Price: $7.75

Technical Manual

COMPUTER
Model 930

SDS 900066C

February 1966

This publication supersedes SDS 900066B
dated June 1965

SCIENTIFIC DATA SYSTEMS • 1649 Seventeenth Street • Santa Monica, Calif. • (213) 871-0960

©1966, Scientific Data Systems, Inc.

LIST OF EFFECTIVE PAGES

Total number of pages is 226, as follows:

Page No.	Issue	Page No.	Issue
Title	Original		
A	Original		
i thru x	Original		
1-1 thru 1-10	Original		
2-1 thru 2-4	Original		
3-1 thru 3-160	Original		
4-1 thru 4-18	Original		
5-1 thru 5-12	Original		
6-1 thru 6-2	Original		
7-1 thru 7-2	Original		
8-1 thru 8-2	Original		
I-1 thru I-4	Original		

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont.)

LIST OF ILLUSTRATIONS (Cont.)

LIST OF TABLES

LIST OF TABLES (Cont.)

LIST OF RELATED PUBLICATIONS

The following publications contain information not included in this manual but necessary for a complete understanding of the Computer Model 930.

Publication Title	Publication No.
SDS 930 Computer, Reference Manual	900064
SDS 930 Computer Examiner Diagnostic System, Technical Manual	900097
930/9300 Computer, Interface Manual	900561
SDS 930 Computer Central Processor, Logic Diagrams	900592
925/930 Computer, Basic Interrupt, Logic Layouts, Current and History	900608
925/930/9300 Computers, General Reference Drawings	900619
930/9300 Memory, Logic Layouts, Current and History	900620
925/930/9300 Computers, Module Reference Data	900623
SDS 930 Computer, Logic Equations, Main Frame and Memory	900636
Models 93280/90 Interrupt Control System, Technical Manual	900667
Models 9228, 9229, 92280, 92290 Interrupt Arming Option, Technical Manual	900668
Models 92200/93200 Time-Multiplexed Communication Channels (TMCC), 925/930/9300 Computers, Technical Manual	900685
Models 925, 930, and 9300, Internal Memory Stack, Troubleshooting Guide	900689
Models 92990/92992 Multiple Access to Memory (MAM), Technical Manual	900695
Direct Access Communication Channel (DACC), Model 92220, Technical Manual	900696
PX18 Power Supply, Technical Manual	900707
PX19 Power Supply, Technical Manual	900708
PX20 and PX21 Power Supplies, Technical Manual	900709
PX22 Power Supply, Service Manual	900710
PX23 Power Supply, Service Manual	900711
PX25 Power Supply, Technical Manual	900713
Model 91500 Memory Interface Connection, Theory of Operation	900808
Data Multiplex Channel (DMC), Model 91602, Technical Manual	900850
Data Subchannel I (DSC-I), Model 91711, Technical Manual	900852
Data Subchannel II (DSC-II), Model 91712, Technical Manual	900853
External Interlace, Model 91800, Technical Manual	900854
Power Fail-Safe, Model 92010, Technical Manual	900858
Model 91880 Real-Time Clock, Technical Manual	900859
Troubleshooting Manual Computer Memories for Computer Models 930/9300	900865
Memory Diagnostic	304001
Instruction Diagnostic	304002
P & S Register Test	304003

SECTION I
GENERAL DESCRIPTION

1-1 INTRODUCTION

1-2 This manual contains information necessary to install, operate, and maintain the Computer Model 930, manufactured by Scientific Data Systems, Santa Monica, California. The material is presented in the form of a general description, a section on operation and programming, a complete theory of operation, a maintenance section, and troubleshooting instructions.

1-3 For a description of equipment associated with the 930 Computer and not described in this manual, refer to the documents in the List of Related Publications in the front of this manual. The list also contains the titles and publication numbers of logic diagrams, parts lists, and reference drawings for the 930 Computer.

1-4 The 930 Computer is a general-purpose, solid-state, digital computer designed for scientific and engineering computation and real-time applications. It is compatible with all SDS 900 series computers, peripheral units, and system components.

1-5 PHYSICAL DESCRIPTION

1-6 The 930 Computer consists of a double-bay central processor cabinet, a single-bay power supply cabinet, a minimum of one single-bay input/output cabinet, and a control console. Additional input/output cabinets are included as necessary to contain the electronics and power supplies for input/output channels and peripheral equipment. The basic 930 Computer is shown in figures 1-1 through 1-3.

1-7 The central processor cabinet contains a computer chassis, two blower units, two distribution panels, and one or two memory doors. The computer chassis consists of five rows of plug-in printed circuit modules. The rows are designated B through F and contain 64 module locations numbered from right to left. A space at the top of the cabinet, which would normally contain row A, is empty. The distribution panels, one mounted on each end of the rack, are used for receiving the outputs from the power supplies and furnish distribution points for the power required by the central processor and memories. These panels are designated A, on the left, and B, on the right, as viewed from the front of the unit.

1-8 A memory door is hinge-mounted on the front of the central processor cabinet. The memory chassis contains five rows of printed circuit modules, A through F, with 32 module locations in each row. The memory stack is located in the center of rows B and C. The first memory is usually mounted on the left-hand side of the central processor cabinet. A second memory may be mounted on the right-hand side of the cabinet. Each memory door contains blower units at the top and bottom.

1-9 If more than two memories are used, the third memory must be placed in a standard single-bay cabinet. In this case, the memory chassis is permanently mounted in the cabinet, and the additional power supplies are placed in the bottom of the cabinet.

1-10 The power supply cabinet contains a power distribution panel, a PX20 Power Supply, a PX21 Power Supply, two PX18's, three PX19's, and two PX25's. When a second memory is included, an additional PX20 and PX21 are installed. An optional PX19 and PX25 are included if necessary for further power demands.

1-11 An input/output cabinet is a standard SDS single-bay cabinet containing module chassis associated with input/output equipment. The basic interrupt chassis is generally mounted in the first input/output cabinet. Any number of input/output cabinets may be used, depending on the number necessary to contain the required input/output equipment.

1-12 The control console is a control panel mounted on a table. The control panel contains grid-controlled neon indicators, to display the states of flip-flops and registers, and pushbuttons, toggle switches, and thumbwheel switches for computer control and register display.

1-13 The optional features of the 930 are added as separate chassis in input/output cabinets or as additional modules in the central processor, the memory door, or the interrupt chassis. The options are listed in table 1-1 with their locations and the SDS publications in which they are described.

1-14 FUNCTIONAL DESCRIPTION

1-15 The 930 Computer executes instructions, stores data and instructions, and processes interrupts. The instructions fall into the following categories:

Load and Store	Shift
Arithmetic	Control
Logical	Breakpoint Tests
Register Change	Overflow
Memory Extension	Interrupt Enable and Disable
Branch	Channel Control
Test and Skip	Channel Tests
	Input/Output

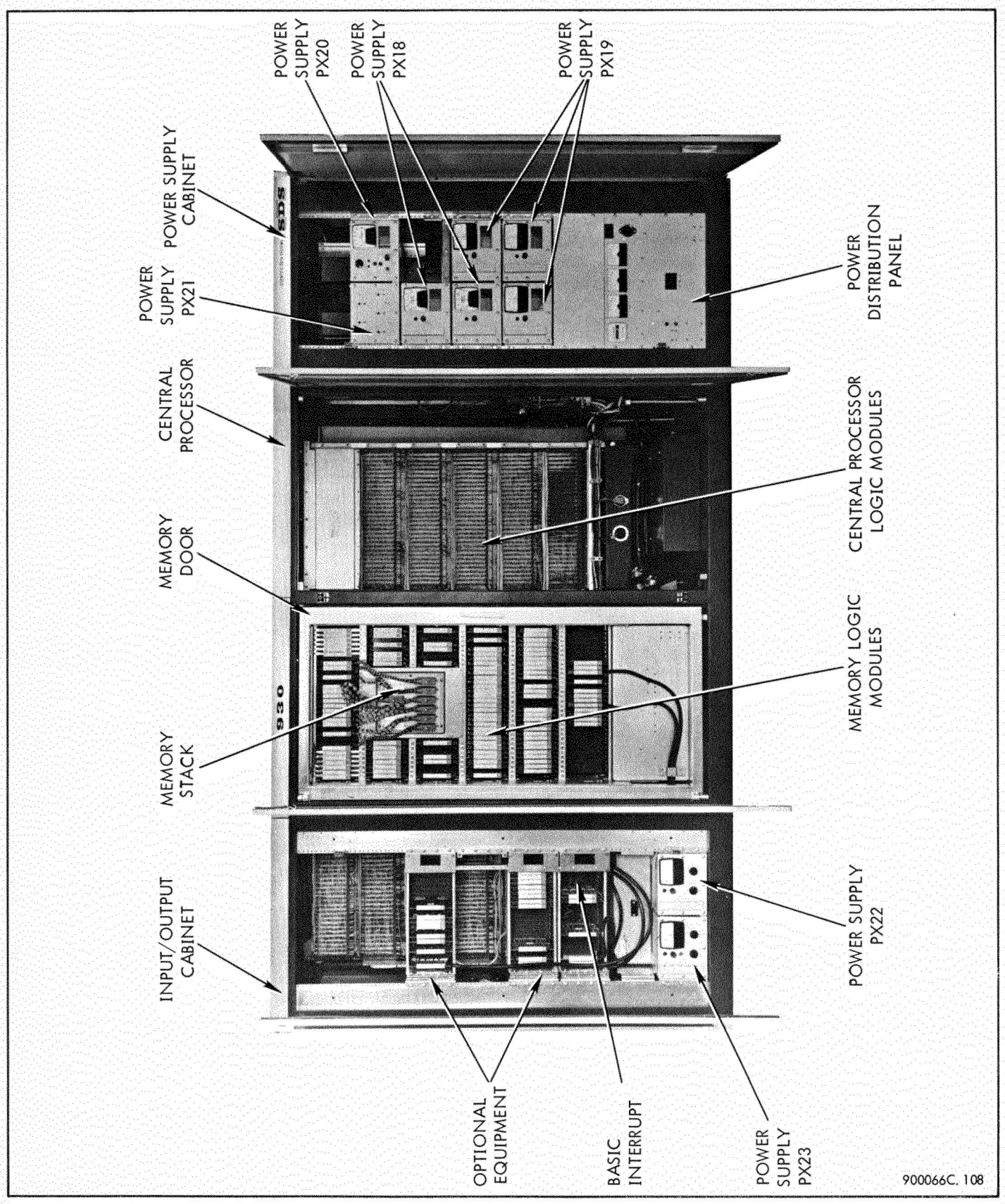


Figure 1-1. Computer Model 930, Front View

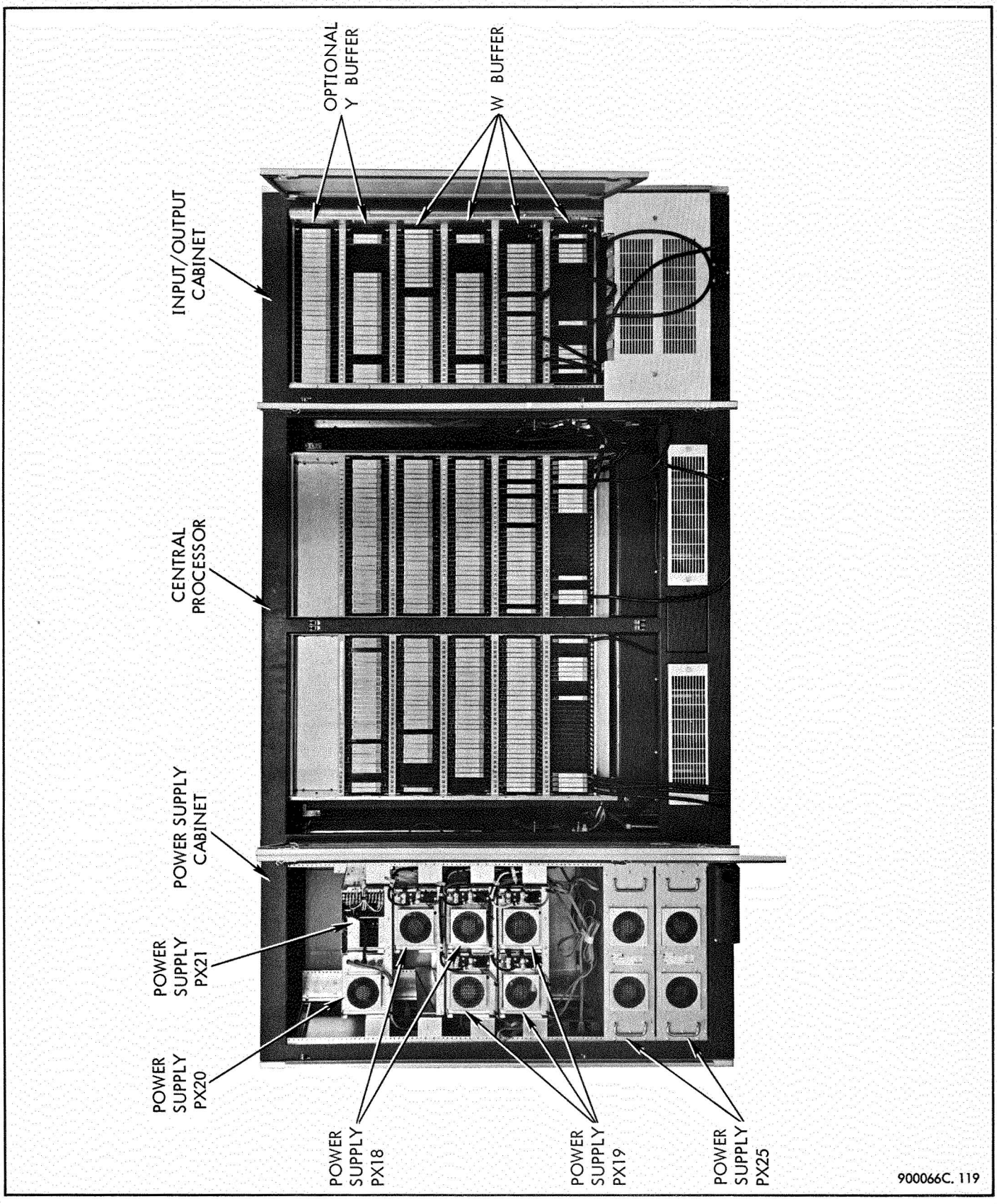


Figure 1-2. Computer Model 930, Rear View

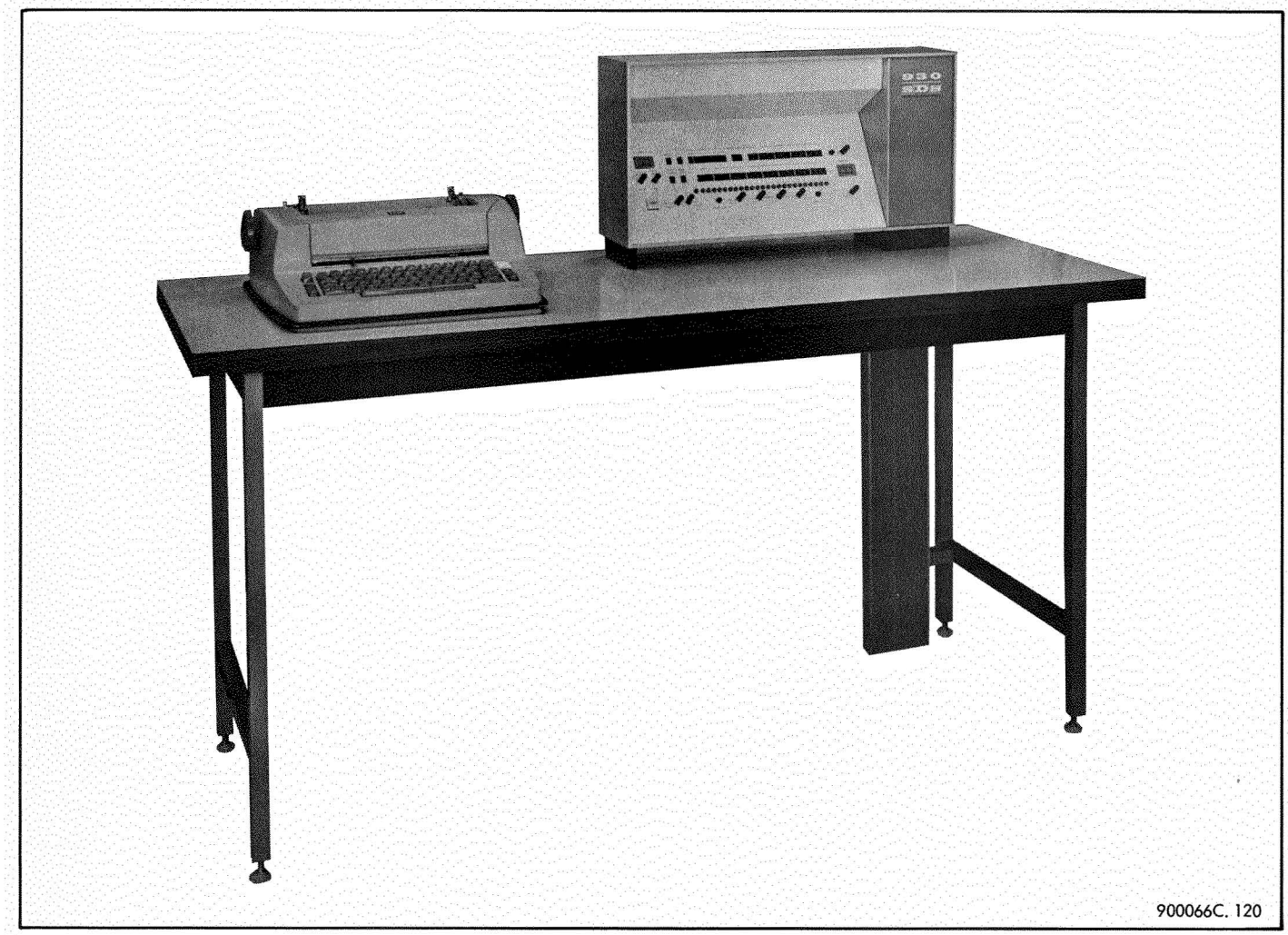


Figure 1-3. Computer Model 930 Control Console

Table 1-1. 930 Computer Options

Model Number	Option	Location	Reference Publication
91210	Memory Interlace Control Unit: For Any TMCC	I/O cabinet	900685
91500	Memory Interface Connection	I/O cabinet	900808
91602	Data Multiplex Channel	I/O cabinet	900850
91711	Data Subchannel I	I/O cabinet	900852
91712	Data Subchannel II	I/O cabinet	900853
91800	External Interlace	I/O cabinet	900854

Table 1-1. 930 Computer Options (Cont.)

Model Number	Option	Location	Reference Publication
91880	Real-Time Clock	Basic interrupt chassis (I/O cabinet)	900859
91903	SDS 930 Addressing Modification Feature	Central processor	This manual, par. 3-90
92010	Power Fail-Safe Option	Basic interrupt chassis (I/O chassis)	900858
92040	4096 Words of Core Memory	Memory door	This manual, par. 3-516
92060	Manual Memory Write Lockout Feature	Memory door	This manual, par. 3-606
92061	Programmed Memory Write Lockout Feature	Memory door	This manual, par. 3-600
92070	Interrupts for Memory Parity and Input/Output Parity	Central processor and basic interrupt chassis	This manual, par. 3-160
92080	8192 Words of Core Memory	Memory door	This manual, par. 3-516
92160	16,384 Words of Core Memory	Memory door	This manual, par. 3-516
92220	Direct Access Communication Channel	I/O cabinet	900696
92280	Arming for 16 Levels of Interrupt	I/O cabinet	900668
92290	Additional Arming for 16 Levels of Interrupt	I/O cabinet	900668
92990	Multiple Access to Memory (MAM)	Central processor and memory door	900695
93201	12-Bit Option for Any TMCC	I/O cabinet	900685
93202	24-Bit Option for Any TMCC	I/O cabinet	900685
93220	One Additional Time-Multiplexed Communication Channel	I/O cabinet	900685
93221	Two Additional Time-Multiplexed Communication Channels	I/O cabinet	900685
93280	Interrupt Control System	I/O cabinet	900667
93290	Priority Interrupt: Two Levels	Priority interrupt chassis (I/O cabinet)	900667

1-15 A block diagram of the functional units of the 930 Computer, including optional equipment, is given in figure 1-4.

1-16 The central processor performs all of the computer functions except storage, interrupt, and serial input/output. Parallel input and output of memory words is handled by the central processor. Word storage, memory control and addressing, and write lockout take place in the memory unit. The basic interrupt system receives interrupt signals from the power fail-safe, memory parity, memory lockout circuits, and the Time Multiplexed Communication Channel (TMCC) and Direct Access Communication Channel (DACC). These interrupts are handled on the basis of priority level, and an interrupt address is sent to the computer for each individual interrupt. The TMCC, W-buffer, communicates with the peripheral equipment using the first path to memory, through the C-register. The interlace control unit for the TMCC allows the program to designate to the TMCC how many words are to be transferred and the memory location of the first word.

1-17 OPTIONAL FEATURES

1-18 The optional Y, C, and D channels of the TMCC provide additional buffered input/output. Direct access to the memory is obtained through the Data Multiplexing System, the Multiple Access to Memory, the Memory Interface Connection, and the Direct Access Communication Channel. The Interrupt Control System and the Interrupt Arming option supply interrupts not covered by the basic interrupt.

1-19 With the parity interrupt option installed, a subroutine is entered when a memory parity error is detected. Blocks of memory may be locked out by the Write Lockout option so that data may not be written in the locked-out portions. The Power Fail-Safe option detects a drop in input power and stores in memory all data contained in the computer registers. A subroutine returns the data to the registers when power is resumed.

1-20 The Real-Time Clock option times the length of a program or subroutine or maintains the time of day for the computer.

1-21 The functions of the optional units are described in detail in the associated technical manuals indicated in table 1-1.

1-22 SPECIFICATIONS AND LEADING PARTICULARS

1-23 The 930 Computer specifications and leading particulars are given in table 1-2.

Table 1-2. Specifications and Leading Particulars

Characteristic	Specification
Power requirements	208 (±20.8) vac, ±10%, 60 (±0.5) cps, 3 phase Y-connected
Recommended service	208 vac, 3 phase Y, 10 kva minimum, with 30-amp circuit breaker in each leg, ganged. 10 kva will permit system expansion.
Service entry	No. 6 AWG wire for phase X, phase Y, phase Z, and Y neutral; No. 8 AWG wire for minimal system. The wires are run through a 1-1/4 inch I.D. conduit and enter at bottom rear of PS bay. Conduit is connected to the ground wire.
Operating temperature	$10^{\circ}C$ - $40^{\circ}C$ ($50^{\circ}F$ - $104^{\circ}F$)
Operating humidity	10% - 90%
Weights	Supported on four double casters
CPU	1150 lb with single memory bank
CPU	1350 lb with dual memory bank
PS cabinet	1200 lb
I/O cabinet	700 lb
Control console	150 lb with I/O typewriter option
Recommended floor area	
Computer	15 ft by 9 ft, 135 sq ft
Control console	9 ft by 6 ft, 54 sq ft
Recommended access area	
Computer	3-1/2 ft front and rear for maintenance
Control console	4 ft front for operation
Memory cycle	1.75 μsec
Logic signal levels	ONE, +4v; ZERO, 0v
Word length	24 bits plus parity bit
Data format	Fixed point, floating point
Coding	Binary

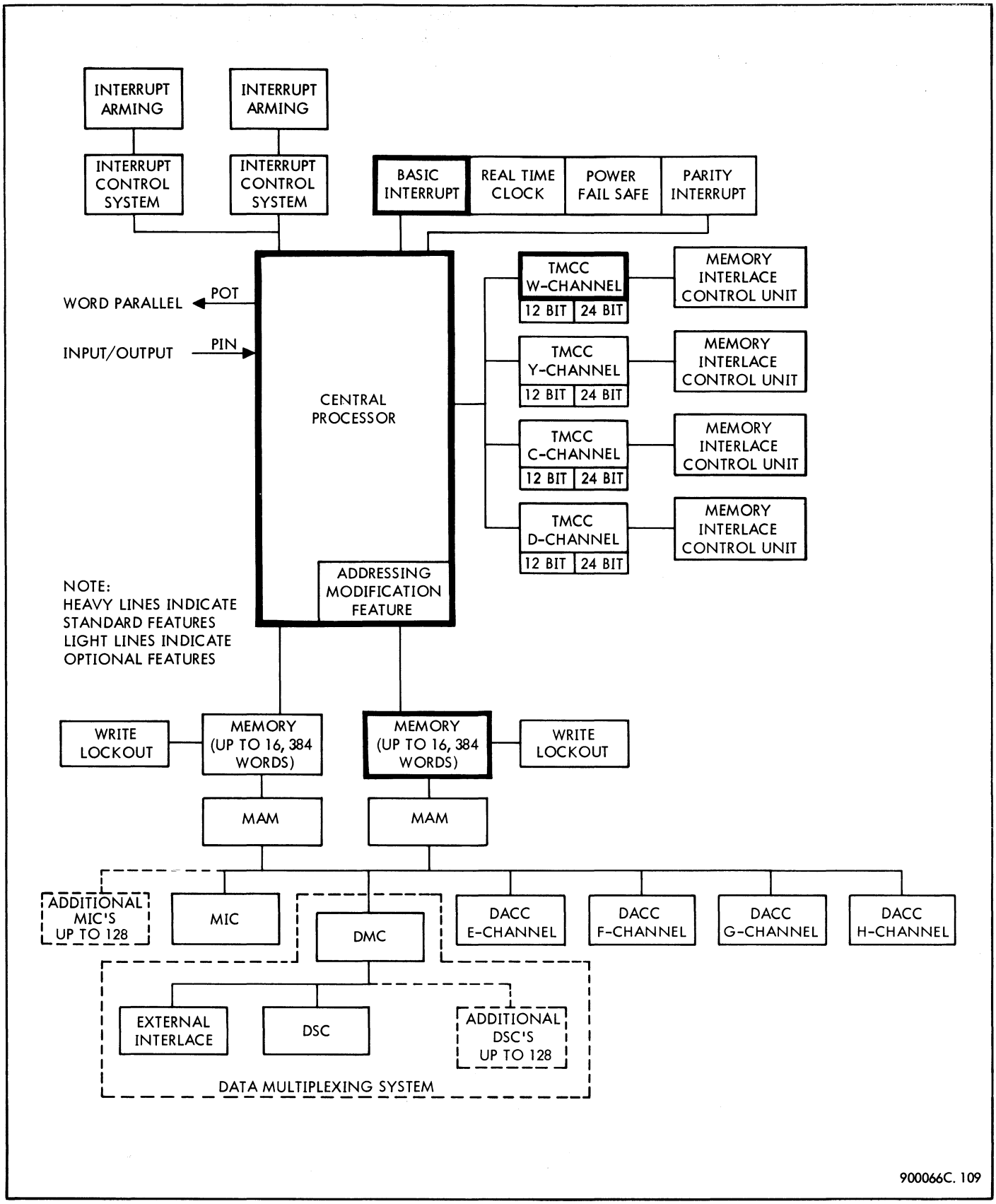


Figure 1-4. Computer Model 930, Functional Block Diagram

1-24 FUSES AND LAMPS

1-25 The fuses and lamps in the 930 Computer are described in table 1-3.

1-26 MODULES

1-27 The modules listed below are located in the central processor, the memory door, or the interrupt chassis. The list automatically includes the following optional features, which are in these three basic 930 Computer chassis: Addressing Modification Feature Model 91903, Manual Memory Write Lockout Feature Model 92060, Programmed Memory Write Lockout Feature Model 92061, Multiple Access to Memory Model 92990, Interrupts for Memory Parity and Input/Output Parity Model 92070, Real-Time Clock Model 91880, and Power Fail-Safe Option Model 92010. The remainder of the optional features are contained in separate chassis in the input/output cabinets.

1-28 Table 1-4 is a list of modules in the central processor chassis.

1-29 Table 1-5 lists the modules in the memory door.

1-30 The variations of Jumpers ZB65 in memory and in the Programmed Write Lockout option are given in tables 1-6 and 1-7.

1-31 Table 1-8 lists the modules in the interrupt chassis.

1-32 For the Addressing Modification Feature Model 91903, BAND NAND IB57 is installed in location 52C of the central processor, and a Dual Flip-Flop FB51 is installed in location 62D of the central processor.

Table 1-3. Fuses and Lamps

Component	Type	Reference Designator	Rating	Location	Circuit Protected
Circuit breaker		CB1	250v, 50 amp	Power supply cabinet	Main power
Fuse	3AG	F1	0.5 amp	Control console	+16v
Fuse	3AG	F2	1.5 amp	Control console	-16v
Fuse	3AG	F1	5 amp	Power Supply PX18	Input circuit
Fuse	8AG	F2	0.01 amp	Power Supply PX18	Meter
Fuse	3AG	F1	10 amp	Power Supply PX19	Input circuit
Fuse	8AG	F2	0.01 amp	Power Supply PX19	Meter
Lamp	Pushbutton switchlight T-1-3/4	S26	28v	Control console	-
Lamp	Triode indicator	DS2 - DS50, DS53, DS54	heater - 1 vac plate - 50 vdc	Control console	-
Lamp	Neon	DS1		Power supply cabinet	-

Table 1-4. Central Processor Chassis Modules

Description	Model	Quantity
Dual flip-flop	FB51	43
Triple flip-flop	FB52	14*
NAND flip-flop	FB54	34#
NAND module	IB52	21
NAND No. 2	IB56	14
BAND NAND	IB57	20
NAND No. 3	IB58	47
Cable driver	AB51	10
Receiver inverter	AB52	2
Receiver inverter buffer	AB53	1
Cable driver No. 2	AB55	13**
Interface +8 to +4	NB50	4##
Shift register	DB50	3
Clock generator	CB50	1
Termination module (+4)	ZB50	20
Indicator interface	NB53	2
Termination module cable	ZB52	2*
Interface	NB57	1
Termination module	ZB68	1***

*Five ZB52's for MAM option
#One FB54 for Parity Interrupt Option
**One AB55 for C or D channel of TMCC
Four AB55's for MAM
##One NB50 for MAM
***Used only when MAM is absent

Table 1-5. Memory Door Modules

Description	Model	Quantity		
		4K	8K	16K
Receiver inverter	AB52	9	9	9*
Cable driver No. 2	AB55	10	10	10#
Z driver	AB56	5	10	20
Dc flip-flop	FB50	13	13	13**
NAND No. 2	IB56	6	6	6##
BAND NAND	IB57	3	3	3

Table 1-5. Memory Door Modules (Cont.)

Description	Model	Quantity		
		4K	8K	16K
NAND No. 4	IB59	4	4	4
Sense amplifier	HB53-3	13	25	25
Z drive transformer cable (assembly 105015)	P1	2	2	4
X-Y transformer cable (assembly 105093)	P3-P4	4	6	8
One-shot multivibrator	OB50-2	2	2	2
Sink switch	OB53	8	8	6
Drive switch	OB54	4	6	8
Termination module	ZB52	4	4	4
Termination module +4	ZB50	1	1	1
Z resistor module	ZB64	10	10	10
Jumper module	ZB65-10	1	-	-***
Jumper module	ZB65-20	-	1	-***
Resistor module	ZB67	2	1	-
Jumper module	ZB65-30	-	-	1***
X-Y resistor module	ZB63	-	1	2
Jumper module	ZB65-50	1	1	1*** ###
Switch flop	SX51	4	4	4****
Termination module	ZB73	1	1	1####

*Three each for MAM. One installed only with first Z option fitted
#Four each for MAM
**Four each for Programmed Write Lockout
##Two each for Programmed or Manual Write Lockout
***ZB65 varies if memory is not first in the system
###Programmed Write Lockout only
****Manual Write Lockout only
####MAM only

Table 1-6. Memory ZB65 Jumper Module Variations

Model	Address Position	ZB65 (Location 8F)	
92040	0-4K	ZB65-10	*
or	4K-8K	-11	
93040	8K-12K	-12	
	12K-16K	-13	#
	16K-20K	-14	
92040	20K-24K	-15	
or	24K-28K	-16	
93040	28K-32K	-17	
92080	0-8K	-20	*
	8K-16K	-22	
or	16K-24K	-24	#
93080	24K-32K	-26	
92160	0-16K	-30	*
or			
93160	16K-32K	ZB65-34	

*Used on first memory in a system
#Replace standard ZB65's

Table 1-7. Programmed Write Lockout ZB65 Variations

Address Position	ZB65 (Location 5F)	
0-4K	ZB65-50	*
0-8K		
0-16K		
4K-8K	-51	
8K-12K	-52	#
8K-16K		
12K-16K	-53	
16K-20K		
16K-24K		
16K-32K	ZB65-54	

Table 1-7. Programmed Write Lockout ZB65 Variations (Cont.)

Address Position	ZB65 (Location 5F)	
20K-24K	ZB65-55	#
24K-28K	-56	
24K-32K		
28K-32K	ZB65-57	

*Used on first memory in a system
#Replace standard ZB65's

Table 1-8. Interrupt Chassis Modules

Description	Model	Quantity	
Jumper	ZK70	1	*
AND Gate Inverter	IH12	1	
Primary Power Detector	SK62	1	#
Counter Flip-Flop	FH15	1	**
Logic Interrupt	SK63	1	##
Priority Interrupt	SK61	10 max.	***

*Omitted if an Interrupt Control Model 93280 System is used

#Used with Real-Time Clock or Power Fail-Safe options, or both

**Used with Real-Time Clock

##Used with Power Fail-Safe option

***Two furnished with input/output buffer. Remainder depend on number of interrupts in system

SECTION II
OPERATION AND PROGRAMMING

2-1 INTRODUCTION

2-2 This section contains operating information not covered in the SDS 930 Computer Reference Manual, publication No. 900064. For programming information, consult the reference manual.

2-3 CONTROLS AND INDICATORS

2-4 The controls and indicators for the 930 Computer are located on the table-mounted control panel shown in figure 2-1. The controls and their functions are listed in table 2-1. The indicators and their functions are listed in table 2-2. The logic of the controls and indicators is described in paragraph 3-164.

2-5 OPERATING INSTRUCTIONS

2-6 TURN-ON PROCEDURE

2-7 Holding in the START pushbutton, press the POWER pushbutton. The POWER pushbutton lights when power is on. Release the START pushbutton. For machines with the auto-start option with the program in memory, it is not necessary to press the START pushbutton.

2-8 DISPLAYING AND CHANGING CONTENTS OF MEMORY

2-9 To display and change the contents of any memory location, use the following procedure:

a. Place the REGISTER thumbwheel switch in the C position.

b. Place the RUN IDLE STEP switch in the IDLE position.

c. Press the CLEAR pushbutton below the REGISTER DISPLAY indicators. This clears the C-register.

d. Using the pushbuttons below the REGISTER DISPLAY indicators, place a Load A-Register instruction, code 76, in the C-register by setting bits 3 through 7. Bits 10 through 23 should contain the address of the memory location to be changed. Press the pushbutton below each bit that should contain a ONE.

e. Place the RUN IDLE STEP switch in the STEP position and release. The computer executes the Load A-Register instruction and places the contents of the effective memory location in the A-register.

f. Place the REGISTER switch in the A position. The indicator lamps display the contents of the memory location to be changed.

g. Press the CLEAR pushbutton.

h. Enter the new data with the pushbuttons below the register display indicators.

i. Place the REGISTER switch in the C position.

j. Press the CLEAR pushbutton to clear the C-register for a new instruction.

k. With the register set pushbuttons, place a Store A-Register instruction, code 35, in the C-register. In the address field, place the address of the memory location to be changed.

l. Place the RUN IDLE STEP switch in the STEP position and release. The computer loads the new data, in the A-register, into the effective memory location.

2-10 Changing Upper 16K of Memory

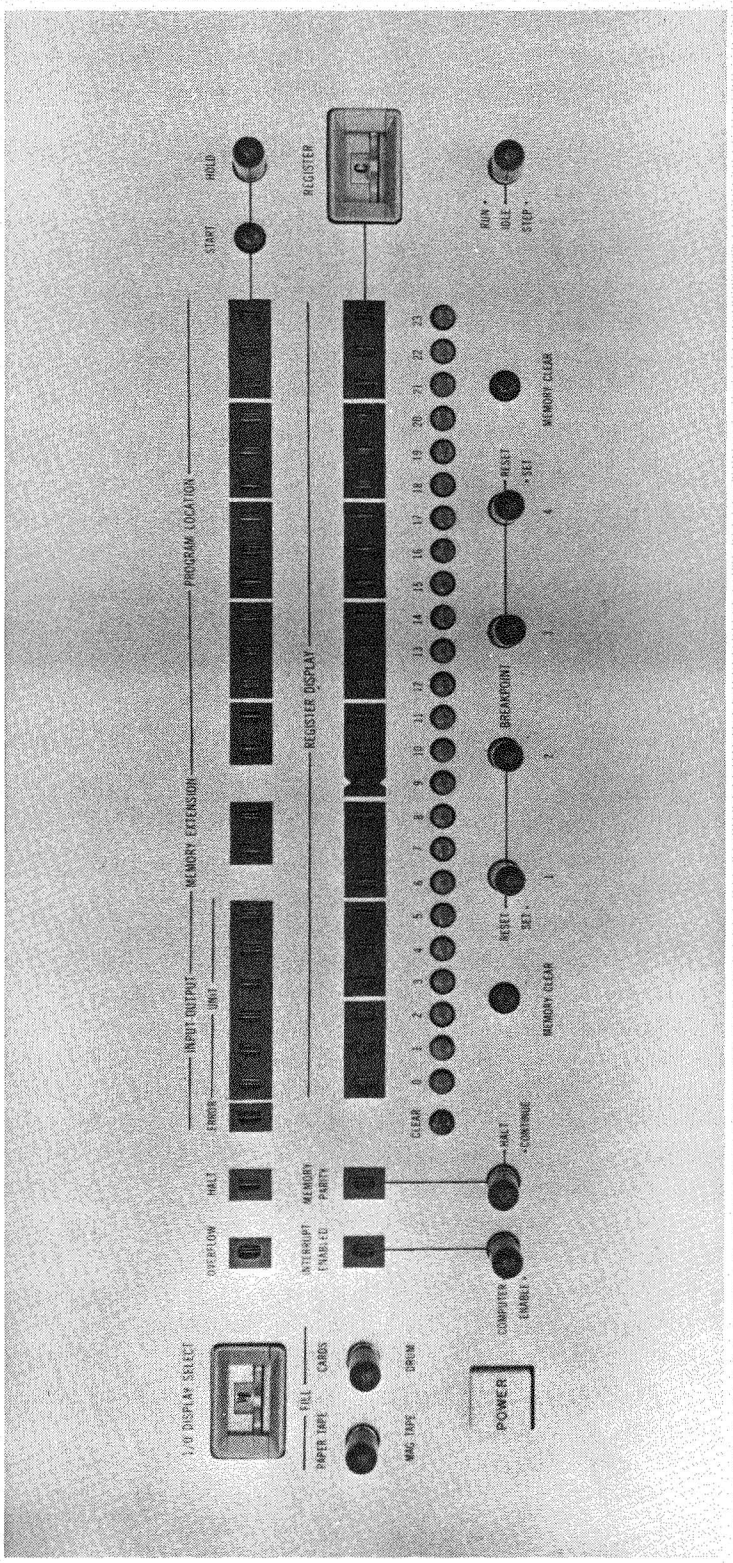
2-11 To change the information in a memory word in the upper 16K of a 32K memory, it is necessary to extend the address to 15 bits by using the memory extension registers, EM2 and EM3. The C-register must be cleared and a Set Extension Register EOD, code 06, entered. The most significant octal digit of the desired address is placed in either bits 18 through 20 or bits 21 through 23 of the EOD. To use bits 21 through 23 (EM2 register) as the most significant octal digit, set a ONE in position 17. If bits 18 through 20 (EM3 register) are to contain the most significant octal digit, set a ONE in position 16.

2-12 After the memory extension register has been set, the procedure for changing memory is the same as for the lower 16K, except that the Load A and Store A instructions must contain a code in bits 10 and 11 which call for one of the extension registers. A ONE ZERO in bits 10 and 11 call for EM2; a ONE ONE in bits 10 and 11 call for EM3.

2-13 As an illustration of the use of the memory extension registers, the following procedure shows how to change the data in location 47000_8:

a. Place the REGISTER thumbwheel switch in the C position.

b. Place the RUN IDLE STEP switch in the IDLE position.



900066C. 110

Figure 2-1. Control Panel

Table 2-1. Controls

Control	Reference Designator	Type	Function
I/O DISPLAY SELECT W, Y, C, D, E, F, G, H	S25	Eight-position thumbwheel switch	Selects channel from which unit address and error are displayed in INPUT OUTPUT indicators.
START	S35	Momentary pushbutton	Resets all channels; clears P-register, overflow indicator, memory parity error indicator; sets up Halt instruction in C-register. Clears, interrupts, and disables the system. Sets EM3 to 3 and EM2 to 2.
HOLD	S38	Two-position toggle switch	Holds current contents of program counter.
FILL			
PAPER TAPE MAG TAPE	S24	Three-position, spring-loaded, center-return toggle switch	Selects paper tape or magnetic tape and starts filling.
CARDS DRUM	S39	Three-position, spring-loaded, center-return toggle switch	Selects cards or drum and starts filling.
REGISTER A, B, C, X	S37	Four-position thumbwheel switch	Selects register to be shown on REGISTER DISPLAY indicators.
CLEAR	S29	Momentary pushbutton	Clears register selected by REGISTER thumbwheel switch.
0-23	S30, S1 thru S23	Momentary pushbuttons	Each pushbutton places a ONE bit in selected position of register displayed.
POWER	S26	Red-illuminated, double-action pushbutton	Connects power to computer. Switch is lit when power is on.
COMPUTER ENABLE	S27	Toggle switch; stationary in COMPUTER, momentary in ENABLE	In COMPUTER position, interrupt system may be enabled or disabled under program control. In ENABLE position, enables interrupt system regardless of program operations.
HALT CONTINUE	S28	Two-position toggle switch	In HALT position, computer enters idle state whenever memory parity error occurs. In CONTINUE position, computer does not change state when memory parity error occurs. With parity interrupt option included, computer processes parity interrupt in HALT position, and does not halt.
MEMORY CLEAR	S40, S41	Two momentary pushbuttons	Operating both switches simultaneously clears the first 16K of memory to ZERO.
BREAKPOINT 1, 2, 3, 4 RESET, SET	S31, S32, S33, S34	Four two-position toggle switches	Program may detect status of these switches by using breakpoint test. Switches control predetermined options within the program.
RUN IDLE STEP	S36	Toggle switch; stationary in RUN and IDLE, momentary in STEP	In RUN position, computation proceeds at machine speed. In IDLE position, computer idles immediately after instruction has been read from memory. In STEP position, computer executes instruction and returns to idle state.

Table 2-2. Indicators

Indicator	Reference Designation	Indication
OVERFLOW	DS24	Shows status of overflow indicator
HALT	DS23	Lights whenever computer executes Halt instruction while in RUN position. Placing RUN IDLE STEP switch to IDLE clears indicator.
INPUT-OUTPUT ERROR	DS54	Reflects status of channel error indicator for selected channel.
UNIT	DS17 through DS22	Contains unit address of peripheral device connected to selected channel.
MEMORY EXTENSION	DS16 (left) DS53 (right)	Lights when EM3 does not contain three. Lights when EM2 does not contain two.
PROGRAM LOCATION	DS1 through DS14	Contains binary indication of location of next instruction to be executed.
INTERRUPT ENABLED	DS50	On whenever interrupt system is enabled.
MEMORY PARITY	DS49	Lights when memory parity error is encountered. Setting CONTINUE ENABLE switch to CONTINUE turns off indicator.
REGISTER DISPLAY	DS25 through DS48	Contains binary indication of contents of register selected by REGISTER thumbwheel switch.

c. Press the CLEAR pushbutton.

d. Using the pushbuttons below the REGISTER DISPLAY indicators, place a Set Extension Register EOD, code 06, in the C-register. Using EM2 for the memory extension, the code is 00620104.

e. Follow steps e through l of paragraph 2-9, using the following code in the address field of the Load A and Store A instructions: 27000.

2-14 TURN-OFF PROCEDURE

2-15 To turn off the computer, proceed as follows:

a. Place the RUN IDLE STEP switch in the IDLE position.

b. Holding in the START pushbutton, press the POWER switch. The light in the POWER switch goes off when the computer is turned off.

2-16 PROGRAMMING

2-17 For complete programming instructions on the 930 Computer, refer to the 930 Computer Reference Manual, publication No. 900064.

SECTION III
THEORY OF OPERATION

3-1 GENERAL

3-2 This section contains a detailed logic description of the 930 Computer central processor and memory. The 92070 Parity Interrupt Option and the 91903 Addressing Modification Option in the central processor and the write lockout option in the memory are also described. Other options are discussed in the applicable manuals referred to in the List of Related Documents. The basic interrupt logic is explained in detail in the Interrupt Control System Technical Manual, SDS 900667.

