 (refer to plate P-54) which outputs a low at ∅3, to clear Instr Fault flip-flop. Clear Spec Int ff Just Honored. The low from 03N41 (refer to plate P-54) is also applied to flip-flops 1XG15 through 1XG18 to clear the Special Int flip-flop just honored due to the output of one of the gates 23G15 through 23G17 in the Special Interrupt Translator (refer to plate P-55). One of these gates will be enabled due to the condition of flip-flops 2X615 through 2X618. Refer to table 1-169 for the condition of the flip-flops.

TABLE 2-167. SCAN AND INTERRUPT/I-SEQUENCE COMMANDS/DESCRIPTIONS (Cont)

Time Notation	Command/Description
T1.3 (Cont)	Clear EF, OD or ID Mon ff Just Honored. The high from 00N41 (refer to plate P-53) is inverted by 01N41 and produces a L ⇒ Clear Monitor to 9gN41 (refer to plate P-56) and the high output is applied to 4gN41, 6gN41 and 5gN41 (refer to plate P-57). Depending upon ID, OD or EF, one of the three gates is enabled and will produce either a L ⇒ Clear ID, OD or EF Monitor. Set 7XMg-ff if E1 Mon Just Honored. The high from 9gN41 (refer to plate P-56) is also applied to 7gN41, which has as the other input a H ⇒ Ext Int, and a L ⇒ Clear EI Monitors is applied from 7gN41 to set one of the 7XMg flip-flops (refer to plate P-38).
T1.4	Set All Int Lockout ff. All Interrupt Lockout flip-flop 0XG70 (refer to plate P-28) is set by a L ⇒ T14 I Seq and a L ⇒ Int Seq at ø4.
T2.1	S1 → S0 and Init CM/Boot if S1 = CM or Boot Address. Gate 30N10 (refer to plate P-24) is enabled by a L ⇒ I, RI, W or I/O, L ⇒ S1 = Cont or Bootstrap Memory Address and a low from 03T21. The high output is inverted by 38N10 and at ø1, 39N10 produces a L ⇒ S1 → S0 to the S0-register (refer to plate P-114); to Control Memory (refer to plate P-122) and to Bootstrap flip-flop OXS09 (refer to plate P-107). Set Bootstrap ff if S1 = 00500-00537$_8$. If S1-register = 00500-00537$_8$, all inputs to 90S05 and 90S09 (refer to plate P-107) will be low. The high outputs are applied to 91S05, which outputs a low and with a L ⇒ S1 → SO, Bootstrap flip-flop OXS09 will set.
T3.4	Disable SR → D_{15-12} and P_{15-12} → D_{15-12}. Gates 40N02 and 31N02 (refer to plate P-18) are disabled by the absence of a L ⇒ Int Seq · B Seqs, I/O, or Hold 2. The low outputs are inverted by 48N02 to disable 49N02, which disables SR → D_{15-12}, and 37N02, which in turn disables 39N02, and thus disables P_{15-12} → D_{15-12}.

2-595. EI Monitor Operations. EI monitor operations differ from those for other interrupts. The clearing of the EI monitor flip-flop is not performed until the program clears the All Interrupt Lockout flip-flop. Also, the EI monitor flip-flop disables detection of the EI request signal on its channel until it is cleared. Both EI monitor and EI request can be locked out by the External Interrupt Lockout flip-flop. Refer to figure 2-148 for a simplified diagram of the EI request and monitor logic on one channel. Operations are initiated by occurrence of the EI request signal.

2-596. EI Monitor Operation Essential Commands. Commands not previously described are described in table 2-168 in detail.

2-597. INTERCOMPUTER OPERATIONS. The I/O data and control transfer technique is altered for intercomputer operations. It is necessary to consider the Resume-Fault interrupt, output available condition, special programming, and ODR/EFR logic configuration.

2-598. Intercomputer Channels. Intercomputer communications can be set up on any of the 16 channels. Except for ODR operations, the I/O functions are normal. Setting up the special intercomputer ODR logic configuration requires shifting the CHANNEL INTERCOMPUTER switch, for the particular channel, to the UP position. The I/O cable connections must be reversed from normal. The input cable from one computer must be connected to the output cable terminals of the other computer and vice-versa. The cross cabling results in the output data lines from one computer appearing as input data line to the other, and some of the acknowledge signals being considered as request, see figure 2-149 for the intercomputer cable description.

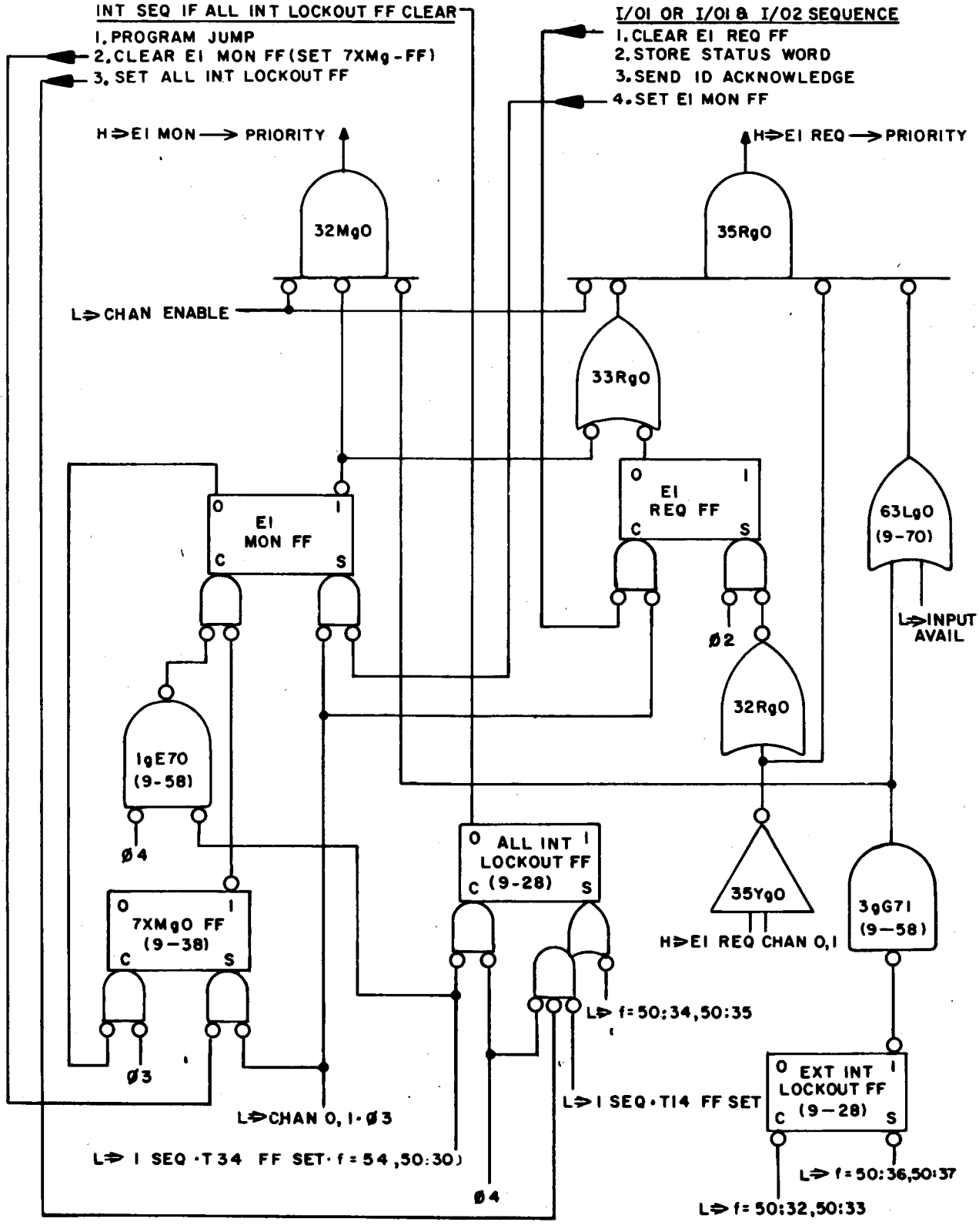
2-599. External Function/External Interrupt Exchange. Refer to figure 2-149. If Computer B does not have the capability to generate an External Function Request, then an EF word and a EF Acknowledge from computer A would appear to computer B as a status word and an External Interrupt Request. Since the EFR line of computer A does not have a logic connection at Computer B, Computer A could execute control over the program of computer B, by means of the EF/status word bit configuration. This coded word could cause the receiving computer by program control to set up an ID buffer to accept data and specify how to process this data. If Computer B does have the capability to generate an External Function Request, Computer A will handle the EFR in the normal manner. The EF Request will also clear the OD Request flip-flop. This is done since the EF Ack sent out by computer A is interpreted as an EI Request. This causes Computer B to send out ID a Ack which is seen by Computer A as a ODR.

2-600. Data Exchange. Refer to figure 2-149. During intercomputer data exchange operations, an output data word and OD Acknowledge from computer A appears to computer B as an input data word and IDR. As computer B accepts each word, it responds with the ID Acknowledge, which appears to computer A as the ODR. This ODR indicates that computer B has accepted the last data word and is a request for the next data word. The transfer process is initiated in the special intercomputer ODR logic of computer A. The setup of the OD buffer by the OUT instruction (f = 50:12) sets the OD Active flip-flop for the particular

channel. The Active flip-flop simulates the first ODR. Thus, the buffer instruction forces out the first data word for intercomputer operation. Each word after the first is not transferred until computer A receives the ID Acknowledge (ODR) from acceptance of the previous word. If computer B does not respond with the ID Acknowledge, within a specified length of time after a data word has been sent to it, computer A will notify its own program by generating the Resume-Fault interrupt.

2-601. ODR Logic. Refer to paragraph 2-574 for intercomputer logic description. Setting of the OD Active flip-flop at buffer initiation simulates the first ODR to start the data-transfer process. During the sending of the associated OD acknowledge, the Resume flip-flop is cleared. It is not set until the ID Acknowledge (ODR) from the data-receiving computer is received. Setting the Resume flip-flop again satisfies the ODR gate to present this signal to priority to be honored for the next word. This process continues until the OD buffer terminates. As each word is outputted, the corresponding OD Acknowledge is not dropped until the ID Acknowledge is received to set the Resume flip-flop. Therefore, while a word is being sent to the data-receiving computer waiting for the acknowledge, output is not available for the particular chassis. When the last word of the OD buffer is outputted, the OD Active flip-flop is not immediately cleared as for normal channel operation. It remains set until the corresponding last ID Acknowledge (ODR) is received. It is necessary for the Active flip-flop to remain set to allow last ODR to be recognized and to set the Resume flip-flop. Setting the Resume flip-flop enables clearing of the OD Active flip-flop.

2-602. Resume-Fault Interrupt. Refer to paragraph 2-580 for a description of the resume-fault interrupt. The Resume-Fault flip-flop is set if the Resume flip-flop, on any of the four I/O chassis, is in the clear state for longer than a specified length of time. For normal channel operation, this flip-flop is set for the approximate durati of the EF or OD Acknowledge. Therefore, intercomputer operation is the only case whereby the flip-flop could remain clear for a relatively long period of time. The Resume-Fault condition exists if the data-receiving computer does not set the Resum flip-flop in time with its ID Acknowledge (ODR). Upon detection of a Resume-Fault interrupt, the program should set the Resume flip-flop so as to provide the output available condition for the particular chas The absence of this condition prevents any OD or EF activity for any channel on the chassis. The Resume flip-flop is set by the SRSM instruction (f = 50:20). The interrupt is timed by the RTC. Thus, it is necessary that the clock not be disabled by the RTC DISCONNECT switch. Otherwise, the program would not be notified of the fault condition and output would remain unavailable for the chassis.

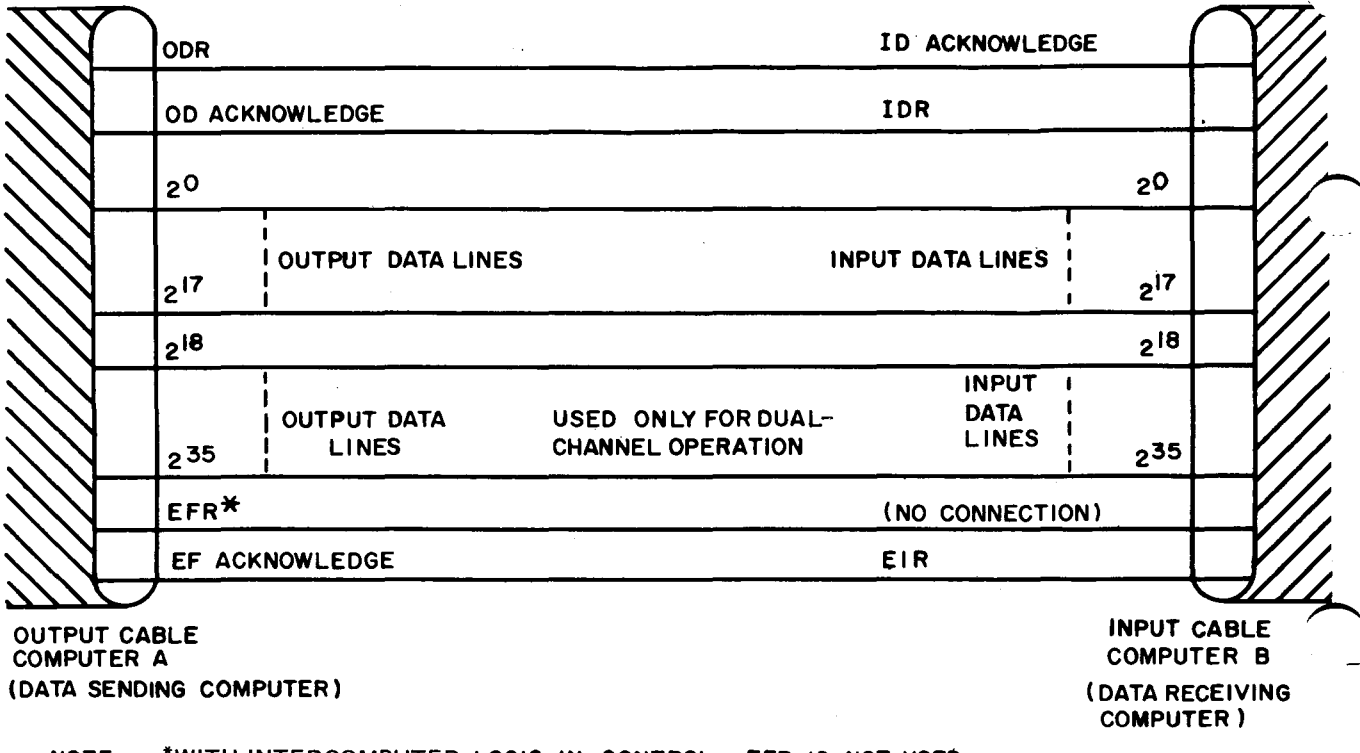


NOTE: UNLESS OTHERWISE NOTED, CIRCUITS SHOWN ARE IN LOGIC DIAGRAMS, PLATE P-61

Figure 2-148. EIR and EI Mon Interaction, Simplified Logic Diagram

TABLE 2-168. EI MONITOR OPERATION COMMAND/DESCRIPTION

Time Notation	Command/Description
T3.4	When All Interrupt Lockout flip-flop OXG70 is cleared (refer to plate P-28), the L⇒ Clear EI Monitor from 01E70 is also applied to 1gE70 (refer to plate P-58) which applies the L⇒ Clear EI Monitor to the EI Monitor flip-flop (refer to plates P-61 through P-64). The other input, L⇒ Clear EI Monitor Channel (channel number) is applied from one of the 7XMg flip-flops (refer to plate P-38) to clear EI Monitor flip-flop. Until EI Monitor flip-flop is cleared, 33Rg0 will output a high to disable 35Rg0 thus disabling detection of the EI request signal. When External Interrupt Lockout flip-flop OXG71C (refer to plate P-28) is set, a L⇒ Lockout External Interrupt is applied to 3gG71 (refer to plate P-58) to send a H⇒ $\overline{\text{Enable EI Monitor}}$ to disable 32Mg0 (refer to plate P-61). The high from 3gG71 is also applied to 63Lg0 (refer to plate P-70) which outputs H⇒ $\overline{\text{Input Avail}}$ to disable 35Rg0 (refer to plate P-61). Both EI Monitor and EI Request are locked out. Refer to logic diagram plate P-64 for the external-sync logic. The external-sync logic is similar to that for a data request. Flip-flop OXG16 provides a one-shot function for the signal to allow it to set the external-sync flip-flop once. This interrupt can be disabled by the DISCONNECT EXTERNAL SYNC switch.



NOTE: *WITH INTERCOMPUTER LOGIC IN CONTROL, EFR IS NOT USED.

Figure 2-149. Intercomputer I/O Cabling

Section 2-3. Detailed Functional Description (Power Distribution)

2-603. POWER DISTRIBUTION

2-604. GENERAL. The computer Power Supply requires 115 VAC 400 Hz 3-phase regulated input. Wattage requirements vary with computer configuration. Power Supply, PS1, is located immediately below the logic and memory drawers, in the lower cabinet area. PS1 furnishes most of the voltages (refer to table 2-3) required for operation of computer logic circuitry, Additional power supplies, located in each memory chassis of memory drawer A3, provide DC voltages (refer to table 2-4) required for operation of computer memory circuitry. Power control panel A5, located in the upper cabinet area immediately above the logic drawers, contains the primary switching and indicating facilities for computer power application and monitoring (refer to plates P-176 through P-180 located in Volume 2 Part 3 of this OP, for the functional schematic diagrams relating to power control and distribution.)

2-605. POWER CONTROL CIRCUIT. The 115 VAC 400 Hz 3-phase input is applied through a single four-conductor cable to line filters FL1, FL2, and FL3 (refer to plate P-177). From these filters power is applied via fuses F8, F9, and F10 (refer to plate P-178) to the contracts A1, B1, C1, and of power control relay K1. The POWER switch A5S4 (refer to plate P-176), located on the power control panel, is a spring-loaded, two position toggle switch shown in the center position. Momentarily setting switch to the ON position provides ØA 115 VAC as an enable K1 Power ON signal, via the temperature and drawer interlock switches (refer to plate P-177), to the coil of K1 (refer to plate P-178). Holding contacts 11-12 of K1 (refer to plate P-178), apply power to K1, via the POWER switch and interlocks, allowing K1 to remain energized after the POWER switch is released. Positioning the POWER switch to OFF opens the holding circuit causing K1 to deenergize. Interlock switch A12S1 (refer to plate P-177) located on the rear of power supply PS1, breaks the power-on circuit when the power supply drawer is extended. The temperature sensing switches A6S3 and A13S6 (if installed), are located within the cabinet interior. These switches deenergize relay K1 if the temperature within the cabinet exceeds 140 degrees F (50 degrees C). The BATTLE SHORT switch, A5S2 (refer to plate P-176), bypasses the temperature and drawer interlocks.

2-606. MAIN DC POWER SUPPLY AND DISTRIBUTION. The main DC power supply, located in PS1 and shown in plate P-178, generates the DC power required for general logic operation. This circuitry is composed of three basic supplies and furnishes +4.5 VDC, -4.5 VDC, -15.0 VDC, and +15.0 VDC. The +4.5 VDC is used only by the main memory power supply. Transistors Q1 through Q4 provide regulation for the +25 VDC and -25 VDC regulated outputs used in the control memory circuitry.

2-607. -15 VDC Supply. The 115 VAC 400 Hz 3-phase is applied via fuses F1 through F3 to transformer T1 (refer to plate P-178). The voltage is then rectified by diodes CR1 through CR6 and filtered by inductor L1 and capacitor

C3. The -15 VDC is then distrubuted by cable W3 to each chassis (refer to plate P-180). Diode CR3 on each chassis provides voltage protection and fuses provide current protection. Noise on the -15 VDC bus is bypassed to ground by capacitor C2 of the 3180 capacitor assemblies in each row of the logic chassis. The -15 VDC is also provided to the power control panel A5 (refer to plate P-176) for operating the fault horn and lights.

2-608. +15 VDC Supply. Diodes CR7 through CR12 rectify the output of T3 to provide the +15 VDC which is filtered by L3 and C2, and distributed by W4 (refer to plate P-180) in the same manner that the -15 VDC is distributed, using CR2 on each chassis for voltage protection and C3 of the capacitor assembly for noise filtering. Diodes CR25 and CR26 also provide protection. The +15 VDC is also applied via the IND or IND-SET position of switch A5S1 (refer to plate P-176) to provide the voltage source for the indicator lights on the logic drawers A1, A2, A4, or A8. Noise suppression is provided by C1 of the 9010 capacitor-resistor assembly at J35G on chassis A1A2, A4A1, and A8A2.

2-609. Transformer T3 provides unregulated -25 VDC, via CR19 through CR21 and +25 VDC, via CR22 through CR24. These voltages are filtered by chokes that are part of T3; and by C4, C6 and C8, C9 respectively. The -25 VDC is regulated by transistors Q1 and Q3 which are controlled by regulator-amplifier 11BA10 (refer to plate P-134).

2-610. Card 11BA10 provides feedback to the regulating transistors to compensate for variations in load current and temperature. The regulated output is filtered by C5 (refer to plate P-178) and provided to control memory via fuse F7. The +25 VDC is regulated the same way by transistors Q2 and Q4, regulator amplifier 12BA10 (refer to plate P-134), and C7 and supplied via F11.

2-611. ±4.5 VDC Supply. The -4.5 VDC is provided from transformer T2 by diodes CR15 through CR18, filtered by L2 and C1 and distributed by W2 (refer to plate P-180) in the same manner as the +15 VDC and -15 VDC is distributed. Zener diode CR1 on each chassis provides the voltage protection and C1 on the capacitor assemblies provides filtering. The +4.5 VDC from T2 and diodes CR27 through CR32 is used by the main memory power supplies.

2-612. MAIN MEMORY POWER SUPPLY. The main memory power supplies, located on each memory (A3A2, and A3A1 if applicable) and shown in plate P-179, produce voltages unique to memory operation. Output voltages from this supply are +3.0, -3.0, -4.5, +6.0 and ±15.0 VDC.