3-3 COMPUTER INTERFACES

3-4 The interfaces between the central processor and memory, central processor and input/output, central processor and interrupt system, and memory and input/output are shown in figure 3-1. Data, address, and memory request transfer paths are shown. Timing signals and POT/PIN connectors are not shown.

3-5 The input/output configuration may consists of one or more direct access channels and one or more time multiplexed communication channels. A direct access channel transfers data directly to or from the memory, always in a time-share mode. The memory start signal and memory address are sent to the memory via the central processor. The time-share mode is optional in the time multiplexed channel. The memory start signal, address, and data are transferred to and from memory via the central processor.

3-6 A normal octal-serial transfer of data between a TMCC and central processor operates as follows: If the TMCC is to store data in the memory, the data is received, three bits at a time, by the central processor via Rwy1 through Rwy3. These bits are stored in an octal register designated Rn1 through Rn3. The contents of Rn1 through Rn3 are in turn transferred in parallel to bits 0-2 of the C-register. The C bits are shifted right, and the cycle is repeated until the proper number of characters are received. A memory cycle is then initiated and the data is stored in memory. If the TMCC is to read data from memory, the data word is read from memory and placed in the C-register. The word is then transferred an octal at a time via C21 through C23 to the input/output channel.

The interrupt request signal (Ir) and its associated address lines (N5-N14) are transferred from the interrupt system to the central processor. The interrupt address is transferred to the S-register in the central processor for memory access.

3-8 Further details on computer interfaces may be found in the Interface Manual, SDS 900561.

3-9 SPEED

3-10 The memory access time in the 930 is 0.7 microsecond. One memory cycle is completed in 1.75 microseconds. Typical execution times are given in table 3-1.

Table 3-1. Execution Times

Operation	Time (microseconds)
Add, fixed point	3.5
Multiply, fixed point	7.0
Add, single-precision floating point	77
Add, double-precision floating point	92
Multiply, single-precision floating point	54
Multiply, double-precision floating point	147

In serial numbers 3101 to 3119 of the 930 Computer the memory cycle time is 1.925 microseconds.

3-11 INSTRUCTION CODES

3-12 The 930 instruction set, with mnemonics and octal codes, is given in table 3-2. Instructions are listed in numerical order.

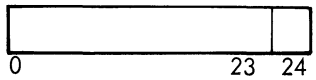
3-13 SIGNAL DESCRIPTIONS

3-14 Descriptions of logic terms used in the central processor and memory are given in tables 3-28 and 3-29 at the end of this section.

3-15 LOGICAL ELEMENTS

3-16 WORD FORMAT

3-17 The 930 Computer memory word contains 24 bits of data and a parity bit. The bits are numbered from left, or most significant end, to right, or least significant end. Because bit 24 is used only for parity check and does not enter into computation, this bit will not be shown on other word formats.



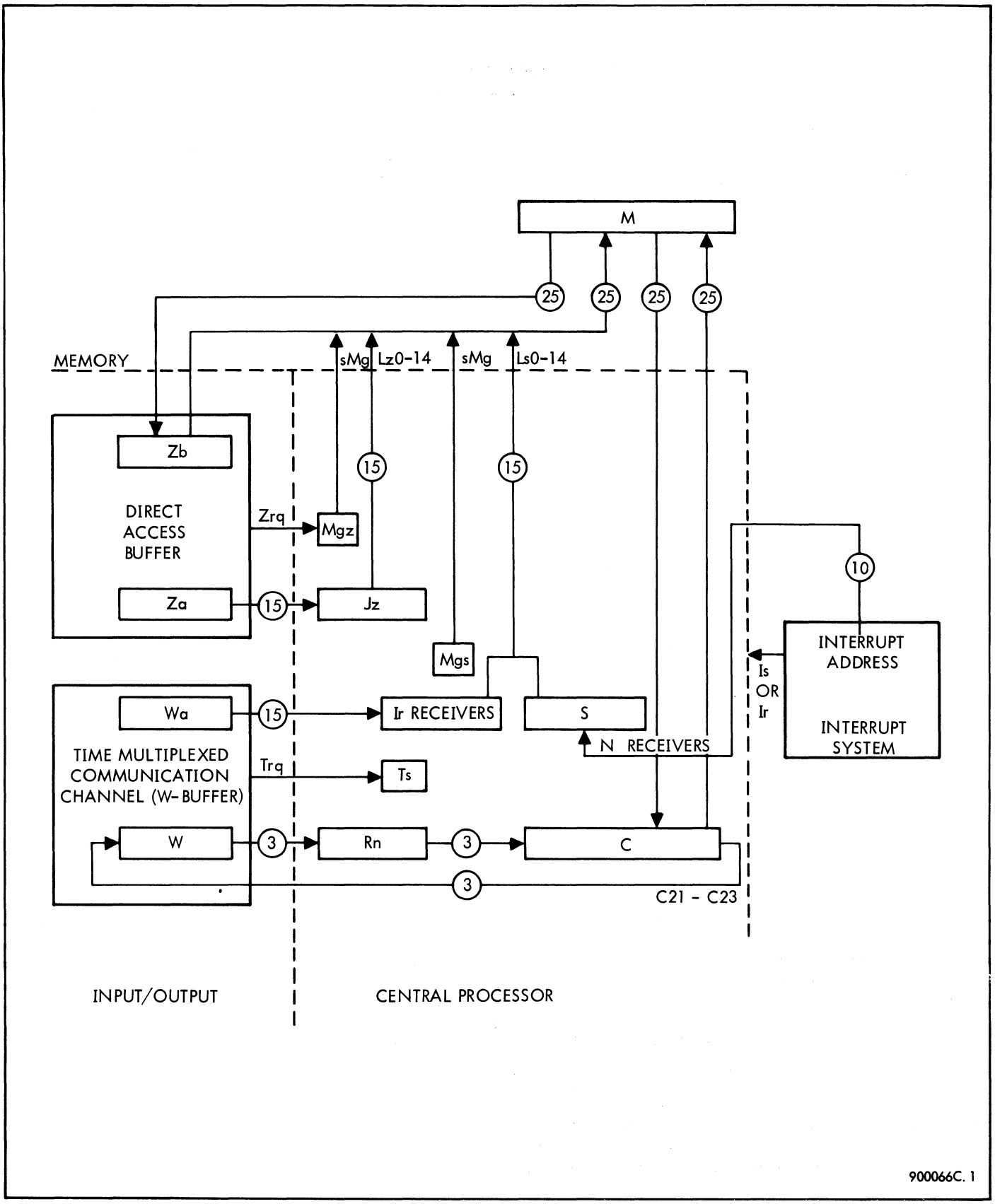


Figure 3-1. Interface Diagram

Table 3-2. Instruction Set

Time	Code	Mnemonic	Function
1	00	HLT	Halt
1	01	BRU	Branch to M
1	02	EOM	Energize M
	03		
	04		
	05		
1	06	EOD	Energize direct buffer
	07		
2+	10	MIY	(M) ⟶ Y when ready
	11		
2+	12	MIW	(M) ⟶ W when ready
3+	13	POT	Parallel out, when ready
2	14	ETR	Extract
	15		
2	16	MRG	Merge: (A) or (M) ⟶ A
2	17	EOR	Exclusive OR
1	20	NOP	No operation
	21		
	22		
1	23	EXU	Execute
	24		
	25		
	26		
	27		
3+	30	YIM	(Y) ⟶ M when ready
	31		
3+	32	WIM	(W) ⟶ M when ready
4+	33	PIN	Parallel in, when ready
	34		
3	35	STA	(A) ⟶ M
3	36	STB	(B) ⟶ M
3	37	STX	(X) ⟶ M
2,3	40	SKS	Skip if M not set
1, 2	41	BRX	(X) + 1 X, branch if X9=1
	42		
2	43	BRM	(P) M, BRU M + 1

Table 3-2. Instruction Set (Cont.)

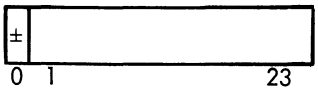
Time	Code	Mnemonic	Function
	44		
	45		
1	46 47	RC*	Register change
2, 3	50	SKE	Skip if (A) = (M)
2	51	BRR	(M) + 1 P (BRU)
2, 3	52	SKB	No skip if (B)(M) = 1 anywhere
2, 3	53	SKN	Skip if (M) negative
2	54	SUB	(A) - (M) ⟶ A
2	55	ADD	(A) + (M) ⟶ A
2	56	SUC	(A) - (M) - Carry ⟶ A
2	57	ADC	(A) + (M) + Carry ⟶ A
3	60	SKR	(M) - 1 M Skip if negative
3	61	MIN	(M) + 1 M
3	62	XMA	(M) (A)(Exchange)
3	63	ADM	(M) + (A) ⟶ M
4	64	MUL	Multiply
10	65	DIV	Divide
2-7	66	RSH*	Right shift
2-4	67	LSH*	Left shift
2	70	SKM	Skip if (A) = (M), on B mask
2	71	LDX	(M) ⟶ X
2, 3	72	SKA	No skip if (A)(M) = 1 anywhere
2, 3	73	SKG	Skip if (A)(M)
2, 3	74	SKD	Difference exponents and skip
2	75	LDB	(M) ⟶ B
2	76	LDA	(M) ⟶ A
2	77	EAX	Eff. address ⟶ X

*Instruction where the address is used in addition to the instruction code to define the specific operation.

3-18 Data Word Format

3-19 Data words in memory are entered in either fixed-point or floating-point form.

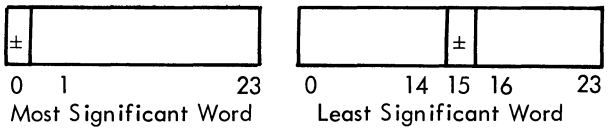
3-20 Fixed-Point Format. Fixed-point data words have the format.



Bits 1 through 23 contain the data, bit 0 being designated as the sign bit. A ZERO in bit position 0 indicates that the number is positive. A ONE in bit position 0 indicates that the number is negative and is represented in two's complement form. Fixed-point numbers are held in memory as 23-bit fractions with an assumed binary point to the left of bit position 1.

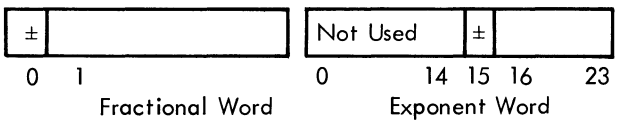
3-21 Floating-Point Format. A floating-point number occupies two 24-bit word locations and consists of a proper fraction with a 9-bit exponent. The leading bit of the fraction and of the exponent is the sign bit. The assumed binary point is just to the left of the most significant bit of the proper fraction. Double-precision or single-precision floating-point arithmetic may be performed.

a. Double-Precision Floating Point. The double-precision floating point word has the following format:



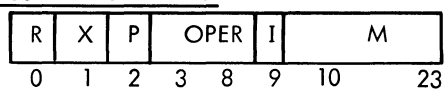
The fraction contains 38 bits of data and a sign bit. Bits 0 through 23 of the most significant word represent the most significant portion of the fraction, and bits 0 through 14 of the least significant word represent the least significant portion of the fraction. Bits 15 through 23 of the least significant word contain the exponent, bit 15 being designated as the sign bit.

b. Single-Precision Floating-Point. The single-precision floating-point word has the following format:



The fractional portion of the single-precision word occupies the 24 bit positions of a data word referred to as the fractional word. The 9-bit exponent is contained in bits 15 through 23 of a second word called the exponent word. Bits 0 through 14 of the exponent word are not used.

3-22 Instruction Word Format



3-23 The instruction word is defined as follows:

Bit	Designation	Use
0	R	Relative address bit
1	X	Index bit
2	P	Programmed operator bit
3-8	Oper	Instruction code
9	I	Indirect address bit
10-23	M	Address information

The relative address bit not sensed by the computer logic during normal computation is used by standard loading programs to enable relocation of certain programs.

3-24 The index bit enables the initial address to be altered by the addition of the contents of the index register. Indexing does not alter execution time.

3-25 The programmed operator bit allows C_{2-8} to be used as a subroutine address . When C2 is a ONE, bits 3 through 8 are not interpreted as an instruction code.

3-26 The indirect address bit enables the contents of the effective address to be read from memory as in normal operation, but the contents of the effective location are decoded as if the instruction contained no instruction code. Address decoding is reinitiated on the contents of the effective location in order to obtain the address of the operand. One cycle is added to the instruction execution time for each level of indirect addressing.

3-27 TIMING

3-28 Clock

3-29 Clock pulses are generated in a crystal oscillator. The clock pulse is approximately 35 nanoseconds wide. The clock signals are passed through two levels of inversion in order to obtain the necessary drive capability. At the second level of inversion, a time-share gating signal, Ts, is provided so that certain portions of the clock distribution are inhibited when a time-share operation is taking place. Gated clock signals are referred to in the logic as Cg. Ungated clock signals, Cc, are delivered continuously to the central processor logic, which must be active during a time-share operation. Advanced clock signals, both gated (Cga) and continuous (Cca), are supplied to some counters and flip-flops to enable the logic to operate at a 1.75-microsecond cycle time.

3-30 Machine Cycle

3-31 Eleven timing pulses are used to define one machine cycle. The pulses are designated T8 through T0, Tr, and Tp. The first clock of each machine cycle is T8, and the last clock of the cycle is Tp. One machine cycle takes 1.75 microseconds.

3-32 Pulse Counter. Each individual timing pulse is defined by a unique state of a pulse counter consisting of flip-flops Q1 through Q6. The logic equations for the Q-counter are as follows:

$$sQ1 = \overline{Q6}$$
$$rQ1 = \overline{Q4}\ Q5$$
$$sQ2 = \overline{Q6}$$
$$rQ2 = Q6$$
$$sQ3 = Q2$$
$$rQ3 = \overline{Q2}$$
$$sQ4 = Q3$$
$$rQ4 = \overline{Q3}$$
$$sQ5 = Q4\ Q3$$
$$rQ5 = \overline{Q1}\ \overline{Q4}$$
$$sQ6 = Q5$$
$$rQ6 = \overline{Q5}$$

The eleven timing pulses are decoded as presented in table 3-3.

Table 3-3. Machine Cycle Timing Pulse Decoding

	Q1	Q2	Q3	Q4	Q5	Q6
8	0	0	0	0	0	0
7	1	1	0	0	0	0
6	1	1	1	0	0	0
5	1	1	1	1	0	0
4	1	1	1	1	1	0
3	1	1	1	1	1	1
2	1	0	1	1	1	1
1	1	0	0	1	1	1
0	1	0	0	0	1	1
Tr	0	0	0	0	1	1
Tp	0	0	0	0	0	1

The equations are the decoding of the Q-counter are as follows:

$$T8 = \overline{Q2}\ \overline{Q6}$$
$$T7 = Q2\ \overline{Q3}$$
$$T5 = Q4\ \overline{Q5}$$
$$T4 = Q5\ \overline{Q6}$$
$$T3 = Q2\ Q6$$
$$T2 = \overline{Q2}\ Q3$$
$$T1 = \overline{Q3}\ Q4$$
$$T0 = Q1\ \overline{Q4}\ Q5$$
$$Tr = \overline{Q1}\ Q5$$
$$Tp = \overline{Q5}\ Q6$$
$$T1 + T0 = Q1\ \overline{Q3}\ Q5$$
$$T7 + T6 = Q2\ \overline{Q4}$$
$$T6\text{-}T0 = Q1\ \overline{T7}$$
$$T6\text{-}T3 = Q2\ Q3$$
$$T6\text{-}Tr = Q3 + Q5$$

3-33 Phase Control

3-34 The internal operations of the computer can be grouped into general functions such as indexing, waiting for input/output, using core memory, testing, or skipping. Each of these functions is performed during its own unique phase of internal control. For example, waiting for input/output is always done in phase 2, and skipping is always accomplished during phase 7.

3-35 Only instructions that wait for input/output will use phase 2. All other instructions will bypass phase 2. Instructions are built up in the hardware by sequencing them through the phases that perform the desired functions and bypassing the other phases. The entire 930 Computer instruction list uses a total of eight phases, designated phase 0 through phase 7. A general description of these phases is given below.

a. Phase Zero (Ø0). This phase is the beginning of all instructions. All instructions start at phase 0 T8 with the new instruction in the C-register. At phase 0 T8 the instruction code (opcode) is transferred to the O-register. Indexing and indirect addressing are performed in phase 0; also, the memory regenerates the instruction (writes it back into core memory because of destructive readout) and fetches the operand. Indirect addressing of the operand will cause the instruction to remain in phase 0 an additional machine cycle for each indirect address bit encountered. Instructions that do not permit indexing or indirect addressing will be in phase 0 for only one pulse time (T8), and the next pulse time (T7) will be in phase 5.

b. Phase One (Ø1). This phase is the setup or preparation phase for the shift (66,67) and divide (65) instructions. Phase 1 and phase 3 inhibit pulsing the memory for an operand. The memory address(s) register is now free to count the shifts or machine cycles during phase 3 of shift or divide instructions.

c. Phase Two (Ø2). This phase is the wait phase for input/output instructions (10, 12, 13, 30, 32, 33). The computer waits in this phase until the data is ready. If the data is ready before the instruction is given, phase 2 is bypassed, except for the parallel input/output instructions (13, 33), where phase 2 lasts at least one machine cycle time for data to be transferred.

d. Phase Three (Ø3). This phase is the execution phase for multiplication (64), division (65), and shifting (66, 67), and usually requires several machine cycles. In shifting, the duration of this phase is dependent upon the number of shifts required.

e. Phase Four (Ø4). This phase is the second cycle time of three-cycle instructions that write new data into core memory. The three cycles are phase 0, phase 4, and phase 7. During phase 4, the contents of the program location (P) register are increased by one to obtain the address of the next instruction, and the word to be stored is shifted serially by octals into the C-register.

f. Phase Five (Ø5). This phase is the execution phase for some single-cycle instructions. The C-register does not shift, and memory parity for the instruction word is not checked.

g. Phase Six (Ø6). This phase is the main execution phase of instructions requiring an operand from core memory, but no memory modification. These instructions are the two-cycle instructions and include conditional skips which may require three cycles.

h. Phase Seven (Ø7). During this cycle, memory regeneration (or new storage) of data words and next instruction access are performed. The contents of the P-register are not increased unless the skip flip-flop is set.

i. End Phase. An End is defined as the last cycle of execution for the particular instruction being performed. The End term designates that the next phase is to be phase 0.

3-36 Summary of Phase Functions. The phase functions in the 930 Computer may be briefly summarized as follows:

Phase 0 – Index and indirect address
Phase 1 – Prepare shift and divide
Phase 2 – Input/output wait
Phase 3 – Multiply, divide, or shift
Phase 4 – Memory destined word to C-register
Phase 5 – Do nothing or external test
Phase 6 – Use operand from memory
Phase 7 – Write new memory word or skip
End – Last cycle of instruction

3-37 Phase Counter

3-38 Successive machine cycles are given phase designations by a phase counter consisting of three flip-flops, F1, F2, and F3. Phases 0 through 7 are obtained by using all eight states of the counter, F3 being the least significant bit:

$\overline{F1}$	$\overline{F2}$	$\overline{F3}$	=	Ø0
$\overline{F1}$	$\overline{F2}$	F3	=	Ø1
$\overline{F1}$	F2	$\overline{F3}$	=	Ø2
$\overline{F1}$	F2	F3	=	Ø3
F1	$\overline{F2}$	$\overline{F3}$	=	Ø4
F1	$\overline{F2}$	F3	=	Ø5
F1	F2	$\overline{F3}$	=	Ø6
F1	F2	F3	=	Ø7

3-39 Advancement of the phase counter and sequence of phases are determined by signals derived from the instruction codes. Usually the Tp signal at the end of a machine cycle advances the phase counter. In some cases a phase consists of partial machine cycle, and the counter is advanced by other timing pulses. The following equations apply (numbers in parentheses indicate applicable instructions):

sF1 = Tp (Sk + Eax + Ø4)

(14-17, 34-37, 54-57, 74-77) + (Tp $\overline{Ia}$ Ø0) 03 04

(41, 43, 50-53, 60-63, 70-73) + (Tp $\overline{Ia}$ Ø0) 01 $\overline{04}$

(10-13, 30-33) + (Tp ($\overline{F1}$ $\overline{F3}$ $\overline{01}$ 03 $\overline{04}$) $\overline{Ia}$ Rf)

+ T8 $\overline{G0}$

+ [Ø0 T8 $\overline{Ia}$ $\overline{C2}$ $\overline{C5}$ $\overline{C8}$ ($\overline{C3}$ + $\overline{C4}$)]

rF1 = (Tp End $\overline{Sk}$)

sF2 = Tp (Sk + Eax + Ø4)

(41, 43, 50-57) + (Tp $\overline{Ia}$ Ø0) 01 $\overline{02}$

(1-, 5-, 7-) + (Tp $\overline{Ia}$ Ø0) 03 (01 + $\overline{02}$)

(30, 32, 33) + (Tp $\overline{Ia}$ Ø0) 03 $\overline{04}$ $\overline{Rf}$

(65) + Ø1 Tp

(66, 67) + Ø1 05 $\overline{Q2}$

(64) + (Tp $\overline{Ia}$ Ø0) (04 $\overline{05}$ $\overline{06}$)

rF2 = (Tp End $\overline{Sk}$)

(30-33) + (Tp $\overline{F1}$ $\overline{F3}$ $\overline{01}$ 03 $\overline{04}$ $\overline{Ia}$ Rf) 02

sF3 = Tp (Sk + Eax + Ø4

+ [Ø0 T8 $\overline{Ia}$ $\overline{C2}$ $\overline{C5}$ $\overline{C8}$ ($\overline{C3}$ + $\overline{C4}$)]

+ T8 $\overline{G0}$

(64,65) + (Tp $\overline{Ia}$ Ø0) $\overline{03}$ 04

(66,67) + Ø0 $\overline{Ia}$ Q4 $\overline{03}$ 04 05

rF3 = (Tp End $\overline{Sk}$)

3-40 The phase counter is always reset to zero during Tp End $\overline{Sk}$, where End indicates the last cycle of an instruction and Sk is the skip flip-flop. The following example explains why Tp End is qualified by $\overline{Sk}$. The phasing for one instruction during which a skip is performed is phase 0 – phase 6 – phase 7 – phase 0. One condition which defines End is F1 F2; therefore, End is true during phase 6 and phase 7 of the example presented. Since a skip is to be performed, Sk is true at phase 6 Tp, and reset of the phase counter is inhibited. The skip flip-flop is reset at phase 7 T0, enabling the phase counter to be reset at phase 7 Tp End.

3-41 A more detailed discussion of control of phase sequence is given in the section on individual instructions.

3-42 HARDWARE MECHANIZATION

3-43 Data is represented in binary form. A logical ONE is represented by a +4 volt signal, meaning that an element, such as a flip-flop, is set, or true. A logical ZERO is represented by a 0-volt signal, meaning that an element is reset, or false.

3-44 Flip-flops in the 930 Computer are ac-clocked, meaning that the flip-flop triggers on the trailing edge of the clocking pulse. This type of clocking enables old data to be read out as new data is being stored in the flip-flop. If both the set and reset sides are pulsed simultaneously, the output is unpredictable. The flip-flops are individually clocked; clock is not a part of the enable term.

3-45 Logic Mechanization

3-46 NAND logic is used in the 930 computer. A typical logical representation of the circuits used is shown in figure 3-2. Some typical gating is also shown. Typical logic symbols are given in table 3-4.

3-47 REGISTERS

3-48 All registers in the 930 Computer are static registers, with the exception of the index register, which is a dynamic register in that it must continually recirculate in order to retain its information. Data transfer, in general, is serial-octal, three bits of data being transferred during each pulse period.

3-49 The registers are shown in the machine block diagram in figure 3-3. Data flow paths are shown, and the number of bits transferred via each path. The registers are identified in table 3-5.

Table 3-4. Logic Symbols

Symbol		Description
(A)	=	the contents of the A-register.
	=	movement of data. For example: $A_{21-23} \rightarrow A_{0-2}$ means bits 21 thru 23 of the A register are transferred to bits 0 thru 2 of the A-register.
(A0 ⊕ A1)	=	exclusive OR of the 0 and 1 bits of the A-register. Odd number of ONES.
sA0	=	set bit 0 of A-register.
rA0	=	reset bit 0 of A-register.
(T6-Tr)	=	duration of time covered by T6 thru Tr, inclusive.
Sc	=	clear the S-register.
Sxp	=	transfer the contents of the P-register to the S-register, or (P)→S.

Table 3-5. Central Processor Registers

Symbol	Description
A	Main accumulator.
B	Extended accumulator when operating on double precision words.
C	Arithmatic and control. All data transferred to or from memory via the main frame passes thru the C register. Also used for parity generation and check.
X	Index register.
O	Opcode or instruction register.
P	Program counter. Normally contains the address of the next instruction.
S	Memory address register, shift counter.
M	Memory data register.
Jz	Address register for direct access buffer.
Rn	3-bit register for receiving an octal of data from input/output.

3-50 Movement of data into the registers must be synchronized with clock. This is accomplished by enable paths.

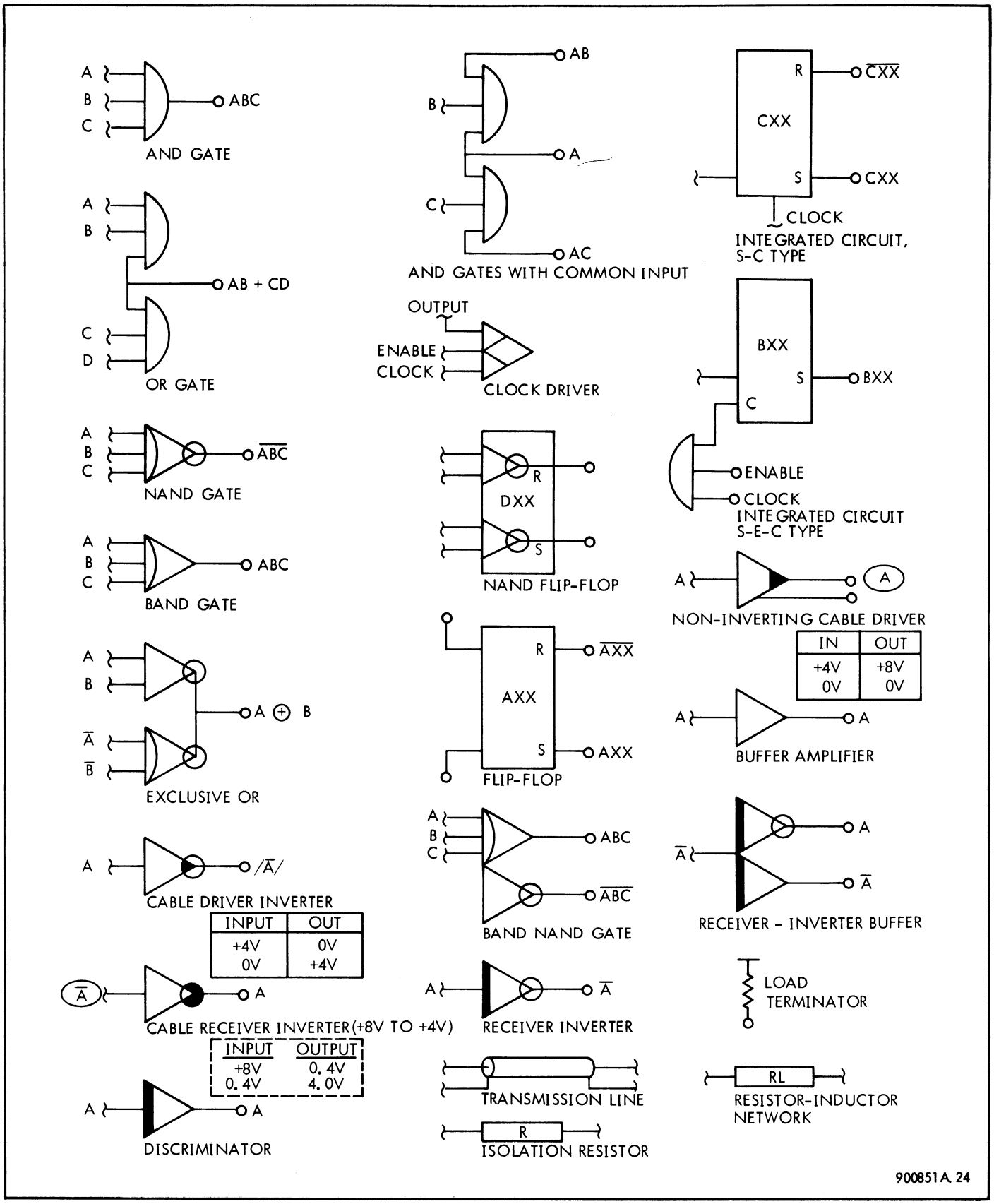


Figure 3-2. Logic Symbol Diagram

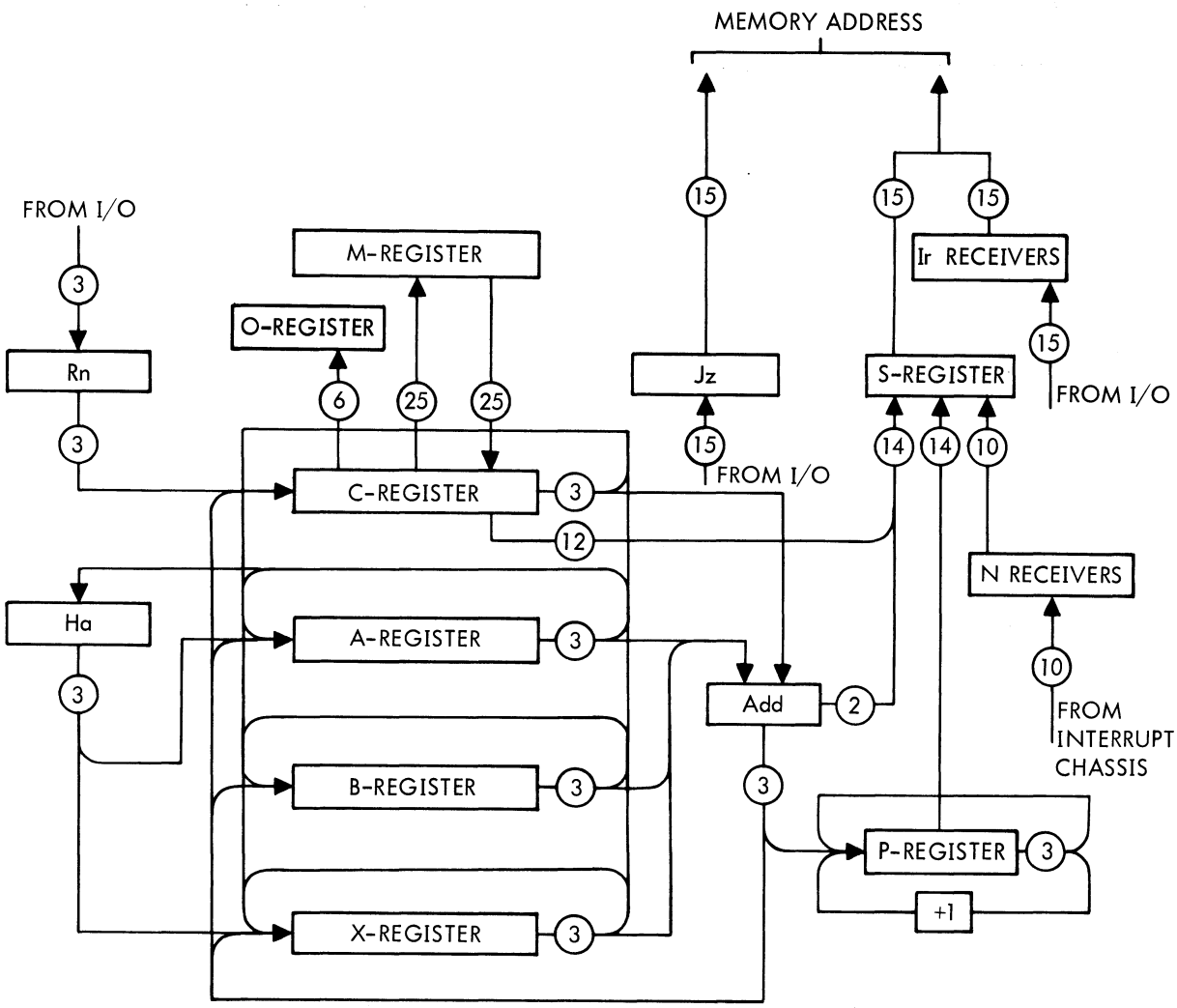


Figure 3-3. Computer Registers, Block Diagram

For example, if the A-register is to recirculate, the information in bits 21 through 23 must be placed in bits 0 through 2. The input gates to A0 through A2 will have Ar3, A21, Ar3, A22, and Ar3, A23, respectively. The enable path is Ar3. One of the qualifying terms for Cr3 is a pulse-counter flip-flop:

$$Ar3 = \overline{F1}\ \overline{F2}\ \overline{O5}\ Q1 + \ldots$$

where

$$Q1 = (T7\text{-}T0)$$

When an enable path does not have a qualifying term directly derived from the timing counter, a flip-flop is used as a qualifying term. The flip-flop is true for a specified duration of time.

3-51 Table 3-6 explains some of the enable paths in the central processor.

3-52 A- And B-Registers (A0 through A23, B0 through B23)

3-53 The 24-flip-flop A-register is the main accumulator of the computer. For double-length numbers, the 24-flip-flop B-register is used as an extension of the A-register; therefore, the B-register can be considered an extended accumulator. Supplementary circuits considered as an integral part of the B-register are used for multiply, divide, and shift operations. When working with double-length numbers, the A-register contains the most significant portion of the word, and the B-register contains the least significant portion. Bit position 0 of the A-register contains the sign bit of the double precision word, and bit position 0 of the B-register contains one of the data bits. Bit position 23 of the B-register is disregarded. When the A- and B-registers are used separately, A0 and B0 contain the sign bits of the respective data words.

Table 3-6. Register Enable Paths

Symbol	Description
Anr	A denotes register concerned; nr means not recirculate.
Ar1	The A-register is shifted right one bit position for each clock pulse period during which this enable term is active, A00→A0, A0→A1, A1→A2, etc. Not an end-around or recirculating transfer path.
Ar3	The A-register is shifted right an octal at a time for each pulse period that this enable term is active: $A_{0\text{-}2} \rightarrow A_{3\text{-}5}$, $A_{3\text{-}5} \rightarrow A_{6\text{-}8}$, etc. May be used to perform end-around shift or recirculation of data contained in A, in which case $A_{21\text{-}23} \rightarrow A_{0\text{-}2}$. If new data is to be shifted into the A-register, $A_{21\text{-}23} \rightarrow A_{0\text{-}2}$ is inhibited by Anr. New data is enabled into $A_{0\text{-}2}$ by proper decoding of the instruction being performed, and A is shifted right an octal at a time until the proper amount of data has been loaded in the A-register. This path is also used as the right three path for the B-register. If A and B are to be exchanged, $A_{21\text{-}23} \rightarrow B_{0\text{-}2}$ and $B_{21\text{-}23} \rightarrow A_{0\text{-}2}$.
Aℓ2	The A-register is shifted left two bit positions each clock pulse period during which this enable is active: A2→A0, A3→A1, A4→A2, etc. This path is also used as a left two shift for the A- and B-register: B0→A22, B1→A23, A0→B22, and A1→B23.
Cxm	Parallel transfers the contents of M-register to C-register.
Ck	ONES complements, or inverts, C-register.
Sc	Clears S-register.
Sxp	Parallel transfers (P) to S.

3-54 Figure 3-4 indicates the enable paths for the A- and B-registers. Enable paths Ar3 and A12 are used on both registers to shift the data right three positions or left two positions, respectively. When either of these paths is active and the A- and B-registers are not being used as a double-length register, the enable path is further qualified at the input gates to the register. Signals Anr and Bnr are used to inhibit recirculation of the respective registers. For example, Bnr inhibits the end-around shift of B_{21-23} into B_{0-2} during the time that the B-register is undergoing a change operation. Enables Ar1 and Br1 shift the data in the respective registers right one bit position, and Rsa transfers the contents of the right shift adder into the B-register in parallel.

3-55 C-Register (C0 through C23)

3-56 The C-register acts as a central transfer station for all information going to and from memory. During certain operations, the C-register performs arithmetic functions. The C-register, parity flip-flops, and enable paths are illustrated in figure 3-5.

Flip-flops C24, Cp, Cpi, and Cpr are used for checking and generating parity and for receiving parity interrupt information. Parity logic is explained in paragraph 3-140.

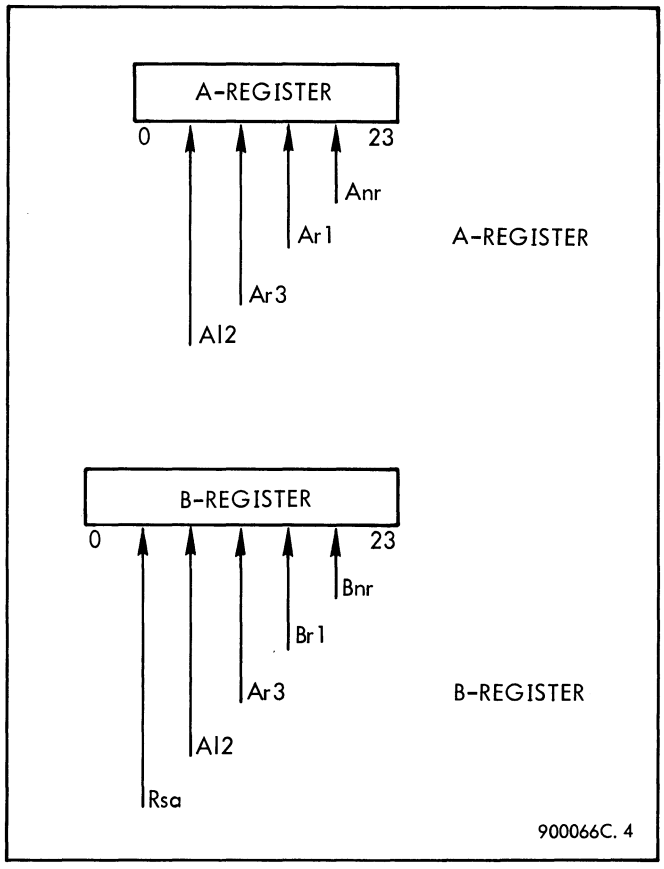


Figure 3-4. A- and B-Register Enable Paths

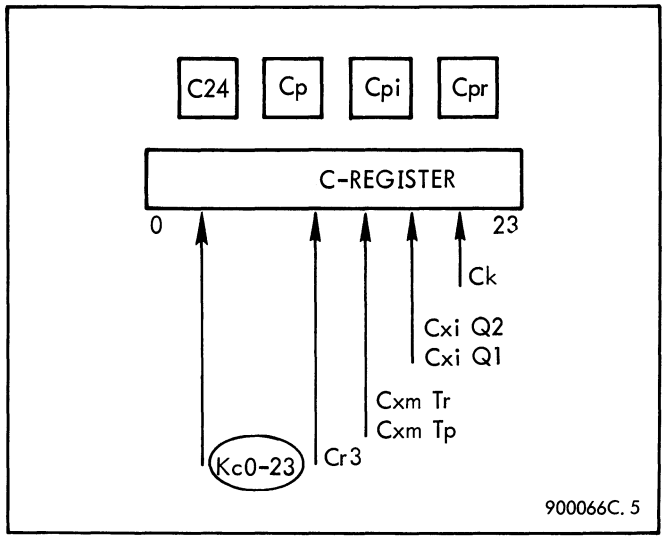


Figure 3-5. C-Register Enable Paths

3-57 Enable signal Ck inverts the contents of the C-register. When transferring data from the memory register into the C-register, Cxm Tr clears the C-register and Cxm Tp enables the data from the selected memory word into the C-register. For end-around shaft during parity checking and generation, Cr3 shifts the C-register right three bits at a time. When loading parallel input data in the C-register, Cxi Q2 clears the C-register, and Cxi Q1 loads the data into the register. The Kc0-Kc23 inputs are dc set terms energized by the control console set pushbuttons.

3-58 Index Register (X)

3-59 The index register is a recirculating register using three dynamic serial flip-flop shift circuits. To hold information, these registers must circulate constantly.

3-60 Each of the three legs of the index register consists of 11 stages, corresponding to the 11 pulse times in a machine cycle. During normal recirculation, the output of the n (now) flip-flop is fed to the write flip-flop, which in turn feeds the dynamic register stages. In the first leg of the index register, the output of Xn1 feeds an additional flip-flop, Xw1. The added delay of this flip-flop is compensated for by disconnecting one of the integrated circuit stages, thereby reducing the number of dynamic stages by one. A diagram of the index register is shown in figure 3-6.

3-61 Each of the recirculating loops of the index register holds eight bits of the 24-bit word. The first loop holds only the most significant bits of each of the eight octal digits in a word. The second loop holds the middle bits of the eight octals. The third recirculating loop holds the least significant bits of each of the eight octals. If the octal word 07030407 is being held in the register, the bits at one pulse time will appear as follows:

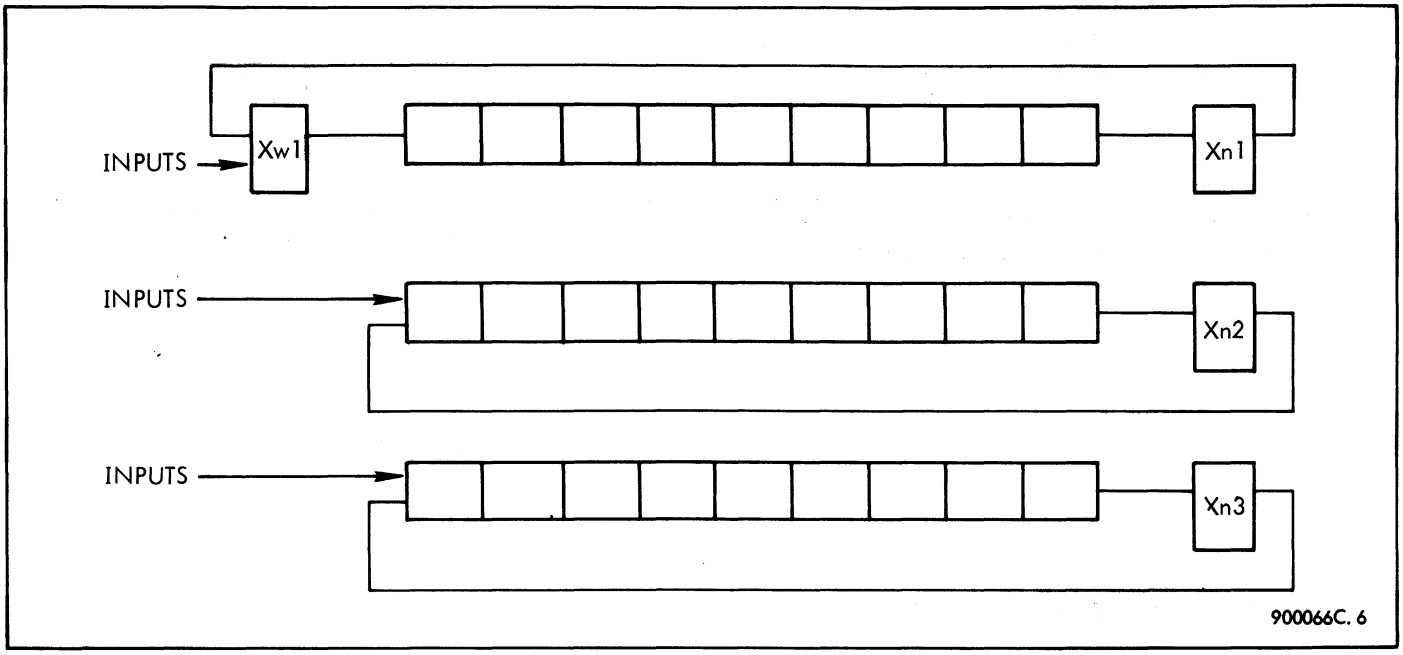
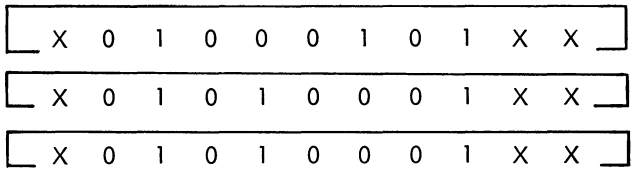


Figure 3-6. Index Register



3-62 The only output available from this register is an octal of data from $Xn_{1-3;}$ outputs are not available from the intermediate stages. Recirculation of old data is inhibited during certain operations by the term Xnr. The index register is used primarily for address modification. If the index bit of an instruction contains a ONE, the computer adds the contents of bits 10 through 23 in the index register to the contents of the address field of the instruction prior to execution of the instruction. This addition does not use any overflow or carry beyond the fourteenth address bit.

3-63 Instruction Register (O1-O6)

3-64 The instruction register, or opcode register, contains the six-bit instruction code during the execution of the instruction. This register does not shift, but only receives information. The output of the O-register is decoded to define the logical operation to be executed. The enable paths for the instruction register are shown in figure 3-7.

3-65 Clear term Oc clears bit 1 and bits 3 through 6 and sets bit 2 of the opcode register, thereby entering a no operation instruction (NOP). Effective transfer from the C-register is $C_{3-8} \rightarrow O_{1-6}$. Bit 2 of the O-register is set

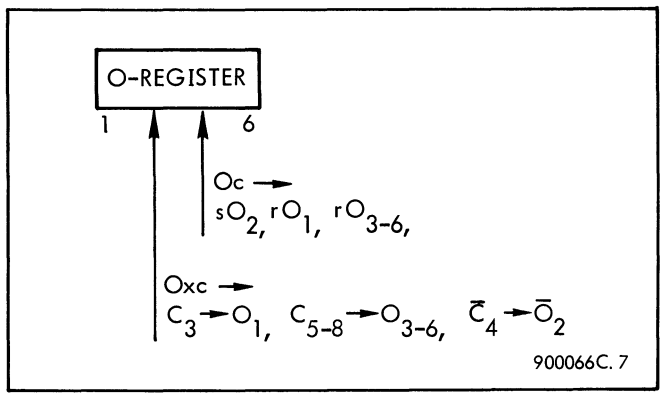


Figure 3-7. O-Register Enable Paths

by Oc; if C4 is false for a particular instruction O2 is reset by the data transfer enable path Oxc, which is active during Ø0 T8. Normally, the O-register has time to stablilize during the remainder of Ø0 before being decoded.

3-66 Sometimes the instruction must be decoded before it is available in the O-register. In these special cases, the instruction is decoded from the proper C-register bits and used to set a control flip-flop. The control flip-flop output is available for gating purposes, and the four or five levels of logic required to decode the output of the O-register are bypassed.

3-67 The Oxc signal is qualified by $\overline{C2}$. If bit position 2 of the C-register contains a ONE, a programmed operator function is implied. During such an operation, the C bits,

which normally define the instruction, are used as a subroutine address, and these bits are not transferred into the O-register.

3-68 Program Counter (P0 through P14)

3-69 The P-register is the instruction address counter, and is the source of the instruction address for the S-register. Transfer from the P-register to the S-register is parallel. The contents of the P-register are increased by one just before the memory is addressed for the next instruction. The circuits that add one to the P-register contents are described in paragraph 3-95.

3-70 New data enters P_{0-2} by octal-serial transfer enabled by Pr3 signal is then used to shift the data to the right until the proper number of octals have been loaded. The address for branch instructions enters P0-2 of the P-register via Add1, 2, and 3. The programmed operator address enters the P-register via $C_{6-8} \rightarrow P_{0-2}$. When an increment index and branch instruction (BRX) or a mark place and branch instruction (BRM) is executed, the address to be saved is copied octal-serially into the C-register via $P_{12-14} \longrightarrow C_{0-2}$.

3-71 Memory Address Register (S1 through S14)

3-72 The S-register holds the address of the memory location to be addressed. When the memory circuitry is actively reading or writing, the S-register must remain in the static state. Information is received in parallel from the P-register (instruction address), the C-register (operand address), or from the interrupt address lines, N5 through N14.

3-73 The S-register is also used as a counter during divide and shift operations. The enable paths are shown in figure 3-8.

3-74 Before data transfer, the S-register is cleared by Sc. The parallel transfer path for the operand address and shift count into the S-register is Sxc. The 12 least significant bits are transferred from bits 0 through 11 of the C-register, and the two most significant bits are transferred through bits 2 and 3 of the three-bit serial adder (Add2 and Add3).

3-75 The transfer path for the next instruction address into the S-register is Sxp. The address is taken from P_{0-11}, sP1, and sP2. The latter two terms are the set inputs for P1 and P2 on the next shift, and represent the two most significant bits of the address.

3-76 Term Sxn enables the interrupt address from the N receivers to be transferred in parallel into the S-register. Only bit positions 5 through 14 of the S-register are filled, because the N receivers contains ten bits.

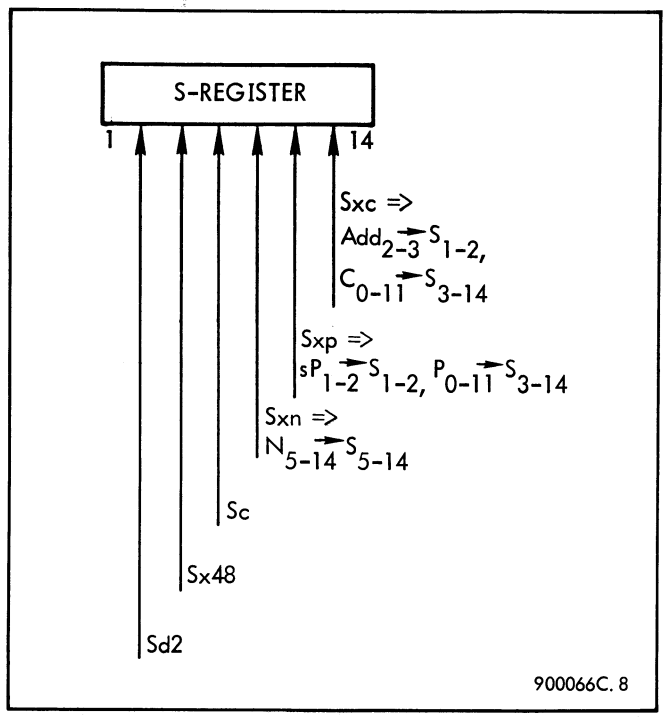


Figure 3-8. S-Register Enable Paths

3-77 The Sd2 signal is used to count the S-register down during divide and shift operations. The Sx48 term forces a 48_{10} into the S-register if the number of shifts required is greater than 48_{10}. The Sd2 and Sx48 signals are explained further in the section on shift and divide.

3-78 Memory Data Register (M0 through M24)

3-79 The M-register, containing 24 bits of data plus a parity bit, is the source of all data received from memory, and is the destination of all data to be stored in memory.

3-80 Address Register For Direct Access Buffer (Jz0 through Jz14)

3-81 The Jz register, which is part of the model 92990 Multiple Access to Memory option, stores the 15 address bits from the direct access communication channel. The 15 flip-flops are set in parallel by signals on receivers Iz0 through Iz14. Outputs of the Jz register are fed to cable drivers Lz0 through Lz14, which send the memory address for the I/O channel to the memory.

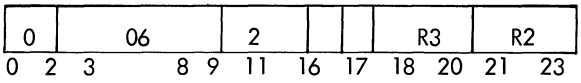
3-82 Input/Output Data Register (Rn1 through Rn3)

3-83 The three-bit Rn register receives an octal digit at a time in parallel from the time multiplexed communication channel on data lines Rwy1 through Rwy 3. The octal data is transferred in parallel into bits 0 through 2 of the C-register.

3-84 Memory Extension Registers

3-85 Two three-bit memory extension registers, EM3 and EM2, allow addressing of memories greater than 16,384 words. The EM3 register consists of flip-flops E3m0 through E3m2. The EM2 register consists of flip-flops E2m0 through E2m2. Either or both of the registers may be loaded, and may be used one at a time to provide the most significant octal of a memory address. The S-register alone can supply only 14 bits of memory address, while the EM registers may increase the address to 15 bits.

3-86 The enable paths for the EM3 and EM2 registers are shown in figure 3-9. The registers are loaded with a Set Extension Register EOD instruction:



The R3 field is the octal number to be placed in the EM3 register; the R2 field is the octal number to be placed in the EM2 register. An octal 2 in bits 9 through 11 enables loading both registers when qualified by bits 16 and 17. A ONE in bit 16 gates C18 through C20 into the EM3 register; a ONE in bit 17 gates C21-23 into the EM2 register. Setting bits 16 and 17 loads both registers:

$$sE3m0\text{-}2 = Eod\ C10\ \overline{C11}\ C16\ T4\ C18\text{-}20$$

$$sE2m0\text{-}2 = Eod\ C10\ \overline{C11}\ C17\ T4\ C21\text{-}23$$

3-87 The EM3 and EM2 registers are used in combination with S1 and S2 of the address register. After the memory extension registers are loaded with a Set Extension Register EOD, the address portion of the instructions calling for the EM registers as part of the memory address must contain the proper code in bits 10 and 11. An octal three in bits 9,10, and 11 activates EM3. An octal two in bits 9, 10, and 11 activates EM2. For example, if the instruction is ADD 34000 the most significant octal digit of the address is replaced by the contents of EM3. If the instruction is ADD 24000, the most significant octal digit of the address is replaced by the contents of EM2. The S-register is loaded by the instruction in the usual manner.

3-88 The address is routed to memory on the memory address lines, Ls0-14. The most significant three bits are decoded from the combination of the EM registers and S1 and S2:

$$Ls0 = \overline{Tsm}\ (S1\ \overline{S2}\ E2m0 + S1\ S2\ E3m0) + \ldots$$

$$Ls1 = \overline{Tsm}\ (S1\ \overline{S2}\ E2m1 + S1\ S2\ E3m1) + \ldots$$

$$Ls2 = \overline{Tsm}\ (\overline{S1}\ S2 + S2\ E3m2 + S1\ \overline{S2}\ E2m2) + \ldots$$

When the program counter is counting successive memory locations and memory is being addressed by means of the S-register, an octal 7 in EM3 has the effect of adding a fifteenth bit at the most significant end of the address when the S-register is filled with 37777. When addressing the lower 16K memory locations, EM3 must be set to 3 and EM2 must be set to 2.

3-89 Bits E3m0 and E2m0 are used independently of the memory address function to select an input/output medium. The Kfp, Kfm, Kfd, and Kfc signals from the PAPER TAPE, MAG TAPE, DRUM, and CARDS FILL switches on the control console place a two-bit code in E3m0 and E2m0. The function of this code is described in the section on the control console in paragraph 3-221.

3-90 Model 91903 32K Addressing Modification Feature. Memory Extension Model 91903 feature is a standard option for the 930 Computer. This feature supplements the standard memory extension system by allowing a fifteenth address bit to be specified in an instruction. Using this fifteenth bit, a full 32, 768-word memory can be directly addressed. Logic diagrams containing the addressing modification feature are given in figure 3-10.

3-91 When the Model 91903 feature is installed, a program-controlled flip-flop determines whether addressing is in the normal mode or the extended mode. An EOD 20000 instruction establishes the normal addressing mode, and an EOD 20400 establishes the extended mode. When the computer is in the extended addressing mode, the instruction word is interpreted as follows:

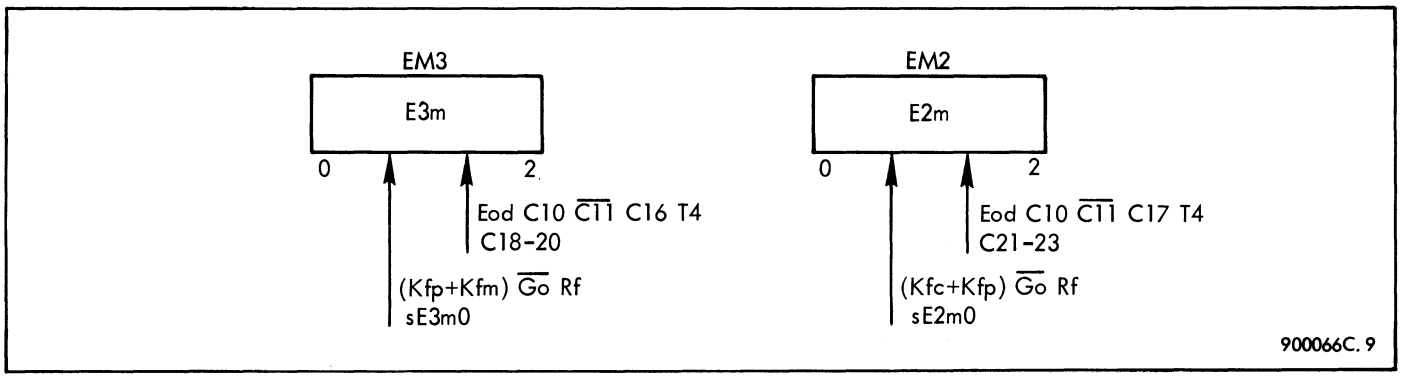


Figure 3-9. Memory Extension Register Enable Paths

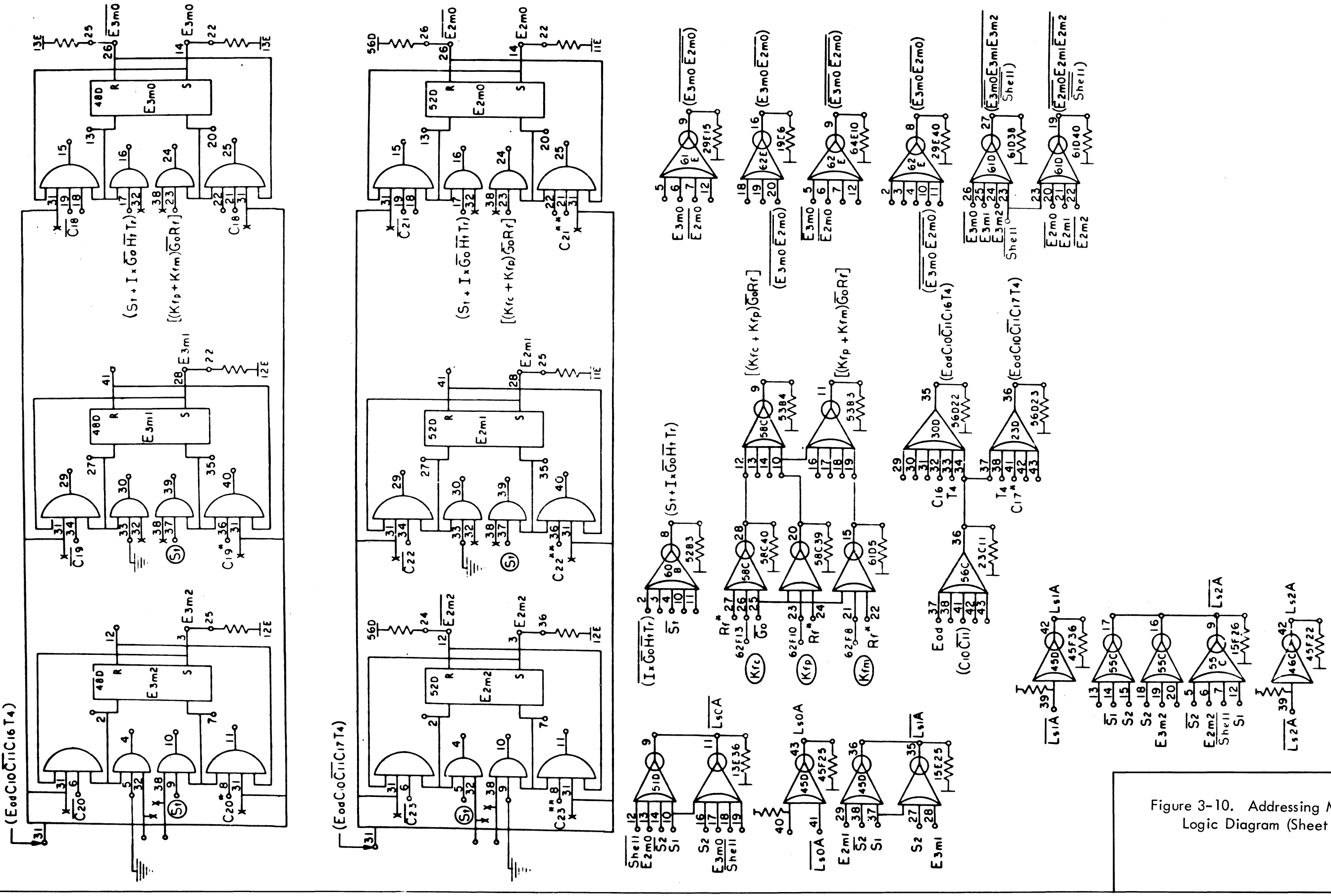


Figure 3-10. Addressing Modification, Logic Diagram (Sheet 1 of 3)

108843-2B

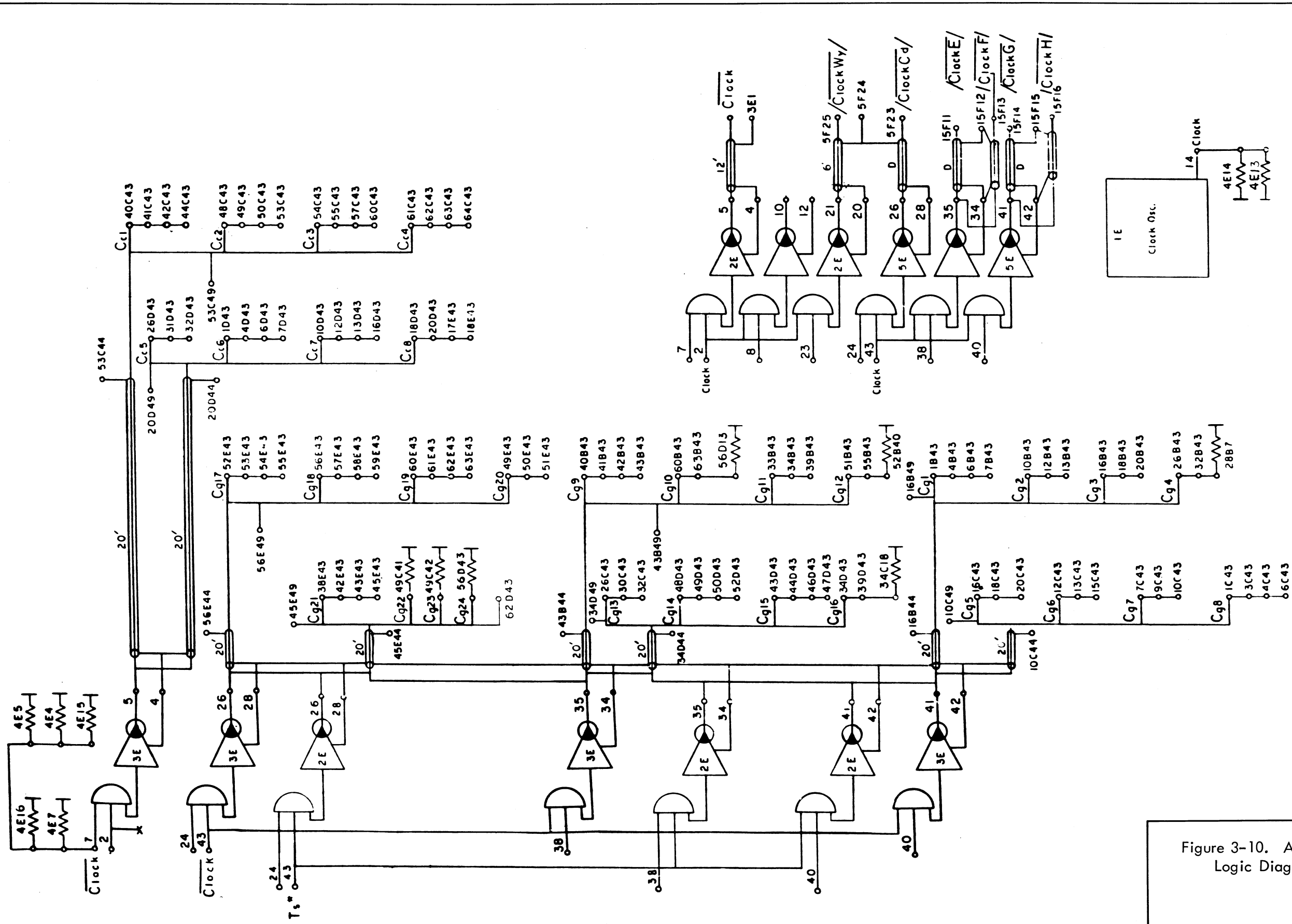
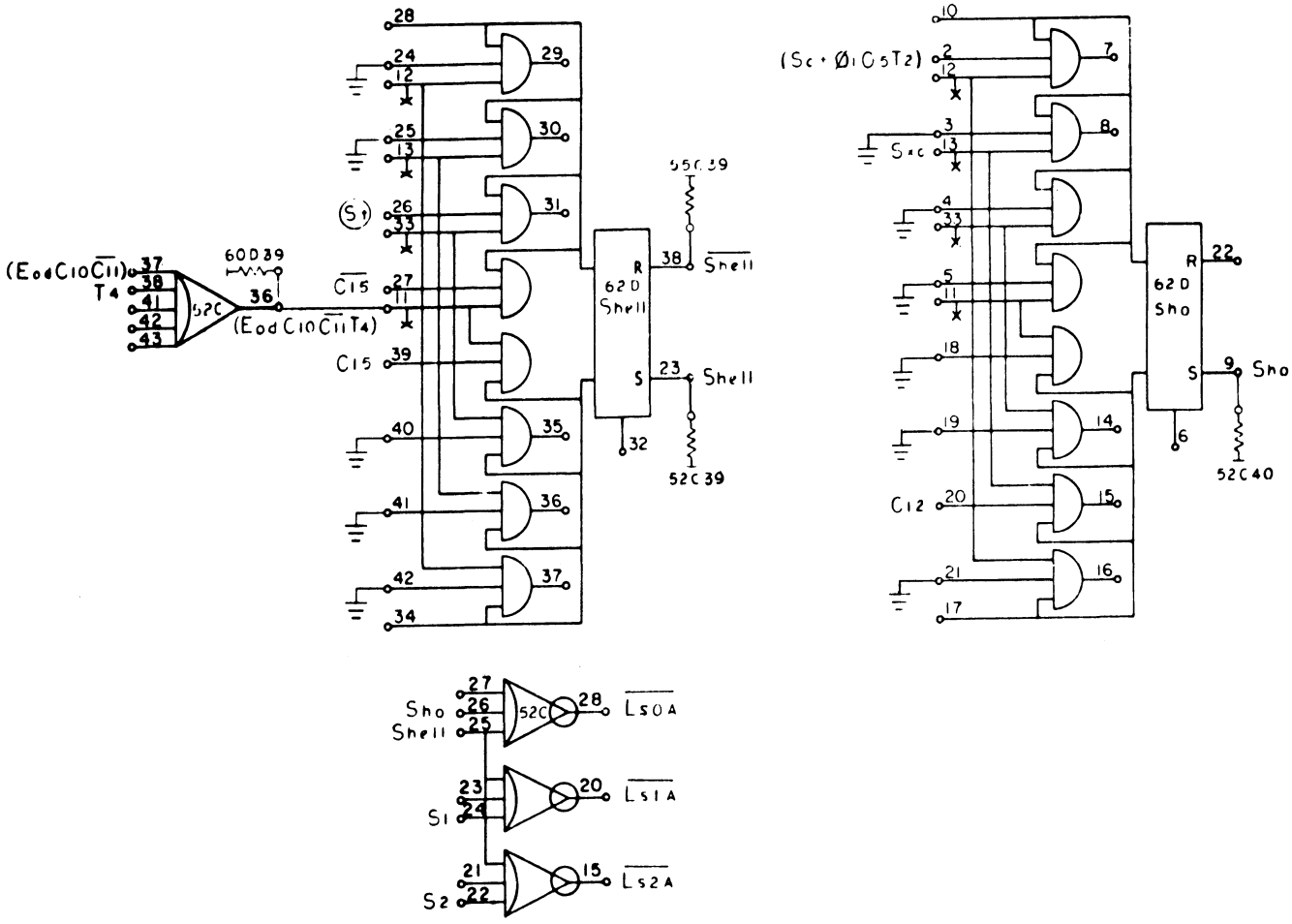


Figure 3-10. Addressing Modification, Logic Diagram (Sheet 2 of 3)

108843-3B



108843-4B

Figure 3-10. Addressing Modification, Logic Diagram (Sheet 3 of 3)

0	1	2 3 ... 8	9	10 ... 23
R	X	Instruction Code	I	Address Field

If R is a ZERO, reference is to memory locations 0 to 16,383. If R is a ONE, reference is to memory locations 16,384 to 32,767 for data only. Bit 0 is thus the most significant bit of the required 15-bit address. Indexing occurs only on the low order 14 bits. Bit 0 is ignored in branch instructions. An instruction sequence cannot be executed in locations above $16,384_{10}$ with the addressing modification feature activated.

3-92 The logic for the 91903 Addressing Modification Feature is shown in figure 7-3. The Shell flip-flop is set by the extended address EOD:

$$sShell = C15\ Eod\ C10\ \overline{C11}\ T4$$

The Sh0 flip-flop is set by bit 0 of the instruction word:

$$sSh0 = C12\ Sxc$$

At T3, when Sxc is true, the C-register has shifted four octals to the right, so that the bit originally in position 0 is now in position 12. The Shell output disables the terms normally generating $\overline{Ls0A}$ and substitues Shell Sh0. This in effect gates bit 0 of the instruction word onto bit 0 of the memory address lines Ls0 through Ls14. The Shell term gates S1 and S2 onto Ls1 and Ls2. Since the EM3 and EM2 registers are not used with the 91903, their terms are not used in addresssing. The Shell term also disables the $\overline{(E3m0\ E3m1\ E3m2)}$ and $\overline{(E2m0\ E2m1\ E2m2)}$ signals which light the memory extension indicators on the control panel.

3-93 ADDERS

3-94 The 930 Computer central processor contains four adding circuits: a right shift adder, a half adder, a full adder, and circuits to add one to the contents of the P-register.

3-95 P+1 Adder (P0 through P14)

3-96 The P+1 adder is the circuitry required to increase the contents of the program counter, or P-reigster, by one. The increment is added by octal ring-shifting to the right for one complete revolution, using the indirect address flip-flop, Ia, as a carry as suggested in figure 3-11, the octal digit in P12 through P14 is increased by one as it shifts into P0 through P2 if Ia is true and remains the same if Ia is false.

3-97 The P-register recirculates during pulse times T7 through T3 in phases 4, 5, 6, and 7:

$$Pr3 = F1\ Go\ Q2 + \ldots$$

Adding one takes place during phases 4 and 6, and sometimes 7 and 0:

$$sIa = T8\ F1\ \overline{F3}\ \overline{Kr}\ (\overline{Ij} + Inr) + \emptyset 0\ \overline{Ia}\ T8\ \overline{C2}\ \overline{C5}\ \overline{C8}\ (\overline{C3} + \overline{C4})\ \overline{Kr} + T8\ \emptyset 7\ Sk\ \overline{Kr}\ (\overline{Ij} + Inr) + \ldots$$

$$rIa = \overline{(P12\ P13\ P14)}\ Q2\ F1 + Tr\ F1 + \ldots$$

3-98 The second set Ia gate is for instructions that are in phase 0 for only one pulse time (T8) and then advance immediately to phase 5. Addition of one is performed in phase 7 only if the skip flip-flop, Sk, is set. Single instruction interrupt channels prevent adding one with Ij, and the HOLD switch on the control console prevents adding one with $\overline{Kr}$. The Inr term in the first and third Ia gate is to allow the P-register to be increased by one twice for a skip instruction being executed when a single instruction interrupt occurs.

3-99 The sum is generated by inputs to P_{0-2} from P_{12-14}. If the contents of P_{12-14} are less than 7, Ia is reset after increasing by one, and during the next octal shift the digit is moved without change. A 7 in P_{12-14} allows Ia to remain set so that a carry is propagated into the next octal digit. The equations for P_{0-2} are as follows:

$$sP0 = Pr3\left[P12\ \overline{(P13\ P14\ Ia)}\ F1\ (Go + Kmc)\ \overline{(\overline{O2}\ \overline{O4}\ \overline{O5}\ O6)} + \overline{P12}\ (P13\ P14\ Ia)\ F1\ (Go + Kmc)\ \overline{(\overline{O2}\ \overline{O4}\ \overline{O5}\ O6)}\right]$$

$$rP0 = \overline{Pr3\left[P12\ \overline{(P13\ P14\ Ia)}\ F1\ (Go + Kmc)\ \overline{(\overline{O2}\ \overline{O4}\ \overline{O5}\ O6)} + \overline{P12}\ (P13\ P14\ Ia)\ F1\ (Go + Kmc)\ \overline{(\overline{O2}\ \overline{O4}\ \overline{O5}\ O6)}\right]}$$

$$sP1 = Pr3\left[P13\ \overline{(P14\ Ia)}\ F1\ (Go + Kmc)\ \overline{(\overline{O2}\ \overline{O4}\ \overline{O5}\ O6)} + \overline{P13}\ (P14\ Ia)\ F1\ (Go + Kmc)\ \overline{(\overline{O2}\ \overline{O4}\ \overline{O5}\ O6)}\right]$$

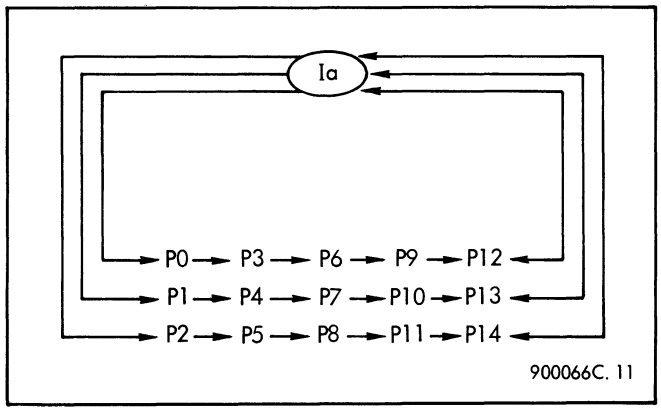


Figure 3-11. P-Register Shift Operation

$$rP1 = Pr3\Big[\overline{P13\ (\overline{P14\ Ia})\ F1\ (Go + Kmc)\ (\overline{O2\ O4\ O5\ O6})} + \overline{\overline{P13}\ (P14\ Ia)\ F1\ (Go + Kmc)\ (\overline{O2\ O4\ O5\ O6})}\Big]$$

$$sP2 = Pr3\Big[P14\ \overline{Ia}\ F1\ (Go + Kmc)\ (\overline{O2\ O4\ O5\ O6}) + \overline{P14}\ Ia\ F1\ (Go + Kmc)\ (\overline{O2\ O4\ O5\ O6})\Big]$$

$$rP2 = Pr3\Big[\overline{P14\ \overline{Ia}\ F1\ (Go + Kmc)\ (\overline{O2\ O4\ O5\ O6})} + \overline{\overline{P14}\ Ia\ F1\ (Go + Kmc)\ (\overline{O2\ O4\ O5\ O6})}\Big]$$

3-100 Half Adder (Ha1 through Ha3)

3-101 The half adder is used for adding one to the A- and the X-registers and complementing numbers in the A- and B-registers. If adder outputs are enabled into a register, the register must be shifting right an octal digit at a time in order to place the data in the proper bit positions.

3-102 The half adder inputs, Hx_{1-3} and $\overline{Hx}_{1-3}$, are derived from the A-, B-, and X-registers with time qualifiers. The carry is stored in a flip-flop, Hz. With Hz set, the number in Hx is increased by one. The equations for the half adder and the carry flip-flop are as follows:

$\overline{Hx1}$ = the inverse of:

(74) + (O3) $\overline{Xn1}$

(41) + $\overline{F1a}\ \overline{\emptyset 1}$ Xn1

(46, 65) + F1a $\overline{O3}\ \overline{A21}$

(65) + $\emptyset 1$ B21

$\overline{Hx2}$ = the inverse of:

+ (O3) $\overline{Xn2}$ + $\overline{F1}\ \overline{\emptyset 1}$ Xn2 + F1a $\overline{O3}\ \overline{A22}$

+ $\emptyset 1\ \overline{B22}$

$\overline{Hx3}$ = the inverse of:

+ (O3) $\overline{Xn3}$

+ $\overline{F1}\ \overline{\emptyset 1}$ Xn3

+ F1 $\overline{O3}\ \overline{A23}$

+ $\emptyset 1\ \overline{B23}$

+ $\emptyset 1$ Q2 $\overline{Q3}$

sHz = T8

rHz = $\overline{(Hx1\ Hx2\ Hx3)}\ \overline{T8}$

Ha1 = $\overline{\overline{Hx1}\ \overline{(Hx2\ Hx3\ Hz)} + Hx1\ (Hx2\ Hx3\ Hz)}$

Ha2 = $\overline{\overline{Hx2}\ \overline{(Hx3\ Hz)} + Hx2\ (Hx3\ Hz)}$

Ha3 = $\overline{\overline{Hx3}\ \overline{Hz} + Hx3\ Hz}$

3-103 In a complement operation (-A→A, -B→B), the the false side of the A- or B-register, which is the one's complement, is presented to the inputs of the half adder an octal at a time. Carry flip-flop Hz is initially set, and the outputs of the half adder are presented to the input gates of A_{0-2} or B_{0-2}. This logic, in effect, adds one to the one's complement of the number to form the two's complement. During phase 1 of divide, the two's complement of the numerator is obtained if the sign of the numerator is negative. During phase 7 of the divide, the two's complement of the quotient is taken if the numerator and denominator are of opposite sign.

3-104 When an SKD instruction is being performed (Skip if $M_{15-23} > B_{15-23}$) the half adder is used to form the absolute value of the number in X. For an Increment Index and Branch instruction (BRX), the half adder is used to add one to X.

3-105 Full Adder

3-106 The full adder, whose outputs are designated Add1-3, is a serial-octal adder used for summing and differencing two numbers and as a transfer path for data between two registers. Add 3 is the least significant bit. The full adder is used to transfer the branch address from the C-register to the P-register, to load the two most significant address bits of an operand address into the S-register, and to add bits 10 through 23 of the index register to the address field in the C-register when indexing is required.

3-107 The full adder addend inputs, designated Xz_{1-3} and $\overline{Xz}_{1-3}$, are derived from bits 21 through 23 of the A-B-registers and Xn_{1-3} of the index register. The augend inputs, designated Yz_{1-3} and $\overline{Yz}_{1-3}$, are derived from bits 21 through 23 of the C-register.

3-108 The adder sums two octal digits and generates a carry into the next more significant digit if the sum is greater than seven. The carry flip-flop, Cz, is set by the term Kz, with time and opcode qualifiers. The equations for the adder and Kz, with time and opcode qualifiers. The equations for the adder and Kz are as follows:

$\overline{Xz1}$ = the inverse of:

+ ($\emptyset 0$ + $\emptyset 7$ 05 + $\emptyset 6\ \overline{02}\ \overline{04}$ + $\emptyset 4\ \overline{05}$ 06) Ix Xn1
(67) (41, 51, 52) (61)

+ ($\emptyset 7\ \overline{05}\ \overline{06}$ + $\emptyset 7\ \overline{05}\ \overline{A00}$ + $\emptyset 6$ 02 04) B21
(64) (65) (74)

+ ($\emptyset 7\ \overline{05}$ 06 A00) $\overline{B21}$
(65)

$$+ \underset{(54\text{-}57)}{\left[(\emptyset 6\ \overline{02}\ 04)\right.} + \underset{(63)}{\emptyset 4\ 05} + \underset{(65)}{\left.\emptyset 1\ \overline{A00}\right]} A21$$

$$+ \underset{(65)}{(\emptyset 1\ A00} + \underset{(73)}{\emptyset 6\ 02\ \overline{04})}\ \overline{A21}$$

$$+ (\emptyset 4\ \overline{05}\ \overline{06})$$
(60)

(equation represents 0's to X input of adder at times control terms are active)

Xz1 = the inverse of:

$$+ \underset{(67)}{(\emptyset 0 + \emptyset 7\ 05} + \underset{(41,51,53)}{\emptyset 6\ \overline{02}\ \overline{04}} + \underset{(61)}{\emptyset 4\ \overline{05}\ 06)}\ (\overline{Ix} + \overline{Xn1})$$

$$+ \underset{(64)}{(\emptyset 7\ \overline{05}\ \overline{06}} + \underset{(65)}{\emptyset 7\ \overline{05}\ \overline{A00}} + \underset{(74)}{\emptyset 6\ 02\ 04)}\ \overline{B21}$$

$$+ (\emptyset 7\ \overline{05}\ 06\ A00)\ B21$$
(65)

$$+ \underset{(54\text{-}57)}{\left[(\emptyset 6\ \overline{02}\ 04)\right.} + \underset{(63)}{\emptyset 4\ 05} + \underset{(65)}{\left.\emptyset 1\ \overline{A00}\right]}\overline{A21}$$

$$+ \underset{(65)}{(\emptyset 1\ A00} + \underset{(73)}{\emptyset 6\ 02\ \overline{04})}\ A21$$

Similarly Xz2, Xz3

$$\overline{Yz1} = \overline{(\overline{\emptyset 7}\ C21 + \emptyset 7\ Rf)}$$

$$Yz1 = \overline{(\overline{\emptyset 7}\ \overline{C21} + \emptyset 7\ \overline{Rf})}$$

$$\overline{Yz2} = \overline{(\overline{\emptyset 7}\ C22 + \emptyset 7\ K0)}$$

$$Yz2 = \overline{(\overline{\emptyset 7}\ \overline{C22} + \emptyset 7\ \overline{K0})}$$

$$\overline{Yz3} = \overline{(\overline{\emptyset 7}\ C23 + \emptyset 7\ Bc23)}$$

$$Yz3 = \overline{(\overline{\emptyset 7}\ \overline{C23} + \emptyset 7\ \overline{Bc23})}$$

sCz =

(56, 57) + (Tr Ø0) 04 05 Xw1

(41, 51, 61) + (Tr Ø0) $\overline{04}$ ($\overline{05}$ 06)

(54, 74) + (Tr Ø0) (04 $\overline{05}$ $\overline{06}$)

(65) + Tr (Ø3 $\overline{05}$ 06) A00

+ Q1 Kz $\overline{(\overline{F1}\ T0)}\left[\underset{(64)}{\overline{(\emptyset 7\ \overline{03}\ 04\ \overline{05}\ \overline{06})\ T7}}\right]$

(73) + (Tr Ø0) 02 03

(65) + Ø1 T8 $\overline{C0}$

rCz =

$$+ \emptyset 0\ T8 + \left\{\overline{Q1\ Kz\ \overline{(\overline{F1}\ T0)}\left[\overline{(\emptyset 7\ \overline{03}\ 04\ \overline{05}\ \overline{06})\ T7}\right]}\right\}$$

which reduces to Q1 $\overline{Kz}$ + $\overline{F1}$ T0 + (Ø7 $\overline{03}$ 04 $\overline{05}$ $\overline{06}$) T7

Kz = the inverse of:

$$\overline{Xz1}\ \overline{Yz1}$$

$$+ \overline{Xz2}\ \overline{Yz2}\ (Xz1 \oplus Yz1)$$

$$+ (Xz1 \oplus YZ1)\ (Xz2 \oplus YZ2)$$

$$\overline{(Xz3\ Yz3 + Xz3\ Cz + Yz3\ Cz)}$$

$$\overline{Add\ 3} = \overline{(Xz3 \oplus Yz3 \oplus Cz)}$$

$$\overline{Add\ 2} = (Xz2 \oplus Yz2)(Xz3\ Yz3 + Xz3\ Cz + Yz3\ Cz)$$
$$+ \overline{(Xz2 \oplus Yz2)}\ \overline{(Xz3\ Yz3 + Xz3\ Cz + Yz3\ Cz)}$$

$$\overline{Add\ 1} = (Xz1 \oplus Yz1)(Xz2 \oplus Yz2)$$
$$(Xz3\ Yz3 + Xz3\ Cz + Yz3\ Cz)$$
$$+ (Xz1 \oplus Yz1)\ Xz2\ Yz2$$
$$+ \overline{(Xz1 \oplus Yz1)}(Xz2 \oplus Yz2)$$
$$(\overline{Xz3}\ \overline{Yz3} + \overline{Xz3}\ \overline{Cz} + \overline{Yz3}\ \overline{Cz})$$
$$+ \overline{(Xz1 \oplus Yz1)}\ \overline{Xz2}\ \overline{Yz2}$$

3-109 Right Shift Adder

3-110 The right shift adder, which is used during multiply and divide, adds the contents of the B-register to the contents of the C-register in parallel. The sum is placed in the B-register, with each bit shifted one position to the right. For example, ignoring carry information, B2 is added to C3 and the sum is placed in B3; B1 is added to C2 and the sum is placed in B2. The sum is shifted into the A-register through A0 as it leaves the B-register through B23.

3-111 The addition of two 24-bit numbers in one pulse period is enabled by the use of an eight-bit carry register, Bc2, 5, 8, etc., through Bc23. Each carry flip-flop is associated with the least significant bit of one of the eight octals in the B-register. A partial sum is placed in the B-register, completed by carry information in the Bc register for each octal.

3-112 A diagram of the adder is shown in figure 3-12. The adder circuits directly associated with the Bc flip-flops are identified with a prime (2', 5', etc.). The Bk terms

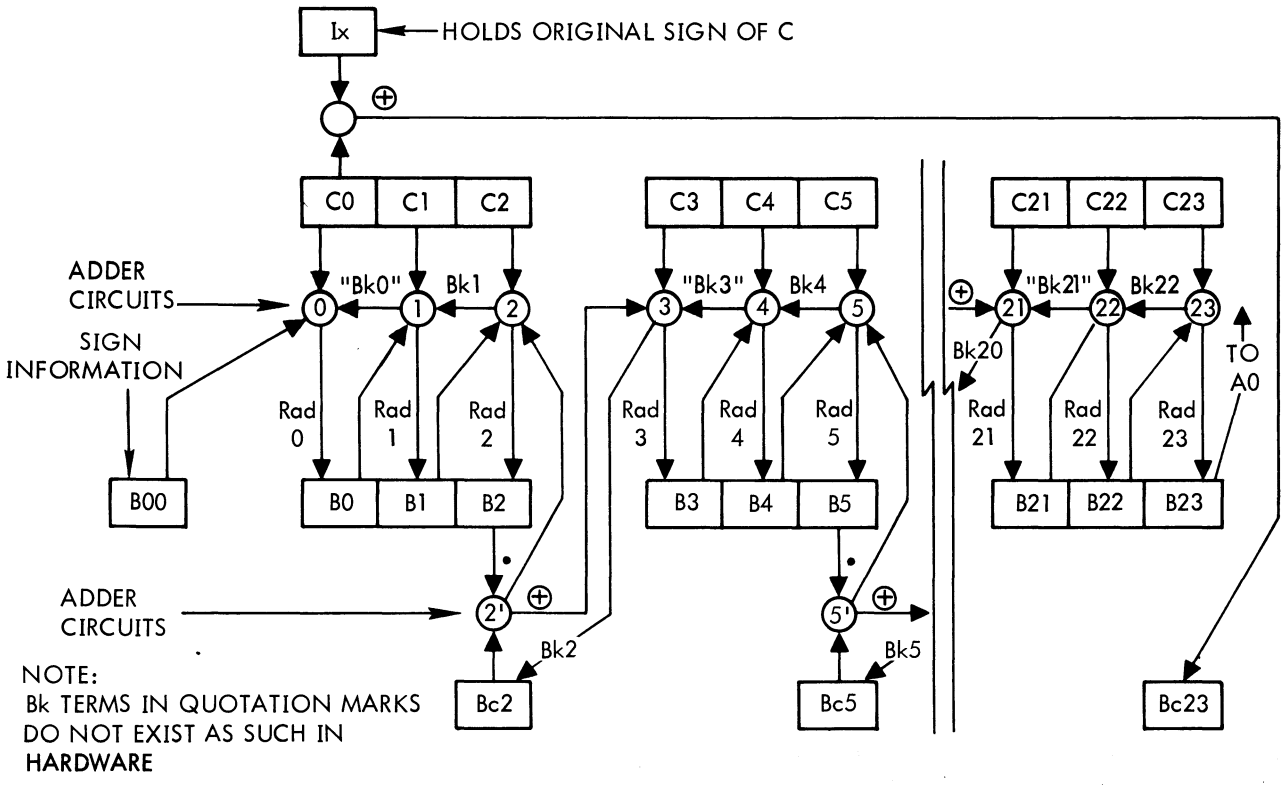


Figure 3-12. Right Shift Adder

represent a carry from one position of the adder to the next more significant bit. All Bk terms entering the most significant bit of an octal are put in quotation marks because they do not exist as such in the hardware, but are simulated by gating logic. The Rad terms represent a sum to the B-register bit, which they feed.

3-113 The transfer of the C-register bits to the adder circuitry is straightforward: the B-register bits feed the adder gates one position to their right. The transfer of B and Bc to the adder circuitry is as follows, using B2 and Bc2 as an example:

a. Expression (B2 ⊕ Bc2) represents a sum of one in the 2^{-2} position of the B-register, and is therefore shifted right with the next add operation into the 2^{-3} adder gates. (Because a binary point is assumed after B0, B1 is 2^{-1}, B2 is 2^{-2}, and B3 is 2^{-3}.)

b. Expression B2Bc2 represents two ones in the 2^{-2} position of the B-register, or a sum in the 2^{-1} position. The effective sum in the 2^{-1} position is shifted right into the 2^{-2} adder gates via the input carry term.