2-613. +6 VDC Power Supply. (Refer to plate P-179), The 115 VAC 400 Hz 3-phase is applied through F1, F2, and F3 to the primary of T1. The output to T1 is rectified by CR1 through CR6, filtered by L1, C6, C8 and C9, fused by F7, and protected by Zener diode CR9. The output (+6 VDC) is further filtered by capacitor assemblies in the 660 cards in the memory chassis (2 cards for every bank).

2-614. +3 VDC Supply. The +3 VDC is provided from the +4.5 supply through R2 and F8 (refer to plate P-179). Shunt regulator Q1 changes the current through R1 and R2 to maintain the +3 VDC. The output of Q1 is controlled by adjustable voltage regulator 00VR03.

2-615. The +3 VDC output is sampled by an error amplifier and compared to a reference voltage. Any difference between the reference

ana _gulator output is amplified by four stages and used to drive the shunt regulator. An increase in the output of the +3 VDC power supply results in more conduction of the shunt regulator to lower the output. A ease in the output of the +3 VDC power supply results in decreased conduction of the shunt regulator to raise the output. The output is filtered by capacitors C3 through C5 and protected by F8 and CR10. The 660 cards provide further filtering.

2-616. ±15 VDC Supply. The ±15 VDC is provided from the main power supply via fuses F4 and F5. The outputs are filtered by acitors C2 and C1 and protected by diodes CR8 and CR7. The 660 cards also provide additional filtering.

2-617. -4.5/-3 VDC Supply. The -4.5 VDC is provided by the main power supply via F6 and protected by Zener diode CR11. The -3 VDC is provided by the diode-resistor network CR12, CR13 and R3, and is fused by F9.

2-618. MISCELLANEOUS CIRCUITS. A running-time meter M1 (refer to plate P-176) records the total time that power is applied through K1 to energize the computer power supplies. A fault alarm LS1 provides an audible indication of a program, voltage, or temperature fault. The BATTLE SHORT switch S2 disables horn activation.

CHAPTER 3

OPERATION

3-1. GENERAL INFORMATION

3-2. The computer is basically an automatic device which is capable of communicating (under control of a set of instructions called a program) with peripheral devices or other computers via input and output channels. The computer can perform arithmetic operations such as addition, multiplication, subtraction, division, scale factoring, and shifting. In addition, it can perform logic operations such as complementation, comparison, parity checking, selective setting, clearing, and substitution.

3-3. The program, entered into the computer from a peripheral device such as a magnetic tape unit or a paper-tape reader, is stored in the computer internal memory. The program is a set of sequential instructions employing arithmetic and logic operations, singly or in combination. In executing the program, the computer reads information from, and writes information into, its internal memory. When so programmed, the computer can interrupt program execution to accept data from peripheral equipments or to output data to readout equipments. Program processing is conducted during intervals between data transfers.

3-4. Operation of the computer consists of loading a program into internal memory, inspecting and altering programs, manually reading or writing into single-memory or consecutive-memory addresses, and observing readout information.

3-5. CONTROLS AND INDICATORS

3-6. Although the computer performs its functions automatically in accordance with program instructions, there are certain controls which affect computer operation. These controls provide the means for setting and clearing registers and selecting operating speeds, optional program jumps or stops, and input and output modes. The controls are located on the front panels of each drawer. The panels are designated: Input/Output Panel (A1 or optional A8), Control Panel 1 (A2), Memory Panel (A3), Control Panel 2 (A4), and Power Control Panel (A5). All front panels except A3 are shown in figures 3-1 through 3-4.

3-7. Indicator/switches on the operator panels are associated with the computer register and control flip-flops. These indicator/switches serve a dual function: Depressing the indicator/switch sets the associated flip-flop in the set or "one" condition and the indicator lights when this state is reached. Thus, the indicators provide visual indications of computer status and are intended primarily for use by maintenance personnel.

3-8. Figures 3-1 through 3-4 illustrate the panels which contain the computer controls. These controls include switches, indicators, and pushbutton indicators listed in tables 3-1 through 3-4.

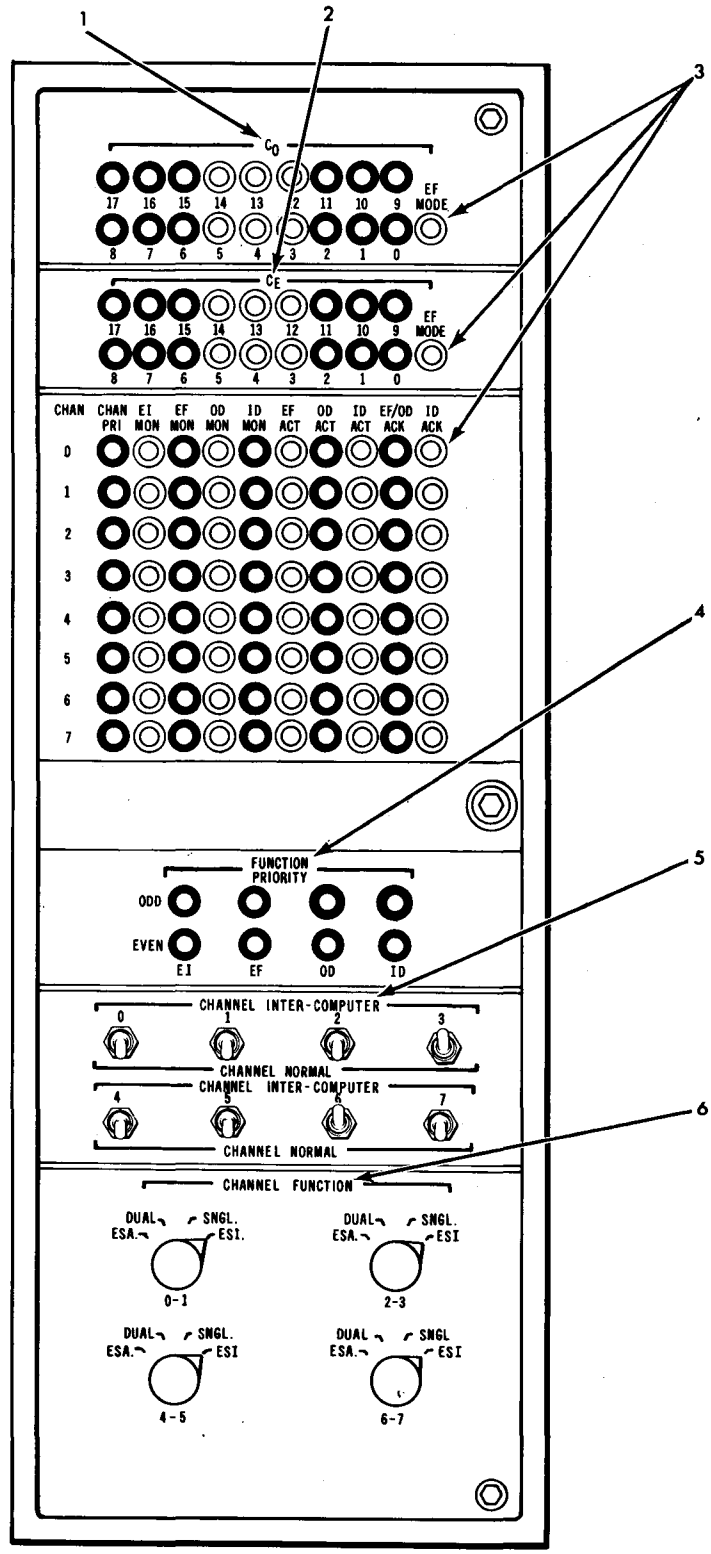


Figure 3-1. Controls and Indicators, Input/Output Panel (A1) or Optional (A8)

TABLE 3-1. CONTROL/INDICATOR FUNCTIONS, INPUT/OUTPUT PANEL (A1 OR OPTIONAL A8)

Figure 3-1 Index No.	Control/Indicator	Type	Function
1	C_O(Register) 0-17	Pushbutton Indicators	Displays contents and allows manual control of 18 bits of communication register common to all odd-numbered output channels. C_O contains upper 18 bits of word in dual channel mode.
2	C_E(Register) 0-17	Pushbutton Indicators	Displays contents and allows manual control of 18 bits of communication register common to all even-numbered output channels. C_E contains lower 18 bits of word in dual channel mode.
3	I/O Channel and Status Grid	Pushbutton Indicators	Indicator/switches indicate function and channel of that coordinate (Channels 0 through 7 or 10 through 17).
	CHAN PRI		Indicated channel is requesting priority.
	EI MON		An external interrupt monitor has been detected and is being processed.
	EF MON		An external function monitor has been detected and is being processed.
	OD MON		An output data monitor has been detected and is being processed.
	ID MON		An input data monitor has been detected and is being processed.

TABLE 3-1. CONTROL/INDICATOR FUNCTIONS, INPUT/OUTPUT PANEL (A1 OR OPTIONAL A8) (Cont)

Figure 3-1 Index No.	Control/Indicator	Type	Function
3 (Cont)	EF ACT	Pushbutton Indicators	External function mode has been made active (able to communicate).
	OD ACT		Output data mode has been made active.
	ID ACT		Input data mode has been made active.
	EF/OD ACK		An external function or output data acknowledge signal has been placed on an output control line.
	ID ACK		An input data acknowledge signal has been placed on an output control line.
	EF MODE		External function mode; when performing any EF/OD function, "on" indicates the EF mode and "off" indicates the OD mode. NOTE: Pressing all above indicator/switches except CHAN PRI will set associated flip-flops.
4	FUNCTION PRIORITY Grid	Pushbutton Indicators	Indicator/switches indicate function and channel group of that coordinate.
	EI (Odd/Even)*		An external interrupt request for priority has been made or is being processed.

*(Odd/Even) indicates odd-numbered or even-numbered I/O channels.

TABLE 3-1. CONTROL/INDICATOR FUNCTIONS, INPUT/OUTPUT PANEL (A1 OR OPTIONAL A8) (Cont)

Figure 3-1 Index No.	Control/Indicator	Type	Function
4 (Cont)	EF (Odd/Even)*	Pushbutton Indicators	An external function request for priority has been made or is being processed.
	OD (Odd/Even)*		An output data request for priority has been made or is being processed.
	ID (Odd/Even)*		An input data request for priority has been made or is being processed. (A request for priority cannot appear simultaneously on odd and even channel group.) NOTE: Pressing FUNCTION PRIORITY indicator/switches will set associated flip-flops.
5	CHANNEL INTER-COMPUTER/CHANNEL NORMAL 0-7	Toggle Switches S1 - S8	Up position: Enables corresponding numbered input and output channels to be used as an intercomputer channel. Down position: Allows normal use of that numbered channel for other peripheral equipment.
6	CHANNEL FUNCTION 0-1 ESA 2-3 4-5 6-7	Switches S9 - S12	Allows the two channels selected to operate in the externally specified address mode (dual mode forced).

*(Odd/Even) indicates odd-numbered or even-numbered I/O channels.

TABLE 3-1. CONTROL/INDICATOR FUNCTIONS, INPUT/OUTPUT PANEL (A1 OR OPTIONAL A8) (Cont)

Figure 3-1 Index No.	Control/Indicator	Type	Function
6 (Cont)	DUAL -	Switches S9 - S12	Connects adjacent input/output channels so they operate in double length (36-bit) mode; for example, switch 0 - 1 in DUAL position allows channels 0 and 1 to operate in dual mode.
	SINGLE		Allows I/O channels to operate in single (18-bit) mode.
	ESI		Allows two channels selected to operate in externally specified index mode (dual mode forced).

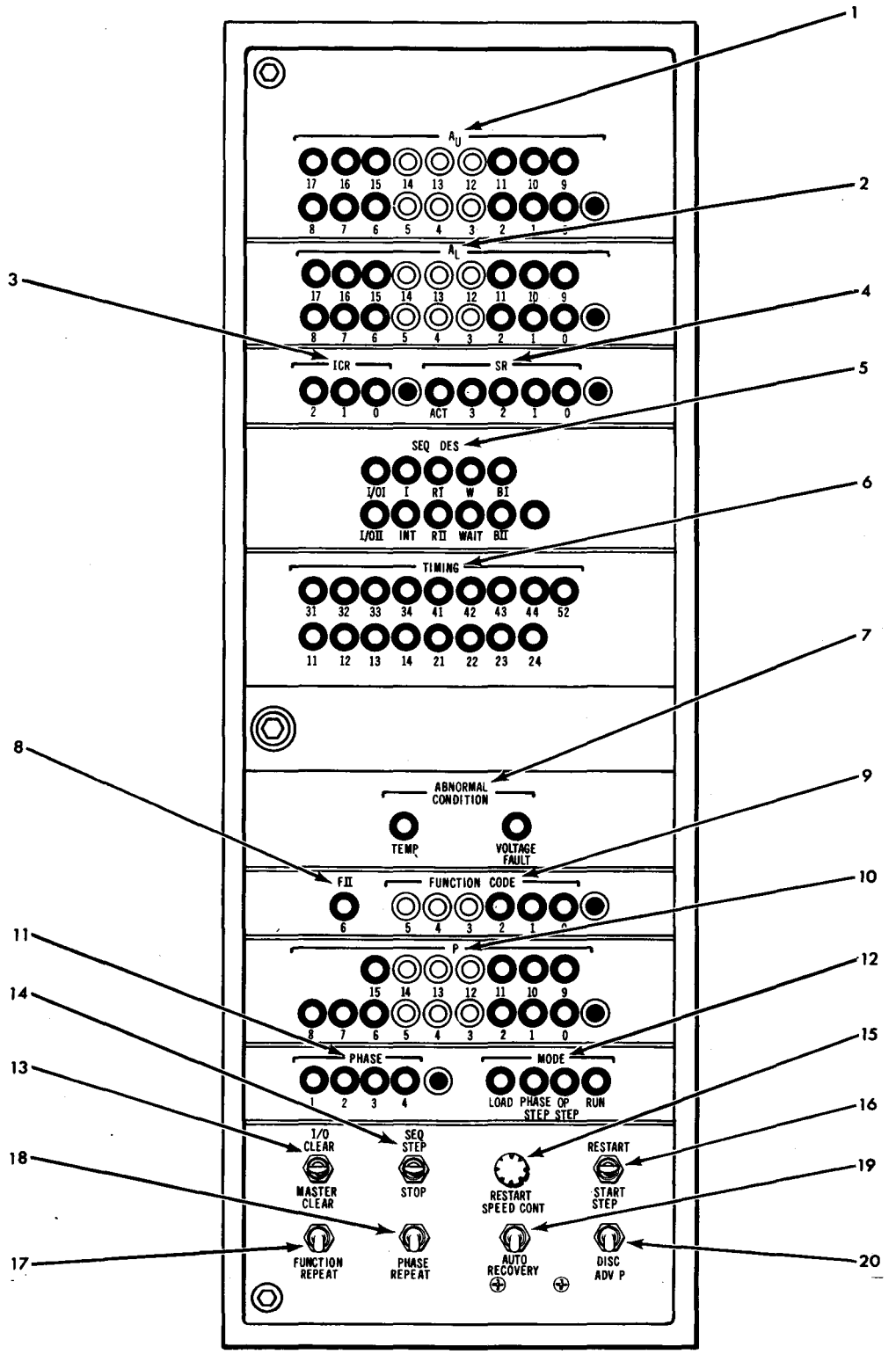


Figure 3-2. Controls and Indicators, Control Panel 1 (A2)

TABLE 3-2. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 1 (A2)

Figure 3-2 Index No.	Control/Indicator	Type	Function
1	A_U (Register) 0-17 and Clear	Pushbutton Indicators	Displays contents and allows manual control of A_U-register (upper 18 bits of A-register). Each bit may be set by pressing appropriate indicator/switch. Clear button clears all 18-bit positions.
2	A_L (Register) 0-17 and Clear	Pushbutton Indicators	Displays contents and allows manual control of A_L-register (lower 18 bits of A-register). Each bit may be set by pressing appropriate indicator/switch. Clear button clears all 18-bit positions.
3	ICR (Index Control Register) 0-2 and Clear	Pushbutton Indicators	Indicates which of index registers is to be used as a modifier. Clear button clears entire register.
4	SR (Special Register) 0-3 and Clear ACT	Pushbutton Indicators	A four-bit register used to specify memory bank currently being used. Clear button clears the register. Comes on when SR is active. When cleared, SR is inactive.

TABLE 3-2. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 1 (A2) (Cont)

Figure 3-2 Index No.	Control/Indicator	Type	Function
5	SEQ DES. (Sequence Designators)	Pushbutton Indicators	
	I/O I		Indicates performance of first I/O sequence.
	I/O II		Indicates performance of second I/O sequence.
	I		Indicates performance of instruction sequence which is common to all instructions. Manually selecting I-sequence clears all other sequences.
	INT		Indicates performance of interrupt sequence.
	R I		Indicates performance of first R (read) sequence.
	R II		Indicates performance of second R (read) sequence.
	W		Indicates performance of W (write) sequence.
	WAIT		Indicates performance of WAIT sequence. (For example, computer waits for an interrupt.)
	B I		Indicates performance of first B-sequence, which reads TACW and stores it in control memory.

TABLE 3-2. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 1 (A2) (Cont)

Figure 3-2 Index No.	Control/Indicator	Type	Function
5 (Cont)	B II	Pushbutton Indicators	Indicates performance of second B-sequence, which reads IACW and stores it in control memory. NOTE: All sequence designator indicators can be manually set.
6	TIMING 11-14 21-24 31-34 41-44 and 52	Indicators	Indicates setting of main timing cycle flip-flops T11 through T52. Thus, as indicators come on and go off, progression of cycle time of computer is indicated. Indicator 52 comes on only during 50:61 and 50:63 instructions.
7	ABNORMAL CONDITION TEMP VOLTAGE FAULT	Indicators	 This indicator comes on when either low temperature thermostat detects an internal air temperature higher than 115°F (46°C). This indicator comes on when any of logic voltages (±5V, -3V) or memory voltage (±10V) fluctuates outside of preset limits.
8	F II (Format 2) 6	Pushbutton Indicators	Bit 2^6 indicator of function code register indicates function code is of Format II type (for example, 50:XX). May be manually set or cleared.

TABLE 3-2. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 1 (A2) (Cont)

Figure 3-2 Index No.	Control/Indicator	Type	Function
9	FUNCTION CODE 0-5 and Clear	Pushbutton Indicators	Bit 2^0 - 2^5 indicators of function code register octally display the function code in F-register. All six bits may be manually set or cleared.
10	P 0-15 and Clear	Pushbutton Indicators	Displays contents and allows manual control of 16 bits of program address register. Each bit may be set by pressing appropriate indicator/switch. Clear button clears all 16-bit positions.
11	PHASE 1-4 and Clear	Pushbutton Indicators	Indicates phase selected. Selecting one phase by pressing indicator/switch enables computer to issue phase pulses in conjunction with phase step mode. Phase pulses are issued individually beginning with selected phase. Pressing Clear button clears all four phases.
12	MODE LOAD	Pushbutton Indicators	Indicates load mode. Pressing MODE LOAD button sets load mode, clears all other modes, and forces a jump to address 00500 for loading at high speed.
	PHASE STEP		Indicates phase step mode. Pressing MODE PHASE STEP button sets phase step mode, clears all other modes, and enables PHASE indicator/switches and computer for clock phase operation. Inhibits memory.

TABLE 3-2. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 1 (A2) (Cont)

Figure 3-2 Index No.	Control/Indicator	Type	Function
12 (Cont)	OP STEP	Pushbutton Indicators	Indicates operation step mode. Pressing MODE OP STEP button sets operation step mode, clears all other modes, and enables computer for execution of one operation at a time. This will be one instruction or one sequence depending on position of SEQ STEP/STOP switch.
	RUN		Indicates run mode. Pressing MODE RUN button sets run mode, clears all other modes, and enables computer for normal high speed operation.
13	I/O CLEAR/ MASTER CLEAR	Toggle Switch	Center position: Neutral. Momentary up position (I/O CLEAR): Clears I/O section of computer only and sets all stages of channel priority. Momentary down position (MASTER CLEAR): With computer not running, and not in phase-step mode, clears all sections, registers and control flip-flops of central computer; with computer running, it clears VOLTAGE FAULT and/or PROGRAM FAULT indicators only.

TABLE 3-2. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 1 (A2) (Cont)

Figure 3-2 Index No.	Control/Indicator	Type	Function
14	SEQ STEP/STOP	Toggle Switch	Center position: Neutral. Momentary up position (SEQ STEP): Enables execution of a single sequence in conjunction with operation step mode. Momentary down position (STOP): Stops high speed operation of the computer; RUN indicator goes off.
15	RESTART SPEED CONT	Variable Resistor	Varies speed of low-speed oscillator.
16	RESTART/START STEP	Toggle Switch	Center position: Neutral. Momentary up position (RE-START): Enables selected mode to be executed at restart speed control rate setting. Momentary down position (START/STEP): Load mode selected - initiates high-speed start at address 00500. Run mode selected - initiates high-speed start at address designated in P. Phase step mode selected - initiates issuance of phase or phases indicated by PHASE indicators. OP step mode selected - initiates execution of one instruction or one sequence, depending upon position of SEQ STEP/STOP switch.

TABLE 3-2. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 1 (A2) (Cont)

Figure 3-2 Index No.	Control/Indicator	Type	Function
17	FUNCTION REPEAT	Toggle Switch	Up position: Forces a repeat of instruction in F-register at mode rate, and disables clearing of S-register in any sequence other than I, except in W-sequence of instruction 50:10-50:13.
18	PHASE REPEAT	Toggle Switch	Up position: Forces repeat at high speed of phase or phases selected by PHASE indicators (Phase step mode must be selected.)
19	AUTO RECOVERY	Toggle Switch	Up position: Computer fault results in a jump to address 00500. Down position: Computer fault results in a jump to address 00000.
20	DISC ADV P	Toggle Switch	Up position: Inhibits incrementing P-Register.

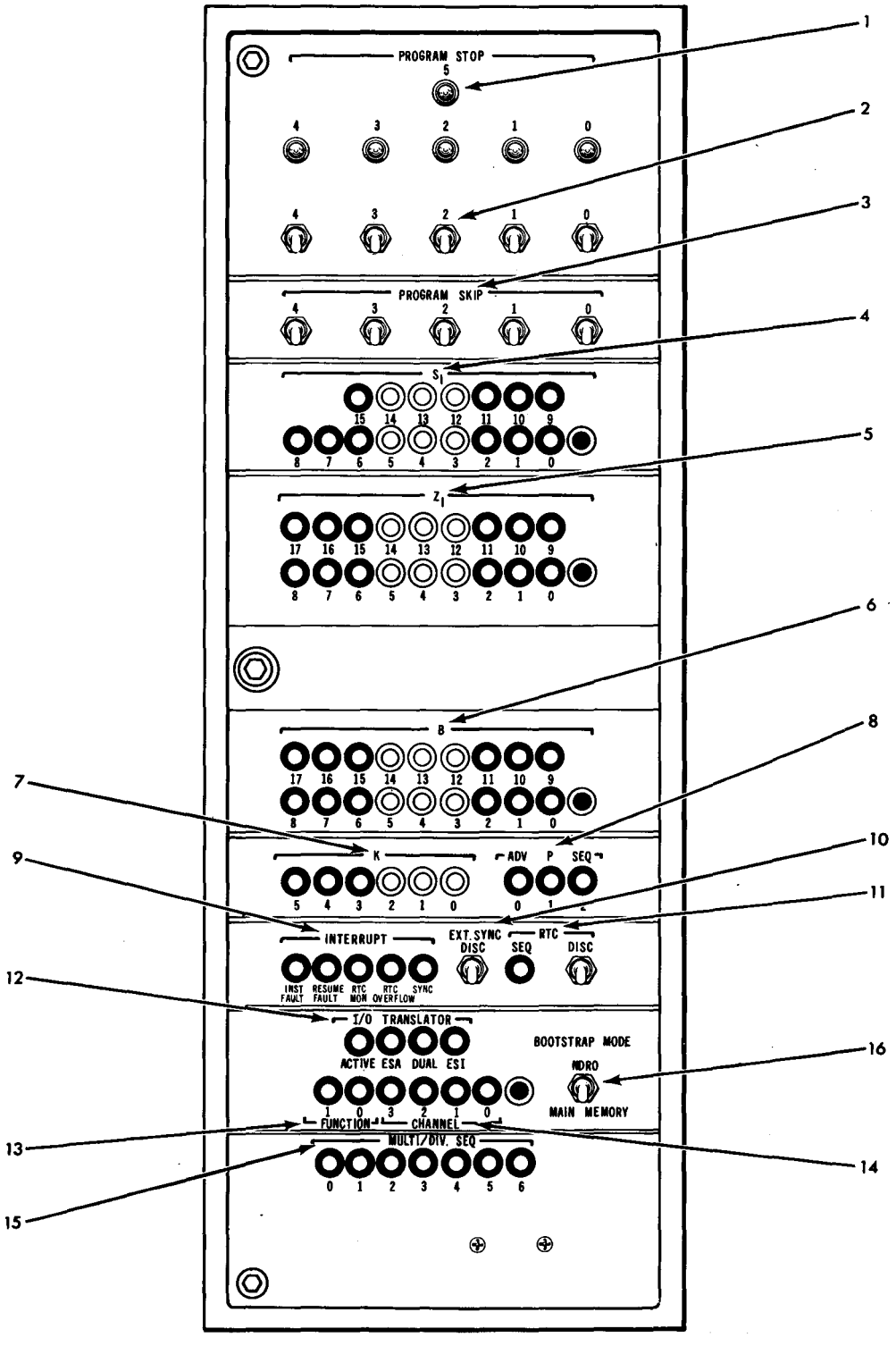


Figure 3-3. Controls and Indicators, Control Panel 2 (A4)

TABLE 3-3. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 2 (A4)

Figure 3-3 Index No.	Control/Indicator	Type	Function
1	PROGRAM STOP 0-5	Indicators	Comes on when a program stop occurs as a result of a 50:56 instruction. Indicator 5 lights for an unconditional stop; the rest are dependent upon stop switches.
2	PROGRAM STOP 0-4	Toggle Switches	On position (up): Enables a program stop for a 50:56 instruction if corresponding bit of instruction is a binary one.
3	PROGRAM SKIP 0-4	Toggle Switches	On position (up): Enables a skip of next instruction on a 50:50 instruction if corresponding bit of instruction is a binary one.
4	S_1 0-15 and Clear	Pushbutton Indicators	Displays contents and allows manual control of the 16 bits of memory address register. Each bit may be set by pressing appropriate indicator/switch. Clear button clears all bit positions simultaneously.
5	Z_1 0-17 and Clear	Pushbutton Indicators	Displays contents and allows manual control of the 18 bits of main memory exchange register. Each bit may be set by pressing appropriate indicator/switch. Pressing Clear button clears all 18-bit positions.

TABLE 3-3. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 2 (A4) (Cont)

Figure 3-3 Index No.	Control/Indicator	Type	Function
6	B 0-17 and Clear	Pushbutton Indicators	Displays contents and allows manual control of 18 bits of buffer control register. Each bit may be set by pressing appropriate indicator/switch. Clear button clears all bit positions simultaneously.
7	K 0-5	Pushbutton Indicators	Displays contents and allows manual control of six-bit register used for shift, multiply, divide, stop, and skip instructions. Must be cleared by MASTER CLEAR switch.
8	ADV P SEQ 0-2	Pushbutton Indicators	Indicates set condition of advance P-sequence flip-flops and operation of sequence. Each flip-flop may be set by pressing appropriate indicator/switch.
9	INTERRUPT INST FAULT RESUME FAULT	Indicators	Indicates that an instruction fault (f = 00, 01, 77) has been detected by fault circuitry, and a fault interrupt address is being generated to inform computer. Indicates that resume signal was not received during a minimum period of 1 second and a maximum of 2 seconds after data was placed on an intercomputer channel.[1]

[1] Dependent upon RTC being operational (RTC DISC switch in down position).

TABLE 3-3. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 2 (A4) (Cont)

Figure 3-3 Index No.	Control/Indicator	Type	Function
9 (Cont)	RTC MON	Indicator	Indicates that memory address 15, RTC word, is equal to memory address 14, RTC monitor; and that an interrupt is being generated to inform computer.[1]
	RTC OVERFLOW		Indicates that 18 bits of RTC word have changed from all binary ones to all binary zeros; and that an interrupt is being generated to inform computer.[1]
	SYNC		Indicates that a synchronizing interrupt has been received from a peripheral device and is being processed.
10	EXT SYNC DISC	Toggle Switch	Up position: External synchronizing input is disconnected. Down position: External synchronizing input may be received.
11	RTC SEQ	Indicator	Indicates that operation of updating real-time clock word is in process.
	DISC	Toggle Switch	Up position: RTC interrupting signal is disconnected (disabled). Down position: RTC interrupting signal is operational (enabled).

[1] Dependent upon RTC being operational (RTC DISC switch in down position).

TABLE 3-3. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 2 (A4) (Cont)

Figure 3-3 Index No.	Control/Indicator	Type	Function
12	I/O TRANSLATOR ACTIVE	Indicator	Indicates performance of any I/O operation or instruction (50:01 - 50:27).
	ESA		Indicates that current I/O operation is in externally specified address mode.
	DUAL		Indicates that current I/O operation is in dual (dual channel, 36-bit words) mode.
	ESI		Indicates that current I/O operation is in ESI (externally specified index) mode.
13	FUNCTION 0 and 1	Indicator	These indicator/switches display binary value which represents I/O functions. Values are assigned as follows: 00 - Ext Interrupt 01 - Ext. Fct. 10 - Output 11 - Input
14	CHANNEL 0-3 and Clear	Pushbutton Indicator	These indicator switches display octal value of active I/O channel. Each bit may be set by pressing appropriate indicator/switch. Clear button clears all bits of CHANNEL and FUNCTION.

TABLE 3-3. CONTROL/INDICATOR FUNCTIONS, CONTROL PANEL 2 (A4) (Cont)

Figure 3-3 Index No.	Control/Indicator	Type	Function
15	MULT/DIV SEQ 0-6	Pushbutton Indicators	Indicates set condition of multiply, divide, shift and scale sequence flip-flops and operation of sequence. Each flip-flop may be set by pressing appropriate indicator/switch.
16	BOOTSTRAP MODE	Toggle Switch	
	NDRO		References NDRO bootstrap program at addresses 00500 through 00537.
	MAIN MEMORY		References main memory addresses 00500 through 00537 which may contain an alternate bootstrap program.

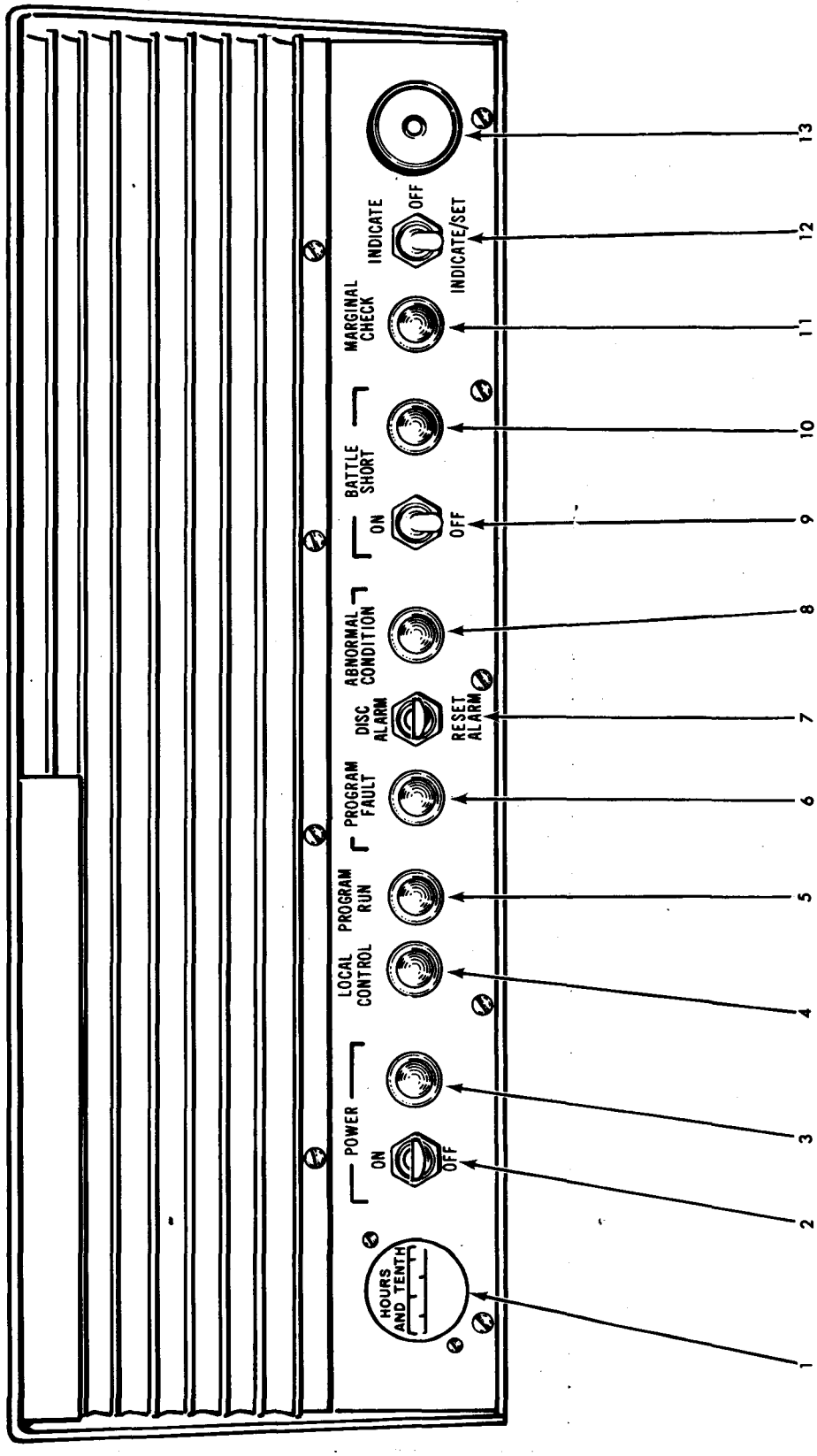


Figure 3-4. Controls and Indicators, Power Panel (A5)

TABLE 3-4. CONTROL/INDICATOR FUNCTIONS, POWER CONTROL PANEL (A5)

Figure 3-4 Index No.	Control/Indicator	Type	Function
1	HOURS AND TENTHS - Running Time Meter	Elapsed Time Meter	Cumulatively records time that power is supplied to computer. Range of meter is 0 to 9999.9 hours and cannot be reset.
2	POWER ON/OFF Switch	Toggle Switch	With proper interlock and operating temperatures, when momentarily in ON position, power is applied to computer; in momentarily OFF position, power is removed.
3	POWER	Indicator	Comes on when power is being applied to computer.
4	LOCAL CONTROL	Indicator	Comes on when computer may be controlled from control panels of computer (not remote control panel).
5	PROGRAM RUN	Indicator	Comes on when computer is in any mode of operation other than stop. (For example run flip-flop is set.)
6	PROGRAM FAULT	Indicator	Comes on when computer detects an illegal function code of 00, 01, 77 (Format I) in F register. Goes off by setting I/O CLEAR/ MASTER CLEAR switch to MASTER CLEAR position.

TABLE 3-4. CONTROL/INDICATOR FUNCTIONS, POWER CONTROL PANEL (A5) (Cont)

Figure 3-4 Index No.	Control/Indicator	Type	Function
7	DISC ALARM/RESET ALARM	Toggle Switch	Center position: Allows horn to sound on program, voltage, or temperature fault. Momentary down position (RESET ALARM): Horn is silenced. Up position (DISC ALARM): Horn is disabled for all faults.
8	ABNORMAL CONDITION	Indicator	Comes on when one of two abnormal condition indicators, TEMP or VOLTAGE FAULT on panel A2 is on.
9	BATTLE SHORT ON/OFF	Toggle Switch	ON position: Disables memory protective circuitry, temperature and blower sensor circuitry, thus removing all automatic computer shutdown.
10	BATTLE SHORT	Indicator	Comes on when BATTLE SHORT switch is in ON position.
11	MARGINAL CHECK	Indicator	Comes on when either CLOCK NARROW/NORMAL switch[1] is in NARROW position.

[1] Located behind front panel on drawer A2.

TABLE 3-4. CONTROL/INDICATOR FUNCTIONS, POWER CONTROL PANEL (A5) (Cont)

Figure 3-4 Index No.	Control/Indicator	Type	Function
12	INDICATE-OFF-INDICATE/SET	Toggle Switch	Center position (OFF): Disconnects voltage to all incandescent lamps on all panels. Down position (INDICATE/SET): Supplies voltage to all incandescent lamps and pushbutton switches on all panels.
13	Fault Horn	Horn	Horn sounds on program, voltage, or temperature fault.

3-9. DAILY TURN-ON

3-10. Daily turn-on is performed as follows:

STEP 1. Position the switches as indicated in table 3-5.

TABLE 3-5. INITIAL SWITCH POSITIONS

Switch	Fig. No.	Position
DISC ALARM/RESET ALARM	3-4	Neutral
BATTLE SHORT - ON/OFF	3-4	OFF
INDICATE-OFF-INDICATE/SET	3-4	INDICATE/SET
I/O CLEAR - MASTER CLEAR	3-2	Neutral
SEQ STEP/STOP	3-2	Neutral
RESTART/START STEP	3-2	Neutral
FUNCTION REPEAT	3-2	Down
PHASE REPEAT	3-2	Down
AUTO RECOVERY	3-2	Down
DISC ADV P	3-2	Down
PROGRAM STOPS	3-3	All Down
PROGRAM SKIPS	3-3	All Down
EXT SYNC DISC	3-3	Down
RTC DISC	3-3	Down

STEP 2. At main power source ensure that appropriate switches are positioned to provide AC power to the computer.

STEP 3. At computer power control panel (figure 3-4), momentarily position POWER switch to ON and observe POWER lamp lights, ABNORMAL CONDITION lamp is out, blowers operate, and HOURS AND TENTHS meter starts.

STEP 4. At Control Panel 1 (figure 3-2), observe ABNORMAL CONDITION lamps (TEMP and VOLTAGE FAULT) are out.

STEP 5. At Control Panel 1 (figure 3-2), Momentarily push I/O CLEAR-MASTER CLEAR switch to MASTER CLEAR.

CHAPTER 4

INSTALLATION

1. UNPACKING INFORMATION

4-2. When shipped by van, the computer will arrive at the site uncrated. When shipped by other means of transportation, or when shipped to an overseas location, the computer will arrive at the site packed in a plywood shipping crate. In either case, the computer is shipped with foam rubber pads taped to the corners of the cabinet to prevent incurring damage to the equipment during loading/unloading operations and during transit. For additional protection, a vinyl bag is placed over the cabinet and sealed at the base to protect the equipment from dust, dirt, and moisture. A skid, bolted to the bottom of the cabinet, affords further protection and permits use of a forklift truck when handling or moving the computer.

4-3. UNPACKING PROCEDURE. The following unpacking procedure applies in its entirety only if the computer is received from the factory in a shipping crate. Should the computer arrive uncrated, omit steps 1 through 4.

STEP 1. Remove all fasteners (screws, nails, metal bands,) from top of shipping crate.

STEP 2. Lift off and remove top section of the crate.

STEP 3. Pry loose and pull away four sides of crate to expose equipment.

STEP 4. Remove all packing material (such as shredded paper) from around computer.

STEP 5. Remove vinyl bag covering computer; remove tape and protective padding from corners of cabinet.

STEP 6. Visually inspect equipment for scratches, dents, and other evidence of shipping damage. Make a note of all shipping damage and report it immediately.

STEP 7. Using a forklift truck, move computer to general area where it is to be installed.

STEP 8. Remove bolts which hold equipment to shipping skid; lift or slide computer from skid.

STEP 9. Check cables, test equipment and spare parts received against shipping invoice for proper type and number; record and report all discrepancies.

STEP 10. Place computer in area specified on installation layout drawing.

4-4. PLACEMENT OF THE COMPUTER

4-5. The computer is a self-contained unit measuring approximately 72 inches high, 31 inches deep, and 26 inches wide (38 inches wide if the computer includes the additional half cabinet). The exact placement of the computer at the site is a matter left to the discretion of the user. However, there are certain environmental factors to be considered when choosing a location. The most significant factors are convenient accessibility/availability of suitable shelter, protection, drainage, clearance, cabling, power, cooling, and installation requirements.

1. Shelter, Protection, and Drainage. The computer requires no special auxiliary structure. However, the selected location must provide protection against possible damage caused by exposure to excessive dust, moisture, and standing water. Also, the area must be sufficiently spacious to

ensure that heat generated by the equipment will be properly dissipated.

2. Clearances. The selected area must provide space for the computer with sufficient clearance around the cabinet to permit the unrestricted movement of personnel and test equipment. The minimum clearances required are: front - 39 inches; rear - 10 inches; sides (door clearance) - 8 inches each side; and top - 12 inches. The 10-inch clearance at the rear of the cabinet is needed to ensure proper air circulation and heat dissipation. The 12-inch clearance above cabinet is required to facilitate signal cable connections. The 39-inch clearance at the front of the cabinet provides the space needed for chassis extension, inspection, testing, and parts removal and replacement. Figure 4-1 illustrates the clearance requirements.

3. Cabling Considerations. When selecting a location for placement of the computer, consideration must be given to the location of the peripheral equipment with which the computer will operate. Generally, the computer should be located close to the peripheral equipment unless a given application requires otherwise. For normal installations, the maximum distance between a peripheral device and a computer with the slow interface option is 300 feet when the signal cables used are balanced, twisted-pair, shielded cables having a maximum capacitance of 20 picofarads. The maximum distance between a peripheral device and a computer with the fast interface option is 50 feet when the signal cables used are balanced, twisted-pair, shielded cables having a minimum characteristic impedance of 120 ohms. All I/O cables connect to the cable entry panel. This panel is located in the top interior of the cabinet directly below the cover figure 4-2. The power cable connects to the power entry jack A14J1 on the top of the computer.

4. Power Requirements. The computer requires a 115-volt(±1%) line-to-line 3-phase 400-cycle(±5%) 1800-watt(4-drawer unit) or 3200-watt (6-drawer unit) alternating current power source. This power is conducted from the source through a junction box or control panel to the computer by means of a four-conductor cable. The cable consists of thre phase wires and neutral wire and connects to jack A14J1 which is located at the top of the cabinet as shown in figure 4-2.

5. Cooling Requirements. The computer system requires 400 cubic feet of cooling air per minute at 32°F (0°C) to 122°F (50°C) for a Mod 1 (4-drawer) unit, or 610 cubic feet of air per minute at 32°F (0°C) to 122°F (50°C) for Mod 0, 2, 3, and 4 (6-drawer) units.

6. Installation Requirements. The type of installation, mobile or stationary, determines the method of installing the computer. When used in a stationary installation, the computer must be bolted to the floor to prevent the cabinet from tipping forward when the pull-out drawers are extended. When used in a mobile installation, the computer must be bolted to the floor and to a wall. The stabilizers needed to fasten the computer to the wall are provided with the computer. Figure 4-3 is an outline dimensional drawing showing the location of the mounting holes in the base and at the rear of the cabinet.

4-6. CABLING CONNECTIONS

4-7. Cabling connections to/from the computer should be made in accordance with the following:

1. Power Cable. The power cable is a four-wire cable consisting of three phase wires and a neutral wire. Before connectir the power cable to either the power source or the computer, check all wires in the cable with an ohmmeter to ensure continuity and absence of short circuits. Then connect the cable as follows.