3-114 The pattern is repeated in each octal group with the following exceptions:

a. B00 feeds the 2^{0} adder gates with sign information. Control of B00 varies with the instruction being performed. Carry information out of the 2^{0} adder is not required.

b. Flip-flop Bc23 will always contain a zero prior to a right shift and add (or a right shift 1); therefore, the 2^{-23} adder gates contain no input carry term.

3-115 The operation performed by the adder is as follows:

$1/2\ (B + Bc) + C \rightarrow (B, Bc)$ if C is positive

$1/2\ (B + Bc) + \overline{C} + 1 \rightarrow (B, Bc)$ if C is negative

The shifting right of the B and Bc terms is equivalent to dividing (B + Bc) by two. The two's complement of a negative number is obtained by inverting the C-register flip-flops with CR to get the one's complement and setting Bc23 to add one to the sum. The setting term for Bc23 is derived by comparing C0 with Ix to determine whether the signs are opposite. At the end of multiply and divide operations, the information in the Bc flip-flops is combined with that in B0 through B23 by using the serial-octal adder.

3-116 The equations for the right shift adder are as follows:

$$sB00 = Rsa\ C0$$

$$sBn = Rsa\ Radn$$

$$rBn = Rsa\ \overline{Radn}$$

where

$$n = 0 - 23$$

$$\begin{aligned}\overline{Rad\ 0} &= Bk1\ (B0 \oplus C1)\ (B00 \oplus C0) \\ &+ B0\ C1\ (B00 \oplus C0) \\ &+ \overline{Bk1}\ (B0 \oplus C1)\ \overline{(B00 \oplus C0)} \\ &+ \overline{B0}\ \overline{C1}\ \overline{(B00 \oplus C0)}\end{aligned}$$

$$\begin{aligned}\overline{Rad\ 1} &= Bk1\ (B0 \oplus C1) + \overline{Bk1}\ \overline{B0}\ \overline{C1} \\ &+ \overline{Bk1}\ B0\ C1\end{aligned}$$

$$\begin{aligned}Rad\ 2 &= \overline{Bk1}\left[B1 + C2 + B2\ Bc2\right] \\ &+ B1\ C2\ (B2\ Bc2)\end{aligned}$$

where

$$Bk1 = B2\ Bc2\ C2 + B2\ Bc2\ B1 + B1\ C2$$

$$\begin{aligned}\overline{Rad\ 3} &= Bk4\ (B3 \oplus C4)\left[(B2 \oplus Bc2) \oplus C3\right] \\ &+ B3\ C4\left[(B2 \oplus Bc2) \oplus C3\right] \\ &+ \overline{Bk4}\ (B3 \oplus C4) \\ &\overline{\left[(B2 \oplus Bc2) \oplus C3\right]} \\ &+ \overline{B3}\ \overline{C4}\left[(B2 \oplus Bc2) \oplus C3\right]\end{aligned}$$

$$\begin{aligned}\overline{Rad\ 4} &= Bk4\ (B3 \oplus C4) \\ &+ \overline{Bk4}\ B3\ C4 \\ &+ \overline{Bk4}\ \overline{B3}\ \overline{C4}\end{aligned}$$

$$\begin{aligned}Rad\ 5 &= \overline{Bk4}\ (B4 + C5 + (B5\ Bc5)) \\ &+ B4\ C5\ (B5\ Bc5)\end{aligned}$$

where

$$Bk4 = B5\ Bc5\ C5 + B5\ Bc5\ B4 + B4\ C5$$

and

$$\begin{aligned}\overline{Bk2} &= \overline{(B2 \oplus Bc2)}\ \overline{C3} + \overline{B3}\ \overline{C4} \\ &\left[(B2 \oplus Bc2) \oplus C3\right] \\ &+ \overline{Bk4}\ (B3 \oplus C4) \\ &\left[(B2 \oplus Bc2) \oplus C3\right]\end{aligned}$$

Similarly

Rad 6, 7, 8; 9, 10, 11; 12, 13, 14; 15, 16, 17; 18, 19, 20; 21, 22, 23

Except that

$$Bk22 = B22\ C23 \text{ only}$$

and

$$\overline{Rad\ 23} = B22 \oplus C23$$

$$sBc2 = Rsa\ Bk2 + \ldots$$
$$rBc2 = Rsa\ \overline{Bk2} + \ldots$$

} Similarly for Bc 5, 8, 11, 14, 17, 20

$$sBc23 = Rsa\ (C0\ \overline{Ix} + \overline{C0}\ Ix)\ 06 + Rsa\ \emptyset 7\ Q1 + \ldots$$

3-117 SHIFT CIRCUITS

3-118 Right Shift One

3-119 Right shift one, Br1, is qualified by multiply, divide, and left or right shift. During multiply or divide, the Bc flip-flops contain meaningful data, but during shift operations the Bc flip-flops contain zero. The B anc Bc terms must be summed and the result shifted right one when Br1 is active. Flip-flop Bc23 always contains a zero prior to Br1; therefore, no adder logic is required for this flip-flop.

3-120 A simplified diagram of the right shift logic is shown in figure 3-13. Flip-flop K0 is used to store information from a left shift two during divide operation. During multiply, B00 is used to extend the sign of the partial product, and during divide, B00 contains meaningful data resulting from a left shift two operation.

3-121 Information in K0 shifts right into B00 only during divide, and B00 shifts into B0 during Br1 of multiply and divide. Flip-flop B0 receives information from bit 23 of the A-register during right shift and from B00 during multiply and divide.

3-122 The summing of information during Br1 in a Bc flip-flop with the proper B-register bit is performed in the following manner, using bit position 2 as an example:

$$B2 \oplus Bc2 \Longrightarrow sB3$$

$$Bc\quad Bc2 \Longrightarrow sBc2 \text{ (inhibit rBc2)}$$

$$\overline{B2} \Longrightarrow rBc2$$

The expression B2 ⊕ Bc2 represents a signal one in bit position 2, which necessitates setting B3 during Br1. The expression B2Bc2 represents two ONE's in bit position 2, or actually a ONE in bit position 1 and a ZERO in bit position 2. These bits are effectively shifted right one position by leaving Bc2 set and resetting B3. Information in B1 is always shifted unchanged into B2. The remaining octals are summed with the corresponding Bc flip-flops in a similar manner with the exception of Bc23, which does not enter into the summing operation, because it is always false prior to a Br1.

3-123 The following equations apply to the B and Bc flip-flops during Br1:

$$sB00 = Br1\ 06\ K0 + \ldots$$
$$rB00 = Br1\ 06\ \overline{K0} + \ldots$$

$$sB0 = Br1\ (O5A23 + \overline{05}\ B00) + \ldots$$
$$rB0 = Br1\ \overline{(O5A23 + \overline{05}\ B00)} + \ldots$$

$$sBn = Br1\ Bn\text{-}1 + \ldots$$
$$rBn = Br1\ Bn\text{-}1 + \ldots$$

where

$$n = 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22$$

$$sBn = Br1\ (Bn\text{-}1 \oplus Bcn\text{-}1) + \ldots$$

$$rBn = Br1\ \overline{(Bn\text{-}1 \oplus Bcn\text{-}1)} + \ldots$$

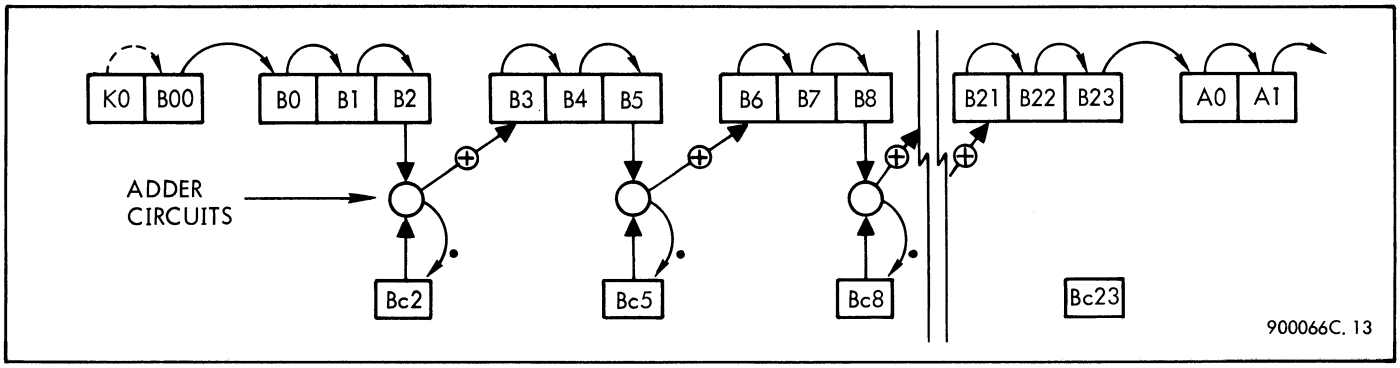


Figure 3-13. Right Shift One, Simplified

where

$n = 6, 9, 12, 15, 18, 21$

$rBcn = \overline{Br1\ Bn}$

where

$n = 2, 5, 8, 11, 14, 17, 20$

3-124 Right Shift Three

3-125 A right shift three operation, enabled by Ar3, shifts the information in the B-register three positions to the right. When the Bc flip-flops do not enter into the operation, the shift is straightforward: $B_{0-2} \longrightarrow B_{3-5}$, $B_{3-5} \longrightarrow B_{6-8}$, etc., or an end-around shift, $B_{21-23} \longrightarrow B_{0-2}$. The end around shift occurs if Bnr (B not recirculated) is false. If Bnr is true, new data is loaded into the B-register through B_{0-2}.

3-126 During phase 7 of multiply and divide, it is necessary to add the information in the Bc register to the information in the B-register while a right shift three is being performed. The operations are shown below:

Case 1. Multiply $B + Bc \longrightarrow A$

Case 2. Divide (positive numerator) $B + Bc \longrightarrow B$

Case 3. Divide (negative numerator) $-(B + Bc) = (\overline{B} + 1) + (\overline{Bc} + 1) \longrightarrow B$

The full adder, Add_{1-3}, is used to perform these operations. The outputs of the B-register are presented to the Xz inputs of the adder. For cases 1 and 2, the contents of B_{21-23} are presented to Xz_{1-3}, and for case 3, $\overline{B}_{21-23} \longrightarrow Xz_{1-3}$. Information in the Bc flip-flops is presented to the Yz inputs of the adder. Since only one Bc flip-flop is associated with each octal of the B-register, Rf and K0 are used to simulate two additional flip-flops to make the Bc input to the adder appear as a three-bit octal. The Yz_{1-3} inputs are actually Rf, K0, and Bc23.

3-127 In cases 1 and 2, where the information in Bc is added unchanged to B, Rf and K0 are zero. Flip-flop Bc23 is set or reset according to the status of Bc20, as indicated in figure 3-14.

In case 3, when it is necessary to present the two's complement of Bc to the adder, Rf and K0 are used. The two's complement of a number can be obtained by passing everything up to and including the least significant ONE and inverting all higher order bits:

n =	001	000	001	000
$\overline{n} + 1$ =	110	111	111	000

(The two's complement of n is $\overline{n} + 1$.) The set term for Bc23 is K0 $\oplus$ Bc2) and the reset term is $\overline{K0 \oplus Bc20}$. When the complment of Bc is taken, Rf and K0 remain reset as long as Bc23 contains a ZERO, and the bit in Bc20 is transferred to Bc23. If Bc23 originally contains a ONE, or if a least significant ONE is inserted in Bc23, Rf and K0 are set. This is equivalent to inverting the two most significant bits in the Bc23 octal when the least significant bit is a ONE. From this time on, the remaining data in Bc must be inverted. When Bc20 is set, a ZERO will be placed in Bc23, and when Bc20 is reset, a ONE will be placed in Bc23:

$sBc23 = K0\ \overline{Bc20}\ Ar3\ \emptyset 7\ \overline{03}\ 04\ \overline{05}\ 06$

$rBc23 = K0\ Bc20\ Ar3$

3-128 Left Shift Two

3-129 Implementation of the left shift two logic is illustrated in figure 3-15. Enabled by Aℓ2, the bits in the B-register are shifted left two positions. The Bc23 flip-flop is always unconditionally reset by Aℓ2.

3-130 Considering bit position 8 as an example, the following process takes place:

$(B8 \oplus Bc8) \Longrightarrow sB6$

$(B8\ Bc8) \Longrightarrow sBc5$

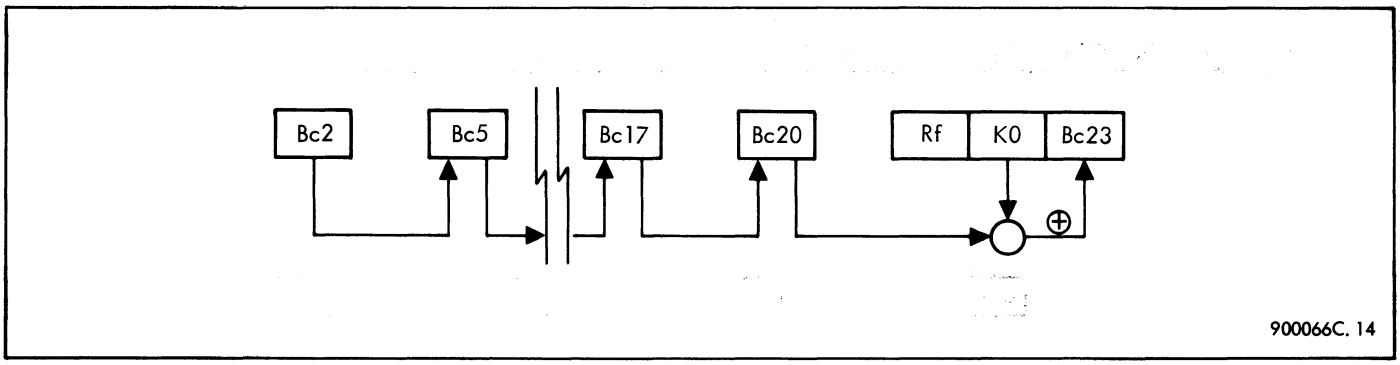


Figure 3-14. Right Shift Three, Simplified Diagram

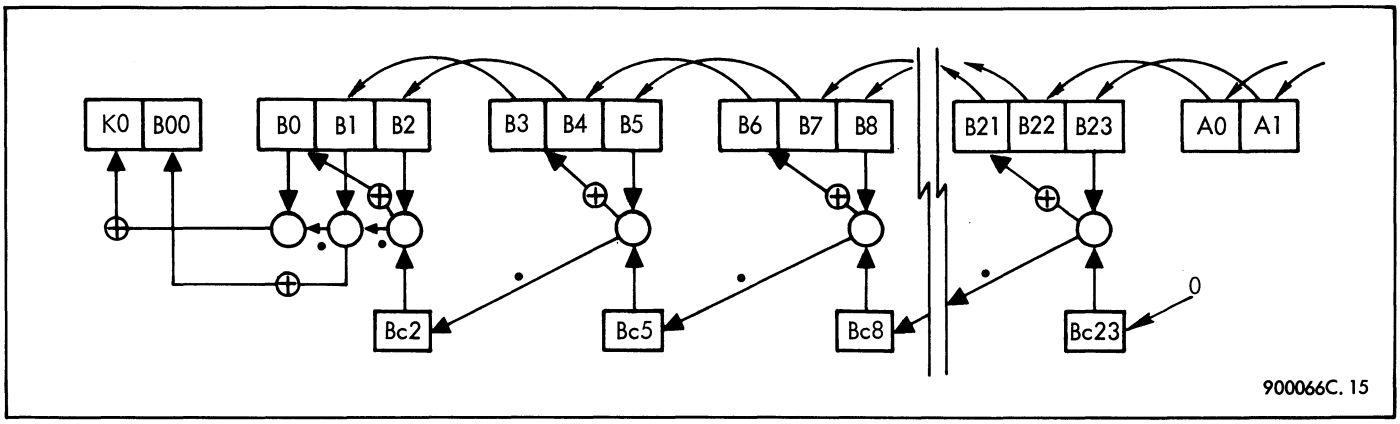


Figure 3-15. Left Shift Two, Simplified Diagram

B8 ⊕ Bc8 represents a ONE in bit position 8, and is therefore shifted left two into B6. B8Bc8 represents two ONE's in bit position 8, or actually a ONE in bit position 7, and is therefore shifted left two into Bc5. Similar logic applies to all octals of the B-register except $B_{0-2'}$ which is equipped with a half adder (see figure 3-15). The net result is that Bc2 is added to B_{0-2} and the result is shifted left two positions. Flip-flop K0 is affected only during the final iteration of divide.

3-131 Sign Detector Circuit

3-132 The sign detector circuit is used to determine the sign of B plus Bc without altering the contents of either register. This process is using during divide operation because it is necessary to complement the contents of the C-register if the sign of C is the same as the sign of B plus Bc. The operation of the sign detector circuit is illustrated in figure 3-16.

Term Bz0 is the output of the sign detector circuit and represents a carry from the sum of the Bc register and bits 1 through 23 of the B-register. If Bz0 is a ONE, a carry into B0 is implied.

3-133 The carry from the sum of B and Bc is B0 ⊕ Bz0. If B0 ⊕ Bz0 = C0, the sign of C is equal to the sign of B plus Bc. The expression B0 ⊕ Bz0 ⊕ C0 implies unlike signs; therefore, $\overline{B0 \oplus Bz0 \oplus C0}$ implies that the sign of C and the sign of B plus Bc are the same.

3-134 MEMORY ACCESS

3-135 Memory accesses may be divided into three types: central processor, interlaced, indirect and interlaced direct access. In central processor access, data is transferred from the C-register in the central processor to the M-register in the memory or from the M-register to the C-register. Interlace indirect operation involves the use of the central processor memory access capability by the input/output channel, and other computer operations are stopped during this access. The memory address is provided by the input/output channel, and data in the C-register is temporarily stored in the input/output channel register while the interlace access is taking place. In interlaced direct access, data is transferred in parallel directly from the M-register to the direct access input/output channel and vice versa. Refer to figure 3-13 for a block diagram of registers and data flow paths.

3-136 The memory is connected to the central processor by a two-cable bus system. One cable carries data signals to and from the memory plus some of the control signals. The other cable carries 15 address signals to the memory plus the balance of the control signals. These cables provide a path for reading operands and instructions from memory and for writing into memory. When two memories are used, both memories share this path; hence only one memory can be accessed in anyone memory cycle.

3-137 The optional direct access path is known as the Multiple Access to Memory feature (MAM) and allows direct access to the memory by the Direct Access Communication Channel (DACC), Data Multiplex Channel (DMC), or Memory Interface Connection (MIC). This path also consists of a two-cable bus system carrying addresses to the memory and data to and from the memory. With two memories connected to the direct access path, only one can be accessed in any one cycle. However, memory overlap can be performed by accessing one memory from the central processor and accessing the second memory via the MAM path.

3-138 Memory cycles may be started by one of two flip-flops, Mgz and Mgs, in the central processor logic. The Mgz flip-flop is set for direct access, and the address is contained in the Jz register, which receives its information from the direct access channel. The address lines from the Jz register to memory are designated Lz_{0-14}. Flip-flop Mgs is set for a central processor or interlace access. Address information is contained in the S-register, for central processor access, or in the Ir receivers, for interlace access. The Ir receivers accept information from the address register

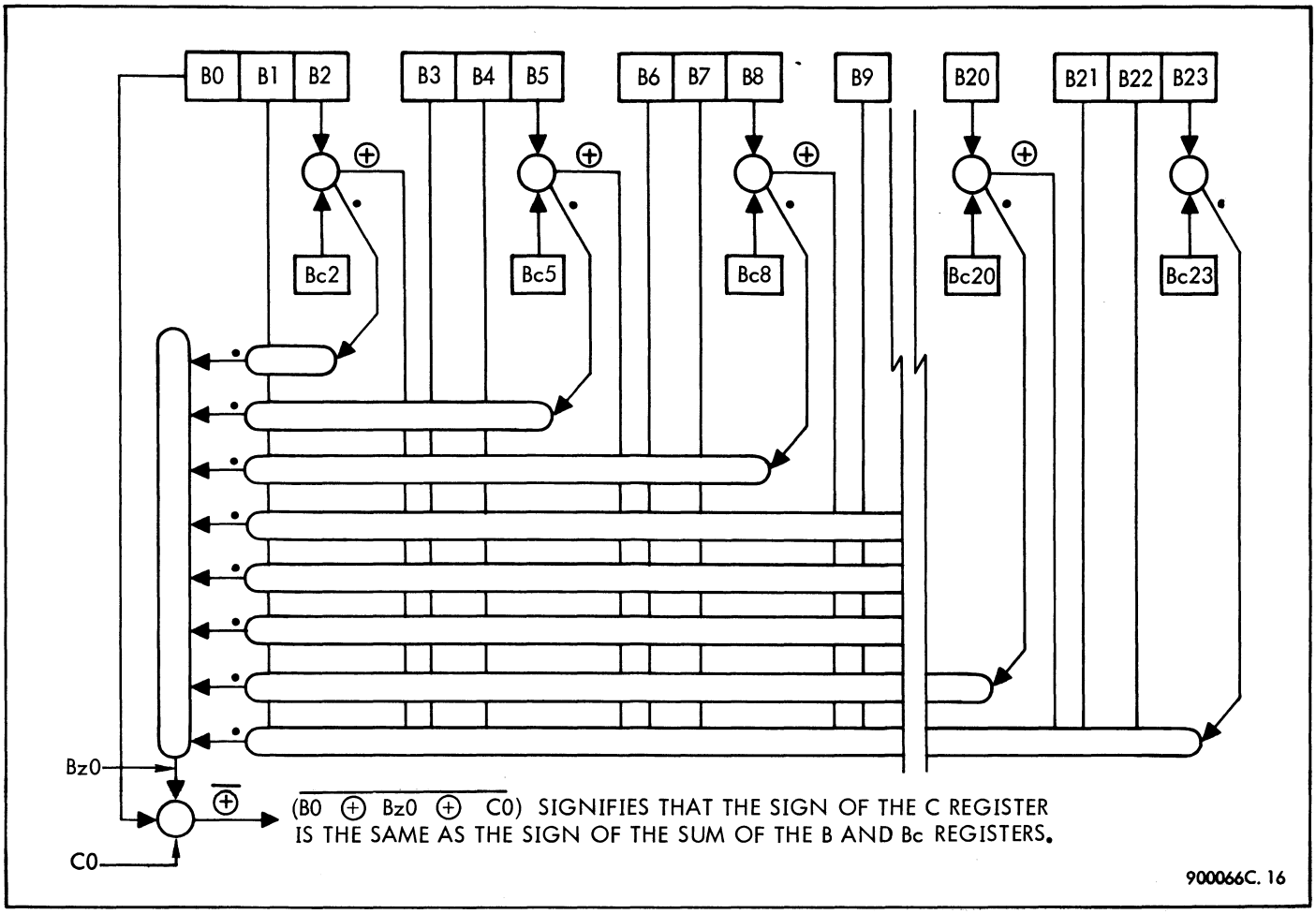


Figure 3-16. Sign Detector, Functional Diagram

in the input/output buffer. The address lines to memory for central processor or interlace access are designated Ls_{0-14}.

3-139 For a description of internal memory operation, refer to paragraphs 3-516 through 3-160.

3-140 Parity

3-141 Parity is checked or generated by the computer for a central processor access. Parity is generated by the central processor for an interlace write operation, but is not checked for an interlace read operation. During direct access input/output, parity is checked or generated by the direct access input/output channel.

3-142 Flip-flops C24, Cp, Cpi, and Cpr, associated with the C-register, are used for checking or generating memory parity or for receiving parity error signals from input/output equipment. A $\overline{Kp}$ term from the control console indicates that the HALT CONTINUE switch is in the HALT position. When Kp is false, the switch is in the CONTINUE position. A $\overline{Pio}$ signal is equal to $\overline{Kp}$ if the parity interrupt option is installed. When the parity interrupt option is not installed, $\overline{Pio}$ is always true. The parity interrupt option is described in paragraph 3-160.

3-143 Flip-flop C24 is used for generating even parity for all data that enters memory via the C-register. Parity is generated during T6-Tr, and C24 is toggled, (that is, set if previously reset and reset if previously set) every time an odd of number of ONE's are observed in C_{0-2} as Cr3 is carrying out a right shift. Since C24 begins in the reset state, it is set at the end of parity generation if an odd number of ONE's entered the C register and reset if the quantity of ONE's was even. The equations for C24 are as follows:

$$sC24 = (T6 - Tr)(C0 \oplus C1 \oplus C2)\overline{(Ts\ \overline{Tsr})}\ \overline{C24}$$

$$rC24 = T8\ \overline{(Ts\ \overline{Tsr})} + (T6 - Tr)(C0 \oplus C1 \oplus C2)\overline{(Ts\ \overline{Tsr})}\ C24$$

3-144 Flip-flop Cp is used to check the parity of data read from memory into the C-register. Enable path Cxm allows the memory parity bit M24 from memory to be copied

into Cp. The data contained in C_{21-23} is then observed for an odd number of ONE's as the C-register is shifted right three bits at a time. Every time C_{21-23} contains an odd number of ONE's, the Cp flip-flop is toggled:

$$sCp = \overline{Ts}\ \overline{Cp}\ C21 \oplus C22 \oplus C23\ \overline{Ht}\ Q1$$
$$(\overline{F1}\ \overline{F2} + \emptyset 7\ \overline{03}\ 04\ \overline{05}\ \overline{06}) + \ldots$$

$$rCp = \overline{Ts}\ Cp\ C21 + C22 + C23\ Ht\ Q1$$
$$(\overline{F1}\ \overline{F2} + \emptyset 7\ \overline{03}\ 04\ \overline{05}\ \overline{06}) + \ldots$$

If Cp is set at the end of the parity check, the data is erroneous. Enable Cr3 is active during Q1, which is true from T7 to T0. The parity checking circuitry is also active during Q1. This allows eight pulse periods for checking the eight octals in the data word.

3-145 Flip-flop Cpi, which is used only when the system contains the parity interrupt option, receives parity error signal Rmp from input/output and transmits an interrupt request to the basic interrupt chassis:

$$sCpi = Rmp\ (\overline{T4} + \overline{T5})\ (Kp)$$

Cpi is transmitted to the interrupt chassis as an I/O parity signal. The input/output parity error signals results when an input/output buffer has been made a direct memory access and the data received contains an odd number of ONE's. When no parity interrupt option is included in the system, flip-flop Cp receives the input/output parity error signal.

3-146 The parity interrupt option allows a distinction to be made as to whether the parity error was detected in the central processor or in the input/output buffer and generates a unique interrupt request for each case. A parity error detected in the central processor is transmitted to the interrupt chassis as Cpu Parity:

$$\text{Cpu Parity} = Cp\ Ht$$

where Ht is the halt flip-flop.

3-147 Flip-flop Cpr is used to generate parity for data that is transmitted octal-serially to the input/output buffers. Bits C_{21-23} are observed for the first octal, and bits C_{18-20} for all subsequent octals of a word. The Cpr flip-flop is set if the octal contains an odd number of ONE's and reset if the octal contains an even number of ONE's. The equations for Cpr are as follows:

$$sCpr = T8\ (C21 \oplus C22 \oplus C23) + Q1\ (C18 \oplus C19 \oplus C20)$$

$$rCpr = Tp + Q1\ \overline{(C18 \oplus C19 \oplus C20)}$$

Since the first octal is observed at T8 time and received by input/output at T7 time the parity bit is generated soon enough to be transmitted with the data. The parity bit to be transmitted with the second octal is generated during T7, and the data is received by input/output during T6 time, etc. It is necessary to generate parity in the central processor, because there is insufficient time for parity generation in the input/output logic after the character is received.

3-148 Time-Share

3-149 Time-sharing is the use of memory for more than one purpose during the same overall time interval by interspersing memory accesses for different operations. Two types of time-share take place in the computer: time-share because of interlace, and time-share because the Ls and Lz lines are requesting the same memory bank.

3-150 Interlace Time-Share. In order to execute an interlace operation, the memory must be time-shared between the central processor and an input/output channel. If the input/output channel requests time-share at the same time that the central processor requires the use of memory, the input/output channel is always given circuitry.

3-151 The C-register may contain valuable information at the time computation is stopped for interlace operation. It is necessary to preserve the contents of the C-register in order to continue computation when an interlace operation is completed. Since interlace information is transferred between memory and the buffer via the C-register, the C-register information must be temporarily moved to another location. This is done by exchanging the contents of the C-register and the contents of the input/output channel register:

$$C_{21-23} \rightarrow \text{cable drivers} \rightarrow \text{input/output channel}$$
$$/Rr_{1-3}/ \rightarrow Rr_{1-3} \rightarrow Rn_{1-3} \rightarrow C_{0-2}$$

where $/Rr_{1-3}/$ is an octal of data transmitted by the input/output channel, Rr_{1-3} are receivers in the central processor, and Rn_{1-3} is an octal register in the central processor from which the data is transferred to the C-register. The contents of the C-register are presented an octal digit at a time to the cable drivers and transmitted directly to the input/output channel. Both the C-register and the input/output channel register are shifting right an octal digit at a time to accommodate the shifting out of old data and the shifting in of new data. If an interlaced write operation is to be performed, the new data in C is ready to be stored in memory at the completion of the exchange. If an interlace read is required, the new data must be obtained from memory. In either case, it is necessary to perform one more exchange between C and the input/output channel register in order to restore the original contents of C and, if a read was performed, to transmit the new data to the input/output channel.

3-152 During time-share, clock is inhibited to all computer logic except that circuitry which is needed to accommodate the transfer of data between the input/output channel and the memory. The time-share control flip-flops are Ts, Tsm, and Tsr. Flip-flop Ts is set during a time-share interlace operation, and much of the computer clock is inhibited with $\overline{Ts}$. The Tsm flip-flop counts the cycles during interlace operation with multiplexed channels and switches the address lines to memory. Flip-flop Tsr is true during a time-share interlace operation with a multiplex channel. The set and reset equations for the time share control flip-flops are as follows:

$$sTs = Mit\ Tr + Trq\ Tr$$

$$rTs = \overline{Mit}\ \overline{Trq}\ \overline{Tsm}\ Tr$$

$$sTsr = \left[\overline{Mit}\ (Trq + Tsm)\right] Tr$$

$$rTsr = \left[\overline{\overline{Mit}\ (Trq + Tsm)}\right] Tr$$

$$sTsm = \overline{Tsm}\ Tsr\ T4$$

$$rTsm = Tsm\ \overline{(Ts\ \overline{Tsr})}\ T4$$

The Mit signal does not pertain to the interlace time-share operation and is assumed to be false. The Trq signal is the interlace request from the input/output channel and enables Ts to be set at Tr time. When Tsr $\overline{Tsm}$ is true, Tsm can be set at the following T4 time. The Tsm signal inhibits the S-register and gates the address in the Ir receivers onto memory address lines Ls_{0-14}. The Tsr signal enables the Rr receiver data to enter the Rn octal register.

3-153 Simultaneous Ls and Lz Access. The Ls and Lz lines may be addressing the same memory bank for one of two reasons: the direct access channel requests the use of memory at the same time as the central processor, or the direct access channel requests the use of memory during TMCC interlace operation. The Mit signal is true, indicating that the Ls and Lz addresses are actively addressing the same memory module. Since direct access always receives priority over other operations, if the direct access channel requests memory at the same time as the central processor, Ts is set and computation is inhibited until all direct access requests have been satisfied. The Tsr and Tsm signals remain false.

3-154 When Mit is true and an interlace operation is being processed, Ts, Tsr, and Tsm are all true. Since direct access receives priority, the time-share interlace operation must be halted until all direct access requests have been satisfied. Since Mit is true, Tsr can be reset, and $\overline{Tsr}$ inhibits Cr3. With Cr3 false, the transfer of data into the Rn registers is inhibited. When the direct accesses have been completed, Mit drops and $\overline{Mit}$ Tsm enables Tsr to be set. With Tsr true, interlace operation may continue.

3-155 INTERRUPTS

3-156 The interrupt system for the 930 Computer is divided into four parts:

a. Basic interrupts, included as part of the basic computer system.

b. Interrupt Control System Model 93280

c. Priority Interrupt Model 93290 (two levels)

d. Interrupt Model 92070 for memory parity and input/output parity.

The Models 93280 and 93290 are described in a separate manual entitled Models 93280/90 Interrupt Control System Technical Manual, SDS 900667. Items b, c, and d are optional.

3-157 The basic interrupts are contained in a separate interrupt chassis designated H. The Interrupts Model 92070 for memory parity and input/output parity consists of one module in location 31E of the central processor and one module in location 2SH of the basic interrupt chassis.

3-158 The interrupts available in the 930 Computer, if all options are included, are shown in table 3-7 with their priority levels. Only the I1w and I2w interrupts are included with the basic computer system.

3-159 The processing of an interrupt signal in the interrupt and main frame logic to produce an interrupt memory address is described in detail in the interrupt Control System Technical Manual, SDS 900667. Each interrupt, including memory parity and input/output parity, is handled in the same manner.

3-160 PARITY INTERRUPT OPTION

3-161 When the parity interrupt option is included in the system, a parity error causes an interrupt subroutine to be entered if the HALT switch is on. With the switch in CONTINUE, the parity error is ignored. The parity interrupt circuits are shown on pages 50, 68, and 86 of the central processor logic diagrams manual and on page 19 of the basic interrupt logic diagrams manual. A parity error signal, Rmp, from the direct access input/output channel, generates (Rmp (Kp)) if the HALT CONTINUE switch is in the HALT position. The halt flip-flop is not set because $\overline{Pio}$ is false. The Rmp (Kp) signal sets the Cpi flip-flop, which is part of the parity interrupt option. The Cpi signal is sent to the parity interrupt circuits as an I/O Parity signal. An interrupt subroutine is entered as described in the Interrupt Control System Technical Manual. If the HALT CONTINUE switch is in CONTINUE, Kp is false, Cpi is not set and an interrupt does not occur.

Table 3-7. Basic Interrupt Levels

Priority Level	Interrupt Signal	Int. Type	930 Memory Address	Description
1	Ion	Is	36	Power fail-safe: Power return
2	Ioff	Is	37	Power fail-safe: Power below safe limit
3	Cpu Parity	Ir	56	Memory parity
4	I/O Parity	Ir	57	Memory parity
5	IL	Is	35	Memory lockout
6	-	-	-	
7	Clock Sync	Is	74	Real-time clock
8	Clock Pulse	Is Ij	75	Real-time clock
9*	Ily	Ir	30	Count equals zero (End-of-Word) TMCC Y channel
10*	Ilw	Ir	31	Count equals zero (End-of-Word) TMCC W channel
11*	I2y	Ir	32	End-of-Record (End-of-Transmission) TMCC Y channel
12*	I2w	Ir	33	End-of-Record (End-of-Transmission) TMCC W channel
13	I1c	Ir	60	Count equals zero (End-of-Word) TMCC C channel
14	I2c	Ir	61	End-of-Record (End-of-Transmission) TMCC C channel
15	I1d	Ir	62	Count equals zero (End-of-Word) TMCC D channel
16	I2d	Ir	63	End-of-Record (End-of-Transmission) TMCC D channel
17	I1e	Ir	64	Count equals zero (End-of-Word) DACC E channel
18	I2e	Ir	65	End-of-Record (End-of-Transmission) DACC E channel
19	I1f	Ir	66	Count equals zero (End-of-Word) DACC F channel
20	I2f	Ir	67	End-of-Record (End-of-Transmission) DACC F channel
21	I1g	Ir	70	Count equals zero (End-of-Word) DACC G channel
22	I2g	Ir	71	End-of-Record (End-of-Transmission) DACC G channel
23	I1h	Ir	72	Count equals zero (End-of-Word) DACC H channel
24	I2h	Ir	73	End-of-Record (End-of-Transmission) DACC H channel

3-162 A memory parity error detected internally causes a Cpu parity error signal to be sent to the interrupt circuits if the HALT switch is on. Parity error signal Cp sets Ht:

$$sHt = Cp\ \overline{\emptyset 2}\ Tr\ \overline{K0}(Kp + Pio) + \ldots$$

The Cp and Ht signals generate the Cpu parity signal. An interrupt subroutine is entered as described in the Interrupt Control System Technical Manual. When an interrupt signal from the parity interrupt circuit sets the interrupt flip-flop in the central processor, Ht is reset:

$$rHt = Kg\ Int\ (\overline{Cp} + Pio)$$

where (Kg) is from the RUN switch and Pio is true when (Kp) is true. Resetting Ht permits access to the interrupt address in memory.

3-163 If the HALT CONTINUE switch is in CONTINUE, Ht is not set and interrupt signal (Cp Ht) is not generated. The parity error is therefore ignored.

3-164 CONTROL CONSOLE LOGIC

3-165 The control console is connected directly to the central processor and contains various switches for control functions and indicators for display purposes. A photograph of the control panel is shown in figure 2-1, and tables 2-1 and 2-2 describe the controls and indicators. Figure 3-17 presents a schematic diagram of the control console.

3-166 CONTROLS

3-167 I/O DISPLAY SELECT Thumbwheel Switch

3-168 This switch determines which of the input/output channels, W, Y, C, D, E, F, G, or H, is to display its unit address and error indication on the control panel. The eight-position switch generates a three-bit code on (Kcc0), (Kcc1), and (Kcc2). The code is transmitted to the input/output channels to select the proper channel.

3-169 START Pushbutton

3-170 The START pushbutton generates a (St) signal, which is routed to the control section of the computer and to peripheral equipment to reset flip-flops and clear indicators wherever necessary to prepare the system for operation. Clearing the C-register with a (St) signal in effect places a Halt instruction in the C-register. The (St) signal also resets the W-buffer, clears interrupts, and disables the interrupt system. The START pushbutton is interlocked with the IDLE switch.

3-171 The following signals are generated in the central processor by (St):

Pr3	Resets P-register
Cr3	Resets C-register

The following control flip-flops are reset in the central processor by (St):

Of	Overflow detector
Cp	Memory parity error detector
Sk	Skip flip-flop
En	Interrupt system enable
Go	Computation enable
Ix	Index flip-flop
Rf	Ready flip-flop
Cpi	Input/output parity
Int	Interrupt recognition
Inr	Interrupt interlock
E3m0	Memory extension register flip-flop
E2m0	Memory extension register flip-flop
E2m2	Memory extension register flip-flop

The following control flip-flops are set in the central processor by (St):

Ht	Inhibit computation
E3m1	Memory extension register flip-flop
E2m2	Memory extension register flip-flop
E2m1	Memory extension register flip-flop

3-172 HOLD Switch

3-173 This switch, when in the HOLD position, drives the (Kr) signal false, thereby inhibiting setting of the Ia flip-flop. Since Ia is the carry flip-flop for adding one to the program counter, holding Ia reset inhibits advancement of the program counter to the next instruction address. A branch instruction changes the program counter because the instruction transfers the new address to the P-register from the C-register.

3-174 FILL Switches

3-175 The two FILL switches enable input from four types of input/output units. Each of the four switch positions generates a control signal, (Kfm), (Kfp), (Kfd), or (Kfc), depending on whether the fill operation is to be from magnetic tape, paper tape, drum, or cards. (A disc file may be used instead of a drum.) When a fill operation is to be performed, the computer is first placed in the idle mode with the RUN IDLE STEP switch, and the START pushbutton is pressed. The (St) signal from the START pushbutton prepares the computer as described in paragraph 3-171.

3-176 The computer is next placed in the run mode. The (Kg) signal from the RUN switch enables Ht to be reset,

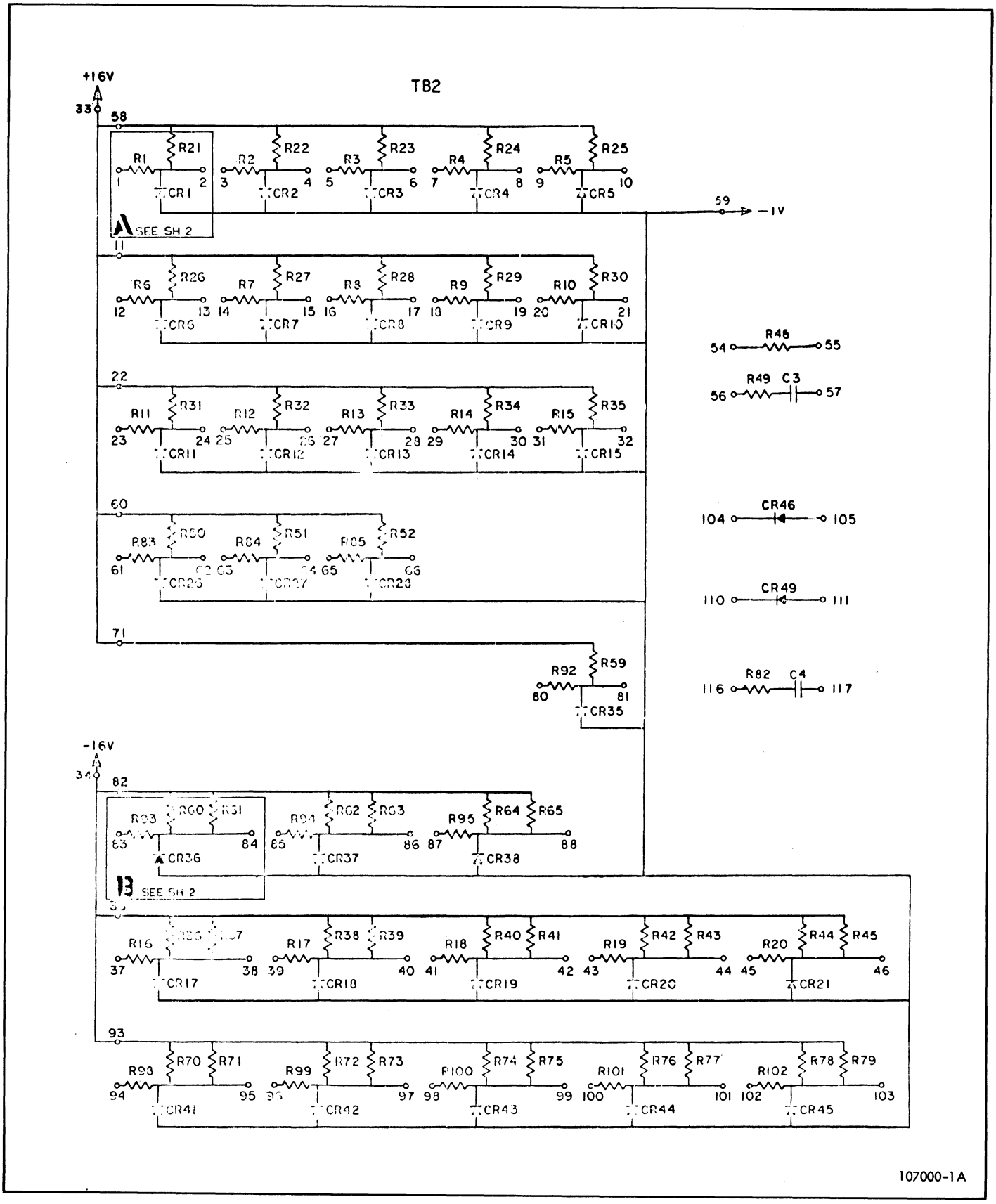


Figure 3-17. Control Console, Schematic Diagram (Sheet 1 of 4)

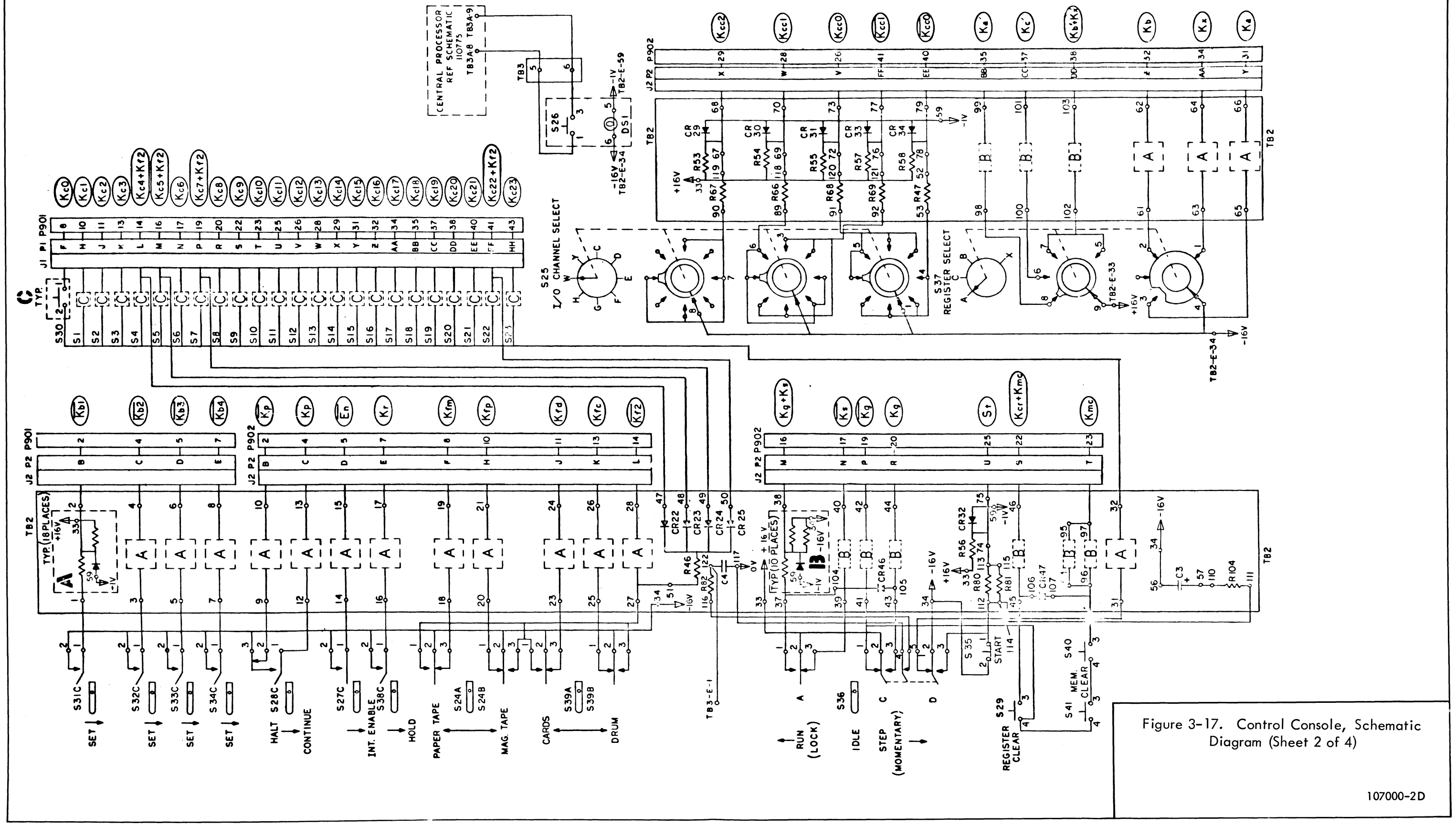


Figure 3-17. Control Console, Schematic Diagram (Sheet 2 of 4)

107000-2D

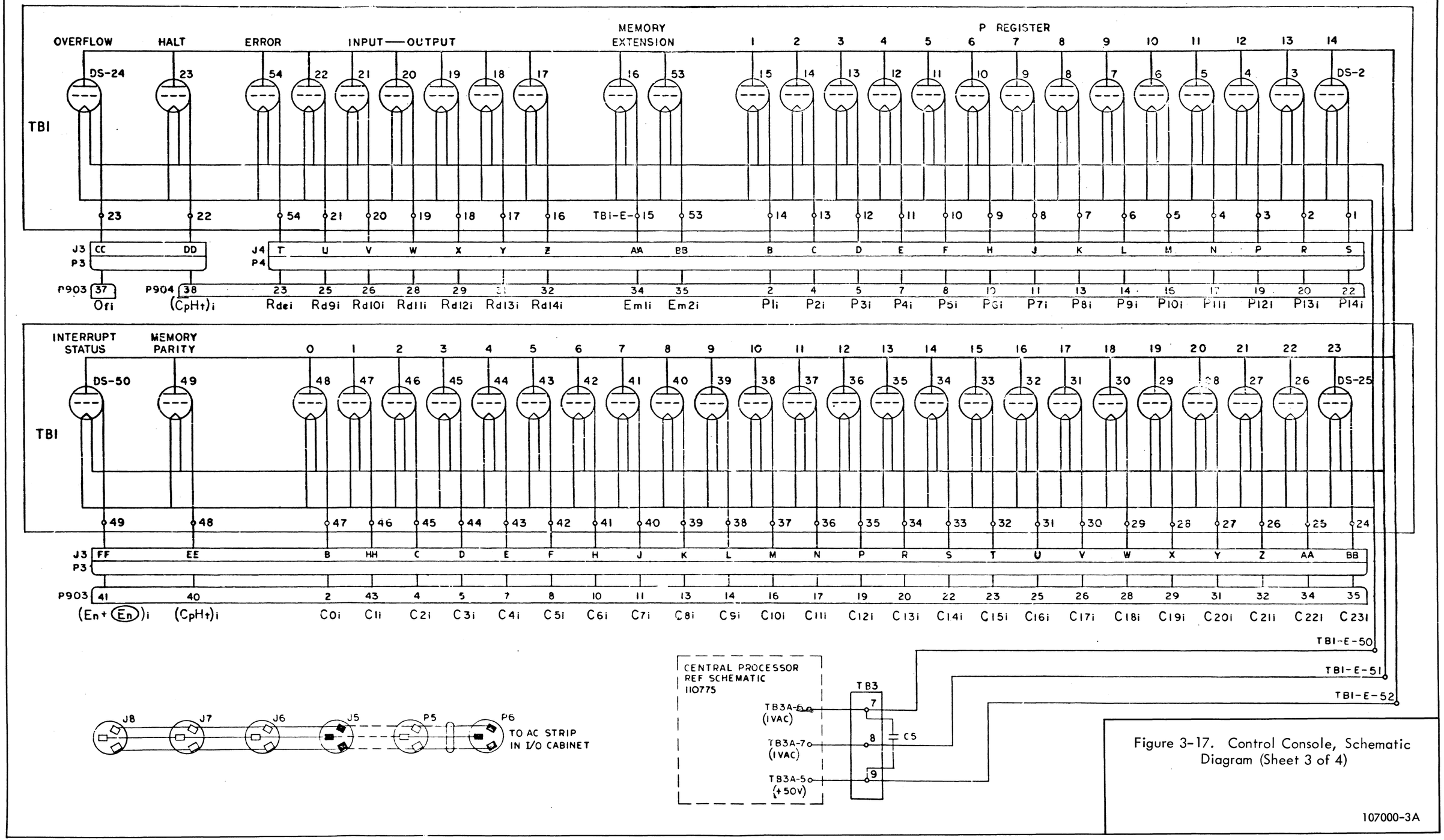


Figure 3-17. Control Console, Schematic Diagram (Sheet 3 of 4)

107000-3A

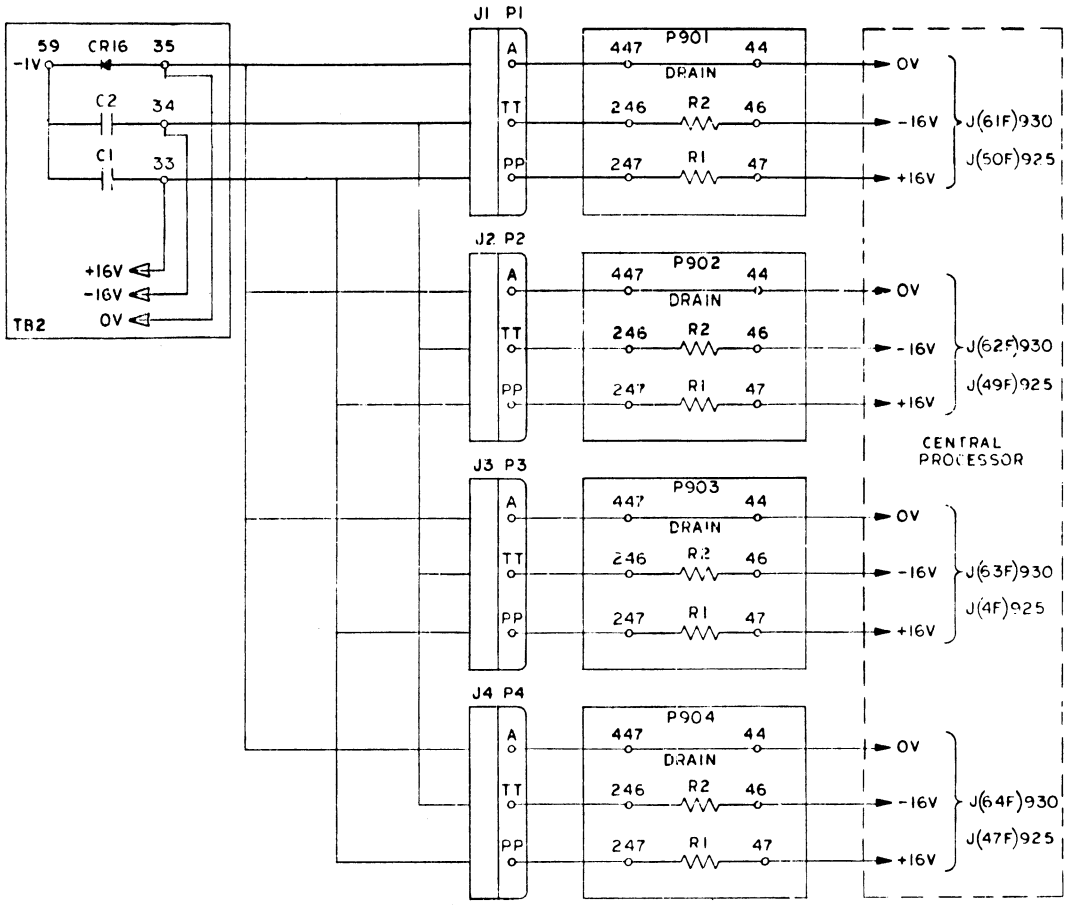


Figure 3-17. Control Console, Schematic Diagram (Sheet 4 of 4)

and $\overline{Ht}$ enables Go to be set. The computer then executes the Halt instruction in the C-register. At the end of the Halt instruction, Go is reset, and a NOP instruction is forced into the O-register as explained in paragraph 3-284.

3-177 When one of the four FILL switches is activated, a Kf1 signal is generated:

$$Kf1 = (Kfc) + (Kfd) + (Kfm) + (Kfp)$$

When Kf1 goes true, Rf is set:

$$sRf = Kf1\ Ht\ \overline{Go}\ \overline{Ix}\ Tr$$

Flip-flops E3m0 and E2m0 are set by the FILL switch signals according to the following equations:

$$sE3m0 = ((Kfp) + (Kfm))\ \overline{Go}\ Rf$$

$$sE2m0 = ((Kfc) + (Kfp))\ \overline{Go}\ Rf$$

Each of four states of E3m0 and E2m0 represents a fill medium as shown in table 3-8.

The states of E3m0 and E2m0 are coded on output lines C19 through C22 so that a six-bit address code may be presented to the peripheral equipment. Since bits 1 and 6 in each of the unit address codes is a ZERO, only bits 2, 3, 4, and 5 are necessary.

3-178 The Ix flip-flop is set at the Tr time following the setting of Rf:

$$sIx = Rf\ Ht\ \overline{Go}\ Tr$$

The unit address codes are transmitted as false signals on the cable drivers. The conditions are as follows:

$/\overline{Ci19}/$ is false if Ix $\overline{Go}$ ($\overline{E2m0}$ $\overline{E2m0}$) is true

$/\overline{Ci20}/$ if false if Ix $\overline{Go}$ (E3m0 $\overline{E2m0}$) is true

$/\overline{Ci21}/$ is false if Ix $\overline{Go}$ ($\overline{E3m0\ E2m0}$) is true

$/\overline{Ci22}/$ is false if Ix Go ($\overline{E3m0}$) is true

3-179 The FILL switch places a WIM 2 instruction, code 03200002, in the C-register. When the switch is in any one of the fill positions, a Kf2 signal is generated, and the signal parallels the manual set switch for bits 4, 5, 7, and 22 of the C-register. These set bits make the code in the C-register a WIM 2 instruction.

3-180 The index register is loaded with 77777771 according to the following equations when the FILL switch is released:

$$rRf = Kf2\ Ix$$

where Kf2 is generated by any one of the FILL switches

$$rGo = (St)$$

rHt = Ix $\overline{Go}$ $\overline{Kf1}$ Tp $\overline{Rf}$ Tem (FILL switch released). The expression Ix $\overline{Go}$ $\overline{Ht}$ drives $\overline{Xnr2}$ low, thereby gating ONE's into the three parallel inputs of the index register. The ZERO's are placed in bits 21 and 22 by qualifying the first and second index register inputs with $\overline{T7}$. The third index register stage is loaded with ONE's during every pulse period with $\overline{Xnr2}$.

3-181 The Ix $\overline{Go}$ expression indirectly generated by the FILL switch simulates an EOM instruction on the cable drivers to the peripheral equipment. The Eom signal line is energized by Ix $\overline{Go}$ $\overline{Ht}$ $\overline{Q2}$ $\overline{Q5}$ (Kg), where (Kg) means that the RUN switch is on. The Ix $\overline{Go}$ expression energizes cable driver Ci14, which specifies binary format, and Ci15 and Ci16, which specify four characters per word. The fact that bit 12 is a ZERO specifies forward direction. Cable drivers Ci19 through Ci22 specify the unit address as described in table 3-8. All information on the cable drivers is zero except that which is listed as being forced. Since Ci10 and Ci11 are false, the buffer control mode is selected by the EOM. The fact that Ci1 and Ci17 are false causes the W-buffer to be selected.

3-182 It is essential that the Eom signal be on the line for one cycle only, during Q2 + Q5, after which time the computer must be ready to accept data as soon as the buffer is ready to transmit it. After the Eom data is transmitted to input/output, the Eom must be removed from the line, and the computer must then perform a normal WIM operation.

Table 3-8. FILL Switch Control Signals

Medium	E3m0	E2m0	Implied Code Bit	Ci19	Ci20	Ci21	Ci22	Implied Code Bit	Unit Address (Octal)
Drum	0	0	(0)	1	0	1	1	(0)	26
Cards	0	1	(0)	0	0	1	1	(0)	06
Magnetic tape	1	0	(0)	0	1	0	0	(0)	10
Paper tape	1	1	(0)	0	0	1	0	(0)	04

A timing diagram of fill operation is given in figure 3-18. The heavy, solid lines are used to separate machine cycles. Parity cannot be checked for any data entered in the C-register from the control console switches, since it is impossible to set the proper status in the parity flip-flop. In this case, instruction parity is checked during the full phase 0 shown, but parity flip-flop Cp is reset by K0 at the phase 0 Tr.

3-183 REGISTER Select Switch

3-184 The REGISTER select switch makes connections which generate Ka', Kb', Kc', and Kx' signals when the A, B, C, and X registers, respectively, are selected according to the letter in the window. Signals Ka, Kb, and Kx are also generated.

3-185 The REGISTER switch selects the register to be displayed in the REGISTER DISPLAY indicators. Registers which may be displayed are A, B, C, and X. Since the contents of the C-register are always displayed, it is necessary when displaying other registers to interchange the contents of the C-register and the register to be displayed. The contents of a register are normally observed while the computer is in IDLE.

3-186 Table 3-8 illustrates the operations that take place when a register is selected for display.

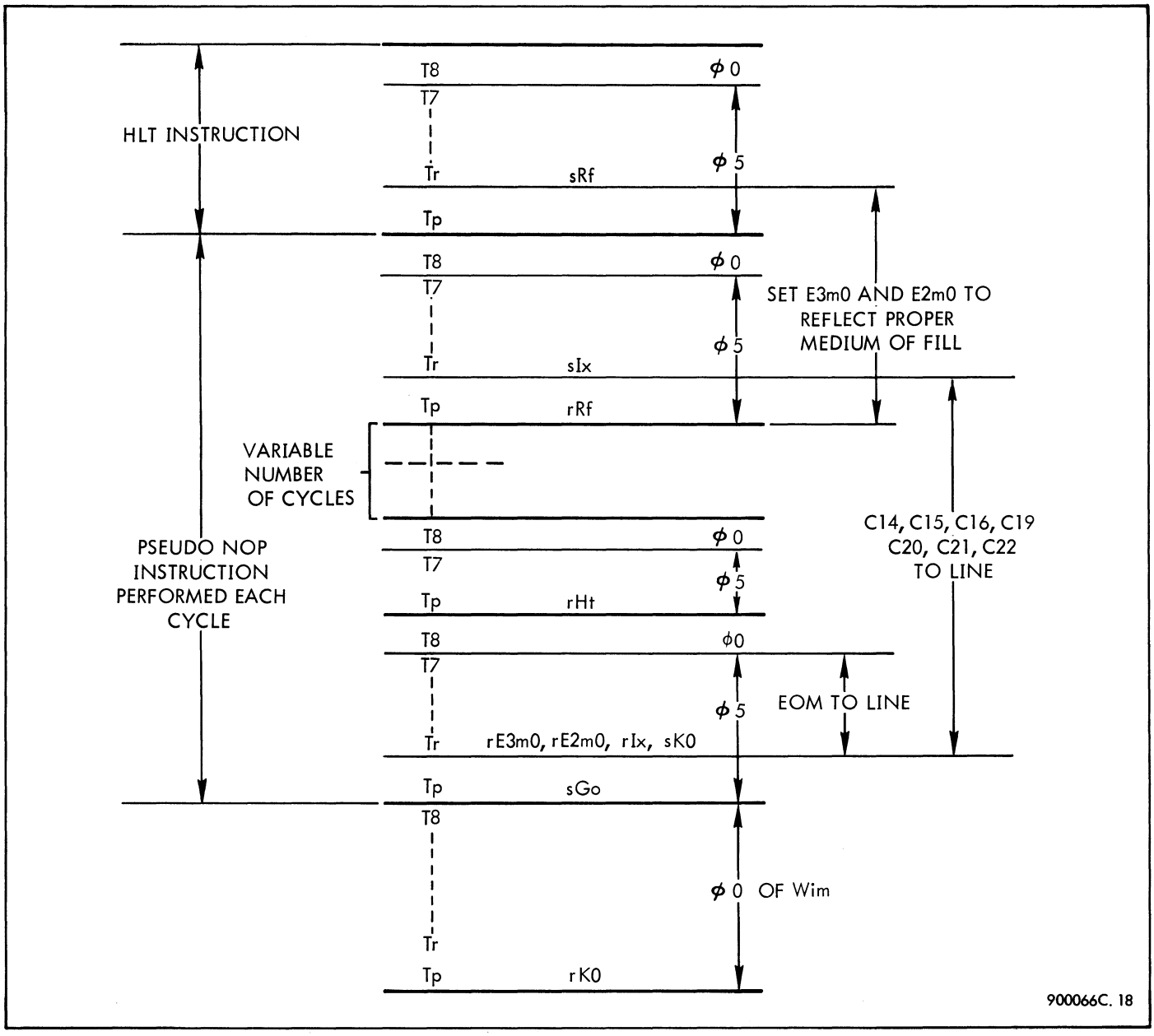


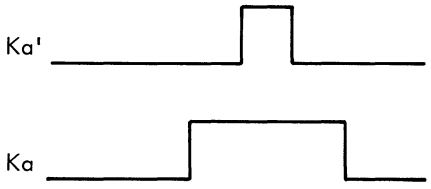
Figure 3-18. Fill Operation, Timing Diagram

Table 3-8. Register Display Sequence

Step	Register Selected	Operation
1	C	-----
2	A	A→C, C→A
3	C	A→C, C→A
4	B	B→C, C→B
5	X	B→C, C→B
		X→C, C→X
6	B	X→C, C→X
		B→C, C→B
7	C	B→C, C→B

When going from step 4 to step 5, the B-and C-registers must first be interchanged before the interchange of the X- and C-registers. The same is true of the X- and C-registers when going from step 5 to step 6.

3-187 Signals generated by the REGISTER select switch are (Ka) and (Ka') for the A-register, (Kb) and (Kb') for the B-register, (Kx) and (Kx') for the X-register, and (Kc') for the C-register. The timing relationship between the prime and non-prime signals is as follows:



3-188 The states of A00 and B00 are used to remember the status of an exchange so that the data can be returned to its original register when the select switch is returned to C. Table 3-9 shows the decoding of A00 and B00.

Table 3-9. Decoding of A00 and B00

A00	B00	Registered Displayed
0	0	C
0	1	B
1	0	A
1	1	X

The equations are as follows:

$$sA00 = Ex\ T0\ \overline{A00}\ (Ka + Kx)$$

$$sB00 = Ar3\ [Ex\ T0\ \overline{B00}\ (Kx + Kb)]$$

3-189 The exchange of data between two registers is enabled by the register exchange flip-flop Ex:

$$sEx = (\overline{Ts}\ Tp\ \overline{Go}\ \overline{Ht})\ \overline{(A00\ B00)}(Ks + Kg + Kc')$$
$$+ (\overline{Ts}\ Tp\ \overline{Go}\ \overline{Ht})\ \overline{A00}\ (Ka')\ \overline{Ks}\ \overline{Kg}$$
$$+ (\overline{Ts}\ Tp\ \overline{Go}\ \overline{Ht})\ \overline{B00}\ (Kb' + Kx')\ \overline{Ks}\ \overline{Kg}$$

where

$$(Ks + Kg)$$

is false in IDLE mode and both $\overline{Ks}$ and $\overline{Kg}$ are true in IDLE mode. The prime signals are used to set Ex in order to eliminate the problem of contact bounce. The non-prime signals are stable during the time that the prime signals may be changing, and Ex and the non-prime signals are used to enable data transfers.

3-190 It is necessary to shift the pertinent registers right three bits at a time in order to exchange data. The Cr3 signal is enabled by Ex, and the not recirculate signals, Anr, Bnr, and Xnr are enabled by Ex(Ka), Ex(Kb), and Ex(Kx), respectively. The not recirculate signals inhibit the previous contents of the register from being shifted back in. The A- and B-registers are shifted right by Ar3, which is qualified by $(01\ C2\ \overline{03}\ 04)$. Since X is a dynamic register, a right shift signal is not needed. The transfer of data from C21 through C23 into A0 through A2, B0 through B2, and X0 through X2 is qualified by Ex with the appropriate register select signal.

3-191 If a register is selected when all the registers contain their original data, A00 and B00 are both false, Ex is set, and the data in C is exchanged with the data in the selected register. If the C-register contains other than its original data when a register is selected, either A00 or B00 or both are set. Flip-flop Ex is set, and the states of A00 and B00 enable the exchange between the C-register and the register previously selected as shown in the following equations:

$$sC0 = A21\ Ex\ A00\ \overline{B00}\ Cr3$$

$$sA0 = C21\ Ex\ A00\ \overline{B00}\ Ar3$$

when C contains its original data, A00 and B00 are reset at T0, and contents of the currently selected register and the C-register are exchanged.

3-192 If the RUN IDLE STEP switch is placed in the RUN position without returning the register contents to their original position, the logic will return the register contents to their original position before executing an instruction. Flip-flop Ex is set by (Ks + Kg) from the RUN switch and the fact that A00 B00 does not contain ZERO ZERO. Exchange of data between the C-register and the register last displayed is qualified by the state of A00 and B00. The Go flip-flop cannot be set until $\overline{A00}\ \overline{B00}$ is true.

3-193 CLEAR Pushbutton

3-194 The CLEAR pushbutton generates a Kcr signal if the RUN IDLE STEP switch is not in the RUN position. The Kcr signal drives Cr3 true, and Cr3 in turn resets C0. When C0 is false, the expression Cr3 $\overline{C0}$ resets all of the other flip-flops in the C-register.

3-195 C-Register Set Pushbuttons

3-196 The C-register flip-flop set pushbuttons, labeled 0 through 23, connect to the output of an A-type control console circuit which has -16 volts on its input. The output of such a circuit is pulled to ground level. When pressed, the pushbuttons transfer this ground level through to create terms Kc0 through Kc23. These signals are applied individually to the dc inputs of the C-register flip-flops. Being dual flip-flops, these register elements set when their dc inputs go to ground.

3-197 The POWER switch makes the connection between the main power source and the power supplies. When power is on, current flows through the indicator in the POWER switch, causing a red glow.

3-198 COMPUTER ENABLE Switch

3-199 The ENABLE position of this switch enables interrupts by driving an (En) signal false, thereby allowing the interrupt flip-flop to be set when an interrupt is received:

$$sInt = \overline{(\emptyset 5\ \overline{01}\ 05\ \overline{Ts}\ Q1)\ T4\ End}$$

$$\left[(Is + Ir\ (En + \overline{(En)})\right]\ \overline{Ts}$$

The En signal is from a program interrupt enable, via an EOM. The Ts and Tr signals are received from the peripheral equipment interrupt system. An /En + $\overline{(En)}$/ signal is sent to the peripheral equipment as a result of a program or manual enable. The En + (En) signal also generates an Sks signal when bits 10 and 21 of the C-register are true and bit 11 is false. This combination of bits implies an Interrupt Enabled Test, IET, in the program.

3-200 HALT CONTINUE Switch

3-201 A (Kp) signal is generated when this switch is in the HALT position; (Kp) is false when the switch is in the CONTINUE position. These signals are used with parity error signals from the central processor or from the peripheral equipment and determine whether or not the computer will stop computation when a parity error occurs or will suppress parity errors. In the CONTINUE position the computer ignores parity errors. In the HALT position, the computer stops. If the system contains a parity interrupt option, the CONTINUE switch operates as described above. In the HALT position, the computer interrupts to the proper subroutine.

3-202 The Kp signal is used with input/output and memory parity error signals to set the parity error flip-flop, Cp, and the halt flip-flop, Ht. This causes parity errors to be recognized and computation to stop. Other circuits check input/output character parity.

3-203 MEMORY CLEAR Pushbuttons

3-204 When the two MEMORY CLEAR pushbuttons on the control panel are pressed simultaneously, a (Kmc) signal is generated, and 16,000 memory locations are cleared to ZERO.

The following signals are derived from (Kmc) :

Mxc – places ZERO's in memory location, since the C-register has previously been cleared.

Pr3 – shifts program counter (B-register) right three bits at a time and recirculates the address.

sIa – sets carry flip-flop so that contents of P-register will be increased by one each memory cycle.

Kmc + Go – adds increment to P-register.

Sxp – loads contents of P-register into address register (S) so that ZERO's may be placed in this memory location.

The Kmc signal also generates an Rℓc memory lockout clear signal to the memory.

3-205 Before operating the MEMORY CLEAR pushbuttons, the computer must be put in the idle mode with the IDLE switch. The START pushbutton must then be pressed to clear the C and the P-registers. Clearing the C-register enters a Halt instruction, and at the end of the Halt instruction a NOP instruction is automatically executed as explained in paragraph 3-284.

If a write lockout option is included, the START pushbutton should be pressed after clearing memory. The reason is given in the write lockout logic description.

3-206 The MEMORY CLEAR pushbuttons clear only 16K locations of memory without program help. In order to clear memory locations higher than 16K, memory extension registers EM3 and EM2 must be used to provide the fifteenth address bit. The extension registers may be set manually by entering Set Extension Register instructions in the C-register with the C-register set pushbuttons. Instruction code 0 06 20354 places a 5 in EM3 and a 4 in EM2 and clears locations 16384_{10} through 24575_{10} when the two MEMORY CLEAR pushbuttons are pressed simultaneously. When the S-register has counted to 20000_8, S1 is true, and the most significant octal of the address becomes 4 (refer to the equations in paragraph 3-88):

$$Ls0 = 1 \ (S1 \ \overline{S2} \ E2m0)$$

$$Ls1 = 0 \ \overline{(S1 \ \overline{S2} \ E2m1 + S1 \ S2 \ E3m1)}$$

$$Ls2 = 0 \ \overline{(\overline{S1} \ S2 + S2 \ E3m2 + S1 \ \overline{S2} \ E2m2)}$$

When the count in S reaches 30000_8, the most significant octal of the address becomes 5:

$$Ls0 = 1 \ (S1 \ S2 \ E3m0)$$

$$Ls1 = 0$$

$$Ls2 = 1 \ (S2 \ E3m2)$$

When the S-register is full, the count is 57777_8, or 24575_{10}.

3-207 Placing instruction code 0 06 20376 in the C-register enters a 7 in EM3 and a 6 in EM2, thereby clearing locations 24576_{10} through 32767_{10} when the MEMORY CLEAR pushbuttons are pressed. When the S-register has counted to 20000_8, the most significant octal digit of the address becomes 6 and the address is 24576_{10}:

$$Ls0 = 1 \ (S1 \ \overline{S2} \ E2m0)$$

$$Ls1 = 1 \ (S1 \ \overline{S2} \ E2m1)$$

$$Ls0 = 0$$

When the S-register has counted to 30000_8, the most significant octal digit of the address becomes 7:

$$Ls0 = 1 \ (S1 \ S2 \ E3m0)$$

$$Ls1 = 1 \ (S1 \ S2 \ E3m1)$$

$$Ls2 = 1 \ (S2 \ E3m2)$$

and the count proceeds to $32,767_{10}$.

3-208 BREAKPOINT Pushbuttons

3-209 The output signals from the four BREAKPOINT switches are designated $\overline{Kb1}$, $\overline{Kb2}$, $\overline{Kb3}$, and $\overline{Kb4}$. These signals, which are true when the switches are reset, are ANDed with bits 15 through 18 of the C-register to correspond with the four breakpoint test instructions;

Breakpoint 1	0	40	20400
Breakpoint 2	0	40	20200
Breakpoint 3	0	40	20100
Breakpoint 4	0	40	20040

An Sks signal is generated as follows:

$$Sks = \left(\overline{Kb1} \ C15 + \overline{Kb2} \ C16 + \overline{Kb3} \ C17 + \overline{Kb4} \ C18\right) C10 \ \overline{C11}$$

The Sks signal sets skip flip-flop Sk. When a Breakpoint Test instruction is received and the breakpoint switch specified in the instruction is reset, the computer skips the next location in sequence and executes the following instruction.

3-210 RUN IDLE STEP Switch

3-211 Four logic signals are generated by the RUN IDLE STEP switch as follows:

$\overline{Kg + Ks}$	true in STEP and RUN positions
$\overline{Ks}$	true in RUN and IDLE positions
$\overline{Kg}$	true in IDLE and STEP positions
Kg	true in RUN position

The Kg signal, true in RUN position, is used in the reset expression for halt flip-flop Ht, and $\overline{KG + Ks}$ sets the Go flip-flop. In this condition, the computer executes the program contained in memory until lit is set or the computer is halted because of operator requirements.

3-212 The IDLE position of the switch generates $\overline{Ks}$ and $\overline{Kg}$. If Go is false, $\overline{Ks}$ and $\overline{Kg}$ reset the halt flip-flop:

$$sHt = \overline{Kg} \ Go \ Tr \ End$$

$$rGo = Tp \ End \ \overline{Sk} \ Ht$$

$$rHt = \overline{Ks} \ \overline{Go} \ \overline{Cp} \ T8 \ \overline{Kg}$$

When Ht and Go are both reset, the computer is in the idle state, and the computer idles immediately after an instruction has been read from memory.

3-213 When the switch is placed in the STEP position, the computer executes one instruction and then halts. When the halt occurs, the next instruction is in the C-register. To execute a STEP operation, the computer must have previously been in the IDLE mode. To execute two instructions in the STEP mode, it is necessary to place the switch in STEP, return the switch to IDLE, and then place the switch in STEP again to execute the next instruction.

3-214 When the STEP switch is activated, the Go flip-flop is set:

$$sGo = (\overline{Kg} + \overline{Ks})\,\overline{A00}\;\overline{B00}\;Tp\;\overline{Ht}\;Tem$$

Only one instruction is executed, because during the end cycle Ht is set as shown by the sHt equation above. The switch must be returned to the IDLE position before another STEP operation, because $\overline{Ks}$ and $\overline{Kg}$ are necessary to reset Ht.

3-215 INDICATORS

3-216 OVERFLOW Indicators

3-217 The OVERFLOW indicator receives an Ofi signal, derived from the true output of overlow flip-flop Of.

3-218 HALT Indicator

3-219 HALT indicator is lit by a (Cp Ht)i signal, generated from the parity error and halt flip-flops. When the halt flip-flop has been set by a Halt instruction, the light goes on as long as no parity error is present. Placing the RUN IDLE STEP switch in the IDLE position generates $\overline{Ks}$ and $\overline{Kg}$ to reset the halt flip-flop and clear the indicator.

3-220 INPUT-OUTPUT Indicators

3-221 These indicators are lit by Rdei and Rd9i through Rd14i, derived from Rde and Rd9 through Rd14 from the selected input/output channel. The Rde signal is an error indicator, and Rd9 through Rd14 contain the unit address code of the input/output equipment.

3-222 MEMORY EXTENSION Indicators

3-223 The two MEMORY EXTENSION indicators are lit by Em1i for the left-hand indicator and Em2 i for the right-hand indicator. These signals indicate that a memory extension register is set to address memory locations above 16K. The Em1i signal on the control console schematic diagram is actually $(\overline{E3m0}\ E3m1\ E3m2)i$ in the logic diagrams. When Em1i is true, the E3m register is set to some value other than 011. The Em2i signal on the control console schematic is actually $(\overline{E2m0}\ E2m1\ \overline{E2m2})i$ in the logic diagrams, and indicates that the E2m register is set to some value other than 010. An octal 3, 2 in E3m and E2m, respectively, indicates that the memory extension registers are not operative and that the lower 16K of memory is being addressed. The START pushbutton loads the memory extension registers with 3, 2.

3-224 PROGRAM LOCATION Indicators

3-225 The PROGRAM LOCATION indicators are lit by P1i through P14i, which are derived from the set outputs of bits 1 through 14 of the P-register. These signals indicate the memory address of the next instruction to be executed.

3-226 INTERRUPT ENABLED Indicator

3-227 The INTERRUPT ENABLED indicator is lit by an (En + Sn)i signal, indicating that the interrupt system has been enabled. The En signal is derived from the enable flip-flop, En, which is set by an Enable Interrupt instruction. The (En) signal is generated from the ENABLE position of the COMPUTER ENABLE switch on the control console.

3-228 MEMORY PARITY Indicator

3-229 The MEMORY PARITY indicator is lit by a (CpHt)i signal, indicating that a parity error has occurred and the halt flip-flop has been set.

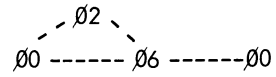
3-230 REGISTER DISPLAY Indicators

3-231 The REGISTER DISPLAY indicators are lit by C0i through C23i, derived from the set outputs of the C-register flip-flops. The indicators display the state of the register selected by the REGISTER thumbwheel switch, which transfers the contents of the selected register to the C-register.

3-232 INTERNAL OPERATIONS

3-233 INSTRUCTION SET

3-234 The instruction set illustrates the functions that can be performed by the 930 Computer. The phases that each instruction employs to perform its particular function are shown in table 3-10. The phasing is straightforward with the exception of those similar to the following illustrations:



The particular phasing chosen is that required for a MIY instruction, which performs (M) Y when the buffer is ready. The logic enters phase 2 to wait for the buffer to be ready. If the buffer is ready during phase 0, phase 2 is skipped and the phasing is effectively Ø0 – Ø6 – Ø0. If a phase is shown above the rest of the phases for a particular instruction, this phase may be bypassed if certain conditions are met.

Table 3-10. Instruction Timing

Mnemonic	Code	Phases	Description
HLT	00	Ø0 - Ø5 - Ø0	
BRU	01	Ø0 - Ø0	Branch to M
EOM	02	Ø0 - Ø5 - Ø0	
EOD	06	Ø0 - Ø5 - Ø0	
MIY	10	Ø0 - Ø2 - Ø6 - Ø0	(M)→Y when ready. Ø2 to wait for buffer ready
MIW	12	Ø0 - Ø2 - Ø6 - Ø0	(M)→W when ready. Ø2 to wait for buffer ready
POT	13	Ø0 - Ø2 - Ø6 - Ø0	Data to buffer furing Ø2. Normal End cycle in Ø6
ETR	14	Ø0 - Ø6 - Ø0	Extract. (A) (M)→A.
MRG	16	Ø0 - Ø6 - Ø0	Merge. (A) + (M)→A.
EOR	17	Ø0 - Ø6 - Ø0	Exclusive OR (A ⊕ M)→A.
NOP	20	Ø0 - Ø5 - Ø0	No operation.
YIM	30	Ø0 - Ø2 - Ø4 - Ø7 - Ø0	Y→M when buffer ready. Ø2 is waiting phase for buffer ready. Data is received during Ø4 and written in memory. Normal End cycle in Ø7.
WIM	32	Ø0 - Ø2 - Ø4 - Ø7 - Ø0	W→M when buffer ready. Ø2 is waiting phase for buffer ready. Data received during Ø4 and written in memory. Normal End cycle in Ø7.
PIN	33	Ø0 - Ø2 - Ø4 - Ø7 - Ø0	Data received during Ø2, stored in memory during Ø4. Normal End cycle in Ø7.
STA	35	Ø0 - Ø4 - Ø7 - Ø0	Store A. Store (A) during Ø4. Normal End Cycle in Ø7.
STB	36	Ø0 - Ø4 - Ø7 - Ø0	Store (B).
STX	37	Ø0 - Ø4 - Ø7 - Ø0	Store (X).
SKS	40	Ø0 - Ø5 - Ø7 - Ø0	Skip if signal not set. Phasing is 0-5-0 if a skip is not performed.
BRX	41	Ø0 - Ø6 - Ø0	Increment index and branch. If X9 = 1 after adding one to the contents of the P-register, control is transferred to the effective location. If not next instruction is taken sequence.
BRM	43	Ø0 - Ø6 - Ø0	Mark place and branch.
RCH	46	Ø0 - Ø5 - Ø0	Register change.

Table 3-10. Instruction Timing (Cont.)

Menmonic	Code	Phases	Description
SKE	50	Ø0 - Ø6 ⟨Ø7⟩ ------ Ø0	Skip if (A) = (M).
BRR	51	Ø0 - Ø6 - Ø0	Return branch. The net result is to return to (effective location) +1.
SKB	52	Ø0 - Ø6 ⟨Ø7⟩ ------ Ø0	Skip if B and M do not contain a pair of ones in any bit position.
SKN	53	Ø0 - Ø6 ⟨Ø7⟩ ------ Ø0	Skip if memory negative.
SUB	54	Ø0 - Ø6 - Ø0	(A) - (M)→A, and preserve carry out of most significant bit in X_0.
ADD	55	Ø0 - Ø6 - Ø0	(A) + (M)→A, and preserve carry out of most significant bit in X_0.
SUC	56	Ø0 - Ø6 - Ø0	(A) - (M) - Carry→A. "Carry" is that information preserved during SUB.
ADC	57	Ø0 - Ø6 - Ø0	(A) + (M) + Carry→A. "Carry" is that information preserved during ADD.
SKR	60	Ø0 - Ø4 - Ø7 - Ø0	Reduce memory, and skip if result is negative.
MIN	61	Ø0 - Ø4 - Ø7 - Ø0	(M) + 1→M. (M) are incremented and new number stored during Ø4. Normal End cycle in Ø7.
XMA	62	Ø0 - Ø4 - Ø7 - Ø0	Exchange (M) and (A). Loads contents of effective memory location in A, and stores old contents of A in effective memory location.
ADM	63	Ø0 - Ø4 - Ø7 - Ø0	Add (A) to (M) and store result in effective memory location.
MUL	64	Ø0 - Ø3 - Ø7 - Ø0	(A) x (M)→AB.
DIV	65	Ø0 - Ø1 - Ø3 - Ø7 - Ø0	$\frac{(A, B)}{M}$, result has quotient in A. Remainder in B.
RSH	66	Ø0 - Ø1 - Ø3 - Ø7 - Ø0	Right shift.
LSH	67	Ø0 - Ø1 - Ø3 - Ø7 - Ø0	Left shift.
SKM	70	Ø0 - Ø6 ⟨Ø7⟩ ------ Ø0	Skip if A = M on B mask.
LDX	71	Ø0 - Ø6 - Ø0	Load index register.
SKA	72	Ø0 - Ø6 ⟨Ø7⟩ ------ Ø0	Skip if A and M do not contain a pair of ones in any bit portion.

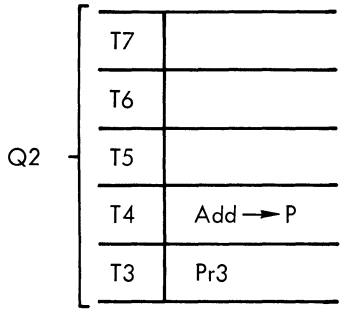
Table 3-10. Instruction Timing (Cont.)

Menmonic	Code	Phases	Description
SKG	73	Ø7 Ø0 - Ø6 ------ Ø0	Skip if A > M.
SKD	74	Ø7 Ø0 - Ø6 ------ Ø0	Skip if $M_{15-23} > B_{15-23}$.
LDB	75	Ø0 - Ø6 - Ø0	Load B-register.
LDA	76	Ø0 - Ø6 - Ø0	Load A-register.
EAX	77	Ø0 - Ø6 - Ø0	Copy effective address into index register.

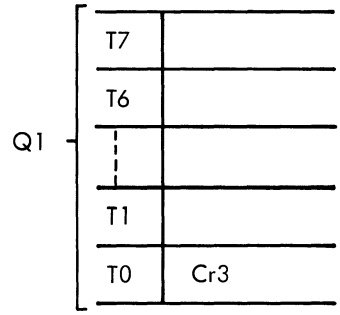
3-235 The octal designation is shown in table 3-10 for each instruction, as well as a description of the function to be performed by the instruction. A detailed explanation of each instruction is contained in the SDS 930 Computer Reference Manual, SDS 900064.

3-236 LOGIC DESCRIPTION OF OPCODES

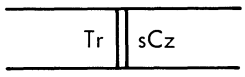
3-237 All of the computer instructions are described individually in this section, with a timing diagram for each. The duration of time during which an operation is active is indicated by a vertical line in the timing diagram. The operation to be executed is listed just to the right of the vertical line. In the following example, Pr3 is active during Q2; therefore, a vertical line is drawn to include timing pulses T7 through T3 (Q2 = T7 – T3). If more than one operation takes place during a specified period of time, both operations are shown as illustrated in the example. The illustration implies that Pr3 is active during Q2, and that during the same time interval Add_{1-3} are presented to P_{0-2}.



3-238 The following example illustrates the case when only one operation is active during a specified period of time. It is implied that Cr3 is active for the duration of Q1, so that the contents of the C-register are shifted right an octal at a time for each of the eight pulse periods.



3-239 When the execution of an operation is dependent upon certain conditions, the operation is denoted by a double vertical line as illustrated below. It is implied that Cz will be set if Tr is true, and that if Tr is not true, Cz will remain reset and the main operation will continue.



Explanation notes are included with each timing diagram.

3-240 Indexing (See figure 3-19)

3-241 Indexing takes place during phase 0 and requires no additional machine cycle time. During the end phase of the previous instruction, the indexed opcode is loaded from memory into the C-register. At Ø0 T8, the Ix flip-flop is set if the index bit (position 1) of the C-register contains a ONE:

$$sIx = Ø0\ T8\ C1\ Go + \ldots$$

Also at T8, the opcode in bits 3 through 8 of the C-register are transferred to bits 1 through 6 of the O-register. Beginning with T7, the C-register is shifted right three positions at a time, and the effective address in bit positions 10 through 23 is presented to the Yz inputs of the serial

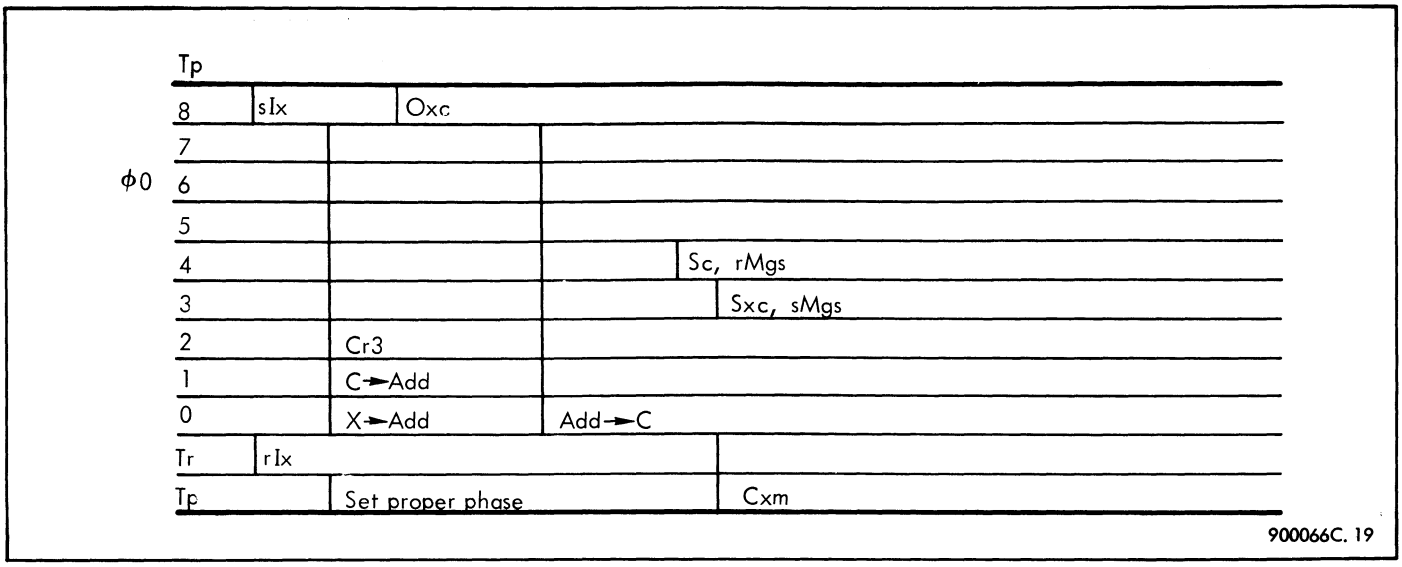


Figure 3-19. Indexing, Timing Diagram

adder an octal at a time through C_{21-23}. With each shift, the adder outputs, Add_{1-3}, are shifted into C_{0-2}. If Ix is set at T7, the index register is presented to the Xz inputs of the adder at the same time that C_{21-23} is presented to Yz. The result is that the effective address in C is increased by the number in the index register and shifted into C. At t4 time, the S-register and the Mgs flip-flop are cleared. At T3 time, C_{0-11} contain the least significant four octals digits of the indexed address, and Add_{2-3} contain the two most significant bits. An Sxc operation is executed at T3 time, transferring in parallel C_{0-11} to S_{3-14} and Add_{2-3} to S_{1-2}.