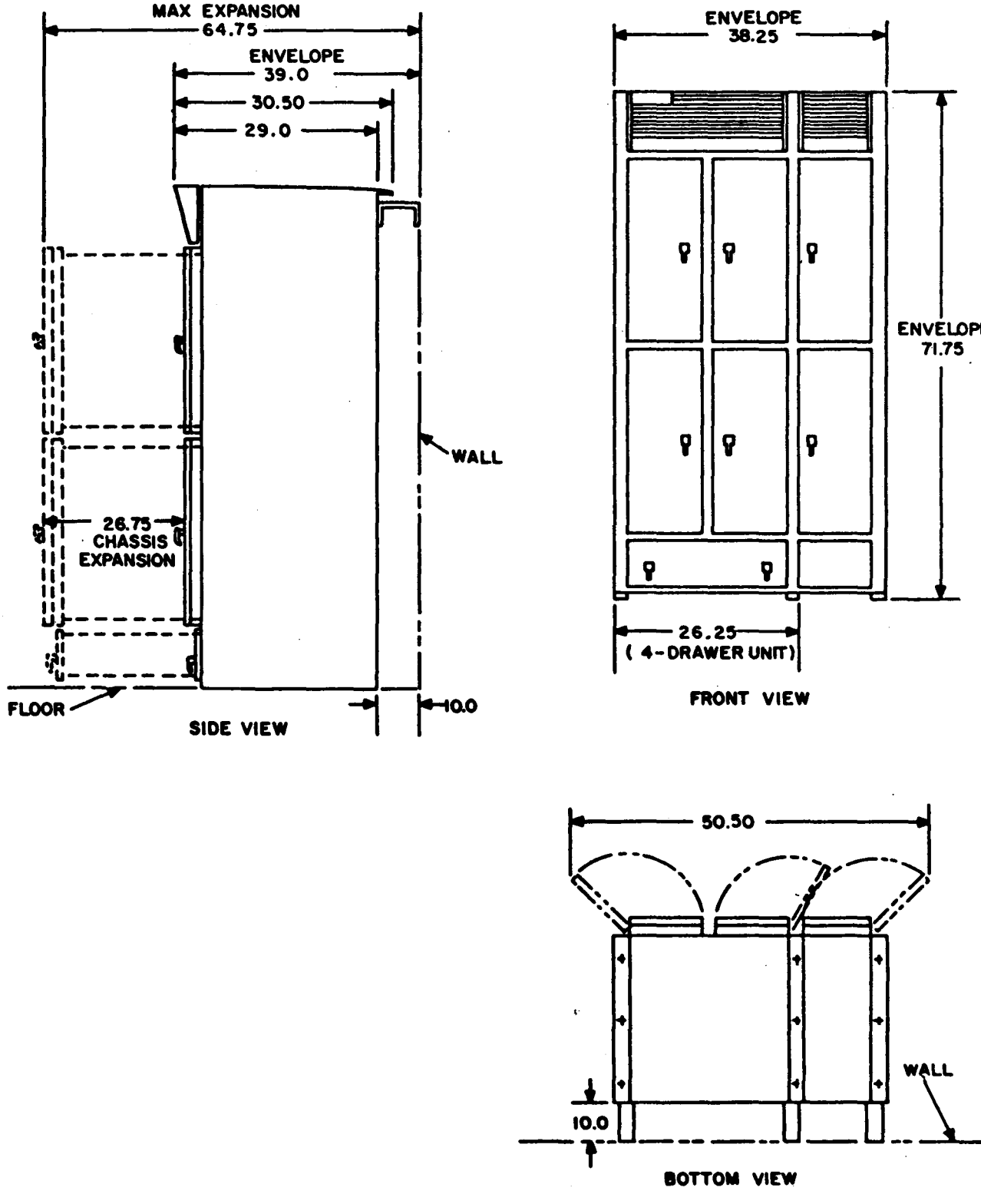


Figure 4-1. Computer Clearance Requirements

CAUTION

Before proceeding with STEP 1, check to ensure that power is off and the power supply to which the cable will be connected has been deactivated.

STEP 1. Prepare the cable for fastening to the jack at the back of the computer.

STEP 2. Remove the plug from jack A14J1.

STEP 3. Open the jack and connect the neutral wire to pin D; then connect the three phase wires to pins A-Ø1, B-Ø2, and C-Ø3.

STEP 4. Connect the other end of the cable to the designated terminals at the control panel.

STEP 5. Verify all connections at A14J1 and at the power supply.

STEP 6. Perform the initial power checks (MRC W-2).

2. I/O Signal Cables. The number of I/O signal cables that connect to the computer depends on the I/O capability of the computer. Each I/O channel requires two 90-conductor (45 twisted-pair lines) cables - one an input cable and the other an output cable. Thus, sixteen cables (maximum) for an eight-channel computer, and thirty-two cables (maximum) for a sixteen-channel computer are required.

3. Optional Cabling Connections. In addition to the I/O signal cables, a single

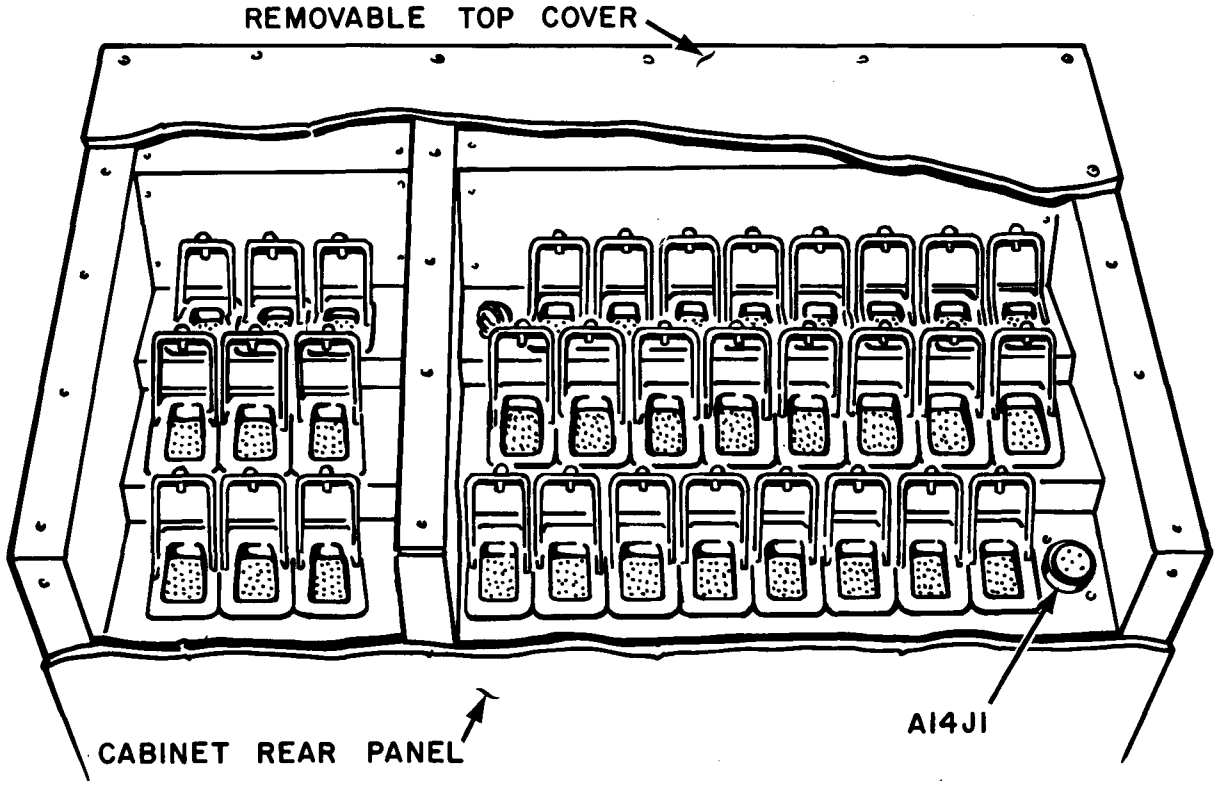


Figure 4-2. Location and Access to Cable Entry Panel

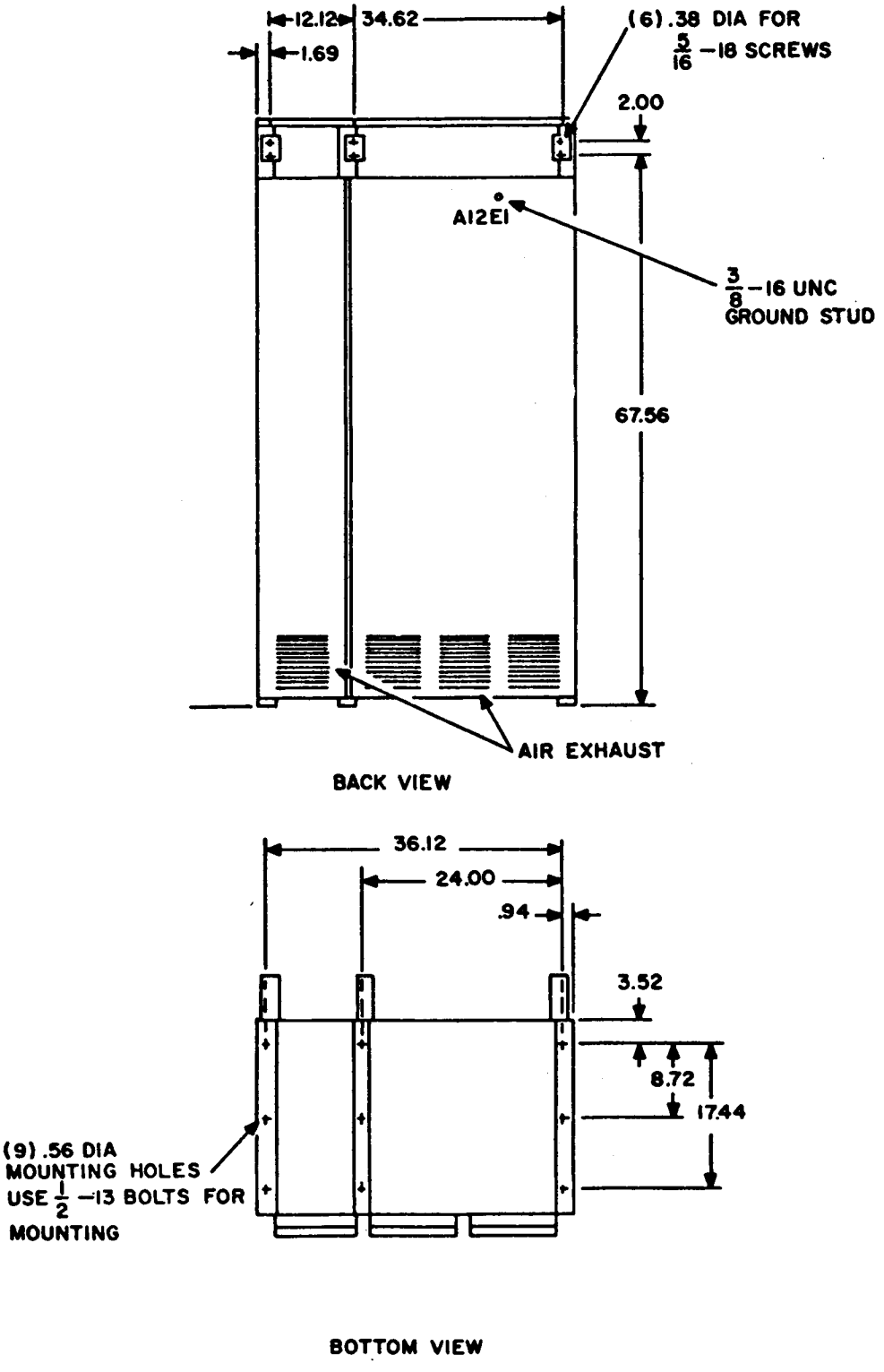


Figure 4-3. Location of Cabinet Mounting Holes

TABLE 4-1. CABLE CONNECTIONS

Computer Connector No.	Cable	Computer Connector No.	Cable
J1	Channel 6-input	J18	External sync
J2	Channel 5-input	J21	Channel 12-input
J3	Channel 4-input	J22	Channel 11-input
J4	Channel 3-input	J23	Channel 10-input
J5	Channel 2-input	J24	Channel 15-input
J6	Channel 1-input	J25	Channel 14-input
J7	Channel 0-input	J26	Channel 13-input
J8	Channel 6-output	J27	Channel 17-input
J9	Channel 5-output	J28	Channel 16-input
J10	Channel 4-output	J29	Channel 17-output
J11	Channel 3-output	J30	Channel 16-output
J12	Channel 2-output	J31	Channel 15-output
J13	Channel 1-output	J32	Channel 14-output
J14	Channel 0-output	J33	Channel 13-output
J15	Channel 7-input	J34	Channel 12-output
J16	Channel 7-output	J35	Channel 11-output
J17	Remote console	J36	Channel 10-output

three-conductor cable can be connected to the computer. This is a special cable used to connect the computer to an external synchronizing device. Also, an additional 90-pin jack at the top of all computers is supplied for use if the optional remote control feature of the computer is to be utilized.

4. Cable Connections. The external synchronizing cable, remote control cable, and all I/O signal cables connect to the computer at the cable entry panel. Cable connections are made as indicated in table 4-1.

5. Pin Connections. Table 4-2 shows the pin assignments for the input/output cables connecting to the peripheral devices (connectors J1 through J16 and J21 through J36). Table 4-3 shows the pin assignments for the signal cable connecting to the remote console (connector J17). Table 4-4 shows the pin assignments for the three-conductor cable connecting the computer to an external synchronizer (connector J18).

TABLE 4-2. I/O CABLE CONNECTOR PIN ASSIGNMENTS (J1-J16, J21-J36)

Pin No.	Input Connector Pin Designation	Output Connector Pin Designation
01	Input data request	Output acknowledge
02	Input data acknowledge	Output data request
03	External interrupt request	External function acknowledge
04	Spare	External function request
05	2^{32}*	2^{32}*
06	2^{33}*	2^{33}*
07	2^{34}*	2^{34}*
08	2^{35}*	2^{35}*
09	2^{0}	2^{0}
10	2^{1}	2^{1}
11	Input data request (R)	Output acknowledge (R)
12	Input data acknowledge (R)	Output data request (R)
13	External interrupt (R)	External function acknowledge (R)
14	Spare (R)	External function request (R)
15	2^{32}(R)*	2^{32}(R)*
16	2^{33}(R)*	2^{33}(R)*
17	2^{34}(R)*	2^{34}(R)*
18	2^{35}(R)*	2^{35}(R)*
19	2^{0}(R)	2^{0}(R)
20	2^{1}(R)	2^{1}(R)
21	Not used	Not used
22	2^{2}	2^{2}
23	2^{3}	2^{3}
24	2^{4}	2^{4}

TABLE 4-2. I/O CABLE CONNECTOR PIN ASSIGNMENTS (J1-J16, J21-J36) (Cont)

Pin No.	Input Connector Pin Designation	Input Connector Pin Designation
25	2^5	2^5
26	2^6	2^6
27	2^7	2^7
28	2^8	2^8
29	2^9	2^9
30	2^{10}	2^{10}
31	2^{11}	2^{11}
32	2^{12}	2^{12}
33	2^2(R)	2^2(R)
34	2^3(R)	2^3(R)
35	2^4(R)	2^4(R)
36	2^5(R)	2^5(R)
37	2^6(R)	2^6(R)
38	2^7(R)	2^7(R)
39	2^8(R)	2^8(R)
40	2^9(R)	2^9(R)
41	2^{10}(R)	2^{10}(R)
42	2^{11}(R)	2^{11}(R)
43	2^{12}(R)	2^{12}(R)
44	Not used	Not used
45	Cable shield	Cable shield
46	Not used	Not used
47	2^{13}	2^{13}
48	2^{14}	2^{14}

TABLE 4-2. I/O CABLE CONNECTOR PIN ASSIGNMENTS (J1-J16, J21-J36) (Cont)

Pin No.	Input Connector Pin Designation	Output Connector Pin Designation
49	2^{15}	2^{15}
50	2^{16}	2^{16}
51	2^{17}	2^{17}
52	2^{18}*	2^{18}*
53	2^{19}*	2^{19}*
54	2^{20}*	2^{20}*
55	2^{21}*	2^{21}*
56	2^{22}*	2^{22}*
57	2^{23}*	2^{23}*
58	2^{13}(R)	2^{13}(R)
59	2^{14}(R)	2^{14}(R)
60	2^{15}(R)	2^{15}(R)
61	2^{16}(R)	2^{16}(R)
62	2^{17}(R)	2^{17}(R)
63	2^{18}(R)*	2^{18}(R)*
64	2^{19}(R)*	2^{19}(R)*
65	2^{20}(R)*	2^{20}(R)*
66	2^{21}(R)*	2^{21}(R)*
67	2^{22}(R)*	2^{22}(R)*
68	2^{23}(R)*	2^{23}(R)*
69	Cable shield	Cable shield
70	2^{24}*	2^{24}*
71	2^{25}*	2^{25}*
72	2^{26}*	2^{26}*

TABLE 4-2. I/O CABLE CONNECTOR PIN ASSIGNMENTS (J1-J16, J21-J36) (Cont)

Pin No.	Input Connector Pin Designation	Input Connector Pin Designation
73	2^{27}*	2^{27}*
74	2^{28}*	2^{28}*
75	2^{29}*	2^{29}*
76	2^{30}*	2^{30}*
77	2^{31}*	2^{31}*
78	Not used	Not used
79	Not used	Not used
80	2^{24}(R)*	2^{24}(R)*
81	2^{25}(R)*	2^{25}(R)*
82	2^{26}(R)*	2^{26}(R)*
83	2^{27}(R)*	2^{27}(R)*
84	2^{28}(R)*	2^{28}(R)*
85	2^{29}(R)*	2^{29}(R)*
86	2^{30}(R)*	2^{30}(R)*
87	2^{31}(R)*	2^{31}(R)*
88	Not used	Not used
89	Not used	Not used
90	Not used	Not used

* When the computer is operating dual channel (for 36-bit word equipment), these pin designations apply. When the computer is operating single channel (for 18-bit word equipment), pin designations do not apply. Pins labeled with an asterisk are considered spares, but cannot be used as such. The (R) implies either the ground return or signal return line of the twisted pair.

TABLE 4-3. PIN ASSIGNMENTS FOR REMOTE CONSOLE CABLE CONNECTOR (J17)

Pin No.	Connector Pin Designation
01	Ground
02	Ground
03	-15V
04	-15V
05	Unassigned
06	Unassigned
07	Unassigned
08	Unassigned
09	Unassigned
10	Unassigned
11	Battle Short indicator
12	Marginal Check indicator
13	Horn Alarm
14	Stop 5 indicator
15	Unassigned
16	Unassigned
17	Unassigned
18	Unassigned
19	Unassigned
20	Unassigned
21	Unassigned
22	Stop 1 indicator
23	Run indicator
24	Remote indicator

TABLE 4-3. PIN ASSIGNMENTS FOR REMOTE CONSOLE CABLE CONNECTOR (J17) (Cont)

Pin No.	Connector Pin Designation
25	Stop 4 indicator
26	Stop 3 indicator
27	Stop 2 indicator
28	Program Fault indicator
29	Stop 0 indicator
30	Unassigned
31	Unassigned
32	Unassigned
33	Unassigned
34	Unassigned
35	Unassigned
36	Unassigned
37	Unassigned
38	Unassigned
39	Unassigned
40	Unassigned
41	Unassigned
42	Unassigned
43	Unassigned
44	Unassigned
45	Ground
46	Unassigned
47	Stop 4 switch
48	Stop 3 switch
49	Stop 2 switch

TABLE 4-3. PIN ASSIGNMENTS FOR REMOTE CONSOLE CABLE CONNECTOR (J17) (Cont)

Pin No.	Connector Pin Designation
50	Stop 1 switch
51	Stop 0 switch
52	Skip 4 switch
53	Skip 3 switch
54	Skip 2 switch
55	Skip 1 switch
56	Skip 0 switch
57	Master clear switch
58	Stop 4 switch (R)
59	Stop 3 switch (R)
60	Stop 2 switch (R)
61	Stop 1 switch (R)
62	Stop 0 switch (R)
63	Skip 4 switch (R)
64	Skip 3 switch (R)
65	Skip 3 switch (R)
66	Skip 1 switch (R)
67	Skip 0 switch (R)
68	Master clear switch (R)
69	Ground
70	Remote ON/OFF switch (ON)
71	Auto recovery switch
72	Load (Mode) switch
73	Stop switch
74	Start switch (NC)

TABLE 4-3. PIN ASSIGNMENTS FOR REMOTE CONSOLE CABLE CONNECTOR (J17) (Cont)

Pin No.	Connector Pin Designation
75	Start switch (NO)
76	Unassigned
77	Abnormal condition indicator
78	Unassigned
79	Unassigned
80	Remote ON/OFF switch (ON) (R)
81	Auto recovery switch (R)
82	Load (Mode) switch (R)
83	Stop switch (R)
84	Start switch (NC) (R)
85	Start switch (NO) (R)
86	Unassigned
87	Unassigned
88	Unassigned
89	Unassigned
90	Unassigned

The (R) implies either the ground return or signal return line of the twisted pair.

TABLE 4-4. PIN ASSIGNMENTS FOR EXTERNAL SYNC CONNECTOR (J18)

Pin No.	Connector Pin Designation
A	External Sync (R)
B	Ground
C	External Sync Signal

The (R) implies either the ground return or signal return line of the twisted pair.

4-8. COMPUTER GROUND. Grounding the computer is important for two reasons: (1) to ensure the safety of operating and maintenance personnel; (2) to maintain the DC level used for I/O communications. The computer is grounded in two places: at A14E1 of the power entry jack, and at ground lug A12E1 located at the rear of the cabinet. The AC neutral wire connects to A14E1 via pin D (A14J1); system ground cables connect to ground lug A12E1 along with all internal circuit grounds. A separate cable should be used for grounding each device peripheral to the computer. These cables should not be junctioned at any point other than at the ground lug A12E1 on the rear of the computer.

4-9. PREOPERATIONAL INSPECTION AND PRELIMINARY TESTING

4-10. VISUAL INSPECTION. The following procedure should be performed after the computer has been installed and the power and I/O signal cables have been connected but before the power has been turned on. The purpose of this inspection is to detect components and parts that may have been damaged when the computer was loaded, unloaded, or in transit. The types of damage to check for are loose circuit cards, loose card retainers, damaged fuses or indicators, and other forms of shipping damage not apparent or detected at the time of unpacking the equipment. Perform the following steps:

STEP 1. Open drawer A3 (memory panel) front cover and remove combination tool from inside of the door.

STEP 2. Unscrew four locking latches and extend drawer; ensure that it locks securely in fully extended position. Check drawer for mechanical defects and damage to components.

STEP 3. Check for loose retainers, circuit cards, and hardware. Reseat any cards that are loose and replace any that are damaged.

NOTE: Omit STEP 4 (drawer A3 only) on computers with a 32K memory or less.

STEP 4. Swing chassis open. Check for loose or broken wire wrap connections. Repair as necessary.

STEP 5. Ensure that no loose hardware (nuts, bolts, washers) is lying inside drawer. Also ensure that all terminal board mounting hardware and insulating materials are securely in place.

STEP 6. Check fuses and replace as necessary.

STEP 7. Close the hinged chassis; then close and lock pull-out drawer.

STEP 8. Repeat steps 1 through 7 for each of remaining drawers except that in step 2, use combination tool and loosen three retaining screws on front panel and open panel to expose test blocks and drawer-retaining bolt. Use combination tool and unscrew drawer-retaining bolt until plugs on rear of drawer are disengaged from cabinet receptacles. Slide drawer forward to fully extended position and ensure that

drawer lock is engaged.

STEP 9. Check power supply drawer to ensure no fuses are cracked or broken. Ensure that all fuses are securely in place.

STEP 10. With a volt-ohmmeter, check power supply to ensure no short circuits exist between TB2-1, 2, 3 or from TB2-1, 2, 3 to ground.

STEP 11. Ensure computer is properly grounded by inspecting ground connections at A12E1 located at rear of cabinet. Separate cables for grounding each peripheral device should be connected here.

STEP 12. Ensure the AC neutral wire is connected to lug E1 located at the rear of the cabinet.

STEP 13. Ensure that all cable connectors (plugs/jacks) are properly connected and tight.

4-11. POWER CHECK. Verify AC and DC voltage levels; perform MRC Q-1.

4-12. FAN ASSEMBLY A12 ROTATION CHECK. After power checks are performed, ensure that fan is rotating in the proper direction by performing the following checks:

STEP 1. Position POWER ON/OFF switch on power control panel to OFF.

STEP 2. Remove and retain retaining screws and springs fastening grill on power panel.

STEP 3. Remove grill and air filter.

STEP 4. Position POWER ON/OFF switch on power control panel to ON.

STEP 5. Standing facing the computer and looking at the left end of the blower observe that blower fan rotates CCW.

STEP 6. If rotation is wrong check phases A, B, C on TB2 Terminals 1, 2, and 3 respectively in computer for proper phase relationship. See MRC Q-1 for access to terminals.

STEP 7. If fan is rotating properly replace grill and air filter and return to normal operating condition.

4-13. OPERATIONAL CHECK. Load and run diagnostic confidence test; perform MRC appropriate to digital computer modification as follows:

Mod	MRC
0 and 3	Q-2
1 and 5	Q-3
2 and 4	Q-4

CHAPTER 5

MAINTENANCE

5-1. INTRODUCTION TO THE DIAGNOSTIC PROGRAM

5-2. Maintenance functions are divided into two basic categories: scheduled maintenance and corrective maintenance. For both categories, the diagnostic serves as the primary tool. In the case of scheduled maintenance, the diagnostic program verifies the operability of the equipment; in corrective maintenance, the diagnostic program together with correlative test procedures isolates the malfunction to a specific module or modules. The diagnostic procedures thus consist of both detection and isolation phases. Detection is defined as noting the absence of proper operation of a function and isolation as translating information available from a detected malfunction into an acceptable statement regarding location of the malfunction. An acceptable statement is one which effectively aids the user in restoring the device under test to its proper state of operation within a reasonable length of time.

5-3. DETECTION PHASE. The detection phase of the diagnostic has two main goals. The first goal is to provide an exercise and test for every function. The function repertoire is the means by which this task is accomplished. The second goal is to provide a reliable indication representative of any function, noted to be in error or of a non-error (i.e., good pass) completion of the program. The maintenance panel is used for displaying these indications.

5-4. ISOLATION PHASE. The isolation phase of the diagnostic consists of correlating the controlled detection displays with the failed hardware. A typical malfunction isolation sequence is the following:

1. The diagnostic program is executed according to procedures in chapter 8.

2. An error condition occurs with a resultant symptom.

3. Reference is made to the fault isolation tables for card replacement. The symptom observed in step 2 requires the observance of register values.

5-5. TESTING PHILOSOPHY. Diagnostic testing philosophy is, in general, based upon the "building block" approach. This approach requires that a small area of hardware logic be tested, and upon successful completion, this area is used as a base for testing another area. It is inherent in this concept that each new area be relatively small and that areas previously tested form the foundation needed to check each new area. The order of testing is determined following an analysis of the hardware; the testing continues until all functions have been exercised.

5-6. LIMITATIONS OF THE DIAGNOSTIC. Under ideal conditions, within each "building block" there exists the capability of automatically asking questions of the logic under test and getting complete answers without endangering program

control in the event of a malfunction. Under other than ideal conditions where loss of program control is anticipated, manual procedures requiring operator intervention are used to provide an adequate presentation to display the result of the attempted function. Prudent use of the diagnostic dictates that no attempt be made during either maintenance phase to exceed the known limitations of the program. Detection and isolation of multiple-fault malfunctions which are independent of each other are also within the capabilities of the diagnostic. The diagnostic also is capable of detecting intermittent and backward-propagating malfunctions or failures incurred in power supply modules and wiring networks.

5-7. BASIC COMPUTER PROGRAMING

5-8. SCOPE. Since the primary tool for troubleshooting the Mk 152 Computer is the Diagnostic Program, it is important that the maintenance technician know how to read the program listing and also, on occasion, be able to do basic programing to assist in isolating malfunctions. There is no intention to produce fully qualified programmers; the intent is only to impart to the maintenance person sufficient programing knowledge to enable him to adequately repair the computer. It should be realized that programing is an art which cannot be taught; only some basic rules can be provided to assist in the development of this art.

5-9. PROGRAM DEVELOPMENT. A computer program represents more than a set of detailed instructions. It is the product developed from a programmer's applied knowledge to a specific problem and results in intelligent control of the computer system capabilities. Programs may be written in direct number systems (machine language), in coded forms, or in symbolic codes which exist in abundance. A program is a complete plan of attack on a specific problem including program description and computer routine written carefully and precisely in the appropriate computer language. The development of a program must begin with problem definition and analysis. The following must be considered in preparation of even the simplest program:

1. Allocation of sufficient storage location to data, instruction, and related information.

2. Conversion of original data to an input medium.

3. Register displays or printouts desired for a failure or for a successful completion of the program.

4. Ability to continuously repeat all or portions of the program without manual intervention.

5. Ability to restart in the event of a hardware failure.

5-10. A program is actually a list of statements; therefore, clarity, completeness, and precision are all prerequisite qualities of good programing. The statements must occur in a sequence that the computer can execute one at a time. The programmer should provide for branching along alternate pathways depending on answers evolving from decision-making processes. Provisions should also be made for looping.

back and repeating a portion of the program where doing so would increase the efficiency of the program. All in all, exactness and close attention to every detail are of prime importance in both the planning and development phases of the programing.

5-11. PROGRAMING LANGUAGE. As mentioned previously, a program may be coded in machine language or it may be coded in a symbolic language. Most programs are written in a symbolic language for several reasons. Programing in machine language is difficult and very time consuming because of the necessity of keeping track of all allocations of memory locations. Since instructions must be written in an exact sequence, a single omission or mistake may require numerous changes. A greater understanding of the machine is also required to use machine language. By coding in symbolic language and using an assembler, the paperwork and bookkeeping is handled automatically and thus it also permits the transfer of programs to any region of memory since symbolic addressing is independent of memory locations. An assembler is itself a computer program which operates on the symbolic input data in a certain format, and produces machine instructions by translating the symbolic operation codes into the specific number system of the computer. The assembler generally translates symbolic codes to machine codes on an item-for-item basis and produces as output the same number of instructions or constants as defined in the input symbolic codes. An assembly language operation has the following general format: (label) * (operator) * (operand) * (notes).

1. The label identifies the particular statement. A label is not required for every statement. In an absolute-addressed program, every word is assigned an absolute address during the code process. The assembling process equates the label to the machine address assigned to the instruction generated by the statement. Only those statements which are referred to by other statements require a label or symbolic address. Where more than one instruction is generated by a statement, the label refers to the address of the first instruction generated. The term "label" is used rather than "address" since it accurately describes the function of the symbolic address. The instructions or words generated from unlabeled statements following one another on the source program tape are ultimately assigned to consecutive memory addresses.

2. The operator together with the operand comprise the program statement. The operator is the symbolic shorthand or octal notation which identifies the basic function to be performed. It causes the assembler to generate a single machine instruction or in some cases a group of instructions. The operand defines, modifies, or completes the operator function. It may be either a constant in octal notation or a symbolic alphanumeric notation referring to a constant (either an absolute address or an item of data).

3. Whenever any statement refers to a label, the operand portion is called a tag. The tag must be identical to the related label but it may be followed by a + or - and an octal integer to provide for increments or decrements. Thus, an instruction may refer to an unlabeled instruction in terms of its sequential position preceding or following a labeled instruction.

4. The notes merely help in stating the purpose of, or clarifying, a portion of the program. They do not, in any way, affect the instructions generated from the statement.

5-12. ASSEMBLING PROCESS. The final products resulting from the assembling process are a program listing and a program tape in a particular format. The program listing consists of at least the symbolic program (address labels and instruction codes) together with remarks, and the corresponding program in machine language. For explanatory purposes, the following two common and useful programs are presented as examples. Each is explained in detail with the symbolic language together with the corresponding machine program.

5-13. Inspect and Change Program. An inspection and change program (refer to Table 5-1) permits the inspection and changing of sequential addresses in memory. The operator selects the starting address and enters the address to be inspected in AL. The program then automatically displays, in AU, the contents of the selected address and comes to a stop. The AU register may then be set to the desired value and upon restarting the program, the new value is stored in the proper location. If AU is not changed, the address contents remain unchanged. In addition to storing the AU value, the program also automatically references and displays the next sequential address and its contents in AL and AU respectively.

5-14. A detailed explanation of the program instruction follows. Index register 01 is selected and the address entered into AL is stored there (index register 01 is assigned to memory address 01). The contents of that address are then loaded into AU. An unconditional stop is then executed. Upon restarting, the contents of AU are stored into the location specified by the index register. AL is then incremented by 1 for use in repeating the program for the next sequential address.

NOTE: An inspect and change program is also contained in the diagnostic program.

TABLE 5-1. INSPECT AND CHANGE PROGRAM INSTRUCTIONS

Label	Symbolic Instruction	Notes	Address	Machine Instruction
START	ENTICR.01	SELECT INDEX REGISTER (B1)	010	50 7201
IC1	STRAL.01	LOAD ADDRESS TO INSPECT IN B1	011	44 0001
	ENTAUB.0	DISPLAY CONTENTS OF ADDRESS	012	11 0000
	STOP	STOP	013	50 5640
	STRAUB.0	STORE CONTENTS	014	47 0000
	ADDALK.1	INCREMENT FOR NEXT ADDRESS	015	71 0001
	JP.IC1	REPEAT FOR NEXT ADDRESS	016	34 0011

5-15. BOOTSTRAP Program (Channel 0). The paper/magnetic tape BOOTSTRAP Program for Computer Mk 152 Series (refer to Table 5-2) occupies locations 500 through 537. The BOOTSTRAP assembly residing in a computer depends upon peripheral requirements. The program described here loads a paper tape into location 540-704 in main memory. The program loaded is usually a loader program which then permits the loading of any size block of information into any area of main memory. The Computer Mk 152 Series paper tapes have a standard format in which the first three words consist of the identifier code, initial address location, and terminal address location respectively. Since the bootstrap unconditionally loads the program into addresses 540-704, the first three data words are irrelevant.

5-16. A detailed explanation of only the Channel 0 Program follows: The first instruction sets the interrupt lockout. Next the SR register is enabled for stack 0 and ICR 01 is selected. Following this, an external function to turn on the paper tape reader and start read is executed. The B-register is then cleared and stored in address 003. A value of 000002 is then stored in address 004. This value is used with an index skip (ISK) instruction to bring in a total of three 6-bit frames in order to build the 18-bit word. The A-register is now cleared and the input initiated. The one-word input is to address 002. At wait, skip inactive (SKPIN) loop is used to determine when the word has been inputted. The initially cleared AL register is then shifted 6 positions left and then selectively set with the input word in address 002. Although the left shift has no effect for the first 6 bits of the word, on the next two passes through the loop the left shift repositions the previous input to allow the building of the total 18-bit word. Next, AL is checked for zero. If AL is zero, AU is loaded with the contents of address 003, the count of words in. If this AU value is still zero, the first word on tape has not yet been reached and the program loops back to boot until the first word is found. If address 003 is non-zero, the program is in the data portion of the tape but has merely found a frame of all zeros (00). In this case, as well as if AL had been non-zero after the input, control passes to count where an index skip is done on address 004. If the skip fails to occur, indicating that a group of 3 frames had not been input to build the total 18-bit word, the program jumps to boot and inputs another frame. If the skip occurs, the total 18-bit word is in AL and the program stores it using the STRALB instruction into address 536 plus B modification. It should be noted that the first words load into addresses 536 and 537 which are NDRO. However, the first words are irrelevant since they are the three format words (76 code, 15-bit initial address, 15-bit terminal address) which are packed into two 18-bit words. A B-register skip is then executed to enable the next word to be stored in the next sequential memory location when program control is returned to next. The entire input process repeats until 147 words have been input and stored in main memory.

5-17. DIAGNOSTIC PROGRAM DESCRIPTION

5-18. The following subparagraphs describe the diagnostic program. Each subparagraph (CONTBASI, AIMAT, MEDIA, and IOTA) is examined. The ZPAC utility package, which facilitates execution of isolation test, is also described.

5-19. CONTBASI. CONTBASI checks the entire arithmetic and control logic. It uses a "building block" approach, assuming a

TABLE 5-2. BOOTSTRAP PROGRAM INSTRUCTIONS

Label	Mneumonic Instruction	Notes	Address	Content
		BOOTSTRAP Assembly No. 7024774-02: paper-tape, level 6, octal, channel 0 program used with I/O Console Mk 77 Series		
Start	SIL*0	Set Interrupt Lockout	500	503400
	ENTSR*10	Enable SR Stack 0	501	507310
	ENT1CR*1	Select ICR 1	502	507201
	EXF*0	Turn On Reader, Start Read	503	501300
	0*READ	Reader Code	504	000534
	0*READ		505	000534
	ENTBK*0	Clear B	506	360000
Next	STRB*3	Total Words Input Count	507	420003
	ENTALK*2	Index for 3 Frames of Data	510	700002
	STRAL*4	Store Index	511	440004
	ENTALK*0	Clear AL	512	700000
Boot	IN*0	Input-Channel 0	513	501100
	0*2	Input Location - 1 Word	514	000002
	0*2		515	000002
Wait	SKPIIN*0	Is Word In?	516	502100
	JP*WAIT	No-Loop Until In	517	340516
	LSHAL*6	Yes-Position Data	520	504606
	SLSET*2	Insert Data in AL	521	510002
	JPALNZ*			
	COUNT	Data Equal Zero?	522	630525
	ENTAU*3	Yes-Check if Data or Leader	523	100003
	JPAUZ*BOOT	And Handle Zero Frames	524	600513
Count	ISK*4	Decrement Frame Index	525	570004
	JP*BOOT	Get Next Frame	526	340513
	STRALB*536	Store Word in Memory	527	450536
	BSK*WORD	All Words Input?	530	560533
	JP*NEXT	No-Do Next Word	531	340507
	JP*540	Yes-Exit	532	340540
Word	0*147	Input Count	533	000147
Read	0*151	Reader Code	534	000151
	0*0	Not Used	535	000000
	0*0	Not Used	536	000000
	0*0	Not Used	537	000000

TABLE 5-2. BOOTSTRAP PROGRAM INSTRUCTIONS (cont.)

Label	Mneumonic Instruction	Notes	Address	Content
		BOOTSTRAP Assembly No. 7024774-10: magnetic-tape, modulus 3,556 BPI, octal, odd parity, channel 13 program used with Digital Data Recorder Mk 19 Series		
Start	SIL*0	Set Interrupt Lockout	500	503400
	ENTSR*0	Disable SR	501	507300
	ENTBK*366	Index for Control Memory	502	360366
	ENTAL*			
	REMOVE	Ignore Interrupt Code	503	120512
Store	STRALB*11	Store RIL in Control Memory Address	504	450011
	BJP*STORE	All RIL's stored?	505	730504
	EXF*13	Yes-Activate DDR I/O Channel	506	501313
	0*540	DDR External Function Codes	507	000540
	0*534		510	000534
	EXFOV*13	Demand Control, Master Clear DDR	511	502713
Remove	RIL*0	Remove Interrupt Lockout	512	503000
Boot	IN*13	Input - Channel 13	513	501113
	0*304	Input Buffer	514	000304
	0*263		515	000263
	EXFOV*13	Select Address 1 (1240 MTU)	516	502713
	LSHA*77	Delay to Allow DDR to Respond	517	504777
	EXFOV*13	Rewind-Read, Clear Write Enable	520	502713
Wait	SKPIIN*13	Is Input Complete?	521	502113
	JP*WAIT	No-Loop Until In	522	340521
Check	ENTB*263	Initial Address of Input Buffer Stored in Memory	523	320263
	ENTAL*265	Enter Checksum	524	120265
	SUBAL*264	Subtract Starting Address	525	160264
Next	SUBALB*0	Subtract Word Stored in Memory	526	170000
	BSK*266	Is Checksum Complete?	527	560266
	JP*NEXT	No - Do Next Word	530	340526
	JPALNZ*			
	START	Yes - Is Checksum Correct? Restart if No	531	630500
	IJP*264	Yes - Exit	532	550264
	0*600000		533	600000
	0*600000	Demand Control, Master Clear DDR EF Word	534	600000
	0*400011	Select 1240 MTU Address 1 EF Word	535	400011
	0*164711	Rewind-Read, Clear Write Enable EF Word	536	164711
	0*164711		537	164711

TABLE 5-2. BOOTSTRAP PROGRAM INSTRUCTIONS (cont.)

Label	Mneumonic Instruction	Notes	Address	Content
		BOOTSTRAP Assembly No. 7024774-15: paper-tape, level 6, octal, channel 13 program used with I/O Console Mk 77 Series		
Start	SIL*0	Set Interrupt Lockout	500	503400
	ENTSR*10	Enable SR Stack 0	501	507310
	ENTBK*366	Index for Control Memory	502	360366
	ENTAL*CODE	Ignore Interrupt Code	503	120537
Store	STRALB*11	Store Code in Control Memory Address	504	450011
	BJP*STORE	All Codes Stored?	505	730504
	EXF*13	Yes - Turn On Reader, Start Read	506	501313
	0*CODE	Reader Code	507	0005
	0*CODE		510	000537
	ENTBK*0	Clear B	511	360000
Next	STRB*123	Total Words Input Count	512	420123
	ENTALK*2	Index for 3 Frames of Data	513	700002
	STRAL*124	Store Index	514	44012
	ENTALK*0	Clear AL	515	70006
Boot	IN*13	Input - Channel 13	516	501113
	0*122	Input Location - 1 Word	517	000122
	0*122		520	000122
Wait	SKPIIN*13	Is Word In?	521	502113
	JP*WAIT	No-Loop Until In	522	3405
	LSHAL*6	Yes-Position Data	523	504606
	SLSET*122	Insert Data in AL	524	510122
	JPALNZ*			
	COUNT	Data Equal Zero?	525	630530
	ENTAU*123	Yes - Check if Data or Leader	526	100123
	JPAUZ*BOOT	And Handle Zero Frames	527	600516
Count	ISK*124	Decrement Frame Count	530	570124
	JP*BOOT	Get Next Frame	531	340516
	STRALB*536	Store Word in Memory	532	450536
	BSK*WORD	All Words Input?	533	560536
	JP*NEXT	No - Do Next Word	534	3405
	JP*540	Yes - Exit	535	34054
Word	0*147	Input Count	536	000147
Code	0*503151	Enable Interrupt/Reader Code	537	503151

minimum of circuitry is working and proceeds to use this circuitry to test the rest of the logic. The first two routines (FAULT and SKIP/STOP) are normally bypassed and run over the rest of CONTBASI. PRELIM, which is the first section usually run, relies initially on only the ENTAL and STRAL instructions being worked and builds from there as indicated in the following explanation of sequential operations:

1. CONTBASI (First Routine). This subroutine checks capability to recognize fault instructions and checks proper referencing of the fault-interrupt entrance address. The routine first stores stop instructions in selected interrupt entrance addresses and stores a jump back to the routine in the fault-interrupt entrance address. When a fault instruction (fc=00) is executed, the proper interrupt entrance address must be selected or the program will stop to indicate an error in the interrupt address translation circuitry. If no interrupt is generated, an error stop also occurs. This procedure is repeated for function codes 01 and 77. Each function code error is recorded and the final error is displayed in the AL register. This routine terminates by looping at address 01000 where a visual inspection for the fault light being set is required.

2. SKIP/STOP Subroutine. This sequence of instructions checks the circuitry associated with the SKIP/STOP control logic. With all keys initially enabled, individual skips are executed on each key to check each key and bit path through the K register. A skip with no bits set is also done to insure that a skip condition is not erroneously enabled. Next, a stop code on all bits is executed and a visual inspection of all stop lights being lit is required. If the stops are proper, all keys must then be released. Upon restarting the program, a stop on all keys is executed to insure that no stop occurs. The routine then proceeds to execute individual skips to insure that no skips occur.