3-242 Indirect Addressing (See figure 3-20)

3-243 If the instruction read from memory during the End cycle of the previous operation contained a ONE in the indirect address bit, C9, indirect address flip-flop Ia is set at Ø0 T8.

$$sIa = Ø0\ T8\ \overline{C2}\ C9\ (\ldots) + \ldots$$

When Ia is set, the logic uses the address portion of the contents of the effective address location as the address of the next instruction or operand. This is done by using Ia or the input gates to F1, F2, and F3 so that phase 0 must be repeated and the address in the accessed location is placed in the S-register. This machine cycle is repeated until the effective location contains a ZERO in bit position 9. The ultimate effective address has been determined at this point. Flip-flop Ia is reset, and the proper phase coding is set up at Ø0 Tp.

3-244 Programmed Operator (See figure 3-21)

3-245 A programmed operator bit in C2 enables the logic to use bits 2 through 8 of the C-register as a subroutine address. Since C2 is a ONE, the location must necessarily be within the range of 100_8 through 177_8.

3-246 If C2 is set, sJu and sEax are true at Ø0 T8. Setting Eax enables the transfer of C_{3-8} to the P-register during T7 and T6; Eax is also used to set P2 at T5 time. On the timing chart, this transfer is shown as $C_{6-8} \rightarrow P_{0-2}$ at T7 and T6. At T5, P2 is the most significant bit of the address being transferred, and this is set by Eax. This transfer effectively places the programmed operator bit in the most significant position of the subroutine address in the P-register. During T7-T4, P_{3-14} are transferred to the C-register via $P_{12-14} \rightarrow C_{0-2}$. During T3, the two most significant bits of the number in P are transferred to the C-register, and Eax causes an indirect address bit to be placed in C0. During T2-T0, the status of the memory extension bits and the overflow flip-flop are transferred to C. During T6-Tr, parity is generated for the new data being transferred into C. At the end of phase 0, the C-register contains in bits 10-23 the memory location from which the programmed operator instruction was taken, and the indirect address bit in C9.

3-247 When the contents of C are transferred into memory at Ø0 Tp, the address in the S-register is 00000, because S has been cleared and has received no new information. The transfer places the last memory address, with an indirect address bit, in location 00000 ready to be called out later during the subroutine to address the operand indirectly.

3-248 A normal End cycle occurs during phase 7. A memory cycle is started, and an instruction is read from the subroutine location, which has been transferred from the P-register to the S-register. At the end of the subroutine,

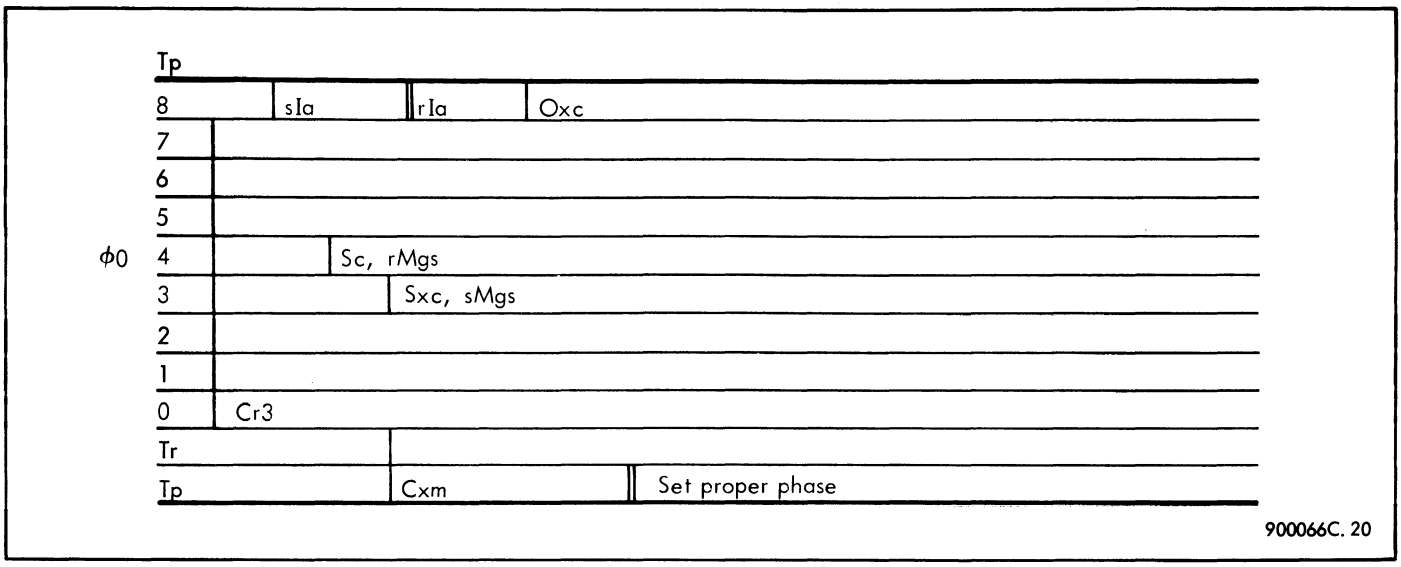


Figure 3-20. Indirect Addressing, Timing Diagram

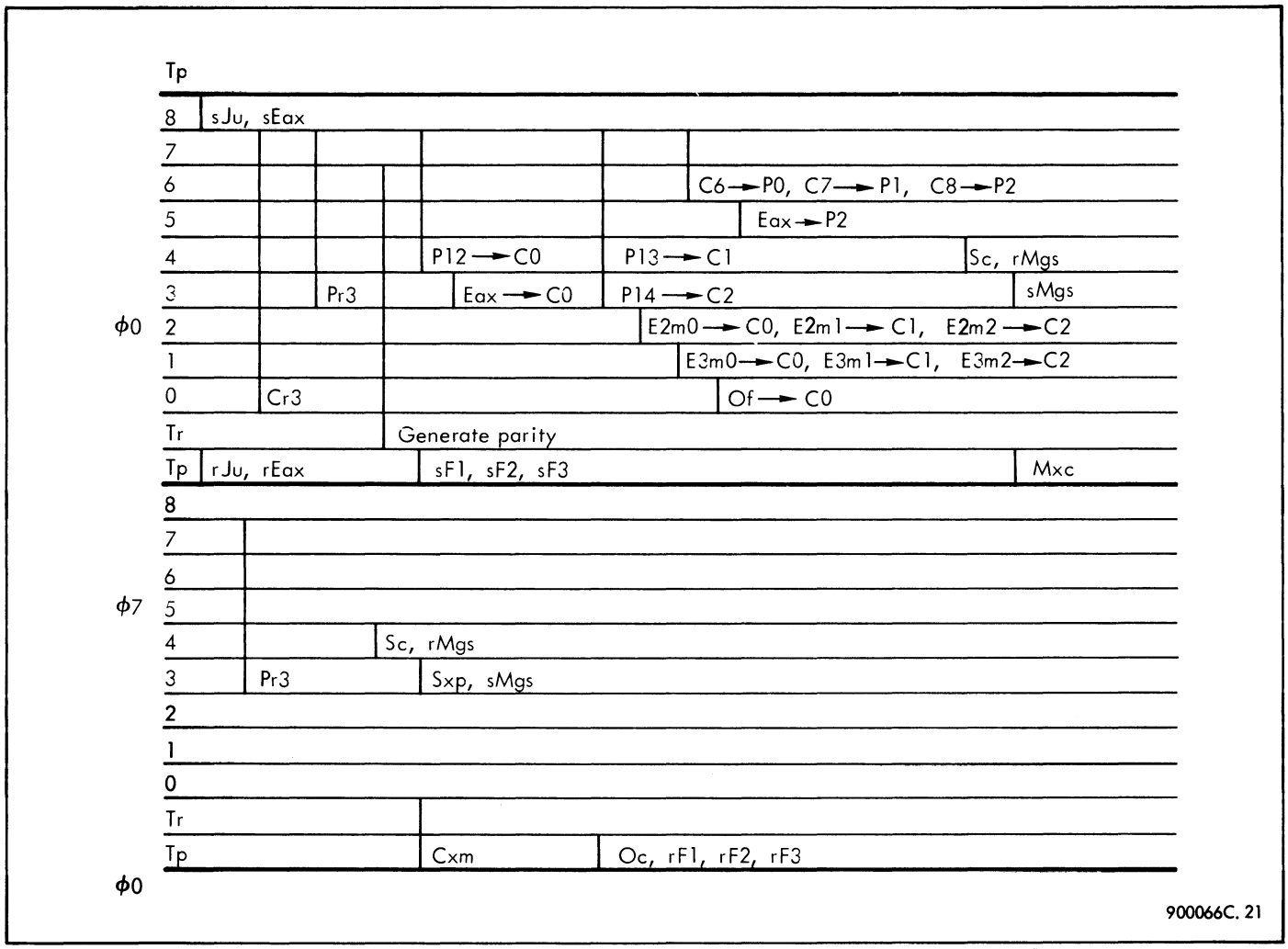


Figure 3-21. Programmed Operator, Timing Diagram

the program branches to location 00000, and the BRR allows the address in this location plus one to be used as the address of the next instruction. The indirect address bit in location 0000 is ignored this time, because the BRR used indirect addressing. Since location 00000 contains the address of the last instruction in the main program, the program is allowed to continue.

3-249 Halt (HLT, Code 00) Ø0 - Ø5 - Ø0 (See figure 3-22)

3-250 The Halt instruction is decoded from the proper C-register bits and used to change the phasing from phase 0 to phase 5 at Ø0 T8. Bits of the C-register are also decoded to set the indirect address flip-flop, Ia, which is used as a carry when adding one to the P-register during Ø5 Pr3. The Halt instruction is decoded from the O-register and used to set halt flip-flop Ht at Ø5 Tr. When Ht is true, the Go flip-flop is reset at Ø5 Tp. Computation is disabled when Go is false. At Ø5 Tp End Sk, a clear O-register signal, Oc, is true and Fl and F3 are reset to place the phase counter at phase 0.

3-251 The Oc signal clears O1 and O_{3-6} and set O2, thereby placing a pseudo-NOP (No Operation) instruction in the O-register. The computer will execute a pseudo-NOP until the Halt condition is removed. The pseudo-NOP instruction is identical to a NOP read from memory, but neither the memory location accessed nor any of the registers are altered

3-252 Parity is not checked during a Halt instruction.

3-253 Branch Unconditionally (BRU, Code 01) Ø0 - Ø0 (See figure 3-23)

3-254 A BRU instruction causes the program to access the next instruction from the effective memory location. The original instruction parity is checked during Ø0 Cr3.

3-255 The C-register bits are decoded to set Ju, the branch control flip-flop, at Ø0 T8. A true signal from Ju enables the effective address in bits 10-23 of the C-register to be transferred to the P-register via the serial adder during Ø0 Pr3. If instruction index bit C1 is a ONE, the BRU instruction is to be indexed, and index flip-flop Ix is set at Ø0 T8. A true signal from Ix gates the outputs of the index register to the input of the serial adder, and the contents of the index register are consequently added to the contents of C. The net result is that the indexed effective address is loaded in P. The contents of P are placed in the S-register, a memory cycle is initiated, and the proper instruction is read from memory.

3-256 Branch Unconditionally and Clear Interrupt (BRU I, Code 01) Ø0 - Ø0 (See figure 3-24)

3-257 The BRU I instruction is the same as a normal BRU instruction except that an indirect address bit is in C9, and the instruction is used to clear the highest active priority level of interrupt in addition to branching to the effective address. The original instruction parity is checked during Ø0 Cr3.

3-258 The effective address in the C-register is transferred to the S-register, and the first memory cycle initiated reads this address from memory. The cycle is repeated as long as the new data read out at Tp contains an indirect address bit.

3-259 When C9 is a ONE, indirect address flip-flop Ia is set at the initial T8. When data read out at Tp contains a ZERO in C9, Ia is reset at the following T8, and the final phase 0 cycle is performed.

3-260 At T8, branch control flip-flop Ju is set and is used to enable Pr3 and the effective address from C to P via Add 1 through Add 3. The Pr3 signal shifts P right 3 bits at a time until the proper number of octals have been loaded in P. The memory cycle initiated in the final Ø0 obtains the next instruction, which is transferred from M to

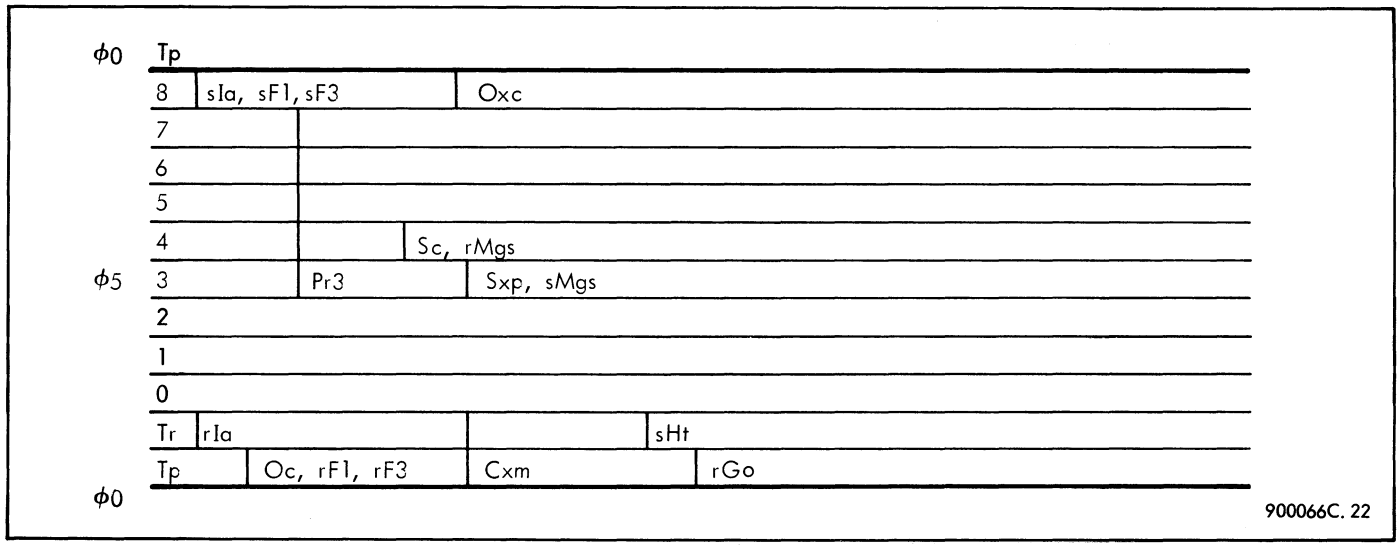


Figure 3-22. HLT Instruction, Timing Diagram

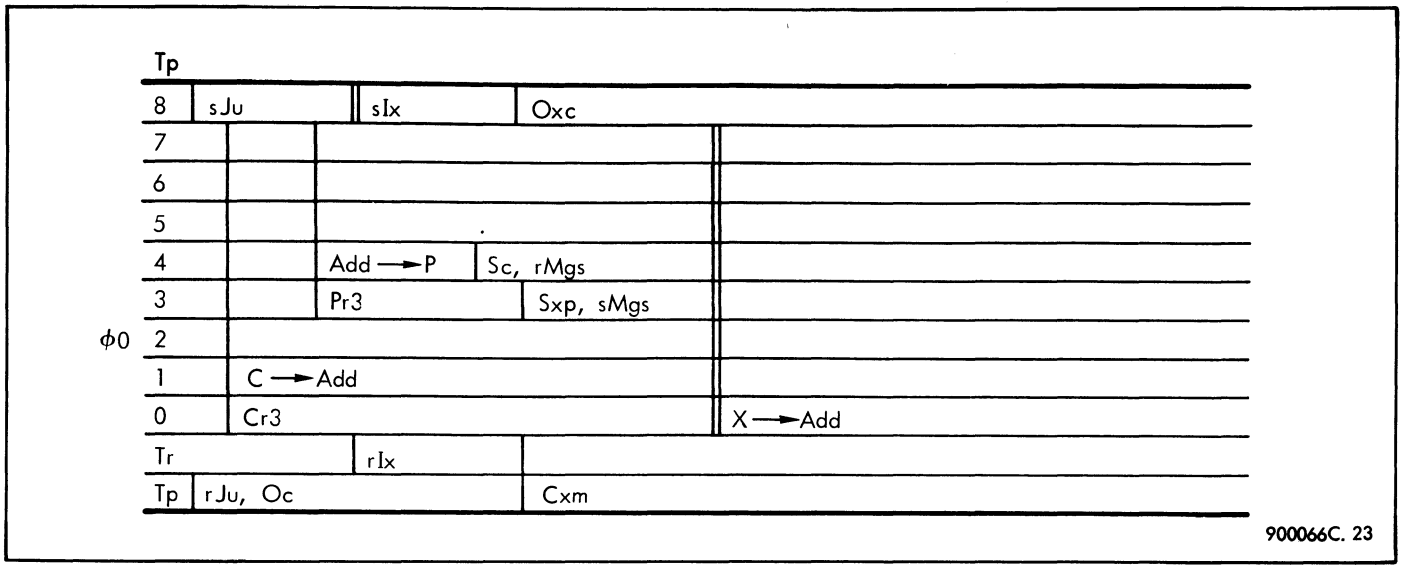


Figure 3-23. BRU Instruction, Timing Diagram

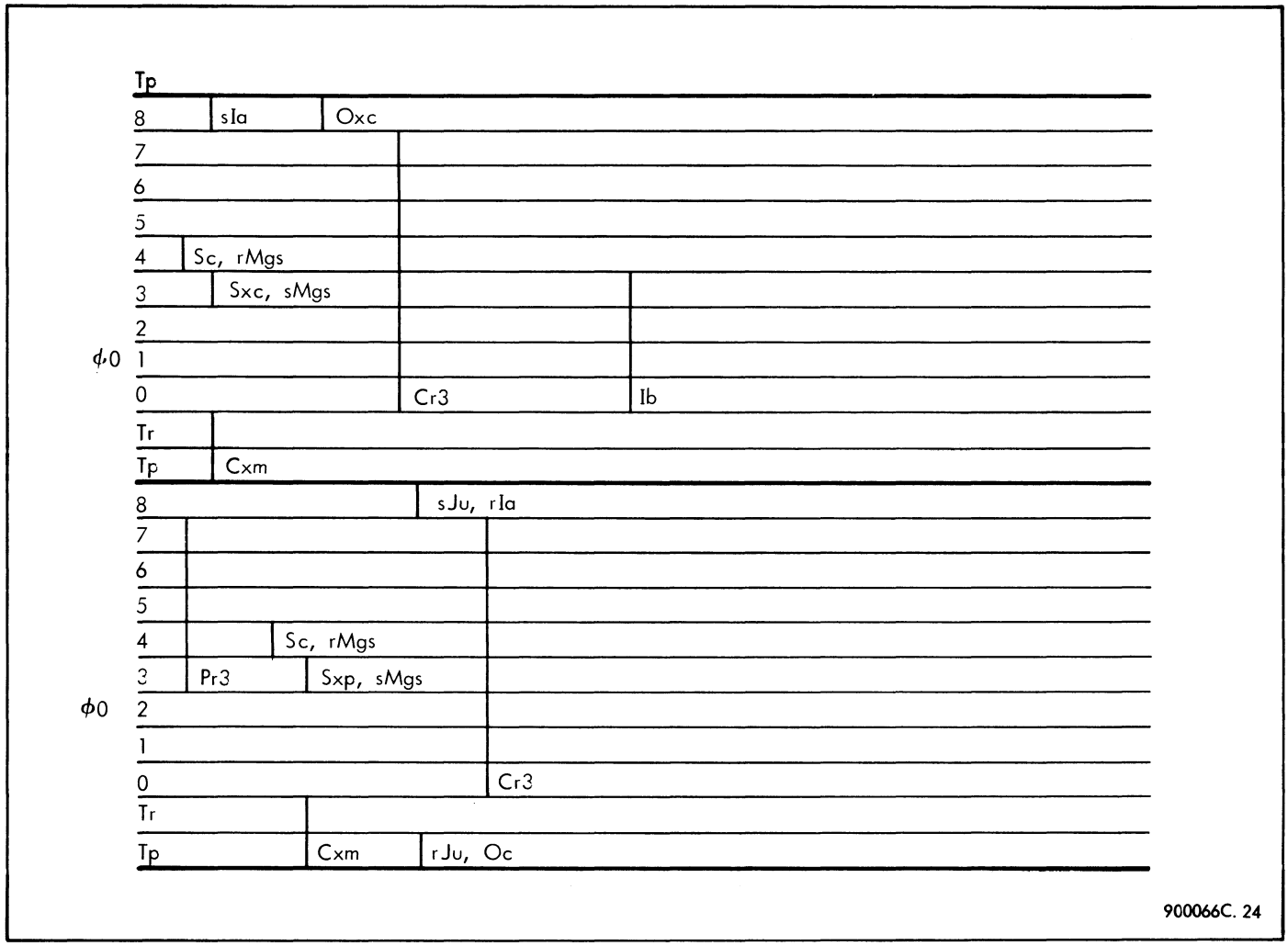


Figure 3-24. BRU I Instruction, Timing Diagram

C during Ø0 Cxm. Flip-flop Ju is cleared during the final Tp.

3-261 The BRU I instruction clears the highest active priority level of interrupt via interrupt subroutine exit signal Ib, which is Ø0 Ia $\overline{O1}$ $\overline{O2}$ $\overline{O4}$ $\overline{O5}$ O6 Ts (T3 - T0).

3-262 Energize Output M (EOM, Code 02) Ø0 - Ø5 - Ø0 (See figure 3-25)

3-263 At T8 the phasing changes from phase 0 to phase 5. The indirect address flip-flop, Ia, used as a carry in adding one to the contents of P-register, enables P + 1 → P during Ø5 Pr3 to read the next instruction from memory. During Ø5 (T7-Tp), a Cen signal is true, gating the contents of the C-register in parallel to the Ci cable drivers for transmission to the external equipment. An Eom signal is presented to a cable driver during (Ø5 (T7 - Tr). Instruction parity is not checked.

3-264 Energize Output to Direct Access Channel (EOD, Code 06) Ø0 - Ø5 - Ø0 (See figure 3-26)

3-265 The EOD instruction operates in the buffer control and input/output control modes and performs the same functions as an EOM on channels E, F, G, and H when present. As a Set Extension Register instruction, code 6 also specifies the contents of extension registers E3m and E2m for addressing memory locations 16,000 and higher.

3-266 Decoding of the opcode changes the phasing from phase 0 to phase 5 at T8. The indirect address flip-flop, Ia, used as a carry in adding one to the P-register, enables

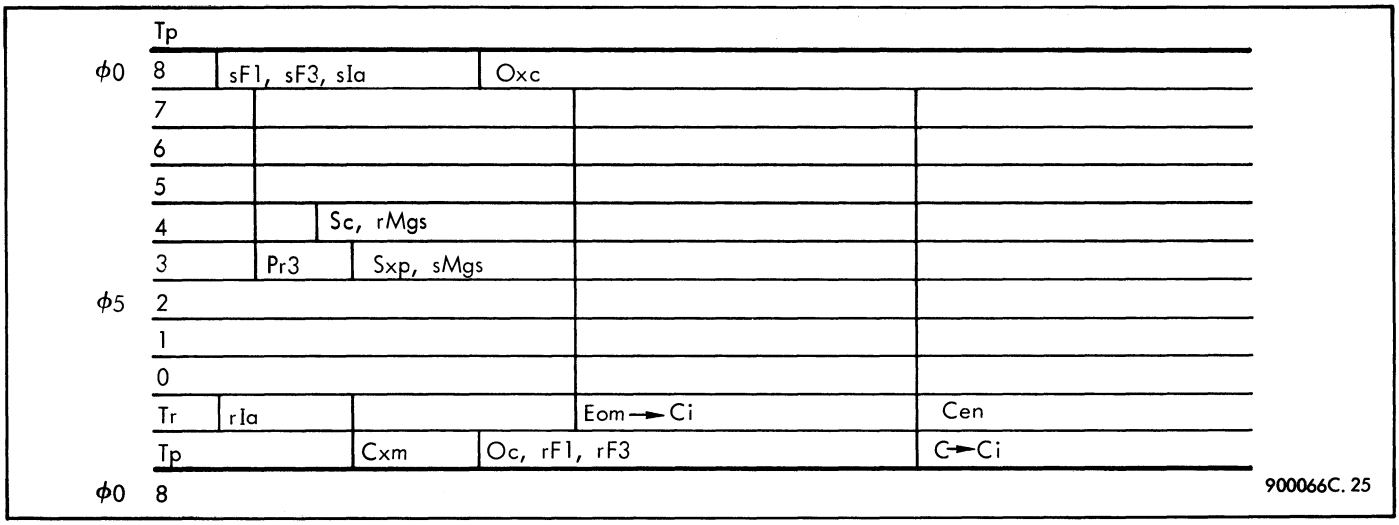


Figure 3-25. EOM Instruction, Timing Diagram

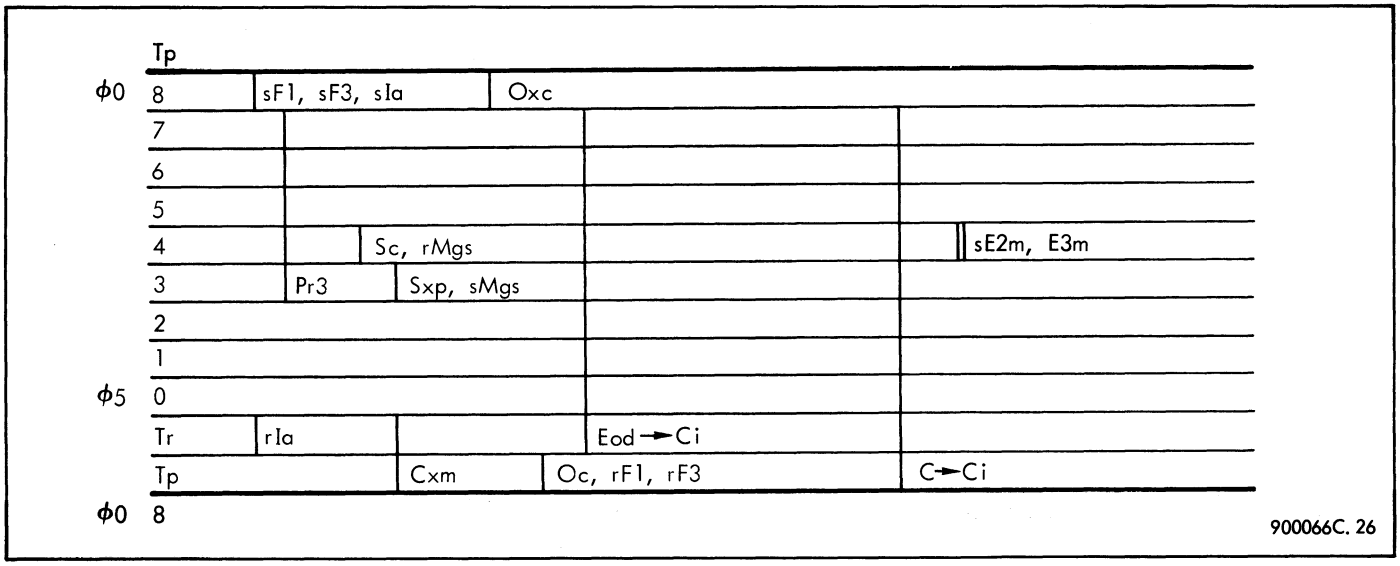


Figure 3-26. EOD Instruction, Timing Diagram

P + 1 during Ø5 Pr3 so that the next instruction may be read from memory. A Cen signal is true, and gates the contents of the C-register in parallel to the Ci cable drivers for transmission to the external equipment. An Eod signal is presented to a cable driver during Ø5 (T7-Tr). Instruction parity is not checked.

3-267 When used to set the memory extension register, the instruction contains an octal 2 in bits 9 through 11. The R3 field, bits 18 through 20, and the R2 field, bits 21 through 23, contain octal codes that provide a fifteenth address bit when the location to be addressed is between 16,384 and 32,767. If bit position 16 contains a ONE, the contents of R3 are loaded at T4 into E3m0 through E3m2. If bit position 17 contains a ONE, the contents of R2 are loaded at T4 into E2m0 through E2m2. If bits 16 and 17 are both set, E3m and E2m are loaded simultaneously. If the number in R3 is to be used as the fifteenth address bit, the next instruction must contain a ONE, ONE in bits 10 and 11. If the number in R2 is to be the fifteenth address bit, the next instruction must contain a ONE, ZERO in bits 10 and 11.

3-268 Memory Into Y Buffer When Empty (MIY, Code 10)

3-269 Refer to MIW instruction, Code 12. The MIY instruction is identical except that Yf Y0 is used in place of Wf W0, and the 05 signal presented to the cable driver as part of the Pwy signal during phase 6 is false. This drives Pwy false and selects the Y buffer.

3-270 Memory Into W Buffer When Empty (MIW, Code 12)

Ø0 ------ Ø6 (Ø2) - Ø0 (See figure 3-27)

3-271 This instruction transfers the contents of the effective memory location to the W buffer. The original instruction parity is checked during Ø0 Cr3. The operand is accessed during Phase 0, and operand parity is checked during Ø6 Cr3.

3-272 The effective address is loaded into the S-register at Ø0 T3, and the contents of this location are loaded into the C-register at Ø0 Tp. If the buffer is ready to receive data at Ø0 T0, the Rf flip-flop is set:

$$sRf = T0\ \overline{F3}\ O5\ \overline{O6}\ \left[\overline{Wf}(W0 + \overline{W9})\right] + \ldots$$

where

$$\overline{Wf}\ (W0 + \overline{W9})$$

is a ready signal from the buffer. Setting Rf changes the phasing from phase 0 to phase 6 at Tp, and phase 2 is skipped:

$$sF1 = Tp\ (\overline{F1}\ \overline{F3}\ \overline{O1}\ O3\ \overline{O4})\ \overline{Ia}\ Rf + \ldots$$

$$sF2 = (Tp\ \overline{Ia}\ Ø0)\ O3\ (O1 + \overline{O2})$$

If $\overline{Wf}$ W0 is false at Ø0 T0, phase 2 is entered because F1 is not set. The logic remains in phase 2 until the buffer is ready, at which time F1 is set at Ø2 Tp, advancing the counter to phase 6.

3-273 During Ø6 Cr3, the contents of the C-register are presented to cable drivers Ci, an octal at a time, for transmission to the W buffer. Since Ia is true during phase 6, a carry is provided to the P-register to increase the program address by one during Pr3. A memory cycle is initiated during phase 6 to place the next instruction in the C-register. Phase 6 and decoding of the opcode generates Rx, which is a signal sent to the buffer during a programmed input/output instruction. The Rx signal and O5 are presented to cable drivers Ci for transmission to the buffer.

3-274 Parallel Output (POT, Code 13) Ø0 Ø2 Ø0 (See figure 3-28)

3-275 The POT instruction presents the contents of the C-register to cable drivers for parallel output to external equipment. The original instruction parity is checked during Ø0 Cr3. The operand is read from memory during phase 0, and operand parity is checked during Ø6 Cr3.

3-276 When the external device is ready, it sends a $\overline{Rt}$ signal, which causes Rf to set sometime during Ø Q2. A ONE in Rf at Ø2 Tp sets F1, enabling the logic to pass from phase 2 to phase 6. The computer stays locked in phase 2 until the external equipment is ready. The POT signal from the central processor and a POT 2 signal generated in the TMCC are available to gate and clock the C lines into the external device. A ONE in carry flip-flop Ia enables adding one to the contents of the P-register during Ø6 Pr3. A memory cycle is started during phase 6 to read the next instruction.

3-277 Extract (ETR, Code 14) Ø0 - Ø6 - Ø0 (See figure 3-29)

3-278 The Extract instruction performs a logical AND between corresponding bits of the A-register and the effective memory location and places the result in the A-register. The original instruction parity is checked during Ø0 Cr3. The operand is read from memory during phase 0, and operand parity is checked during Ø6 Cr3.

3-279 The extract operation is performed during Ø6 Ar3 Cr3 as follows:

A21 C21 → A0, A22 C22 → A1, A23 C23 → A2

At Ø6 T7 the carry bit in Ia enables adding one to the contents of the P-register during Ø6 Pr3. A memory cycle is started during phase 6 to place the next instruction in the C-register.

3-280 Merge (MRG, Code 16) Ø0 - Ø6 - Ø0 (See figure 3-30)

3-281 The Merge instruction performs a logical inclusive OR between corresponding bits of the A-register and the

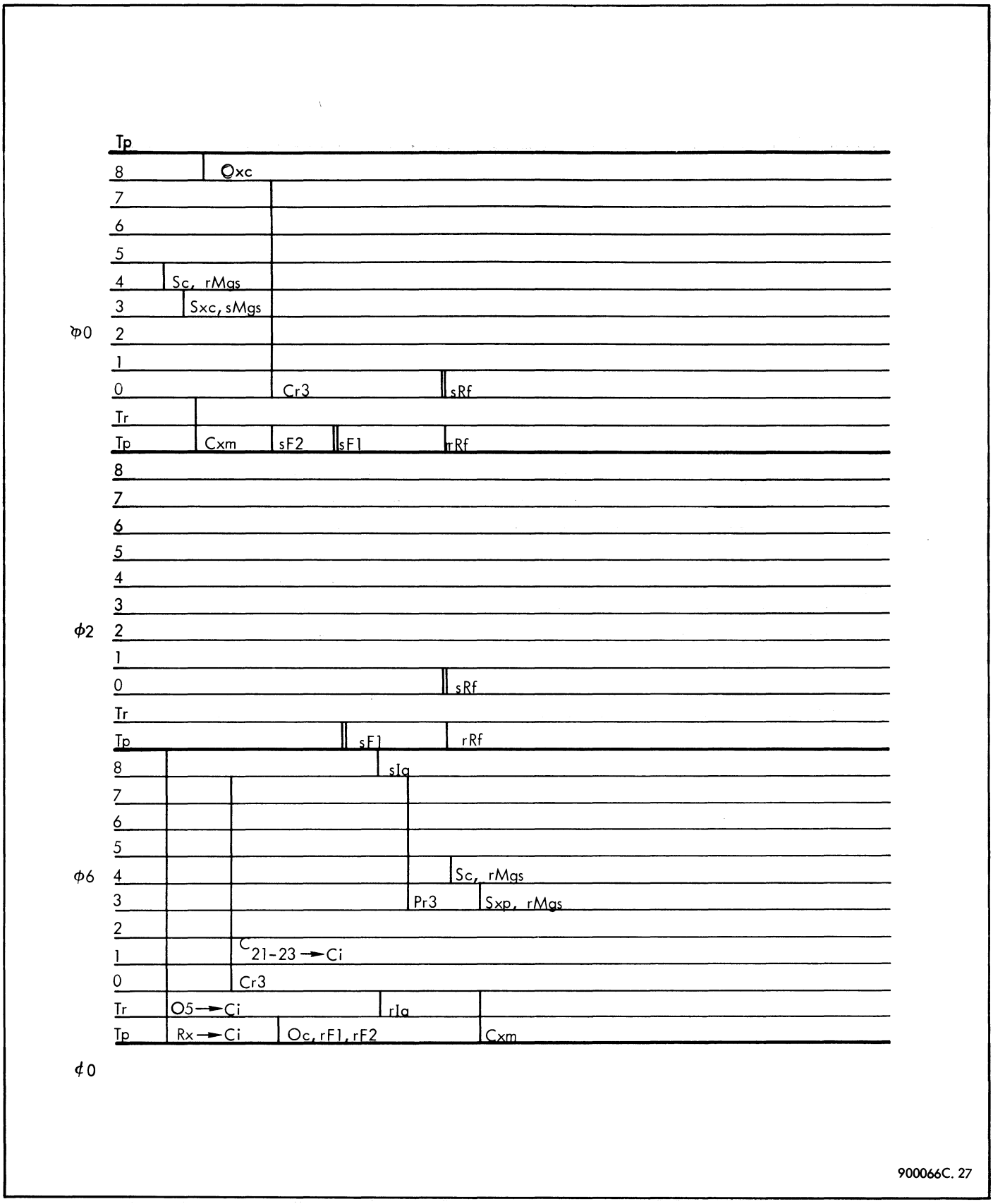


Figure 3-27. MIW Instruction, Timing Diagram

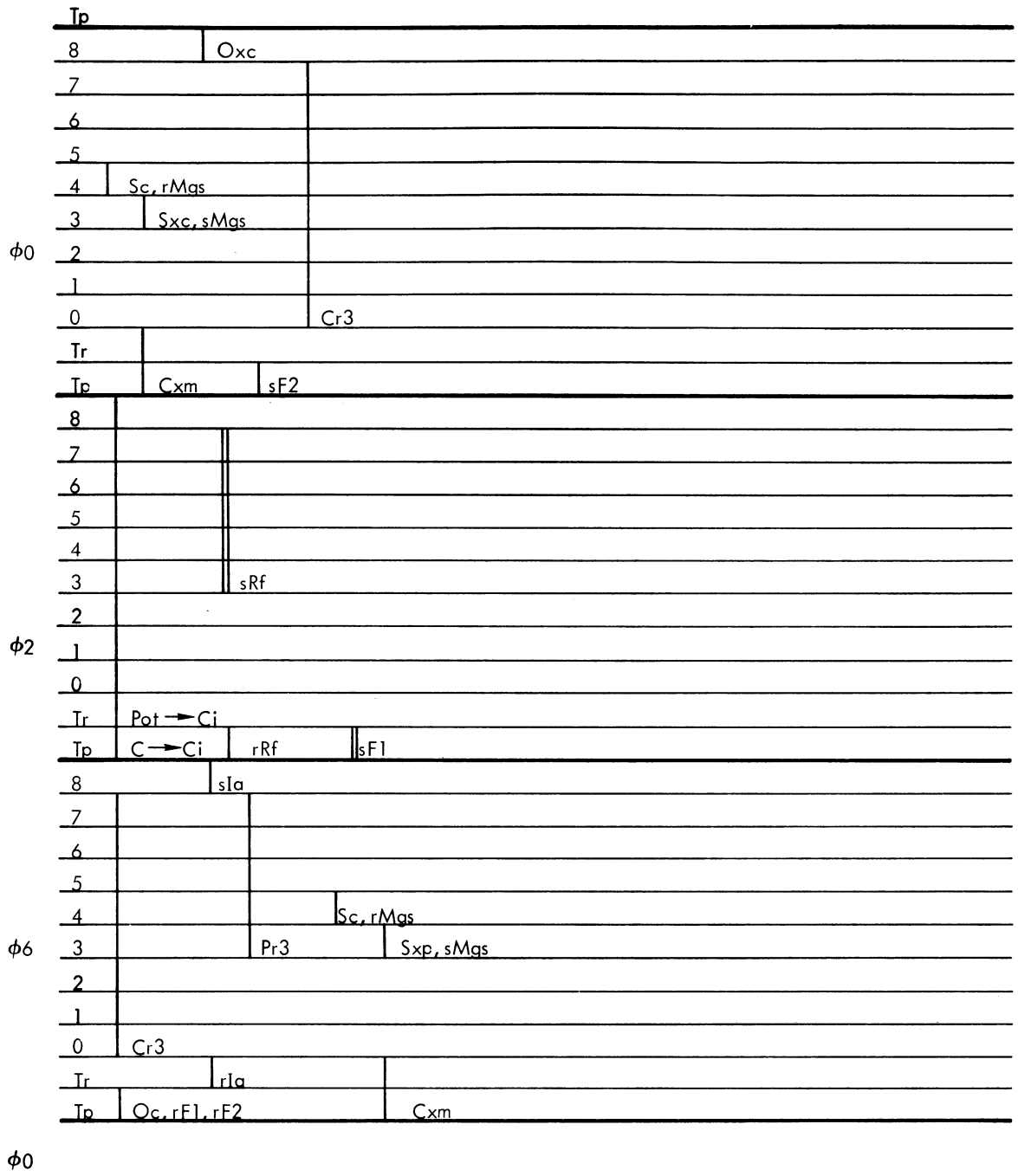


Figure 3-28. POT Instruction, Timing Diagram

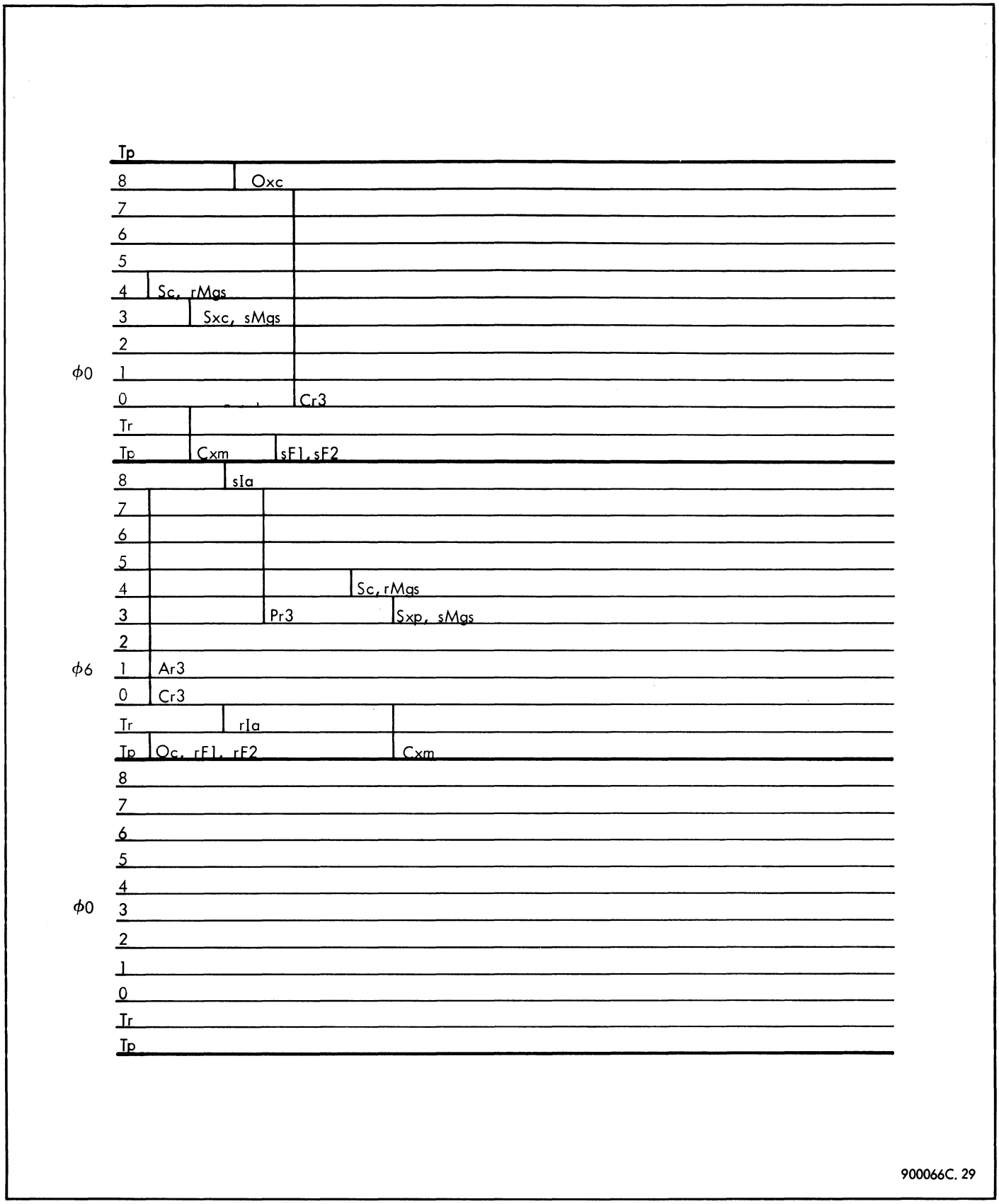


Figure 3-29. ETR Instruction, Timing Diagram

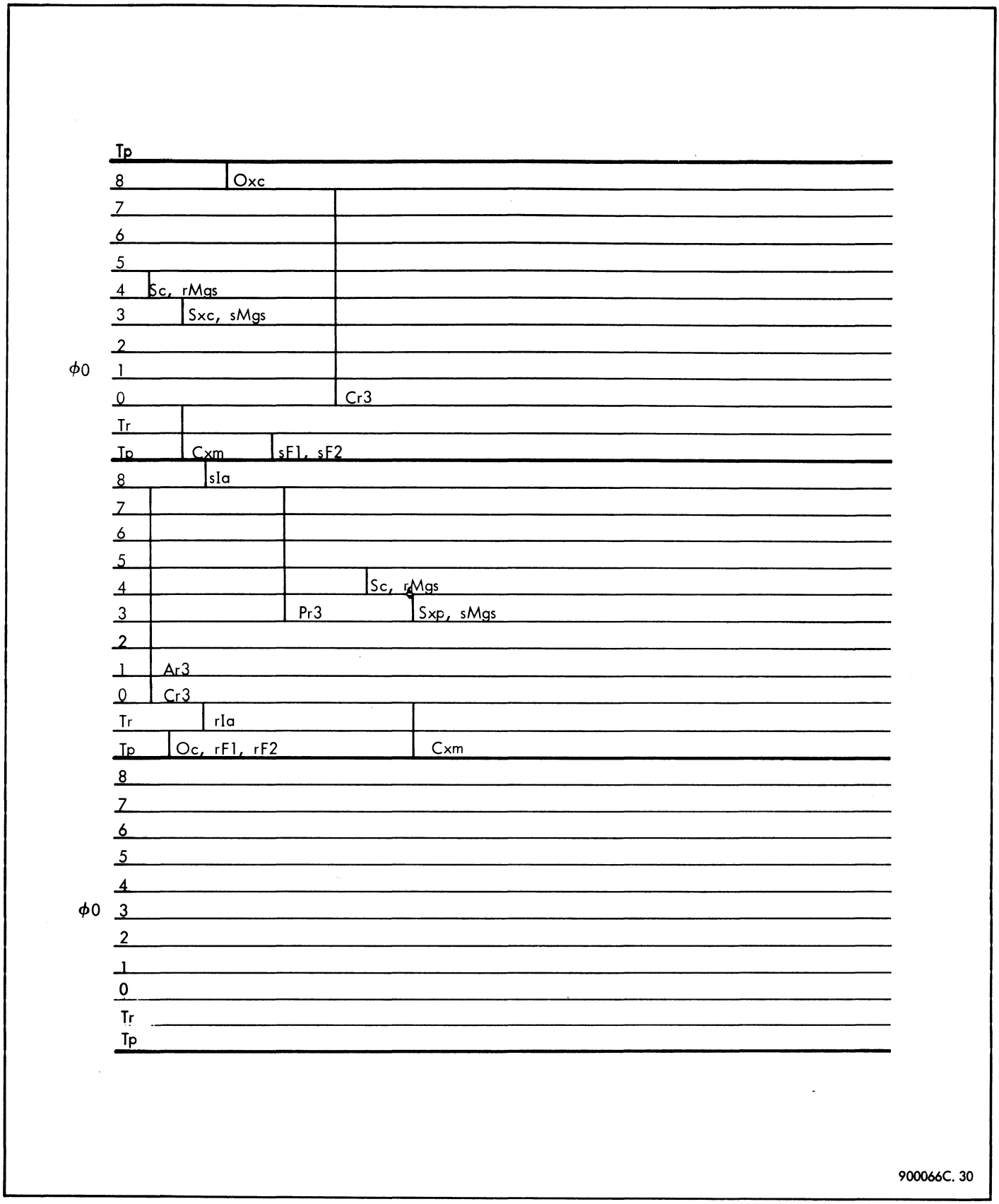


Figure 3-30. MRG Instruction, Timing Diagram

effective memory location and places the result in the A-register. The original instruction parity is checked during Ø0 Cr3. The operand is loaded into the C-register during phase 0 and operand parity is checked during Ø6 Cr3. While the C-register and the A-register are shifting right during phase 6, the merge operation is performed as follows:

$$[(A21 \oplus C21) + A21\ C21] \longrightarrow A0$$

$$[(A22 \oplus C22) + A22\ C22] \longrightarrow A1$$

$$[(A23 \oplus C23) + A23\ C23] \longrightarrow A2$$

A ONE in carry flip-flop Ia enables the P-register to be increased by one during Ø6 Pr3. A memory cycle is initiated during phase 6 to read out the next instruction.