3. PRELIM Subroutine. This sequence of instructions checks functions required to execute the executive and functions used for verification purposes in the remaining subroutines. The fault-interrupt entrance address is first loaded so that if any malfunction causes an erroneous fault interrupt, a controlled analysis of the fault occurs. The first hardware area checked by this routine is the compare stage circuitry and conditional jump for the equal condition. The return jump instruction is then checked. This is accomplished by first insuring that the jump occurs to Y + 1 rather than Y and that P+1 is properly stored at address Y. The indirect jump is then checked by verifying that the conterts of address Y go to P so that proper addressing control occurs. After this the routine checks the P register bits for being dropped by transferring control to the upper address area of the stack and executing a jump back to the program. Following this is the executive which properly references each subroutine sequentially.

4. DEEDEE Subroutine. This sequence checks the ability to properly enter the AU register and the ability to properly store the AU register. Function translation is checked along with the data paths into and out of the AU register. Each bit position is verified for both the set and cleared states. The patterns used are 353535 and its complement, 424242. If an error stop occurs, AU contains the result of the ENTAU instruction and AL contains the result of the STRAU instruction.

5. KONG Subroutine. This sequence checks the ENTALK, ADDALK, and the

conditional jump instructions. The ENTALK instruction is checked by first insuring that a pattern of 7777 is sign extended to set all bits in AL and then a 3777 pattern is used to insure that zeros are extended into the upper bits of AL. Following this the ADDALK function code is checked; the pattern used is not significant. The final portion of this routine checks the conditional jump instruction for both AL and AU. All cases of the zero, not zero, positive, and negative are checked to insure that when the condition is satisfied the jump will occur and when it is not satisfied the next instruction is executed. A single error stop is used for all conditional jump errors. The AL register has a unique bit set for each conditional jump error detected.

6. SLUP Subroutine. This sequence checks selective and complement instructions. Complementary patterns are used to verify all logic involved. The selective set is the first instruction checked. The AL register is loaded with a 353535 pattern and then selectively set with a 424242 pattern. The patterns are then reversed and the procedure repeated. Next the selective clear instruction is checked by loading AL with a 070707 pattern and selectively clearing with a 707070 pattern. These patterns are then reversed and the procedure repeated. The selective complement function is then checked by using the same patterns as for the selective set. Following this the selective substitute is checked by loading AU with a 707070 pattern, AL with a 353535 pattern, and using a memory pattern of 424242. The result is verified for inserting the memory pattern into AL for those bits which are set in AU. This procedure is then repeated by using a 070707 mask in AU. The complement AL function is now checked by complementing a 424242 pattern and the complementing a pattern of all zeros. Next the complement AU function is checked by complementing both a 353535 and a 424242 pattern and then conplementing a pattern of all zeros. Finally, the double length CPA instruction is checked by using a 424242 pattern in both AU and AL and also by complementing patterns of all zeros in both registers.

7. ICEBERG Subroutine. This sequence of instructions checks the following functions: enter and store ICR register, ENTBK, ENTB, BJP, BSK, ISK, and the enter with B modification instructions. The enter and store ICR instructions are checked in combination, one at a time from register 10 down thru register 1. The first check determines that the STRICR does not clear the upper bits of the stored location and that a value other than zero is stored. When an incorrect ICR is translated for, an error display is built into AL which sets the bit in AL corresponding to the register(s) that failed. The ENTBK function code is checked next by using a 7777 pattern and verifying that a sign extension occurs to set all bits. All B registers are then checked for their ability to be cleared. Following this, the ENTB function code is checked; the pattern used is not significant since the B register has been verified for holding data. The B register jump (BJP) and skip (BSK) circuitry is now tested. With the B register set to 000001, a BJP is executed. If no jump occurs, the program proceeds to an error isolation routine which analyzes the $B \pm 1$ and $B = \emptyset$ evaluation network. If the jump occurs, another BJP is executed. This time the jump should not occur since the B register should have been decremented to 000000 by the first BJP instruction. If all is not proper, the same error routine is again referenced. The BSK is checked by first skipping on a value of 000001 with $B = 0$ and insuring that the skip does not occur. The BSK is then executed again and this time verified for the skip occurring.

An error detected in the BSK function also references the same error isolation routine as used by the BJP instruction. The program next checks the index skip instruction. This is done in the exact same manner as the BSK by first insuring that no skip occurs and next verifying that equality exists so the skip does occur. Again, the same error isolation routine is referenced if a malfunction is detected. The final portion of this subroutine checks the enter with B modification function codes, namely, ENTAUB, ENTALB, ENTBB, and ENTBKB. The patterns used are not significant since they have been previously tested.

8. CAMEO Subroutine. This sequence serves to check all B modified portions of instructions that have been checked previously as well as the store zero (CL), store address (STRADR), compare mask (CMSK), and indirect return jump (IRJP) functions. The routine first checks the store zero function by storing into a non-zero address. The next portion of the routine then checks the following functions: CLB, STRBB, STRALB, STRAUB, and CMALB. Various patterns are used in checking the function codes but the patterns are significant only for insuring, for example, that a STRAUB cannot be interpreted as a STRALB; thus only the function code is being tested. Following this, the STRADR is checked by insuring that the lower 12 bits of AL (7777) are transferred to an address that is cleared and that the upper 6 bits of the address are left cleared. The compare mask instruction is then checked by loading AU with a 077777 pattern, AL with 677777, and then comparing with a 177777 pattern and insuring that the equality condition is met. The compare mask with B modification is checked in a similar manner. The jump with B modification (JPB) is checked by merely insuring that program control is transferred to the proper address. The indirect jump is checked by first insuring that proper addressing control is maintained (the next instruction executed is specified by the contents of the address to which the IRJP was executed plus one). The second portion of the IRJP check verifies that the initial P register value plus 1 is stored at the proper address (address specified by contents of address to which IRJP was executed). The final portion of this subroutine checks the selective substitute with B modification instruction.

9. SUPEREG Subroutine. This sequence of instructions serves to check the SR functions and the ability to enter any bank and return to the main program bank properly. The enter and store SR instructions are tested first. The SR register is verified for holding data by using 10 and 07 patterns. A 30 pattern is then used to verify the upper SR bits. Enter and store AL instructions with B modification are used to check that data is properly transferred to other banks. An indirect jump is then used to transfer program control to another bank and the indirect jump stored in the other bank is used to return to the main program bank. This procedure checks the upper P register bits for being dropped. It should be noted that whether all P register bits can be checked depends on the size of the memory unit. The final portion of this routine checks the SR to P enables by using B modification to store data in bank 1 and using SR modification to retrieve the same data.

10. KAYADDER Subroutine. Initially, this sequence checks the scale factor instruction and then uses this to check the K adder. The scale factor shift portion of this routine checks that AL and AU are properly shifted and that address 17 contains the proper shift count. To accomplish this, address 17 and AU are initially cleared an AL is set to 000001. A scale factor with a

maximum count (77) is then executed and AU, AL, and address 17 verified for values of 200000, 000000, 000035 respectively. It should be noted that if the scale factor produces no effect on AU and AL, the program exits to the shiftless routine which analyzes shifting problems. If the scale factor executes properly, the remainder of the K adder is checked by running all possible counts through the adder. This is done by repeatedly incrementing the count and comparing the result until the maximum K count is achieved. If an error is detected, the program stops immediately and displays the incorrect count.

11. SHIFTLESS Subroutine. This sequence tests capability to shift properly. The AL register shifts are thoroughly tested before executing any AU or A register shifts. The AL register is initially tested for dropping a bit in any bit position by setting all bits of AL and then performing a left shift of AL by 1. The check for picking up a bit in any bit position requires AL being set to zero and then performing a left shift of AL by 1. Next AL and AU are entered with patterns such that upon shifting AL, the error is detected. Following this, the left shift end-around circuitry is checked. Once the left shift AL is verified, the right shift AL is checked for proper sign entension capabilities. After the AL shifts have been verified, the AU shifts are tested. The AU shift section first checks the left shift for picking up and dropping bits and then the end-around shifting. The right shift similarly checks the data transfers and then the sign extension circuitry. With both AL and AU shifts completely checked, the A register shifts are tested. This is accomplished by using patterns such that when a left shift is performed, the upper bit of AL must be transferred to change the state of the lower bit of AU and similarly for the upper bit of AU being gated end-around into the lower bit of AL. This is done for both the set and cleared states of each bit. The right shift is then checked for sign extension and for the proper data in both AU and AL. The final portion of the subroutine (PICKAPLUM) performs a complete analysis of the B = 1 and B = $\emptyset$ circuitry.

12. ADDSUBAD Subroutine. This sequence uses the ADDAL instruction with various test patterns to check the main adder circuitry. The borrow generation logic is checked using the ADDAL instruction. Each pattern used has a unique error stop and the failing pattern is displayed in AL. The ADDA and the ADDAB function codes are also checked. Following this, the DKPNOV is executed to clear the overflow designator, and the conditions are established to check the SKPOV, SKPNOV, and SKPNBO instructions for both the conditions in which the skip should occur and in which the next instruction should be executed. The SUBAL, SUBALB, ADDALB, SUBAB, and SUBA function codes are then tested using specific patterns.

13. MUDDIVE Subroutine. This sequence serves to check the multiply and divide circuitry as well as the round (RND) function. The multiply operation uses combinations of four separate patterns which are chosen to cover all possible sign combinations as well as exercise the additional multiply logic. Since a program loop is used to select the patterns, the MULALB and DIVAB functions are used. The patterns, in order of use, are as follows: (133331)(611116), (133331) (232232), (544445) (611116), and (544445) (232232). The first pattern is loaded into AL and second pattern is the multiplier. The divide checks uses the same routine and patterns as the multiply did. The first pattern is loaded into both AL and AU and the second pattern is the divisor. In both the multiply and divide tests, the AU and AL register are each verified for proper results. If an error

is detected, a bit is set in the BENERROR location. After all four multiply and four divide operations are completed, the flag is checked. If the flag is clear, the program proceeds; otherwise, an error stop occurs and AL has a bit set for each error detected. After the found function is checked, the MULAL and DIVA functions are checked.

14. PARITY Check. This is a subroutine which checks the parity functions and the odd and even parity evaluation network by setting all bits in AU, loading AL with the various patterns, and then executing a SKPEVN with 14 different patterns. If an error is detected, a bit is set in an error flag. Two separate error flags are used, one each for odd and even patterns. After all patterns have been run, the even error flag is loaded in AL and the odd error flag is AU. The program then checks for the registers being cleared; if not cleared, an error stop occurs with the error flags displayed.

15. INK4 (Intercomputer Modify and Move Routine). This routine is utilized when an intercomputer load is to be made from an operational computer to a computer under test. This routine can transfer the diagnostic program to either bank 0 or bank 1. INK4 modifies the diagnostic program to allow for execution from memory bank 1 locations (except IOTA which is run in its normal location after being moved by the MEDIA program). This routine is also the vehicle for sending the diagnostic programs to the computer under test.

5-20. AIMAT. The AIMAT routine provides automatic analysis and testing of intermittents by monitoring the controlled recycling of the diagnostic program at the termination of each pass through the program. Monitoring is accomplished by means of direct return jumps if AIMAT is referenced by CONTBASI or by indirect return jump if referenced by MEDIA or IOTA. Return jumps to AIMAT are coded into the diagnostic program at the end of the program indicating a good pass (no error detection), and at points in the program for error detection. Return to the diagnostic program is imposed by AIMAT.

1. The diagnostic program is divided into two parts. The first part (addresses 706 through 2446) is not under the control of AIMAT but tests the necessary functions for AIMAT to be referenced and executed. A malfunction in this part of the program causes an immediate termination of the program. The second part of the program causes an immediate termination of the program. The second part of the program (addresses 2447 through 27400) is controlled directly by AIMAT. At the end of the diagnostic program AIMAT is referenced, this will signify a complete pass through the diagnostic with no error detected. In this way AIMAT has control over the diagnostics in the case of an error detection or a no-detect. AIMAT determines which of the two cases exists based on the address which referenced AIMAT. The execution of a RJP or IRJP instruction to AIMAT terminates the pass through diagnostic program. A pass is the execution of the diagnostic from CONTBASI routine (address 706) through to the execution of a RJP or IRJP to AIMAT instruction.

2. The entry to AIMAT from a point in the diagnostic program between addresses 2447 through 27400 causes AIMAT to alter the diagnostic program such that tests beyond the point of entry will not be executed by the program. This is done to concentrate further testing on the area of the program affected by the malfunction.

3. Program truncation is accomplished by replacing the reference to the next test with an IRJP to the RUNENT subroutine. RUNENT is the no-detect entry point once a detect has occurred and is executed on every subsequent no-detect pass through the truncated program. Its function is relative to the function performed by the jump to AIMAT instruction at the end of the diagnostic program; that is to indicate a no-detect entry point to AIMAT. If the diagnostic program is executed error free it is automatically recycled by AIMAT five times in an attempt

to detect an intermittent malfunction before terminating in an error-free condition. If the program detects a malfunction, it notes the detection and automatically modifies itself to cycle all tests up to and including the subroutines that detected the malfunction. The program continues to recycle these tests until either five consecutive detects with identical P register values are produced or ten no-detects passes are made. If five consecutive detects are made with identical P register values, the program again modifies the diagnostic by storing an error stop instruction back into the diagnostic at the point of the malfunction detection and recycles the tests to run to this error stop. If five consecutive detects with identical P-registers value cannot be identified after twenty attempts the program terminates by coming to a stop 5 and displaying the P value in the C odd register and the B, AU, AL, and CE values in the respective registers.

4. The following tests the important program stops and then respective P registers values:

a. The stop associated with the five consecutive detects with identical P values would be back in the diagnostic at the point of the detected malfunction.

c b. P = 5052,this is the END OF TEST stop, it indicates that the entire diagnostic has cycled six times without a detected malfunction.

c. P = 5246, this is the INTERMITTENT ERROR STOP. It indicates that the program could not find five consecutive detects with identical P values, or two consecutive detects in twenty attempts, or two equal P values in twenty attempts, or that after five consecutive detects with equal P values there has been a new malfunction detected, or that after a detect has occurred the program makes twenty passes through the no-detect routine RUNENT. The intermittent error stop is at P = 45246, when the program is in bank 1 and after intercomputer load of the diagnostic program into bank 1.

d. P = 5013 This is a maintenance aid stop.

5. A historical data area is provided by AIMAT which provides data on P, AU, AL, and B register values for the symptom of the last malfunction detected and the P values for symptoms of previous malfunction detected during the course of program execution. This data is stored in the following addresses:

a. First P-5371
b. First AU-5372
c. First AL-5373
d. First B-5374
e. Last AU-5331
f. Last AL-5332
g. Last B-5333
h. Last P-5334
i. Second from last P-5335
j. Third from last P-5336
k. Fourth from last P-5337
l. Fifth from last P-5340
m. Number of Attempts - 5350
n. Number of Detects - 5352
o. Number of Consecutive Detects - 5354
p. Number of No-detects - 5376
q. Number of times subroutine was recycled - 5401
r. Number of passes allowed - 5362

6. To extend the testing of AIMAT to further isolate an intermittent, the following locations may be changed. Address 5362, normally set to 5, may be set to any number to cycle the diagnostic test for as long as the operator wishes.

P = 5356, normally set to 20, allows for 20 attempts to find 5 consecutive detects before termination. May be set to any number by operator to enable isolation of intermittent malfunction.

P = 5377, normally set to 10. This is the number of no-detects passes allowed to cycle before termination. It may be enlarged but should not exceed the maximum number of attempts (P= 5536 normally 20). This is a RUNENT routine.

5-21. MEDIA. Media is the memory test for Digital Computer Mk 152 Series. It can test any of the available memory options such as any control memory or main memory size (up to 65K) or any bootstrap type (paper or magnetic tape). MEDIA tests all of main, control and bootstrap memories. All interrupts are locked out when MEDIA is run. MEDIA requires the operator to manually enter parameters in AL to the octal number of memory stacks. The program consists of an executive and several subroutines as follows:

1. CONTROL Subroutine. The executive first references the CONTROL subroutine which checks control memory by storing and reading back a pattern of of all ones and then a pattern of all zeros. Addresses 000-077

are tested first, then 200-277, and finally 400-477 and 600-677 are tested if 256 words of control memory are available. If an error is detected in any group, an isolation routine attempts to correlate the error to a specific hardware area.

2. BOOT Subroutine. The executive next references the BOOT subroutine. From here the TRANBOOT and DIOBOOT subroutines are referenced. The TRANBOOT subroutine checks the bootstrap address translation for open or shorted transistors and the DIOBOOT subroutine checks for diode failures on bootstrap address translation cards. Each subroutine uses a bootstrap base address which is incremented by one if testing the group selection cards and by 10 if testing the diode selection cards. For group selection, locations 500 through 507, 510 through 517, and so on are tested against each other for bit problems. For diode selection, location 510, 520, and 530 are tested against each other as are 511, 521, 531 and so forth. In this manner, the group and diode selection tests are independent of each other. The remainder of the bootstrap test consists of checking for a single bit being dropped or picked up throughout bootstrap memory. This is done by the BOOTSTRAP and BOOTPICK subroutines respectively.

3. MAIN Subroutine. Following the check of bootstrap memory, the executive references the MAIN subroutine. This subroutine tests that area of memory designated by the limits set in AU and AL in the executive. Locations 40000 through the highest address in memory are tested first. After locations 40000 through the upper memory address have been tested, the executive repeatedly references the MAIN subroutine to test locations 100-177, then 300-377, 540-577, and finally locations 400-477 and 600-677 as they are not designated as control memory. The MAIN subroutine does no testing itself but references three separate subroutines (ALLONES, ALLZEROS, WORSTPTTN) to accomplish the testing as follows:

a. ALLONES Subroutine. The ALLONES subroutine stores a pattern of all ones into all memory locations in the area being tested. It then reads the patterns back and verifies them individually. If an error is detected, the BITSDROP routine is referenced to determine if only a single bit is being dropped or if more than one bit is dropped. If a single bit is dropped, the SBDROP subroutine is referenced. If multiple bits are dropped the MUDROP subroutine is referenced. Both of these subroutines first reference the SELECT subroutine to determine in which stacks the failure occurred. The SBDROP subroutine then references the SBDROPA, SBDROPB, or SUBDROPC routines depending on the size of main memory. Each routine then branches to an error stop. At each error stop, the AU and AL registers display significant data which is defined in the program listing. The MUDROP subroutine also references three separate routines (MUDROPA, MUDROPB, MUDROPC) and performs a similar analysis. If more than 32K of memory exists, the program may reference corresponding routines (MUDROPD, E, F and SBDROPD, E, F).

b. ALLZEROS Subroutine. The ALLZEROS subroutine is identical to the ALLONES subroutine except it checks for bit being picked up by using a pattern of all zeros An identical approach is now used in that a BITSPICK routine is referenced to determine if a single bit or multiple bits are being picked up. From here on the SBPICK or MUPICK subroutines are referenced and the testing proceeds until an error stop is referenced by one of the routines for single bit failures or by one of the routines for multiple bit failures.

c. WORSTPTTN Subroutine. The WORSTPTTN subroutine tests the stability of the memory.

4. MOVE Subroutine. When the above tests are completed for each memory area, the program is moved by the MOVE subroutine from bank 0 to bank 1. Program control is then transferred to bank 1 and the MAIN subroutine is referenced to test that area of memory in which the program was originally located. Following a successful test of the memory area, the program is moved back to bank 0.

5-22. IOTA. IOTA is the input/output test, and execution of this program requires a strict off-line attitude, no request on any channel, no response from any peripheral device, and no interference between odd-even Channel pairs. IOTA can be run in either the non-jumpered or the jumpered mode. To minimize the removal of I/O external cables, the test has been written to perform the non-jumpered mode first. If an error is detected, the jumpered mode does not have to be performed. If an error is not detected, then the jumpered mode should be run in selected pairs as defined in Chapter 8. Only the non-jumpered tests are run if skip 3 is set. Input/output test routines consist of the following:

1. IOTA Executive. This routine controls the order of execution of all the other main tests in the program. Part of IOTA is located in the lower part of memory stack 2 with the bulk of the test in stack 1. The UPRAM (unpack parameters) routine unpacks and stores parameters entered during the parameter entry stop in CONTBASI. The CHSET (channel set) routine determines the value of the channel parameter that was entered. From this, CHSET sets a value in the index register that is used in looping when checking all channels, and also storing a value in CHTBL (channel table) locations so that the proper table is referenced for various I/O sizes. The HONINT (honor interrupt) routine clears all interrupts by storing RIL's (remove interrupt lockout) in all interrupt entrance addresses. The TERMINAL (terminate all) routine terminates all actives. The SRSMAL (set resume all) routine sets resumes on all channels.

2. I/01 Routine. This sequence checks for a constant input, output, and external function active. The program does the checking by doing a skip if the function being checked is inactive (SKPIIN, SKPOIN, SKPFIN). If the function is active, an error is recorded without coming to an error stop immediately. All channels are checked in order, with the highest channel being checked first. After checking input, output, and external function active on all channels, the error storage locations are checked to see if any errors were recorded. If an error was recorded, the program comes to an error stop with the contents of the error storage locations displayed in AU, AL and the low even C register. If no errors, the test goes to the next routine. The JMPRD (jumpered) routine finds the lowest even non-jumpered channel and uses it in the display on C routine. RECORDI, RECORDO, and RECORDF are error recording routines. The DISPOC (display on C routine) causes one of the error storage locations to be displayed on C_E lower register.

3. I/02 Routine. This sequence checks for the false running of the real time clock, false input data requests, false external function requests, and also checks the ability to clear the active flip-flops. The RTC portion of the routine clears RTC word location in control memory and then waits until this location has had time to be incremented by the false running of the real time clock. An error stop occurs if the RTC word was falsely incremented. The false request portion of the routine first initializes the I/0 control memory locations. The input function is then set active (SIN). Control memory is now checked to see if the contents have been changed and if so, an error stop occurs. Next, a skip on inactive

each channel) is then set depending on whether or not the skip occurred. The program then proceeds to test the remaining channels. After all channels are tested, the error flag is checked for any errors. If an error occurred, it is displayed in AL and the program comes to an error stop. If the flag is clear, no error occurred and the program exits to the next routine.

5. I/0456 Routine. The sequence checks for the false termination of all lower channels when the highest channel in the group is terminated. Initially, all functions (Input, Output, External Functions) are set active by the STALAC (set all active) routine. The input is then terminated on the highest numbered channel and all lower numbered channels are checked for false termination. The highest numbered channel is then set active and a lower channel is terminated. This time the highest numbered channel is checked for false termination. For both cases, an error stop occurs if an error is detected. When this subroutine has been executed for input termination, the I/0 instructions are modified and the same exact subroutine is then used to check output termination. Finally, the same subroutine is used to check external function termination.