3-282 Exclusive OR (EOR, Code 17) Ø0 - Ø6 - Ø0
(See figure 3-31)

3-283 The EOR instruction performs a logical exclusive OR between corresponding bits of the A-register and the effective memory location and places the result in the A-register. The original instruction parity is checked during Ø0 Cr3. The operand is read from memory during phase 0, and operand parity is checked during Ø6 Cr3. While the A-register and the C-register are circulating during phase 6, the exclusive OR is performed as follows:

$$(A21 \oplus C21) \longrightarrow A0$$

$$(A22 \oplus C22) \longrightarrow A1$$

$$(A23 \oplus C23) \longrightarrow A2$$

A ONE in carry flip-flop Ia enables adding one to the contents of the P-register while the P-register is circulating in phase 6. A memory cycle is started during phase 6 to access the next instruction.

3-284 No Operation (NOP, Code 20) Ø0 - Ø5 - Ø0
(See figure 3-32)

3-285 A NOP instruction moves the program to the next memory location without affecting the A-, B, or X-register or memory. Instruction parity is not checked. A ONE in

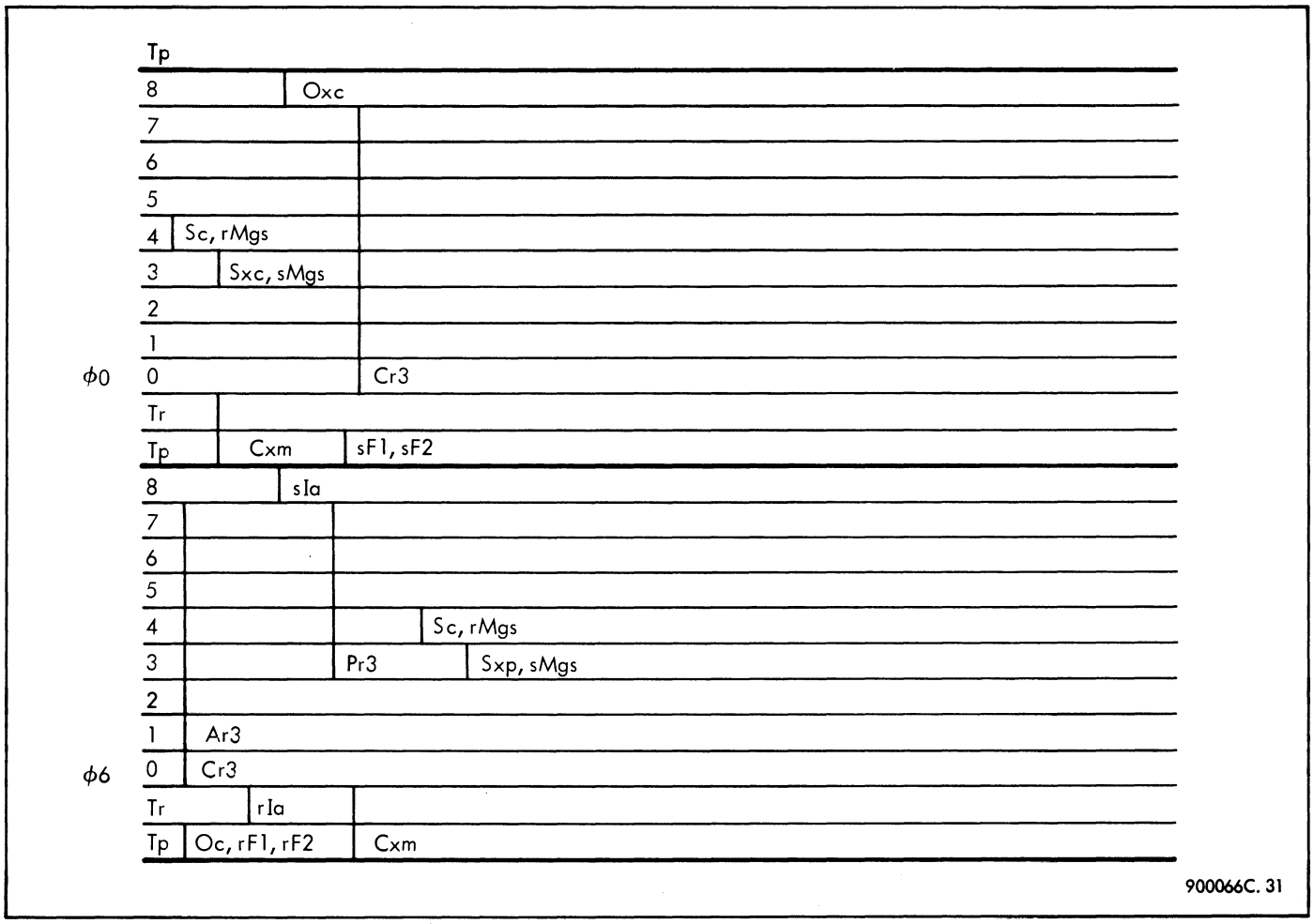


Figure 3-31. EOR Instruction, Timing Diagram

carry flip-flop Ia at Ø5 T7 enables adding one to the contents of the P-register while the P-register is recirculating in phase 5. A memory cycle is started during phase 5 to read out the next instruction. In this instruction, phase 5 appears as an end cycle.

3-286 Pseudo-NOP Instruction. When a Halt instruction has been executed, a pseudo-NOP instruction is executed until the halt condition is removed. During a pseudo-NOP instruction, the computer registers are not altered. The C-register contains ZERO's, and the NOP code has been placed directly in the O-register with an Oc signal. Because the C-register is cleared, Ia cannot be set, and the P-register remains in its original state. Since Go has been reset by the Halt instruction, Cxm cannot come true, and the contents of the current memory location cannot be transferred to the C-register.

3-287 Execute (EXU, Code 23) Ø0 - Ø0 (See figure 3-33)

3-288 The Execute instruction causes the contents of the effective memory location to be executed as an instruction without altering the contents of the program counter. The original instruction parity is checked during Ø0 Cr3. The instruction contained in the effective memory location is read out during phase 0. The contents of the P-register remain unchanged until the end cycle of the next instruction taken from memory. No end cycle occurs during the EXU instruction.

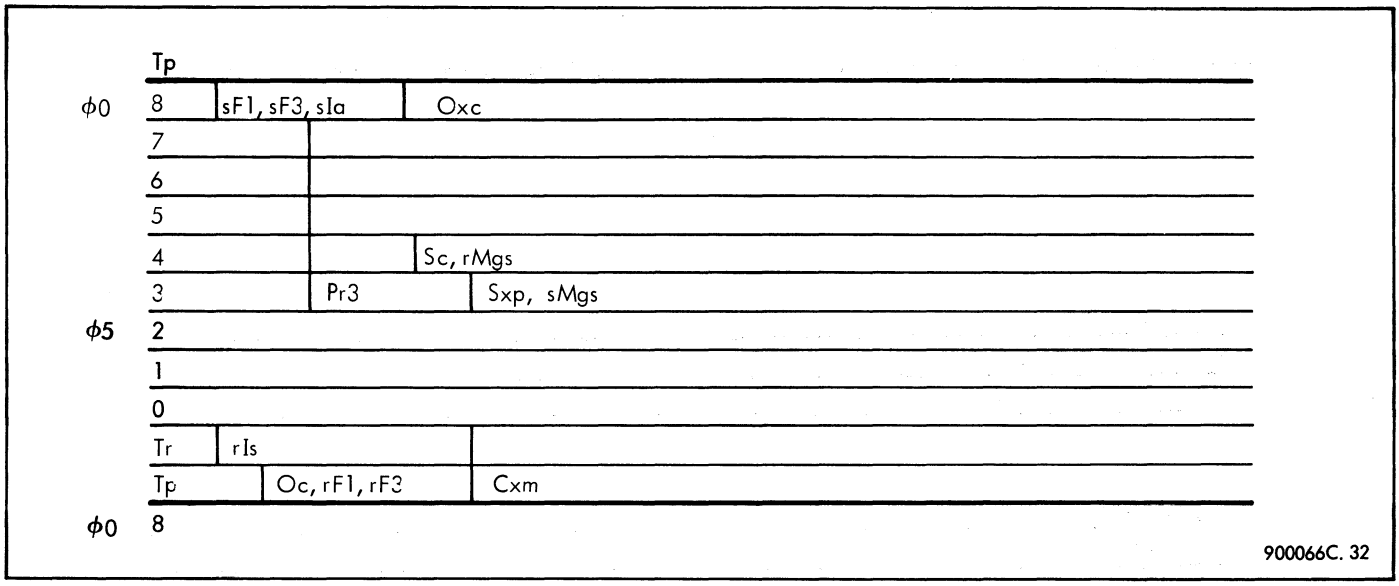


Figure 3-32. NOP Instruction, Timing Diagram

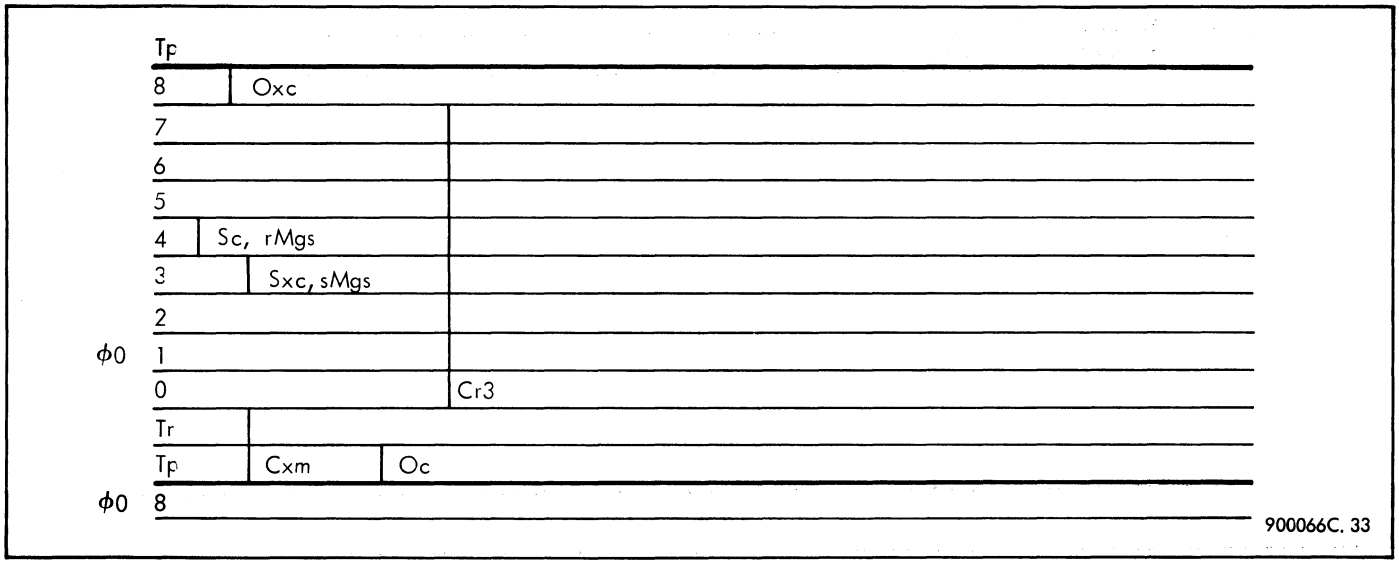


Figure 3-33. EXU Instruction, Timing Diagram

3-289 Y Buffer Into M When Full (YIM, Code 30)

Ø0 ------ Ø2 - Ø4 - Ø7 - Ø0

3-290 The YIM instruction is identical to WIM, code 32, except that Yf Y9 is used in place of Wf W9, and $\overline{O5}$ instead of O5 is transferred to a cable driver during phase 4.

3-291 W Buffer Into M When Full (WIM, Code 32)

Ø0 ------ Ø4 - Ø7 - Ø0 (See figure 3-34)

3-292 The WIM instruction transfers the contents of the channel W word buffer into the effective memory location. The original instruction parity is checked during Ø0 Cr3.

3-293 The effective address is loaded into the S-register at Ø0 T3, and the contents of this location are loaded into the C-register at Ø0 Tp; however, the operand obtained in this access is disregarded. If the buffer is ready to transmit data at Ø0 T0, the Rf flip-flop is set:

$$sRf = \left[\overline{Wf}\ (\overline{W0} + \overline{W9})\right]\ O5\ \overline{O6}\ \overline{F3}\ T0 + \ldots$$

where

$$\overline{Wf}\ (W0 + \overline{W9}$$

is a ready signal from the buffer. Setting Rf changes the phasing from Ø0 to Ø4 at Tp:

$$sF1 = Tp\ (\overline{F1}\ \overline{F3}\ \overline{O1}\ O3\ \overline{O4})\ Ia\ Rf + \ldots$$

If $\overline{Wf}\ \overline{W9}$ is false at Ø0 T0, phase 2 is entered:

$$sF2 = (Tp\ \overline{Ia}\ Ø0)\ O3\ \overline{O4}\ \overline{Rf} + \ldots$$

The logic cycles through phase 2 until the buffer is ready, at which time the phase counter advances to phase 4 at Ø2 Tp:

$$rF2 = (Tp\ \overline{F1}\ \overline{F3}\ \overline{O1}\ O3\ \overline{O4}\ \overline{Ia}\ Rf)\ O2$$

3-294 During phase 4, the buffer data word is shifted into the C-register while Cr3 is true. The data comes three bits at a time from the Rn register as follows:

$$sC_{0-3} = Rn_{1-3}\ \ Cr3\ Rx + \ldots$$

where Rx is a signal sent to the buffer during programmed input/output instructions.

3-295 Parity flip-flop C24 is reset at Ø4 T8, because this flip-flop is used to generate parity as the data is shifted into the C-register. During phase 4 a memory cycle is initiated to write the new data into memory. During Ø4 Pr3, carry flip-flop Ia is true, enabling the contents of the P-register to be increased by one. Flip-flop Ia is reset at the end of phase 4 so that at the following phase (phase 7) the P-register may recirculate without changing and the next instruction may be read from memory. During phase 4, Rx and O5 are presented to cable drivers for transmission to the buffer and O5 is transmitted to the external equipment in the form of a Pwy signal.

3-296 Parallel Input (PIN, Code 33) Ø0 - Ø2 - Ø4 - Ø7 - Ø0 (See figure 3-35)

3-297 This instruction stores the contents of 24 input lines in parallel in the effective memory location. The original instruction parity is checked during Ø0 Cr3.

3-298 During phase 0, the effective address when the data will be stored is transferred from the C-register to the S-register. The information in the effective location is placed in the C-register, but is disregarded. The phase counter is advanced to phase 2 by setting F2:

$$sF2 = (Tp\ \overline{Ia}\ Ø0)\ O3\ \overline{O4}\ \overline{Rf}$$

During Ø2 Q2, the C-register is cleared, and during Ø2 Q1 a Cxi signal reads the data if any from the Cd input lines into the C-register. A PIN signal is presented to a cable driver during Ø2 Q1. The logic remains in phase 2 until the buffer is ready to transmit data. The Rf flip-flop is set sometime during Ø2 Q2 when the buffer is ready.

$$sRf = Q2\ \overline{F3}\ O6\ F2\ Rt$$

where Rt is ready signal from the buffer. A true signal from Rf at Ø2 Tp allows the phase counter to advance from phase 2 to phase 4.

3-299 At Ø4 T8, C24 is reset so that this flip-flop can be used to generate parity for the PIN data word while the C-register is circulating. During T7 through Tr of phase 4, PIN complete signal Rti is presented to a cable driver for use by the buffer. A memory cycle is started to store the data in the C-register in the memory. At T8, carry flip-flop Ia is set so that the contents of the P-register may be increased by one while Pr3 is active. The new contents of the P-register provide the address of the next instruction. At Tp of phase 4, the input data in the C-register is transferred to the M-register, to be placed in memory at the effective address specified by the S-register. At the same time, F2 and F3 are set to advance the counter to phase 7.

3-300 In phase 7, the P-register recirculates but does not receive an increment, because Ia is reset. At T3 the next instruction address in the P-register is placed in the S-register, and at Tp the instruction is transferred from memory to the C-register. Also at Tp, and End signal generated because F1 and F2 are set causes the three flip-flops in the phase counter to be reset to phase 0.

3-301 Store A (STA, Code 35) Ø0 - Ø4 - Ø7 - Ø0 (See figure 3-36)

3-302 This instruction stores the contents of the A-register in the effective memory location. The original instruction parity is checked during Ø0 Cr3.

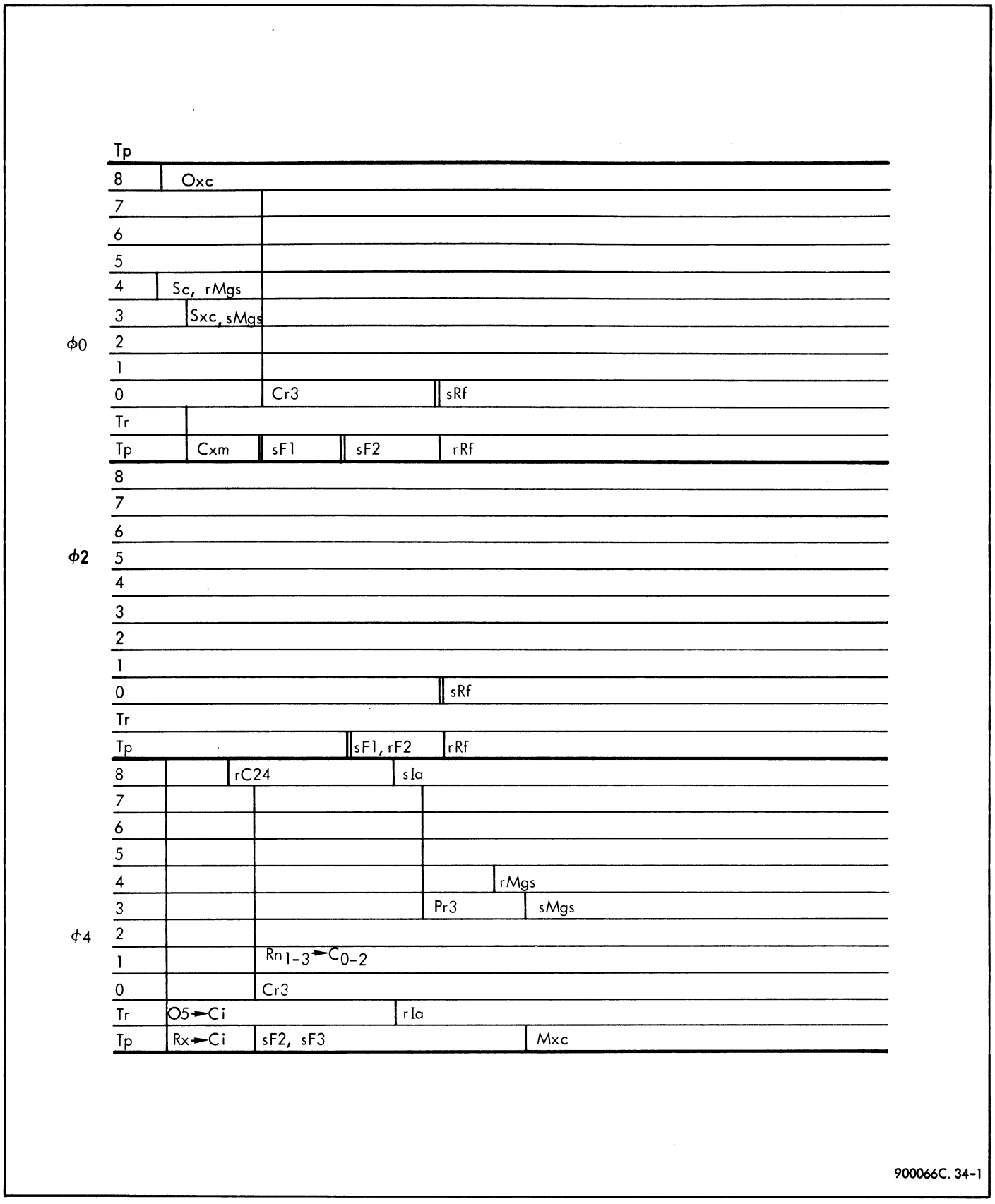


Figure 3-34. WIM Instruction, Timing Diagram (Sheet 1 of 2)

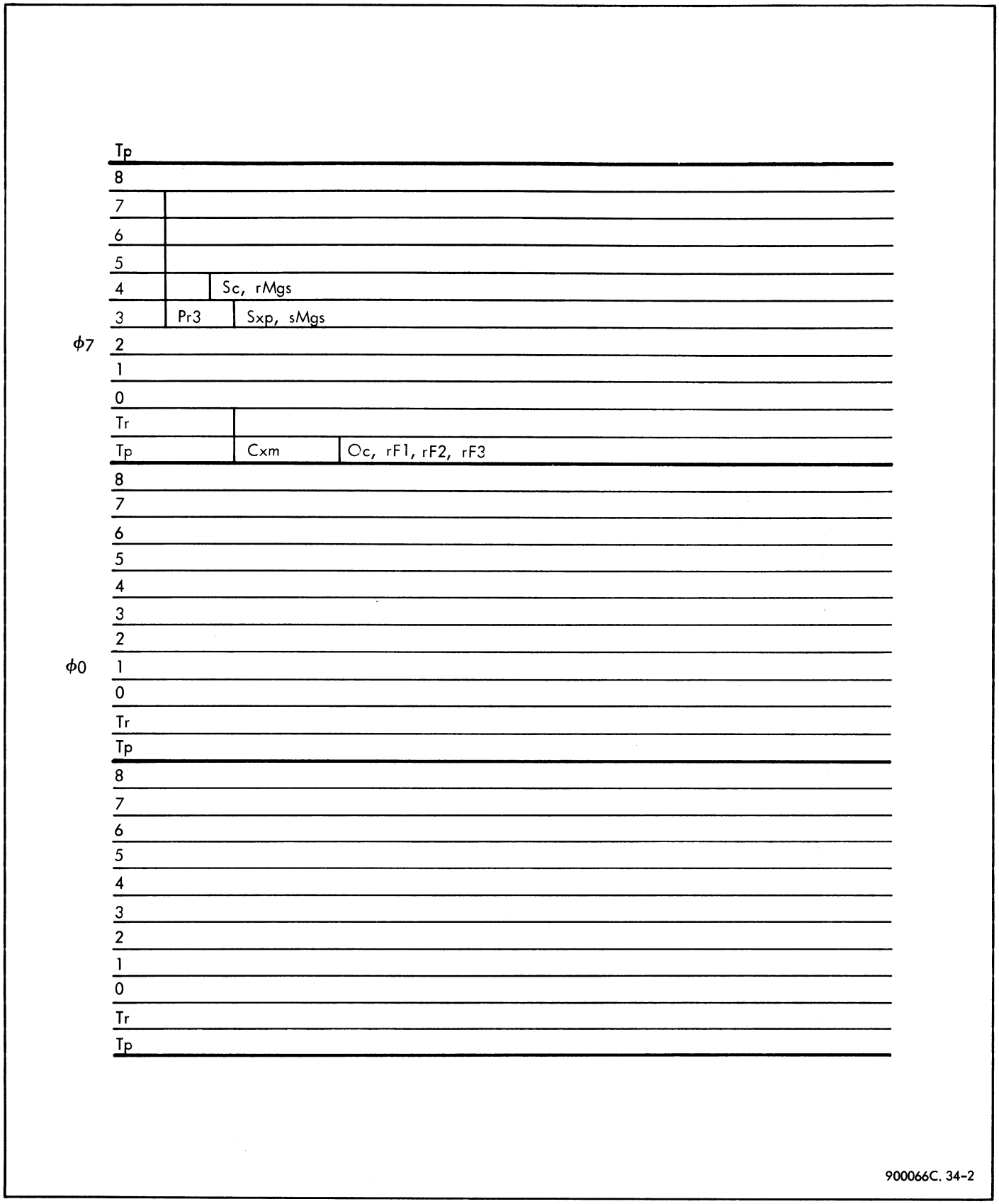


Figure 3-34. WIM Instruction, Timing Diagram (Sheet 2 of 2)

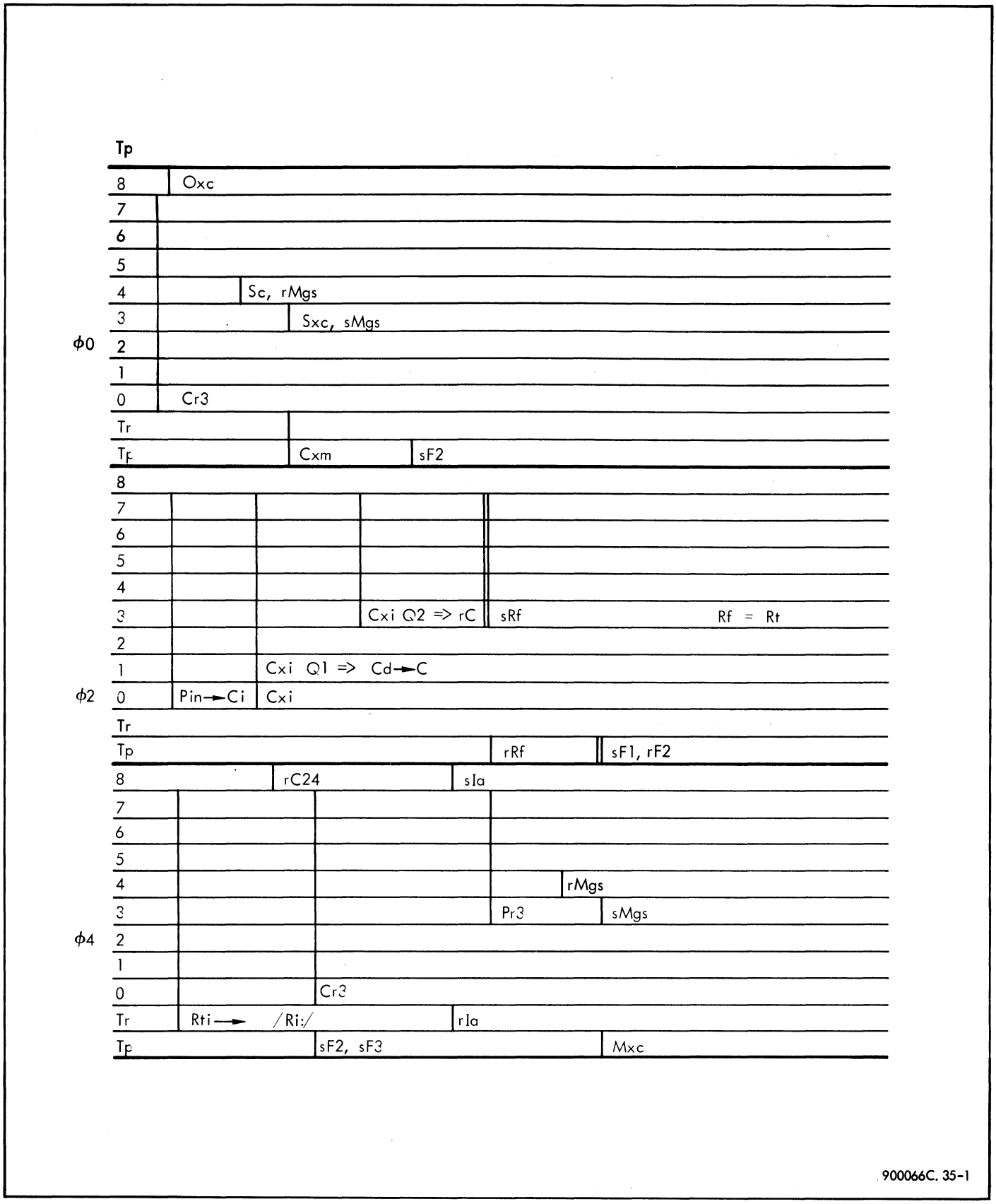


Figure 3-35. PIN Instruction, Timing Diagram (Sheet 1 of 2)

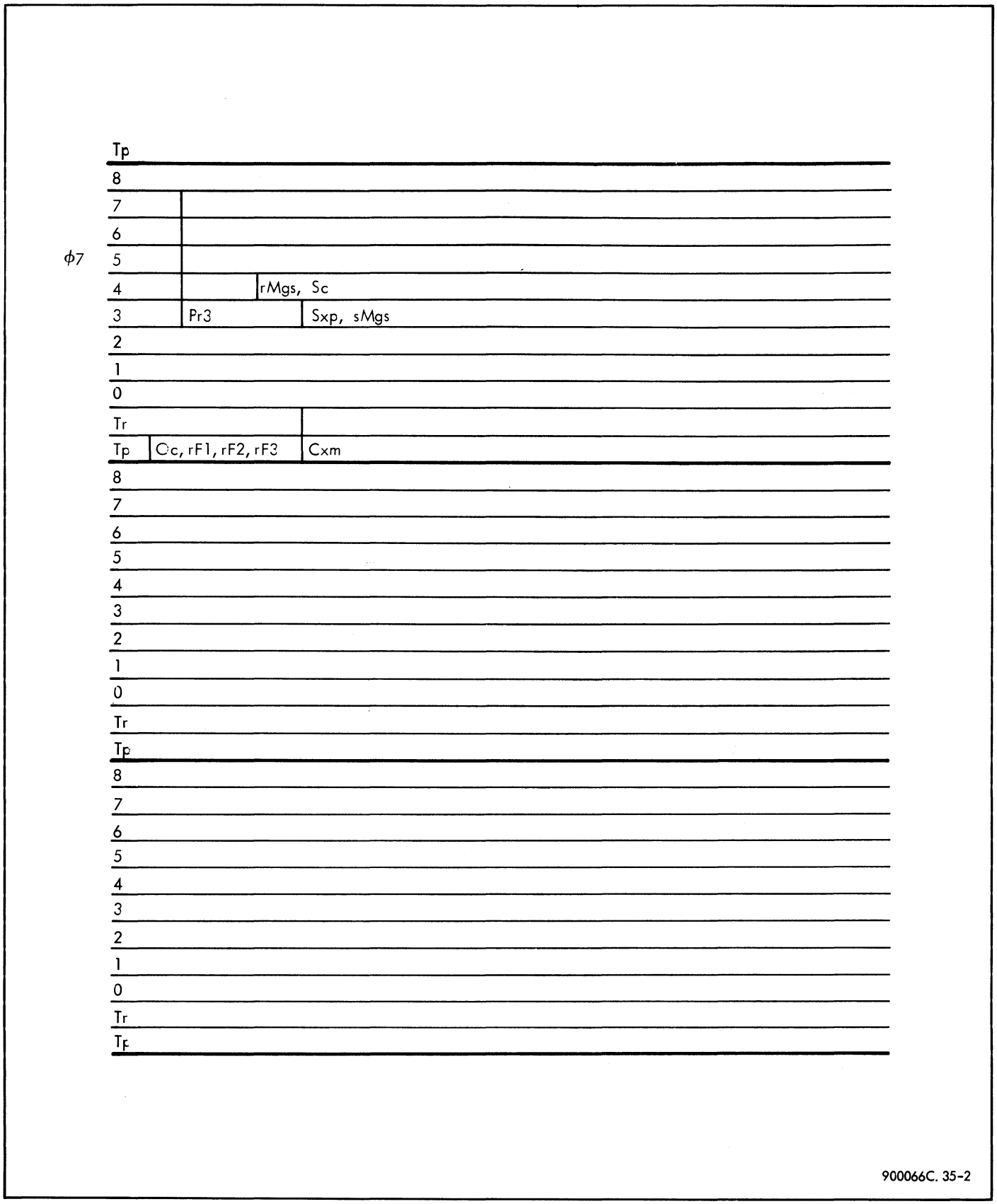


Figure 3-35. PIN Instruction, Timing Diagram (Sheet 2 of 2)

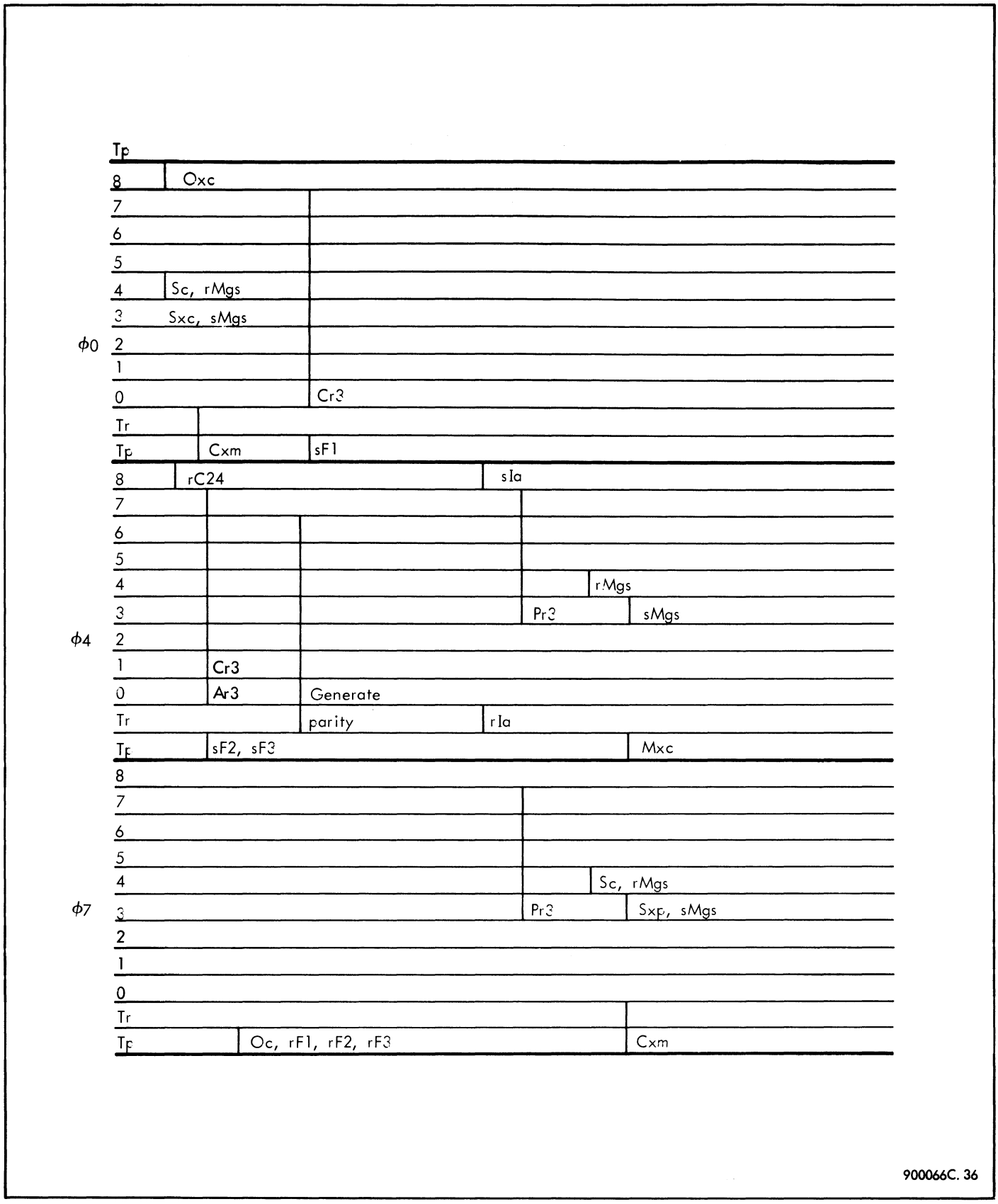


Figure 3-36. STA Instruction, Timing Diagram

3-303 At Tp of phase 0, the contents of the effective memory location are placed in the C- register, but the data is disregarded. The S-register contains the address of the location where the contents of the A-register are to be stored.

3-304 In phase 4, the A-register and the C-register are shifted right three bits at a time, and the contents of the A-register are placed in the C-register by octals. Parity flip-flop C24 is reset at T8, because parity is generated for the memory word while the contents of the A-register are being transferred to the C-register. A memory cycle is started in phase 4, placing the contents of the C-register in memory at the location specified by the S-register, which has not been altered since phase 0. Carry flip-flop Ia is set at T8, enabling the contents of the P-register to be increased by one while circulating right three bits at a time.

3-305 During phase 7, the contents of the P-register are transferred to the S-register, and this P + 1 address is loaded into the C-register at Tp. The next instruction is now in the C-register ready for execution at phase 0.

3-306 Store B (STB, Code 36) Ø0 - Ø4 - Ø7 - Ø0
(See figure 3-37)

3-307 This instruction stores the contents of the B-register in the effective memory location. The sequence of events is identical to those in the Store A instruction. The Ar3 signal is used to circulate the B-register right three bits at a time.

3-308 Store Index (STX, Code 37) Ø0 - Ø4 - Ø7 - Ø0
(See figure 3-38)

3-309 This instruction stores the contents of the X-register in the effective memory location. The sequence of events is identical to those in the Store A and Store B instructions. Since the X-register is continually circulating, no control signal is necessary for this register during the transfer of X into C. The octal digits from the X-register are taken from Xn_{1-3}, which represents the right-hand three bits of the X-register at any time.

3-310 Skip if Signal Not Set (SKS, Code 40)

Ø0 - Ø5 ------ Ø7 ------ Ø0 (See figure 3-39)

3-311 The SKS instruction is used to test the states and responses of data channels and their attached peripheral devices, as well as internal and external indicators. If the signal tested is true, the program skips the following instruction. If the signal tested is false, the program executes the next instruction in sequence. The instruction operates in four different modes, depending on the states of bits 10 and 11 of the instruction, as follows:

C10	C11	Area of Test
0	0	Special internal test
0	1	Channel and device test
1	0	Internal test
1	1	Special system test

An external test requires two cycles if no skip is entailed and three cycles if a skip is called for. An internal test requires one cycle if no skip is involved and two cycles for skipping. In either case, the phasing changes from phase 0 to phase 5 at Ø0 T8. Carry flip-flop Ia is set at Ø0 T8.

3-312 The internal test will be considered first. When the P-register is circulated by Pr3 in phase 5, the contents are increased by one so that the next instruction may be executed if a skip is not performed. The contents of the C-register are presented to cable drivers Ci so that the test conditions may be made available to the data channel and external equipment. If the signal tested is set, the Sk flip-flop remains reset and a normal End cycle occurs during phase 5. The contents of the P + 1 memory location are transferred to the C-register, and F1 and F3 are reset, placing the phase counter at phase 0.

3-313 If the signal tested is not set, Sk is set at Ø5 Tr by an Sks signal, which is derived from an Skrz signal from the input/output channel. The resetting of F1 and F3 are inhibited, and F2 is set so that the phase counter advances to phase 7. Clearing of the O-register (Oc) is inhibited, so that the SKS opcode is still operative in phase 7. The instruction transferred from memory to the C-register during phase 5 is ignored.

3-314 During phase 7, carry flip-flop Ia is set so that the contents of the P-register are increased by one, placing the instruction from the P + 2 location in the C-register on memory access. During the following phase 0, the P + 2 instruction is executed, the P + 1 instruction having been skipped.