6. I/0789 Routine. This sequence of instructions checks for erroneously setting a channel active. This routine is similar to the previous routine (I/0456) in that one subroutine is used to check all functions (input, output, external function) for being erroneously set active. Input is tested first by terminating all channels by using the TERMAL subroutine and then setting the highest numbered channel active. The lower channels are then verified for still being inactive. The highest channel is then terminated and a lower one is set active. The highest numbered channel is now checked for still being inactive. For both cases, and error stop occurs if an error is detected. Following the check for input activation, the I/0 instructions are modified to handle output and the same routine is referenced. Finally, the external function is checked by looping through the same routine after again updating the I/0 instructions to external functions.

7. I/010 Routine. This sequence checks for proper interaction of function enables. All functions are initially terminated and then the EF is set active. The input and output functions are then tested for still being inactive. This procedure is then repeated for all channels from the highest through the lowest. The error flag is checked and if any error occurred, the flag is displayed and an error stop occurs.

8. I/011 Routine. This sequence checks the proper storage of BCW in control memory for input, output, and external function. A check is also made of false termination occurring without doing a force. The program initially terminates all functions and then clears all control memory buffer control word storage locations using the CLCWAR routine. The input function is then set active and checked for false termination. The BCW storage is then verified for proper operation. The terminal address control word (TACW) is checked first. If it is stored in the proper location, the storage of the initial address control word (IACW) is checked. If the TACW is not stored properly, the SRCHM (search memory) routine is used to search for it in other areas of control memory and selected main memory address. After completing the check on input BCW storage, output and external function BCW storage is verified. After checking all three functions on one channel, the test is repeated until all channels have been tested.

9. I/012 Routine. This sequence checks the EXFOV circuitry and the processing

of requests as well as premature and normal buffer termination. The program first initializes control memory and then generates a two-word EF buffer and checks for early termination after only one force. A delay is set up using a left shift on AU until it becomes negative and then a skip on function inactive is executed. If the skip occurs, the buffer is terminated prematurely and the program proceeds to analyze the buffer control word equality network (B≠Z0) before coming to a stop. If the buffer did not terminate prematurely, the program checks that the resume flip-flop is set. Following this the proper current address control word (CACW) is checked to see if it was updated. If it was not changed, the program searches for other control memory locations being altered. If the CACW was updated, the program verifies that it was properly incremented. A second EXFOV is then generated and after a delay, the resume is again verified for being set. Next a skip on function inactive is executed to verify that the buffer did terminate properly. If it did not terminate, the program analyzes if the buffer stayed active because of CDM. If the buffer did terminate properly, the entire above procedure is repeated for all channels. The final portion of the routine checks the B≠0 evaluation circuitry by using control words of 170000 and 007777 to check all bit positions.

10. I/013 Routine. This sequence checks the OUTOV circuitry and false input data requests. The routine is organized similar to the previous routine for testing the EXFOV circuitry. After control memory is initialized, the input area is set up and then a two-word output is done with only one force. After a sufficient time delay, the program checks for a false input data request by checking the BCW area and then a check is made for premature termination of the output. Next the output BCW is checked for normal incrementation by 1. The second and final word is then forced and a check is made for normal termination. The entire procedure is then repeated for the other channels.

11. I/014 Routine. This sequence checks the proper functioning of a backward buffer and a backward buffer with CDM (Continuous Data Mode). An output of one word is done with a backward buffer and the EF buffer control word address is checked to verify the proper order of the control words for the backward buffer. Next CDM is enabled. An output is now done and the program checks that the buffer does not terminate. It checks that CDM loads the new BCW and that the output is still active. It also checks that the BCW is properly transferred to the output buffer area.

12. I/016 Routine. This sequence tests for proper processing of output and EF monitor interrupts. The STINK routine stores stops in control memory except for the interrupt address desired, so that if a monitor interrupt references the wrong interrupt entrance address, the stop instruction will be read up and executed. Both output and external function monitor interrupts are checked for each channel.

13. RLJPEI Routine. This sequence tests for the proper releasing of the interrupt lockout by the RIL and IJPEI instructions. An output with monitor is done on channels 0 and 2. Channel 2 is then forced and a RIL is executed immediately. After a sufficient delay, the program checks that the monitor interrupt was honored. Next channel 0 is forced and the program verifies that this interrupt is locked out because of the previous interrupt. An IJPEI is then executed and the program checks that the lockout is released and the interrupt honored. The final portion of the routine checks that an output with force keeps the lockout set.

14. I/015 Routine. This sequence checks the ability of the SIL instructions t

lock out interrupts. An EF and OUT with monitor is done on each channel in turn with interrupts locked out and a check is made to see if interrupts remain locked out. Stop instructions are stored throughout control memory to note the error if any interrupt is honored.

15. I/017 Routine. This test checks channel priority. Using the output function with monitor and interrupts locked out, the highest channel and the next lowest channel are checked to see that the monitor interrupt was honored on the highest channel first. After the highest channel is checked against each of the lower channels in turn, the highest channel number is decremented by one. This is then considered the highest channel and it is checked against all the lower channels in turn until all combinations have been checked. (For example, channels are checked in pairs as follows: 17-16, 17-15, 17-14, 17-13,.... 17-0, 16-15, 16-14, ...16-0, ...2-1, 2-0, ... 1-0,) If an error is detected, the upper channel being used is displayed in bits 09-12 of AL and the lower channel is displayed in bits 00-03.

16. I/018 Routine. This sequence checks the proper honoring of interrupts when two functions are on the same channel (all channels are checked). An OUT and EF are done on the highest channel first and then a check is made to see that the EF function is honored before the OUT function. Stop instructions are stored in the output monitor registers to detect an output monitor interrupt being honored.

17. WAITCK Routine. This sequence checks the wait for interrupt (WTFI) instruction. After doing a forced output with monitor, a WTFI instruction is executed. Immediately following the WTFI instruction is a stop so if WTFI does not function properly, the error stop occurs.

18. CMPLKO Routine. This sequence checks the ability of the compare instructions to lock out interrupts. An output function is done and forced with interrupts locked out, then a RIL and compare are done immediately to see if the compare locked out the interrupt.

19. EXFEIC Routine. This sequence initially checks that no false external interrupts are honored and then checks that no false external function acknowledges are generated. This is done by using the STINK subroutine to store STOP instructions in control memory. The routine verifies the ability of the computer to correctly recognize and interpret the EF and EI functions associated with the communication process. Proper interrupt recognition is also tested. If any error is detected while attempting EXF/EI transfers, the basic IN/OUT control and EI lockout circuitry is tested to gather data for isolating the malfunction. External interrupt lockout circuitry is tested by the LOCKOB routine.

20. INOUTC (Input and Output Contol Test). This program tests the ability to do INPUT and OUTPUT on one channel at a time with the higher jumpered channel first. The ability to recognize and input data request and data is tested. The ability to recognize an output data request through the ODR gates is tested along with the ability to terminate an INPUT. The presence of the correct buffering mode is tested.

21. CSDATA (C Register, Input Selector and Data Test). This routine checks the transfer of data on each of the jumpered channels. A four-word transfer is done using a pattern of all zeros, a pattern of all ones, and 525252 and 252525 patterns respectively. The first word is then checked for being all zeros and then the 525252 and all ones patterns are checked. After testing the high numbered channel, the lower channel is tested. If an error is detected the

program continues to test whether a single bit or multiple bits were dropped or picked up. Prior to coming to the error stop, error displays are loaded into AU and AL to indicate the type of error (picked or dropped bits, single or multiple bits, high or low numbered channel, data of last channel failure) that occurred. The error codes are defined in the program listing.

22. MONTIN (Input Monitor Interrupt Test). This routine checks the ability to properly honor the input monitor interrupt on both jumpered channels. The higher numbered channel is tested first. Stop instructions are stored throughout control memory and an instruction to return to the main program is loaded into the appropriate input monitor register. An input with monitor is then initiated and the program verifies that the input monitor interrupt is generated, the appropriate interrupt location referenced, and the input terminated. Any error causes the program to stop.

23. ATIME (Acknowledge Timing Test). This routine checks the acknowledge timing for the OD/EF and ID acknowledges if slow interface. It also checks the proper timing of two consecutive force instructions with either fast or slow interface. The program uses the SET ACTIVE and SKIP INACTIVE instructions to insure that the EF, OD, and ID acknowledges are each up for a minimum of 10 micro-seconds and have dropped after 18 micro-seconds.

24. CDMIN (Input Continuous Data Mode Test). This routine checks CDM operation. The program checks that CDM is not detected on the first IDR and that CDM prevents the active from being terminated by the second IDR. The reload word is checked for correctness and the program also verifies that the buffer address control word storage was modified.

25. FUNPGA (Group Function Priority Test). This routine checks the function priority circuitry of the group associated with each jumpered channel. The priority alternation of input and output functions is also checked. This routine is not run if Digital Computer Mk 152 Series is in the standard 1218 mode. The function priority is tested using the PR2345 subroutine. Since priority alternated between the output and input requests, the PR2345 subroutine is first referenced when the output request was the last condition present. Priority is tested between the external interrupt, external function request, output request, and input request. The PR2345 subroutine is next referenced when the input request was the last condition present. The entire routine is then repeated for the group associated with the lower channel.

26. SPECIA (Special I/0 or Real-Time Clock and Intercomputer Test). This routine assumes that the single-channel normal tests have been successfully completed. The jumpered channels are placed in intercomputer mode and the RTC is enabled manually. The RTC circuitry is then checked. The ability to properly increment the RTC and to receive all the associated RTC interrupts is tested. If a failure occurs, the intercomputer resume fault interrupt is requested for added malfunction isolation data. If an error is detected, a specific bit is set in AL to indicate the different RTC failures that were detected. The meaning of each bit is stated in the program listing. The intercomputer portion of this routine tests the input and output operations for both channels. Proper resume operation is checked.

27. DUALMODE (Dual Mode Test). This routine checks the data transfer of the 36 bits word in the computer dual-mode operation. All bit patterns are transferred

from and back into the computer in approximately 40 seconds. All transfers are checked and compared, and proper computer dual mode of operation is verified.

5-23. ZPAC UTILITY PACKAGE. The ZPAC utility package performs limited functions on Digital Computer Mk 152 Series with either a Digital Data Recorder Mk 19 Series or an Input/Output Console Mk 77 Series. The main functions performed provide for facilitating execution of the diagnostic tests. Secondary functions performed by the utility package provide aid in the building of new test tapes. The formatting of the MT is compatible with bootstrap assembly No. 7046680-00 on any channel. The functions provided are the following:

1. Absolute - Bioctal Paper Tape Load.
2. Absolute - Bioctal Paper Tape Dump.
3. Magnetic Tape Load (NTDS Format).
4. Magnetic Tape Dump (NTDS Format).
5. Buffer Control Word Compatibility Changer.
6. Channel Changer, PT or MT.
7. Instruction Word Format Selector.
8. Inspect and Change

5-24. SCHEDULED MAINTENANCE

5-25. GENERAL INFORMATION. This paragraph contains information on maintenance test procedures for Digital Computer Mk 152 Series. These procedures are prescribed in the Planned Maintenance Sub-System (PMS) portion of the ships' 3-M Material Maintenance Management System (3M) Manual, OPNAVINST 4790.4.

5-26. PMS is applicable to all naval vessels and includes installed equipments except medical equipment, Fleet ballistic missiles and nuclear power plants and associated test equipments.

5-27. The purpose of PMS is to provide all maintenance and material managers throughout the Navy with the means to plan, acquire, organize, direct, control, and evaluate manpower and material resources expended or planned for expenditure in support of maintenance. Thus, it is essential that all hands recognize the importance of the System, and understand the role it plays in assisting management to improve the material readiness of equipment in the Fleet.

5-28. SCHEDULED TESTS. Each ship is provided with a simple and standard means for planning, scheduling, controlling, and performing planned maintenance of all equipment. The maintenance actions prescribed are the minimum required to maintain the equipment in a full operable condition, within design specifications. If performed according to schedule, these maintenance actions will provide the means to identify parts requiring replacement prior to failure. PMS procedures are, therefore, preventive in nature in that they are designed to prevent future equipment failures which might otherwise result in repeated maintenance actions.

5-29. These PMS procedures and the periodicity in which they are to be accomplished are developed for each piece of equipment based on good engineering practice, practical experience, and technical standards. These procedures are contained on cards designated MRCs (Maintenance Requirement Cards). MRCs provide the detailed procedures for performing the preventive maintenance and state exactly who, what, when, how and with what resources a specific requirements is to be accomplished.

5-30. TEST AND SERVICING PROCEDURES

5-31. A Maintenance Index Page (MIP) contains a brief description of the requirements on the MRC(s) for each item of equipment, including the periodicity code, the manhours involved, the minimum required skill level and, if applicable, the related maintenance requirements. The MIPs for all equipments in a department are contained in a Departmental Master PMS Record. This manual also contains an index of the effective MIPs, called a LOEP (List of Effective Pages), and a correction page to record changes made. The department Master PMS Record is used by the department head as a scheduling tool when scheduling maintenance on the PMS scheduled forms, and also as a cross-reference guide. Additionally, each work center has a Work Center PMS Record which is identical to the Departmental Master PMS Record, except that it contains only those MIPs and LOEPs applicable to the work center. The division officers, work center supervisors, and maintenance personnel use these records for cross-reference purposes.

5-32. The planning and scheduling of maintenance requirements are accomplished on the cycle, quarterly, and weekly schedules. Transferring maintenance requirements from the MIPs for each work center contained in the Departmental Master PMS Record and scheduling them on the Cycle Schedule creates the ship's overhaul to overhaul maintenance schedule. Quarterly and Weekly Schedules are prepared, using the Cycle Schedule as a guide. Maintenance requirements indicated on the Weekly Schedule are assigned to specific personnel for accomplishment. Scheduled maintenance sections are crossed over with an "X" when they are completed, and actions not completed are circled and arrowed to a new schedule date. Quarterly Schedules are updated in the same manner.

5-33. Fleet maintenance personnel are provided with a PMS Feedback Report (FBR) form to report discrepancies and problems, and to request PMS software. All PMS FBRs are sent to Navy Maintenance Management Field Offices (NMMFOs) or to Type Commanders (TYCOMs) depending on the category of the FBR.

5-34. Planned Maintenance During Overhaul (PMDO) is designed for use during shipyard overhaul. PMDO provides planned maintenance procedures for the deactivation and reactivation of entire systems (instead of routine maintenance requirements) during the overhaul period. PMDO also provides reduced planned maintenance requirements for other equipment which will be used during the overhaul period. A PMDO package, consisting of separate layup and startup sections, is installed in a ship by the TYCOM 3-M Installation and Assistance Team prior to the commencement of overhaul. The layup portion of the package is completely self-contained and can be used without reference to other documentation. However, the startup portion makes extensive use of PMS MRCs (Maintenance Requirement Cards) and is, therefore, dependent upon adequate PMS coverage. The ship is responsible for determining exactly which shipboard systems will be scheduled for PMDO vice PMS and during what specific time period.

5-35. MAINTENANCE REQUIREMENTS. MRCs providing specific procedures for performing all required scheduled maintenance actions on Digital Computer Mk 152 Series are listed on the current MIP (SYSCOM Control No. 5CZ1000).

5-36. MRC facsimiles are no longer provided in OP's. Distribution of all scheduled maintenance materials is made by the Naval Maintenance Management Field Office, to Fleet units through the appropriate Type Commander.

5-37. CORRECTIVE MAINTENANCE

5-38. UNSCHEDULED MRCs. Corrective Maintenance (CM) of the Computer Mk 152 is performed in accordance with procedures delineated on unscheduled Maintenance Requirement Cards which contain no man-hours or rates. These cards are prepared as specified in NAVMATINST 4790.5.

5-39. Background Information. Corrective Maintenance procedures consist of actions required to restore malfunctioning equipment to an operational condition within predetermined parameters. They are intended to provide exact guidance during performance of alignment, adjustment, repair, or replacement procedures on faulty equipment after fault isolation techniques (see Volume 2) have successfully localized the trouble to a specific equipment. CM procedures were originally developed as part of the Planned Maintenance Systems for Surface Missile Systems (PMS/SMS). Prepared in MRC format, they were published as element (volume, chapter, etc.) 4 of PMS/SMS OP's. CM procedures are now available to the Fleet (and to shore activities if needed), on unscheduled 5" x 8" MRCs which are intended to enhance shipboard usability and to correlate corrective maintenance documentation with actual shipboard equipment configuration. Changes to existing PMS/SMS OP's were promulgated to delete CM procedures presented in book form and to re-locate related CM data.

5-40. YARD/TENDER CORRECTIVE MAINTENANCE

5-41. Maintenance information in this publication is limited to organizational-level maintenance. Procedures and instructions for performing maintenance which is beyond the capabilities of ships force will be made available to Yards/Repair Facilities by the In-Service Engineering Agent as Ordnance Data (OD's), specifications of various types, engineering drawings, and similar data.

★U.S. GOVERNMENT PRINTING OFFICE: 1975—0-683-611/4759

a. Inventory CM materials against the letter of transmittal.

b. Validate each work center CM manual to ensure accuracy of CM applicability as reflected on the List of Effective Page (LOEP) (see Table 5-3).

c. Annotate the LOEP as required to accommodate the particular ships CM requirements, organizations and work centers.

d. Using the annotated LOEP, inventory each work center's CMIP's.

e. Using CMIP's, verify that the correct CMC's have been provided.

f. Review the Master CM manual for completeness and organization.

g. Distribute CM documentation to work centers and ensure a sufficient number of CMC containers are installed.

h. Report any discrepancies in the CM material provided using the standard PMS feedback report form (OPNAV Form 4790.7).

Upon installation, the corrective maintenance manual and the related CMC's will constitute the only authorized CM procedure for the affected equipment as listed on the LOEP.

2. Weapons Department. The corrective maintenance material may be forwarded to a by a letter of transmittal for installation by Weapons Department personnel in lieu of a Documentation Installation Team. In this case, it is recommended that the Weapons/Fire Control Officer ensure the CMC's and associated materials are properly installed by accomplishing the following:

a. Inventory the complete package against the material transmittal which will include the following items in the quantity specified on the transmittal:

(1) One master and the required quantity of work center corrective maintenance manuals containing an introduction to the CMC program, the applicable List of Effective Pages and the applicable Corrective Maintenance Index Pages.

(2) One master deck of all CMC sets and one set of applicable CMC's for each work center.

(3) One master file box and one wood file box for each work center for CMC storage.

b. Distribute the corrective maintenance manuals to the leading petty officer of each applicable work center. Reference to equipment covered in each manual will indicate work center applicability.

c. Instruct each recipient of a work center manual to review the front matter describing the CMC program and to verify the applicability of CMIP's contained in the manual for his work center.

d. Conduct a joint meeting of all work center personnel, the system chief and the petty officer delegated the responsibility of maintaining the master CMC file. During this joint meeting, the following should be accomplished:

(1) Annotate the applicable work center codes in the work center column of the LOEP in the master and each work center corrective maintenance manual. Refer to Table 5-3.

NOTE: CMIP's may be transferred to other work center manuals depending on Weapons Department organization. If CMIP's are removed from one manual and inserted in another, make pen and ink changes to the applicable LOEP's of each manual involved.

(2) Distribute CM card sets to work center petty officers in accordance with the LOEP of the master manual and instruct each work center petty officer to inventory each card set against the appropriate CMIP. The four unique alphabetic characters of the syscom MRC Control Number on the CMIP and the appropriate CMC must agree

TABLE 5-3. SAMPLE MASTER LOEP

CMS/SMS DATE 1 June 1972

HULL NO. DDG- XX

CORRECTIVE MAINTENANCE

LIST OF EFFECTIVE PAGES

MIP NO.	EQUIPMENT	UNITS INSTALLED	WORK CENTER
5ABA000/U6-0	GMLS Mk 11 Mod 0	1	
5BBB000/U9-0	G&GM Director Mk 73 Mod 1	2	
5BBC000/U8-2	Radar Set AN/SPG-51C	2	
5CZ1000/U1-3	Digital Computer Mk 152 Series	2	
5FB6000/U1-0	Test Set Mk 474 Mod 2	1	
5FBR000/U1-2	Dynamic Tester Mk 37 Mod 0	1	
5FBT000/U1-0	Radar Test Set AN/SPM-15A	1	
5FBU000/U1-1	Signal Comparator Mk 3 Mod 2	1	
5HBD000/U1-0	Channel Selector Mk 4 Mod 1	1	
5ZBAAFR/U3-5	W.D.E. Mk 1 Mod 0	1	
5ZBAAH5/U1-0	Input/Output Console Mk 77 Series	2	
5HZ1000/U1-0	Motor Generator Set Mk 9 Series	2	
	Enter applicable work center codes in this column		

for each "R" card in the set.

e. Any deficiencies found in the CMC documentation during the implementation procedures should be reported immediately by the standard PMS feedback report, Form 4790-7. In addition, one copy of the material transmittal should be signed acknowledging receipt of the CMC package and returned to the sender (NSMSES/Port Hueneme).

5-40. CMC Storage Containers. All of the CM storage containers are standard Navy stock items and are available as follows:

Size	FSN
5" x 8" x 9 1/2"	7520-285-3148
5" x 8" x 15"	7110-273-8772

The number of CMC's will determine the size and number of containers required at each location.

5-41. Shipboard Implementation of Control Procedures. A master copy of the SMS Corrective Maintenance Manual containing a master LOEP and copies of all applicable CMIP's and CMC's is provided for Weapon's Department use. To ensure

maximum effectiveness and availability of CMC's, it is essential that certain control procedures be implemented. The following control procedures are recommended:

1. Maintain the master and work center CM card sets in CMIP number sequence and within each set, file cards in "R" number sequence.
2. Verify the entry of CMC changes in both the master and work center CM manual and files.
3. Order replacement of shopworn or damaged CMC's.
4. Report any deficiencies or discrepancies in CM documentation.

NOTE: Discrepancies or deficiencies in CM materials determined by the using activities are to be reported by the standard PMS feedback report form (OPNAV FORM 4790-7). It is important that feedback comments be as specific as possible. All blocks of the feedback form are to be completed. Replacement or additional copies of CMIP's and CMC's should be requested using the same OPNAV FORM 4790-7.

5-42. List of Effective Pages. The LOEP in the master SMS Corrective Maintenance Manual, is a list of equipment applicable to a department for which CM material is provided. The LOEP contains the following information (table 5-3).

1. The date of preparation of the LOEP.
2. Maintenance index page number.
3. The equipment component and/or system.
4. Number of units installed.

5-43. Corrective Maintenance Index Page. The CMIP, OPNAV Form 4700-3, contains the same information as the PMS MIP except recommended skill levels and man hours are not provided. The CMC's listed on the CMIP are to be used only when required and are not to be scheduled.

5-44. Differentiation of CMIP from PMS MIP is maintained by the letter U (Unscheduled) immediately after the slash following the Equipment Identification Code (EIC). The U is followed by a number representing the change level of the CMIP resulting from equipment configuration changes (ORDALTS). The last number (preceded by a dash) represents the change level of the CMIP resulting from changes other than equipment modifications.

5-45. Corrective Maintenance Cards. Corrective maintenance procedures are provided on 5" x 8" cards, OPNAV Form 4700-1. These cards provide maintenance personnel detailed guidance for the performance of a specific corrective maintenance action. The first page of all CMC's are overprinted with the notation "CORRECTIVE MAINTENANCE, DO NOT SCHEDULE." CMC's use the letter prefix designator "R" as the MRC code and are numbered sequentially 1 through 99. Equipment with more than 99 CMC's are numbered using a alphanumeric system for CMC's above R-99. This system uses the letters A through Z (excluding I and O) followed by the numbers 0 through 9 to represent numbers greater than 99. For example, R-98, R-99, R-A0, R-A1 thru R-A9, R-B0, and so on.

5-46. COMPUTER CORRECTIVE MAINTENANCE.

5-47. CM ACTIONS. All procedures required for performance of organizational-level non-scheduled maintenance of the Digital Computer Mk 152 Series are

available on CMC's. The original issue of CMC's was listed on syscom MIP Control No. 5CZ1000/U1-0 dated July 1971. A synoptic presentation of the non-scheduled maintenance actions listed on CMIP 5CZ1000/U1-3 dated April 1972 follows:

1. R-1. This CMC provides procedures for removing and reinstalling a logic drawer. These actions:

a. Expose test blocks during troubleshooting.

b. Allow access to plug-in modules for inspection or replacement.

c. Permit opening of a chassis to gain access to the wirewrap back panel and inter-module wiring.

d. Provide for removal of a drawer from the cabinet so that repairs may be effected.

2. R-2. This CMC provides procedures for removal/reinstallation of Power Supply PS-1. These actions are necessary to gain access to power supply components for adjustment, repair and replacement as well as for removal of PS-1 from the computer cabinet.

3. R-3. This procedure covers the steps required to remove the air filter from the computer cabinet for cleaning or replacement.

4. R-4. This procedure covers the steps required to remove and replace defective circuit cards from the computer drawer.

5. R-5. CMC R-5 provides necessary procedural steps for accomplishing the following CM requirements:

a. Replace Main Memory Stack.

b. Replace Control Memory Stack.

c. Replace Bootstrap Assembly.

6. R-6. This procedure provides coverage of main memory adjustment. X + Y Read/Write currents are adjusted to 820 Ma PP. Sense bias is adjusted to -6VDC.

7. R-7. This procedure consists of the steps required to make control memory voltage and current adjustments. Voltage adjustments are made for +17VDC, -16VDC, and for -7VDC. Read current is adjusted to 400-500 Ma P. Write current is adjusted to 300-400 Ma PP.

8. R-8. This procedure covers adjustment of the main memory strobe timing pulse. An integrated memory test program is used to compare the strobe timing pulse with the sen amplifier output. The required equipment is combination tool.

9. R-9. This CMC provides procedural steps for the following maintenance requirements.

a. Adjustment of Clock Cycle Timing. This procedure covers the steps required to adjust cycle period to 250 nanoseconds.

b. Adjustment of Phase Pulse Timing. This procedure covers the steps required to check the duration of the phase pulse. With NARROW/NORMAL switch in the NORMAL position, the phase pulse duration should be within the limits of 80 to 120 nanoseconds. With the NARROW/NORMAL switch in the NARROW position, the phase pulse duration should be within the limits of 50 to 70 nanoseconds.

10. R-10. This MRC provides procedures for adjusting the +3VDC power supply.

11. R-11. This MRC contains step-by-step instructions for adjusting the 104-volt and the 101-volt memory protection circuitry.

5-48. YARD/TENDER CORRECTIVE MAINTENANCE.

5-49. Maintenance information in this publication is limited to organizational-level maintenance. Procedures and instructions for performing maintenance which is beyond the capabilities of ships force will be made available to Yards/Repair Facilities by the In-Service Engineering Agent as Ordnance Data (OD's), specifications of various types, engineering drawings, and similar data.

APPENDIX A

A-1. DIGITAL COMPUTER TERMINOLOGY. Many terms which have an accepted meaning in the English language have a specific meaning when applied to computer operation and circuitry. This glossary is intended to delineate these terms and their specific meanings as they are written in this manual.

ABORT - A condition within the computer which results in the following sequential instruction being skipped or omitted.

ACCESS TIME - The time interval, characteristic of a memory or storage device, which exists between the instant information is requested and the instant this information is available to requesting circuitry.

ACCUMULATOR - A 36-bit addressable register, consisting of two 18-bit registers, AL and AU, used in arithmetic processes and commonly labeled the A register. A register is the principal arithmetic register and is used to hold sum, difference, product, or remainder during the final steps of arithmetic functions.

ACKNOWLEDGE - Indication of acceptance or issuance by the computer of signals on input or output lines to peripheral equipment. Abbreviated as ACK.

ADDER - A device which forms as an output the sum of two or more numbers presented as inputs. Related to the accumulator.

ADDRESS - A coded number specifically designating a computer register or a location within an internal storage device. Stored information is referenced by its address.

ADDRESSABLE - Capable of being referenced or entered by an address or instruction.

AND - A signal circuit with two or more input lines in which the output line supplies the desired signal only when all inputs are coincident signals. Synonomous with an AND gate and AND circuit.

ARITHMETIC - A section within the computer where arithmetic processes are performed and operands or results are temporarily stored.

AUXILIARY ROUTINE - A routine designed to assist in operation of the computer and in debugging other routines.

BINARY - A characteristic, property, or condition in which there are but two possible alternatives.

BINARY ARITHMETIC - A numbering system which uses two as its base and only the digits one and zero are recognized.

BINARY CELL - An information-storing element capable of two possible stable states.

BINARY NUMBER - A single digit or group of characters or symbols representing the total, aggregate, or amount of units by utilizing base two. Using the digits 1 and 0

to represent a quantity.

BINARY POINT - The radix point or root of the binary system.

BISTABLE - The capability of assuming either of two stable states; hence, capable of storing one bit of information.

BIT - An abbreviation of binary digit. A single character in a binary number representing the condition of a stage of storage within a computer. A unit of information capacity of a storage device.

BOOTSTRAP - A technique for loading the first group of instructions of a routine into storage; then using these instructions to load the remainder of the routine.

BOOTSTRAP MEMORY - A permanently wired, nondestructive readout memory containing the load routine for loading programs into storage. The program stored in nondestructive readout memory.

BORROW - An arithmetically negative carry; the additional subtraction of a one from the next partial difference. A borrow occurs in direct subtraction by raising the low order digit of the minuend by one unit of the next higher digit. A borrow occurs in binary subtraction when a digit of the minuend is a zero and the corresponding digit of the subtrahend is a one.

BUFFER - Transfer of data between the computer and peripheral equipments; intermediary storage between two data handling systems having different access times or formats.

BUFFER REGISTER - Final holding register for computer output to external devices or initial holding register for input to the computer; normally referred to as the C registers.

CAPACITY - Maximum size of a numb that can be processed in the storage locations of a computer.

CARRY - A condition which arises in addition when the sum of two digits in the same digit place equals or exceeds the base of the number system in use; a digit to be added to the next digit column; the process of forwarding the carry digit.

CHASSIS - A divisional unit of the computer containing plug-in circuit cards and/or memory devices and the necessary wiring and test points for maintenance, assembly, and repair.

CHECKSUM - Sum used in a summatio check.

CLEAR - To erase the contents of a storage device by replacing the contents with blanks or zeros. To reset a storage memory device to the zero state. The signal used to perform the clear function such as master clear.

CLOCK - Basic timing or master timing device used to provide the basic sequencing pulses for the operation of the computer.

CODE - A system of symbols for meaningful communication; a machine language program.

CODED PROGRAM - Procedure for solving a given problem through use of computer logic. The coded program may vary in detail from a simple outline of the procedure to be followed to an explicit list of coded instruc s.

COMPARE - To examine the representation of a quantity to discover its relationship to zero. To examine two quantities for the purpose of discovering identity or relative magnitude.

COMPILER LANGUAGE - A set of symbols used to abbreviate computer instructions and functions.

COMPLEMENT - A quantity expressed to the base N which is derived from a given quantity by a particular rule; frequently used to represent the negative of a given quantity. A complement on N, obtained by subtracting each digit of the given quantity from N-1, adding unity to the least-significant digit, and performing all resultant carrys. For example, the one's complement of 100011 is 011100; the nine's complement of 456 is 543.

CONTROL - Circuits within the computer which effect the completion of a function in the prepared sequence; the proper interpretation of an instruction and the issuance of proper commands to sequentially command computer logic to final completion of this interpretation.

CONTROL MEMORY - A storage device with assigned address locations used to hold information for control of computer logic operations.

COUNTER - A device capable of increasing or decreasing the value of its content upon receipt of coded input signals.

DEBUG - To isolate and remove all malfunctions from computer circuitry; to correct mistakes in a computer program or routine.

DIGIT - One of a set of numerical characters used as coefficients of powers of the radix in the positional notation of numbers.

ENABLE - (noun) A signal or pulse which allows other conditions to be acted upon; (verb) to apply the signal, to provide an enable signal.

EXECUTE - To perform indicated operations on specified operands.

FAULT - A condition which arises from the application of an illegal or improper signal; resulting from the detection of an improper or illegal condition.

FIXED POINT ARITHMETIC - A type of arithmetic in which operands and results of all arithmetic operations must be properly scaled so as to have a magnitude between certain fixed values. A method of calculation in which operations occur in an invariant manner and in which the computer does not consider the location of the radix point.

FLOW DIAGRAM - A graphical presentation of a sequence of events or operations; a flow chart.

FLOATING POINT ARITHMETIC - A form of number presentation in which quantities are represented by one number, the mantissa; multiplied by a power of the number base, the characteristic. A method of calculation which automatically accounts for location of the radix point.

FUNCTION CODE - That portion of the instruction word which specifies to the controlling logic the particular operation to be performed.

GATE - A circuit, which yields an output signal, that is dependent upon some function of its past or present input signals; AND gate or OR gate.

HALF-SUBTRACT - The bit-by-bit subtraction of two binary numbers with no regard for borrows, abbreviated HS. The complement for half-subtract is "half-subtract not" which is abbreviated HS.

HARDWARE - The physical equipments or devices that comprise a computer and its associated peripheral units; contrasted with software.

HOLD - The function of retaining information in a storage device after this information has been transferred to another device; contrasted with clear.

INDEX REGISTER - A register which contains a quantity that may be used for address or operand modification; sometimes referred to as B register or B box. One of seven registers contained in control memory and used for operand and address modification during specific instructions.

INFORMATION - A collection of facts or other data as derived from the processing of data.

INPUT - Information or data transferred from, or to be transferred from, a peripheral device to internal storage of the computer. To transfer data or information into the computer.

INPUT/OUTPUT - A section within the computer which provides a method of communication with peripheral equipments, abbreviated I/O.

INPUT/OUTPUT REGISTERS - Registers within the computer that are used for the storage of information to be transferred from or transferred to the computer.

INSTRUCTION - A set of characters which defines an operation to be performed by the computer and contains the required addresses, operands, and other necessary information; same as instruction word.

INSTRUCTION DESIGNATORS - Those parts which constitute an instruction word and represented by the letters f, m, k, and y. The letter f designates format or function code; m, minor function code when applicable; k, modification designator; and y, operand address.

INSTRUCTION REPERTORY - The set of instructions which a computing system data processing system is capable of performing.

INTERFACE - A common boundary between automatic data processing systems or between parts of a single system.

INTERNAL STORAGE - Storage facilities forming an integral physical part of the computer from which instructions and operands may be processed.

INTERRUPT - 1) Internal; an internally generated signal which indicates termination of an input or output buffer or acceptance of an external interrupt. 2) External; a signal from an external device which indicates an unusual condition which requires computer attention.

JUMP - An instruction or signal which, conditionally or unconditionally, specifies a location of the next sequential instruction and directs the computer to that instruction; used to alter the normal sequential control of the computer.

LEAST-SIGNIFICANT DIGIT - The first digit of a number counter from the right; the lowest-order digit of a number.

LOAD - To enter information into the storage area of a computer; to insert information into a register; to insert a program into a computer.

LOGIC - The systematic scheme which defines interactions of signals in the design of an automatic data processing system.

LOGICAL PRODUCT - Bit-by-bit multiplication of two binary numbers.

LOGICAL SUM - Bit-by-bit addition of two binary numbers with no regard for carrys; abbreviated LS. The complement of the logical sum is the "logical sum not", abbreviated $\overline{LS}$.

MAIN MEMORY - Core storage area of the computer used for normal word or program storage. That area of the computer used for instruction storage and for addressable operands.

MALFUNCTION - A failure in the operation of the hardware of a computer; failure or casualty in an equipment which degrades its operation or causes equipment to become inoperative.

MASK - A machine word that specifies which parts of another machine words are to be operated upon.

MASKING - The process of extracting a nonword group or a field of characters from a word or string of words.

MASTER CLEAR - To clear all normal storage locations of a computer prior to an operation. To set all major flip-flops to store a zero.

MASTER CLOCK - Primary timing signals within a computer. The source of primary timing signals.

MEMORY - A device into which information may be introduced, retained, and extracted for use at a later time.

MODE - A computer system of data presentation. A selected type of computer operation such as read or write mode.

MODIFY - To alter a portion of an instruction so its interpretation and execution will be other than normal. To alter a subroutine according to a defined parameter.

MODULE - An interchangeable plug-in item containing components. A printed circuit card upon which are mounted the electronic components.

MODULUS - The maximum quantity of permissible numbers that may be used in a process or system. The modulus for the group of numbers from -15 to +15 is 31.

MOST-SIGNIFICANT DIGIT - From left, the first digit of a number other than zero; the highest-order digit of a number.

NONADDRESSABLE - Pertaining to a storage location incapable of being referenced by an instruction word.

NONDESTRUCTIVE READOUT - A memory device that stores information which has been preset at manufacture and cannot be changed or altered; reading information from a memory device that does not destroy the contents of the device; a reading of information

from a register without changing that information within the register.

NONVOLATILE STORAGE - Storage mediums such as magnetic tapes, drums, cores, and discs, which retain information in the absence of power, and may be made available upon restoration of that power.

NUMBER - The total, aggregate, or amount of units; a figure or word, or a group of figures or words used to graphically represent an arithmetic sum or total.

OCTAL NUMBERS - A numbering system using eight symbols, 0 through 7, as its base; numbers in which the base has been set at eight.

OPERAND - A quantity entering or arising in an instruction; an argument, result, parameter, or indication of the location of the next instruction; the address portion of an instruction.

OPERATION - A defined action; action specified by a single computer instruction or group of instructions.

OPERATOR - A mathematical symbol which represents an arithmetic process to be performed upon an operand. One who operates the computer.

OUTPUT - A transfer of information from the internal storage system of the computer to peripheral devices or external storage; the process of an information transfer from the computer.

OVERFLOW - The condition that arises when the result of an arithmetic operation exceeds the capacity of the alloted storage area; overcapacity; information contained in an item which is in excess of a given or stated amount.

PAPER TAPE READER - A device capable of sensing information punched on paper tape in the form of a series of holes. A photoelectric readout device, abbreviated PT Reader.

PARALLEL - To handle simultaneously in separate facilities; to operate on two or more parts of a word or item simultaneously; contrasted with serial.

PARALLEL TRANSFER - To transfer characters of a word simultaneously over separate lines.

PARTIAL CARRY - Execution of the carry process in which carrys that arise as a result of a carry are not transmitted to the next higher stage.

PARITY BIT - A bit, normally a one, that may be added to a word to insure the total number of ones in that word is odd.

PARITY CHECK - The process of che ıg the number of ones contained in a given word or group of words or instructions.

PASS - A complete cycle of reading, processing, and writing. A complete machine operation or run.

PERIPHERAL EQUIPMENT - Auxiliary machines placed under control of the central computer. Equipments used by, and in conjunction with, the main computer system.

POSITIONAL NOTATION - A method of expressing a quantity, using two or more figures wherein the successive right to left figures are interpreted as coefficients of

ascending integer powers of the radix.

PRECISION - The degree of exactness with which a quantity is stated; the degree of discrimination or detail.

PRESET - To set contents of a storage location to an initial value.

PROGRAM - A complete sequence of machine instructions, routines, and operands necessary to solve a problem.

RADIX - The number of individual characters used in a numbering system; radix for the decimal system is ten; radix for the octal system is eight; radix for binary system is two.

RADIX POINT - The period that separates the integer digits from the fractional digits of number of the digital position involving the zero exponent of the radix from the digital position involving the minus-one exponent of the radix. The decimal point for the decimal system; the octal point for the octal system; or the binary point for the binary system.

RANDOM ACCESS - Pertaining to the process of obtaining information from or placing information into storage where the time required for such access is independent of the location of the information most recently obtained or placed in storage.

READ - To sense information; to extract information, usually from a memory location.

READER, CARD - A device capable of recognizing information in the form of holes punched in a card.

READER, MAGNETIC TAPE - A device capable of reading information recorded on magnetic tape in the form of magnetized dots.

READER, PAPER TAPE - (See paper tape reader).

REAL TIME - A measurement of elapsed time relative to a specific time or event.

REAL-TIME CLOCK - An oscillator and the associated circuitry capable of recording elapsed time measured in minutes, seconds, milliseconds, or other fractions of actual time periods.

REGISTER - A number of bistable stages used for holding or storing information. The number of stages determines the modulus of the number system which may be represented by the computing system; an addressable storage location. (See definition of accumulator, buffer register, index register, input/output register).

REPERTOIRE - Instructions or functions capable of being executed by the computer; repertoire of instruction (see definition of instruction repertory).

RESET - To clear; to set to zero; to return a device to zero or to an initial or arbitrarily selected value.

RESUME - The input acknowledge signal generated by a receiving computer upon completion of sampling input information lines.

RISE TIME - Time required for the leading edge of a generated or transmitted pulse to rise from one-tenth to nine-tenths of its final or terminal value. Rise time is proportional to the time constant of the circuit.

ROUTINE - A set of coded instructions

arranged in proper sequence to direct the computer to perform a desired operation or sequence of operations. Subdivision of a program consisting of two or more functionally related instructions.

SCALE - To shift a binary number either right or left in a register to retain the number for future computations or for later storage within the computer.

SCALE FACTOR - Coefficients used to multiply or divide quantities in a problem so they fall within a given range of magnitude such as from +1 to -1.

SCAN - A cycle or sequence which routinely interrogates all requests and interrupts; to determine priority for real-time clock; input and output on a basis of channel number and type of request.

SENSE - To examine relative to a criterion; to read information from magnetic cores.

SEQUENCE - An orderly progression of items of information or of operations in accordance with set rules.

SERIAL - One at a time; pertaining to time-sequential transmission.

SERIAL TRANSFER - To transfer words of information or bits of a word in a serial manner; to transmit bits of a word or elements of information in succession over a single line; contrasted with parallel transfer.

SET - To place a binary cell or flip-flop in the state of storing a one; to change the state of a storage device to output a value other than zero.

SHIFT - To move characters of a unit of information columnwise left or right; to multiply or divide a number by a power of the base of notation; to move information left or right in the arithmetic section of a computer.

SHIFT REGISTER - A computer register in which the contents may be shifted either left or right.

SIGN - A symbol that distinguishes negative quantities from positive quantities.

SIGN BIT - Sign digit; bit used to designate the algebraic sign of a number.

SIGNIFICANT DIGITS - A set of digits from consecutive columns beginning with the most-significant digit other than zero and ending with the least-significant digit whose value is known and considered relevant.

SINGLE ADDRESS CODE - Consisting of instructions containing a coded representation of the operation to be performed and a single address of a word in storage or an operand. The instructions of the single address code contain a maximum of one address for reference in storage.

SOFTWARE - The totality of programs and routines used to extend capabilities of computers; programs and routines such as compilers, assemblers, narrators, routines, and subroutines.

STORAGE - Pertaining to a device in which information may be placed and retained for future use; synonymous with memory.

SUBROUTINE - A set of instructions necessary to direct the computer to perform a well defined mathematical or logical operation; a subunit of a routine such as multiply

or divide subroutine.

TAG - A unit of identification whose composition differs from that of the other members of a set so that it can be used as a marker or label.

TRANSFER - Conveyance of control from one mode to another by means of instructions or signals; conveyance of data from one place to another.

TRANSLATE - To change information from one form of representation to another without significantly affecting the meaning; to transform.

TROUBLESHOOT - To search for the cause of a malfunction or erroneous program behavior to isolate and correct the malfunction or error.

VOLATILE STORAGE - A storage device in which the stored information is lost in the event of a power loss or shutdown; such as a flip-flop register.

WIRED PROGRAM - A program permanently wired into storage; see definition of bootstrap memory.

WORD - An ordered set of characters that occupies one storage location and is treated by the computer circuitry as a unit. A word is normally treated as an instruction by the control section and as a quantity by the arithmetic section. Normal word length for the computer is 18 bits.

WRITE - To record data in a register, location, or other storage medium or device; to introduce information into some form of storage.