3-315 When the Sks is an external test, contents of the C-register must be presented to the cable drivers for two machine cycles. A ONE in C11 causes flip-flop A00 to be set at the first Ø5 T7 and A00 is used to count cycles. At T7 of the second phase 5, A00 is reset. A ONE in A00 enables Bc23 to be set at Ø5 T6 and reset at the following T6. A ONE in Bc23 inhibits the End signal during the first phase 5 so that phase 5 must be repeated. A ZERO in Bc23 enables the End signal during the second phase 5, allowing the logic to proceed to phase 7 if other conditions permit. Skss signal during phase 5 notifies the external device that a test is in process. The duration of Skss may be controlled by bit 9 in the instruction:

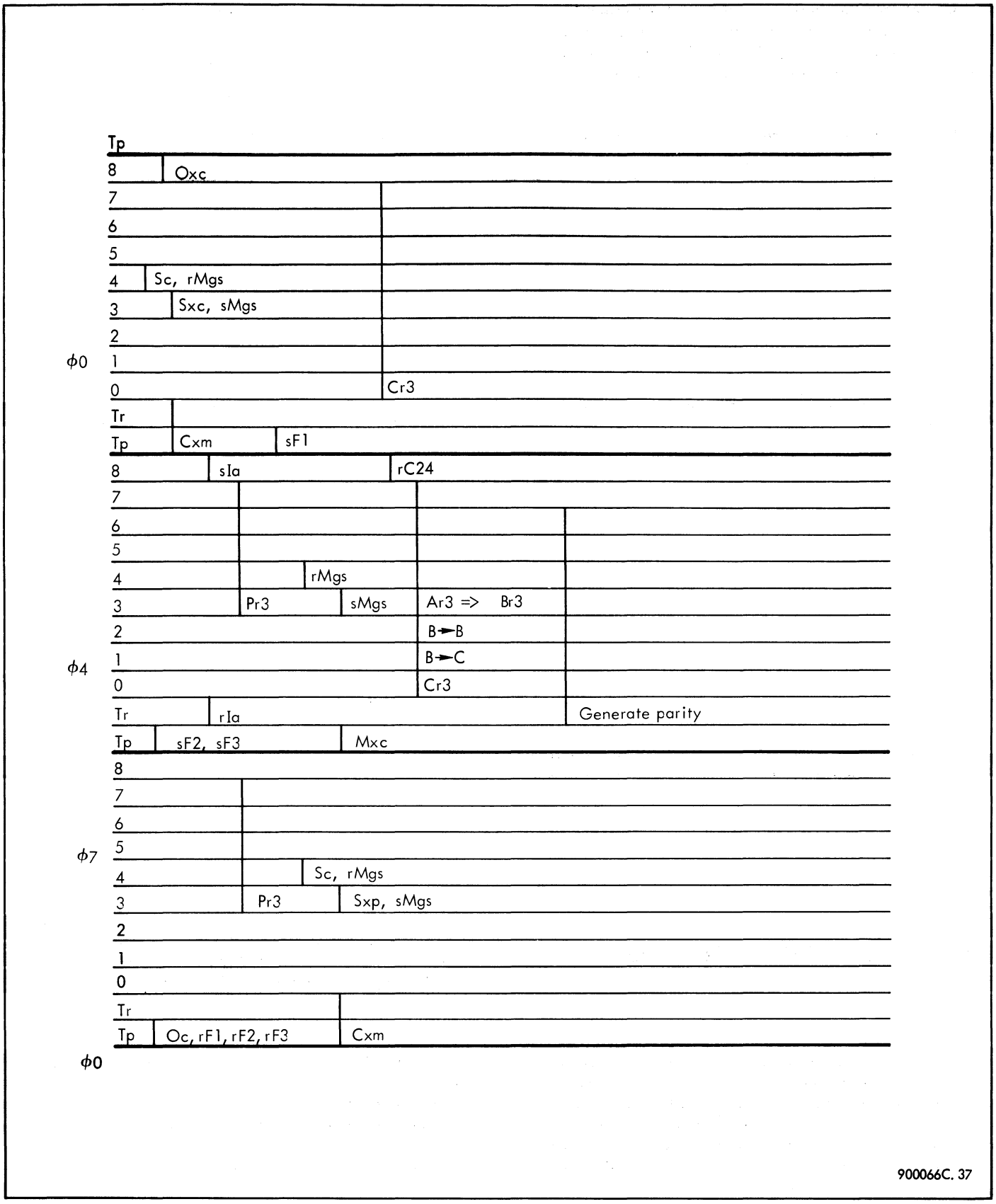


Figure 3-37. STB Instruction, Timing Diagram

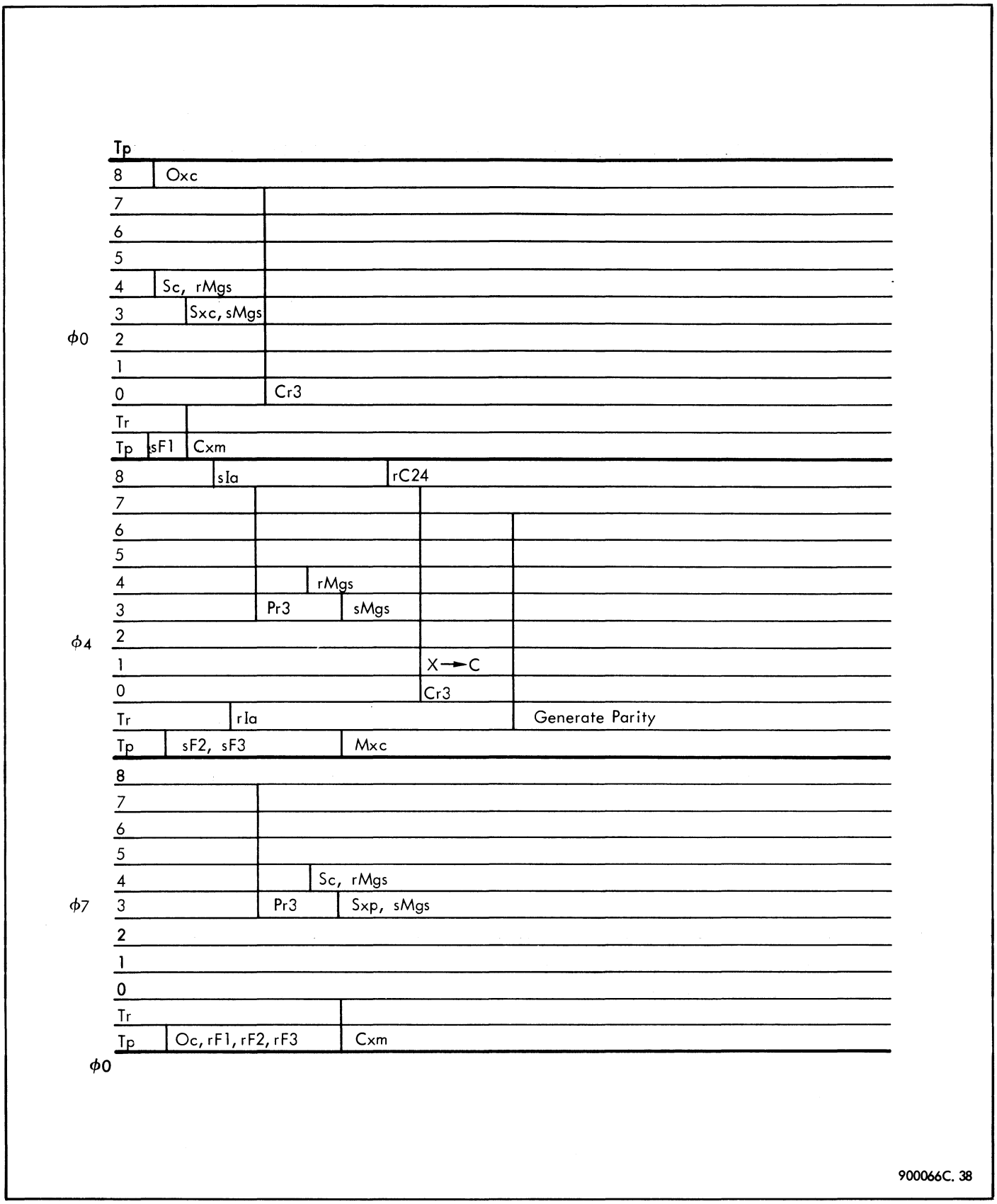


Figure 3-38. STX Instruction, Timing Diagram

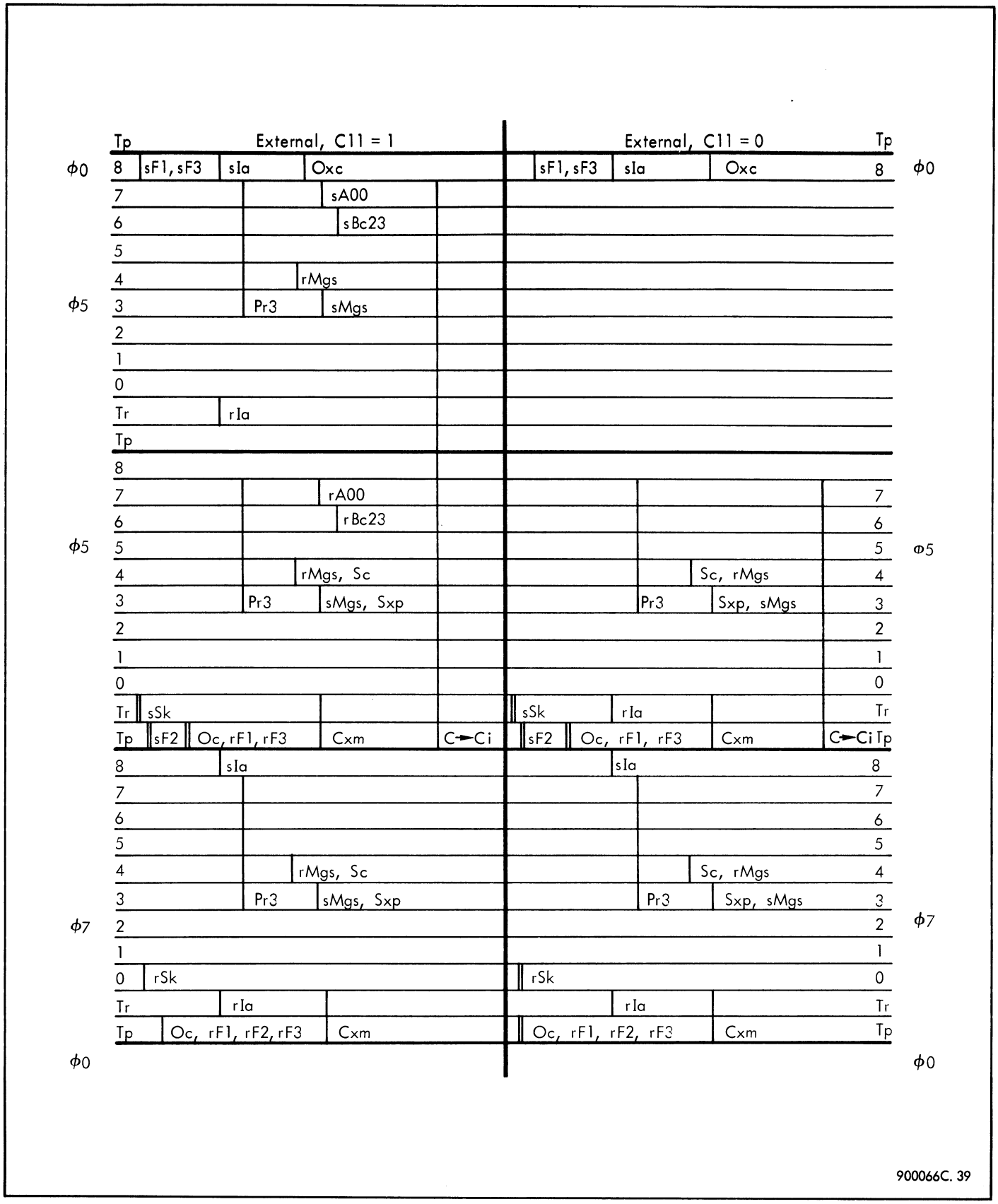


Figure 3-39. SKS Instruction, Timing Diagram

$$Skss = \emptyset 5\ O1\ \overline{O4}\ \left[\overline{A00}\ (Q3 + Q5) + \overline{C9}\right]$$

3-316 During both phase 5 cycles, the contents of the C-register are presented to cable drivers Ci so that the test conditions may be made available to the data channel and external equipment. During the first phase 5, the contents of the P-register are increased by one so that the next instruction may be read from memory during the second phase 5 and executed in the event that a skip is not performed. If the signal tested is set, the Sk flip-flop remains reset and a normal End cycle occurs during the second phase 5. The contents of the P + 1 memory location are transferred to the C-register, and F1 and F3 are reset, placing the phase counter at phase 0.

3-317 If the signal tested is not set, Sk is set at Ø5 Tr. The resetting of F1 and F3 are inhibited, and F2 is set so that the phase counter advances to phase 7. Clearing of the O-register (Oc) is inhibited, so that the SKS opcode is still operative in phase 7. The instruction transferred from memory to the C-register during phase 5 is disregarded.

3-318 During phase 7, carry flip-flop Ia is set so that the contents of the P-register are increased by one, placing the instruction from the P + 2 location in the C-register on memory access. A normal End cycle occurs during phase 7. During the following phase 0, the P + 2 instruction is executed, the P + 1 instruction having been skipped.

3-319 Increment Index and Branch (BRX, Code 41) Ø0 - Ø6 - Ø0 (See figure 3-40)

3-320 The BRX instruction adds one to the index register and then transfers control to either the effective address or to the next location in sequence, depending on the resulting state of bit 9 of the index register. If the BRX instruction is indexed, the transfer of control is to the effective address determined by the value of the index register immediately prior to the execution of BRX. The original instruction parity is checked during Ø0 Cr3.

3-321 At Ø0 T8, the Ju flip-flop is set to control the operations necessary for a branch instruction. Half adder carry flip-flop is unconditionally set, allowing one to be added to the contents of the index register. From T7 to T3, during Pr3 and Cr3, the current program address in the P-register and the effective address in the C-register are interchanged. The contents of the P-register are transferred directly into the C-register, enabled by a Ju signal. The contents of the C-register enter the serial adder through Yz_{1-3}. The contents of the serial adder are gated into the The contents of the serial adder are gated into the P-register by the decoded opcode and the Ju signals. During this transfer, no addition takes place in the adder if the index bit in the BRX instruction is not set because Xz_{1-3} have no inputs and carry flip-flop Cz is not set. If index bit 9 in the BRX instruction contains a ONE, the contents of the index register are added to C via Xz_{1-3}, thereby indexing the effective address. When right shifting of the P-register is completed, the branch address in P is transferred to the address register (S) by a Sxp enable path. The contents of the effective address are now ready to be placed in the C-register if a branch is to be performed.

3-322 From T7 to T0, the contents of the X-register are gated through the half adder and back into the X-register, increased by one through carry flip-flop Hz. The half adder outputs (Ha_{1-3}) enter the index register through Xw1, (Xn2) In1, and (Xn3) In1.

3-323 At T2 of phase 0, the ninth bit of the index register is interrogated to determine whether a branch is to be executed. Bit 9 is at this time in write flip-flop Xw1. If Xw1 contains a ONE, flip-flop O1 is reset, changing the BRX instruction to a simple branch instruction (BRU, code 01). An End signal is generated, causing the remainder of the cycle to function as a normal end cycle. The effective address is placed in the C-register at Ø0 Tr-Tp and the Ju flip-flop is reset. Setting of F1 and F2 is inhibited because O1 is false; therefore, the program stays in phase 0 and the instruction in the effective address is executed.

3-324 If Xw1 contains a ZERO at Ø0 T2, O1 remains set, and this condition causes the P + 1 instruction to be accessed as follows: Serial adder carry flip-flop Cz is set at Tr. The end signal is not true; therefore, a memory access is inhibited, and the current program address remains in the C-register. A true signal from O1 enables the setting of F1 and F2, placing the phase counter in phase 6 and enabling and end cycle to take place. During phase 6, the current program location in the C-register is transferred to the P-register through the full adder, which adds one because Cz is set. This P + 1 address is placed in the S-register at T3 and is the contents of the P + 1 memory location are placed in the C-register at Tp. The phase counter enters phase 0, Cz is reset at T8, and the P + 1 address is ready for execution.

3-325 Mark Place and Branch (BRM, Code 43) Ø0 - Ø6 - Ø0 (See figure 3-41)

3-326 The BRM instruction stores the contents of the P-register (the address of the BRM instruction itself) in the effective memory location and transfers control to the effective memory location plus one. The instruction also stores the status of the overflow indicator in bit 0 of the effective location and E3m and E2m in bits 3 through 5 and 6 through 8, respectively. The original instruction parity is checked during Ø0 Cr3.

3-327 At T8 of phase 0, the Ju flip-flop is set for a branch function, C24 is reset for parity generation, and the opcode is placed in the O-register. The P-register and the C-register shift right three bits at a time, and their contents are interchanged by shifting P into C and C into the serial adder, from which the information goes into P.

3-328 At T4, 12 bits have been shifted out of the P-register, and only two more are left; therefore at T3 it is

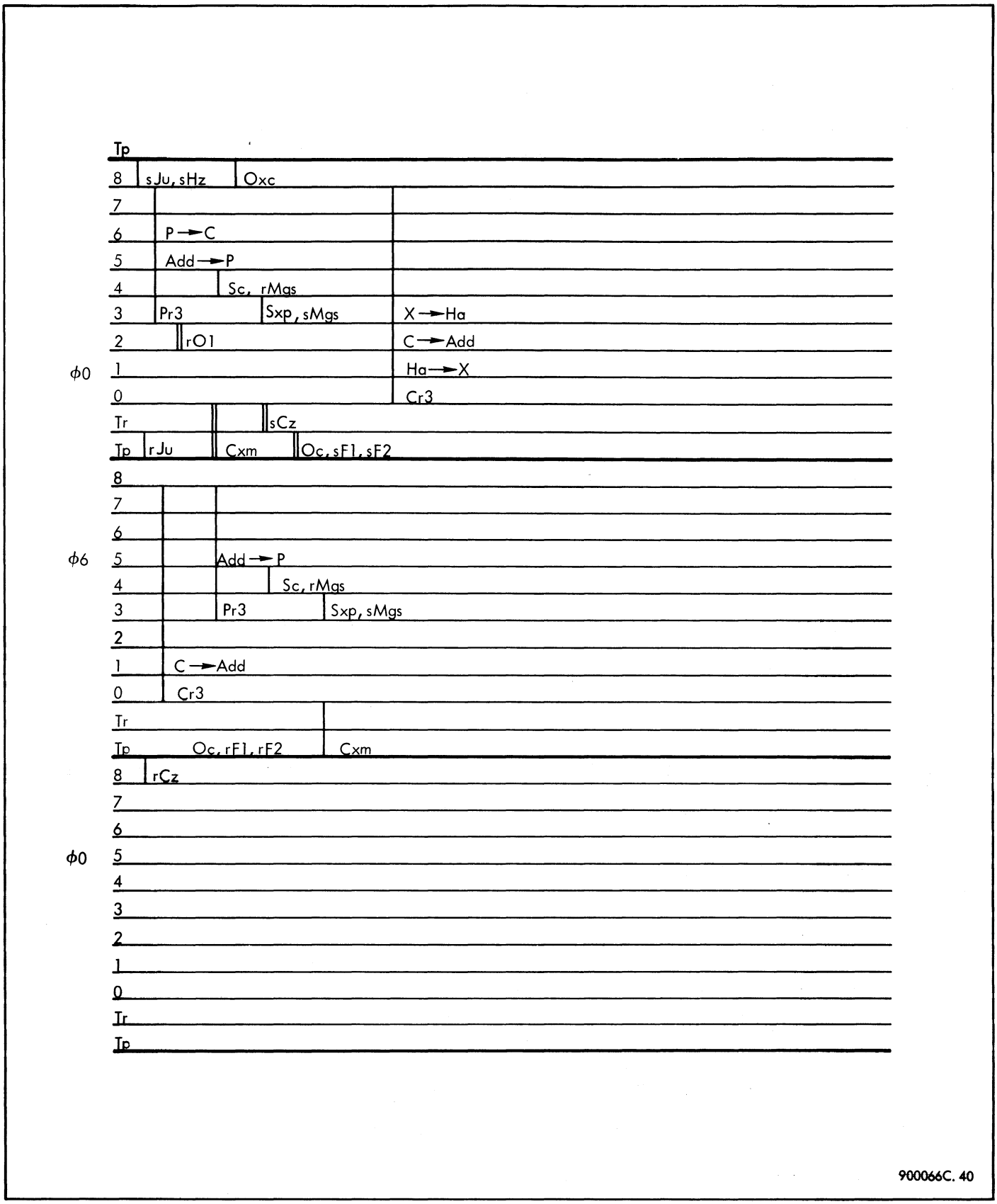


Figure 3-40. BRX Instruction Timing Diagram

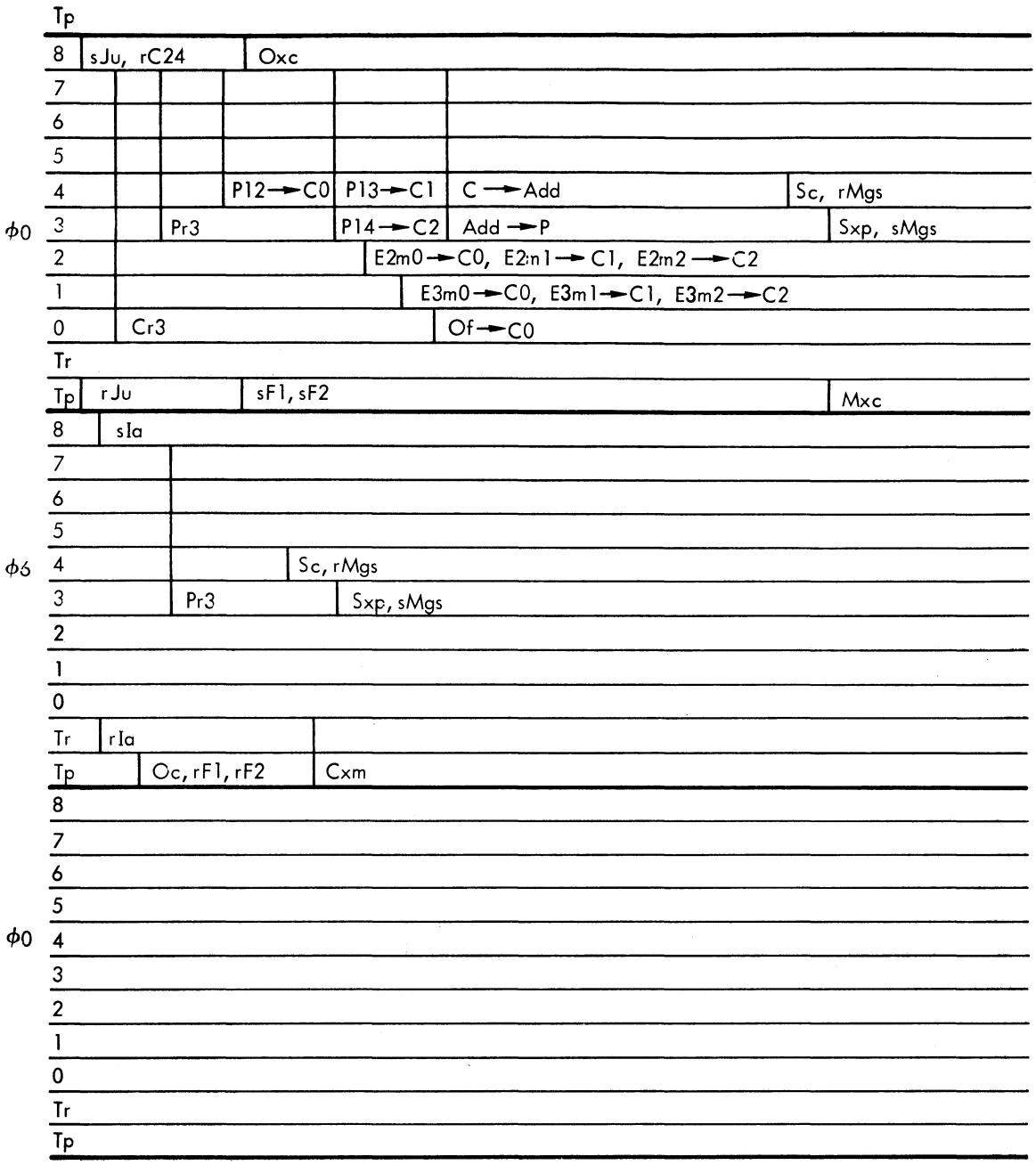


Figure 3-41. BRM Instruction, Timing Diagram

only necessary to shift P13 and P14 into the C-register. Bit 12 of the P-register always contains a zero at this time. The shifting of P into C is qualified by $\overline{Q6}$ (T7 - T4) for P12, and by Q2 (T7 - T3) for P13 and P14:

$$sC0 = P12\ \overline{Q6}\ Ju\ \overline{Ts} + \ldots$$

$$sC1 = P13\ Q2\ Ju\ \overline{Ts} + \ldots$$

$$sC2 = P14\ Q2\ Ju\ \overline{Ts} + \ldots$$

3-329 At T3, the P-register contains the effective address, which is placed in the S-register for access. The C-register contains the current program address, which is stored in memory at the effective location by Mxc at Tp.

3-330 At T2 of phase 0, the contents of memory extension flip-flops $E2m_{0-2}$ are placed in the C-register, their destination being bits 6 through 8. At T1, $E3m_{0-3}$ are transferred to the C-register, to be placed in bits 3 through 5 at the next clock pulse. At T0, C0 is set or reset depending on the state of the overflow flip-flop.

3-331 An end cycle follows in phase 6. Because carry flip-flop Ia is set, P + 1 operation is performed during right shift of the P-register, and at T3 this address is transferred to the S-register. The memory access at Tp, therefore, places the contents of the effective address plus one in the C-register for execution at the following phase 0.

3-332 Register Change (RCH, Code X460XXXX) Ø0 - Ø5 - Ø0 (See figure 3-42)

3-333 A Register Change instruction performs one of 19 register change operations, depending on the number in the first and the last four octal positions of the instruction code. A 46 in the opcode field is common to all of the instructions. The instructions and their functions are listed in table 3-11.

3-334 The functions of the individual bits in the register change instruction other than the opcode bits are presented in table 3-12. Combinations of these bits are used to perform simultaneous operations.

3-335 The original instruction parity is not checked.

3-336 A typical Register Change instruction is given in the timing diagram in figure 3-42, using the Exchange A and B instruction (XAB, code 0460014) as an example.

3-337 At T8 of phase 0, carry flip-flop Ia is set in order to increase the contents of the P-register by one, Ju is set so that the contents of the P-register may be transferred to the S-register, register change flip-flop Rc is set by decoding of the opcode in bits 3 through 8 of the C-register. Timing signals and decoding of the opcode enable setting F1 and F2 to advance the phase counter to phase 5.

3-338 Both the A-register and the B-register are shifted right three bits at a time during Q1 by the Ar3 term. A true output from Rc and the state of bits from the C-register allow the least significant three bits of the B-register to be transferred to the most significant three positions of the A-register and the least significant A-bits to be transferred to the most significant B-positions, as in the following equations:

$$sA0 = B21\ Rc\ C20 + \ldots$$

$$sB0 = A21\ Rc\ C21 + \ldots$$

A true output from carry flip-flop Ia causes one to be added to the contents of the P-register during Q2. A true signal from Ju enables the contents of the P-register to be transferred to the S-register at T3 so that at the end of phase 5, the next instruction in sequence will be taken from memory.

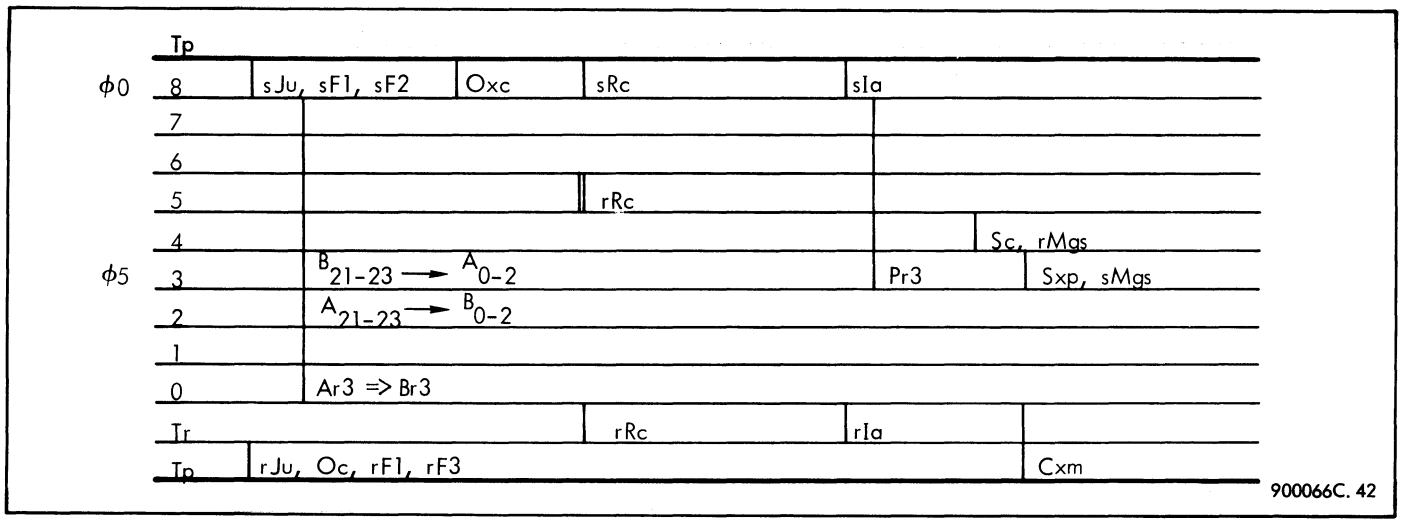


Figure 3-42. RCH Instruction, Timing Diagram

Table 3-11. Register Change Instructions

Instruction Code	Mnemonic	Name	Function
0 46 00001	CLA	Clear A	Clears contents of the A-register to zero
0 46 00002	CLB	Clear B	Clears contents of B-register to zero
0 46 00003	CLR	Clear AB	Clears contents of both A- and B-registers to zero
0 46 00004	CAB	Copy A into B	Copies contents of the B-register into A-register
0 46 00005	ABC	Copy A into B, Clear A	Copies contents of A-register into B-register and simultaneously clears A-register to zero
0 46 0010	CBA	Copy B into A	Copies contents of B-register into A-register
0 46 00012	BAC	Copy B into A, Clear B	Copies contents of B-register into A-register and simultaneously clears B-register to zero
0 46 00014	XAB	Exchange A and B	Copies contents of A-register into B-register and simultaneously copies contents of B-register into A-register
0 46 00020	CBX	Copy B into Index	Copies contents of B-register into index register
0 46 00040	CXB	Copy Index into B	Copies contents of index register into B-register
0 46 00060	XXB	Exchange Index and B	Copies contents of index register into B-register and simultaneously copies contents of B-register into index register
0 46 00122	STE	Store Exponent	Copies least significant nine bits of B-register into index register, extends bit 15 of index register (exponent sign) into bit 0, then clears nine least significant bits of B
0 46 00140	LDE	Load Exponent	Clears nine least significant bits of B-register and copies least significant nine bits of index register into least significant nine bits of B-register
0 46 00160	XEE	Exchange Exponents	Exchanges lest significant nine bits of B-register and index register. New bit 15 of index register (exponent sign) extends into bit 0
0 46 00200	CXA	Copy Index into A	Copies contents of index register into A register
0 46 00400	CAX	Copy A into Index	Copies contents of A-register into index register
0 46 00600	XXA	Exchange Index and A	Copies contents of index register into A-register and simultaneously copies contents of A-register into index register

Table 3-11. Register Change Instructions (Cont.)

Instruction Code	Mnemonic	Name	Function
0 46 01000	CNA	Copy Negative into A	Copies two's complement of contents of A-register into A-register
2 46 00200	CXA	Copy Index into A	Copies contents of index register into A-register

Table 3-12. Functions of Register Change Instruction Bits

C-Register Bit	
1	Clear X
14	-(A) ⟶ A
15	(A) ⟶ X
16	(X) ⟶ A
17	Process bits 15-23 only
18	(X) ⟶ B
19	(B) ⟶ X
20	(B) ⟶ A
21	(A) ⟶ B
22	Clear B
23	Clear A

3-339 Register Transfers. Transfer of data from one register to another is enabled by the Ar3 signal during T7 through T0 of phase 5, the type of transfer being selected by gating Rc with the bit of the C-register that specifies the particular transfer as shown in table 3-12. For example, during the Exchange A and B instruction, bits 21 through 23 of the B-register are gated into bits 0 through 2 of the A register by Rc and C20. Bits entering the index register are transferred an octal digit at a time through Xw1, (Xn2) In1, and (Xn3) In1, which are the parallel inputs to the three legs of the index register.

3-340 Register Clearing. The X-register is cleared under the control of C1, which is a set term for index flip-flop Ix. Decoding of the opcode is used with the true output of Ix to bring down the Xnr term, thereby resetting Xw1:

$$rXw1 = \overline{\overline{Xnr}\ Xn1} + \ldots$$

When Xw1 is reset, the inputs to the second and third legs of the index register are made false:

$$\overline{(Xn3)\ In1} = \overline{Xw1} + \ldots$$

$$\overline{(Xn2)\ In1} = \overline{Xw1} + \ldots$$

3-341 The A-register is cleared during register change when C22 contains a ONE. An Anr signal is generated from RcC22, and the Anr signal in turn allows the clearing of A0-3:

$$rA0\text{-}2 = Ar3\ \overline{(A21\text{-}23\ \overline{Anr})}$$

As the A-register bits are shifted right, the register is filled with ZERO's.

3-342 Clearing of the B-register is enabled with a RcC23 signal, which generates Bnr. The Bnr signal allows B0-2 to be reset, with the result that the B-register is filled with ZERO's as the bits shift right:

$$rB0\text{-}2 = Ar3\ \overline{(B21\text{-}23\ \overline{Bnr})}$$

3-343 Complementing Contents of the A-Register. The false outputs of bits 21 through 23 of the A-register are presented to the half adder with O3. Carry flip-flop Hz is set at T8 to add one to Ha1-3. The outputs of the half adder are the two's complement of A21 through A23, because one has been added to the one's complement. The half adder outputs are transferred to the A-register while the register is recirculating three bits at a time.

3-344 Processing Bits 15 Through 23 Only. When only bits 15 through 23 are to be transferred, C17 is true, and this signal causes Rc to be reset at T5. At this time, the least significant three octals have been shifted, and no more transfers may take place as long as Rc is false.

3-345 Skip if A Equals M (SKE, Code 50)

Ø7

Ø0 - Ø6 ------ Ø0 (See figure 3-43)

3-346 The SKE instruction compares the contents of the A-register with the contents of the effective memory location. If the contents of A equal the contents of the effective location, the computer skips the next instruction in sequence and executes the following instruction. If the contents of A do not equal the contents of the effective location, the computer executes the next instruction in sequence.

3-347 The original instruction parity is checked during Ø0 Cr3. The effective address in the C-register is placed in the S-register at Ø0 T3, and the contents of the effective location are placed in the C-register at Ø0 Tp. Phase 6 is entered, and operand parity is checked during Cr3.

3-348 Skp flip-flop Sk is set at T8 of phase 6. The state of this flip-flop at the end of phase 6 determines whether or not an instruction is to be skipped.

3-349 While the A- and C-registers are recirculating during phase 6, A21-23 are compared with C21-23 as follows:

$$(A21 \oplus C21) + (A22 \oplus C22) + (A23 \oplus C23)$$

If this expression is true at any time, the contents of the A-register do not equal the contents of the effective location, and Sk is reset. The contents of the P-register, which have been increased by one during phase 6 by carry flip-flop Ia, are placed in the S-register. This P+1 instruction is placed in the C-register at Tp. Tp End Sk term resets Fl and F2, and the P+1 instruction is executed at the following phase 0.

3-350 If the exclusive OR expression in paragraph 3-349 remains false throughout the recirculation of the A- and C-registers, the contents of the A-register equals the contents of the effective memory location and Sk remains set. Resetting of F1 and F2 is inhibited, and F3 is set, placing the phase counter in phase 7. The instruction in the C-register is disregarded, the contents of the P-register are again increased by one, and this P+2 address is placed in the S-register. The instruction contained in the P+2 location is placed in the C-register at Tp, ready for execution at the following phase 0.

3-351 Return Branch (BRR, Code 51) Ø0 - Ø6 - Ø0 (See figure 3-44)

3-352 The BRR instruction copies the contents of the effective memory location into the C-register and adds one to the number copied. The least significant 14 bits of the resulting number are stored in the P-register. The overflow flip-flop is set if C0 contains a ONE.

3-353 The original instruction parity is checked during recirculation of the C-register in phase 0. The effective address is placed in the S-register at T3, and at Tp the contents of the effective location are placed in the C-register. The full adder carry flip-flop, Cz, is set at Ø0 Tr.

3-354 At T7 of phase 6, overflow flip-flop Of is set if a ONE-bit is contained in position 0 of the C-register. The contents of the C-register shift through the full adder, which adds one because the carry flip-flop is set. The outputs of the adder are applied to bits 0 through 2 of the C-register. During T7 through T3, the 15 least significant bits of the number going through the adder are shifted into the P-register, and at T3 the least significant 14 are transferred to the S-register for memory access. The contents of the effective address plus one are now ready to be placed in the C-register at Tp.

3-355 Skip if B and Memory Do Not Compare Ones

Ø7

SKB, Code 52) Ø0 - Ø6 ------ Ø0 (See figure 3-45)

3-356 The SKB instruction compares the contents of the B-register, bit by bit, with the contents of the effective memory location. If the contents of the B-register and the contents of the effective location do not have a pair of ONE's in any corresponding bit positions, the next instruction in sequence is skipped and the following instruction is executed. If the contents of the B-register and the contents of the effective location have at least one pair of ONE's in corresponding bit positions, the next instruction in sequence is executed.

3-357 The original instruction parity is checked during recirculation of the C-register in phase 0. The effective address is placed in the S-register at T3, and at Tp the contents of the effective location are placed in the

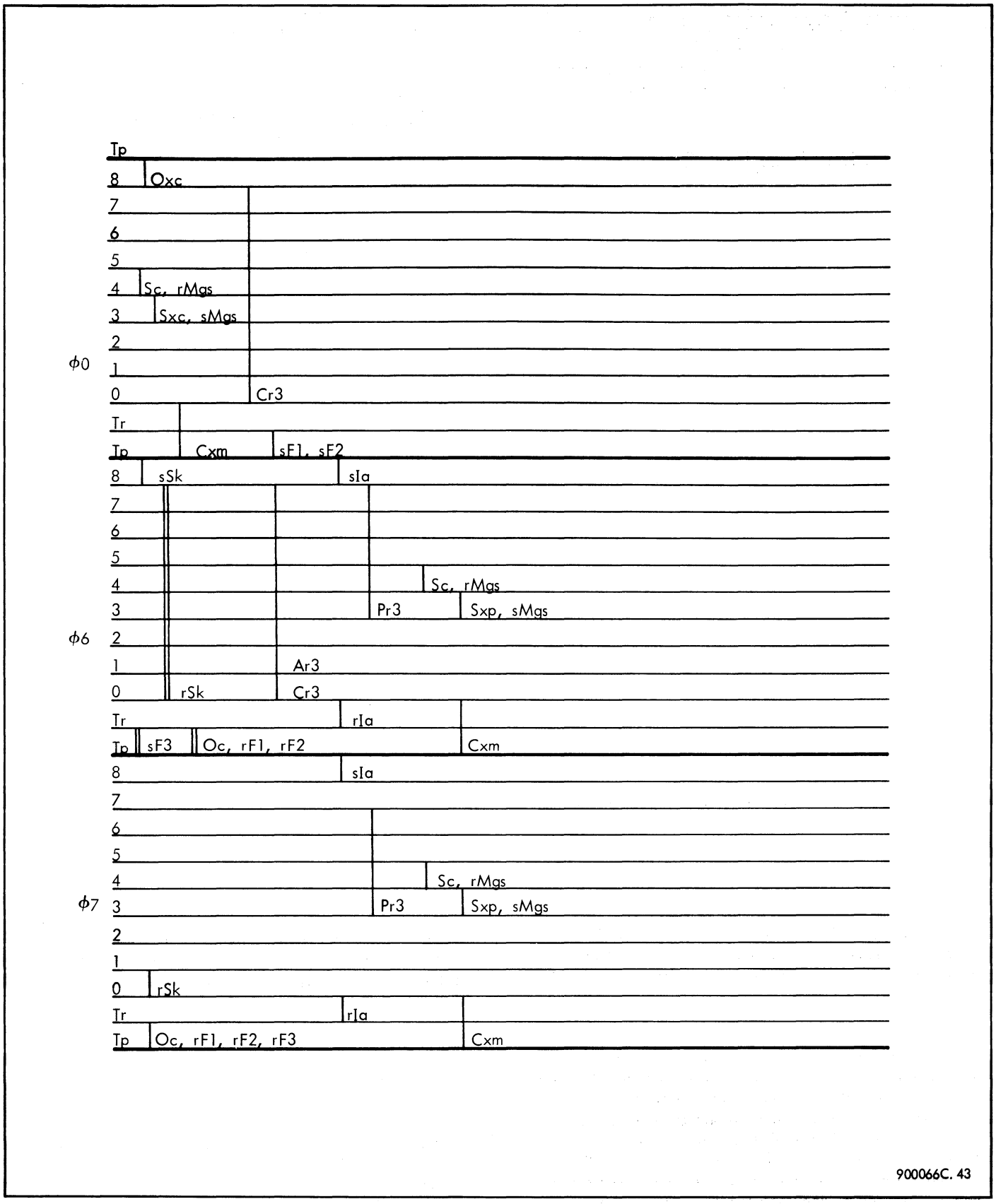


Figure 3-43. SKE Instruction, Timing Diagram

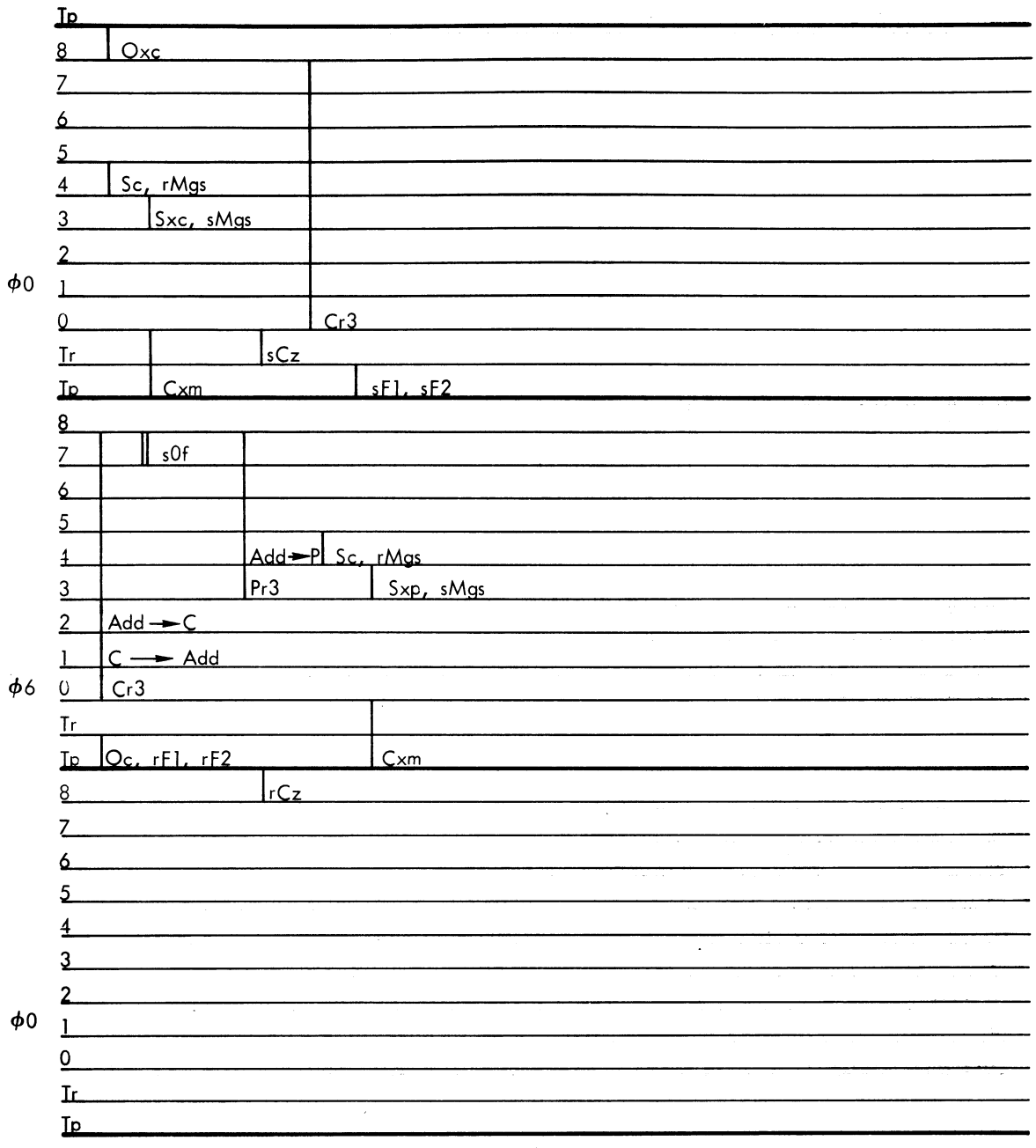


Figure 3-44. BRR Instruction, Timing Diagram

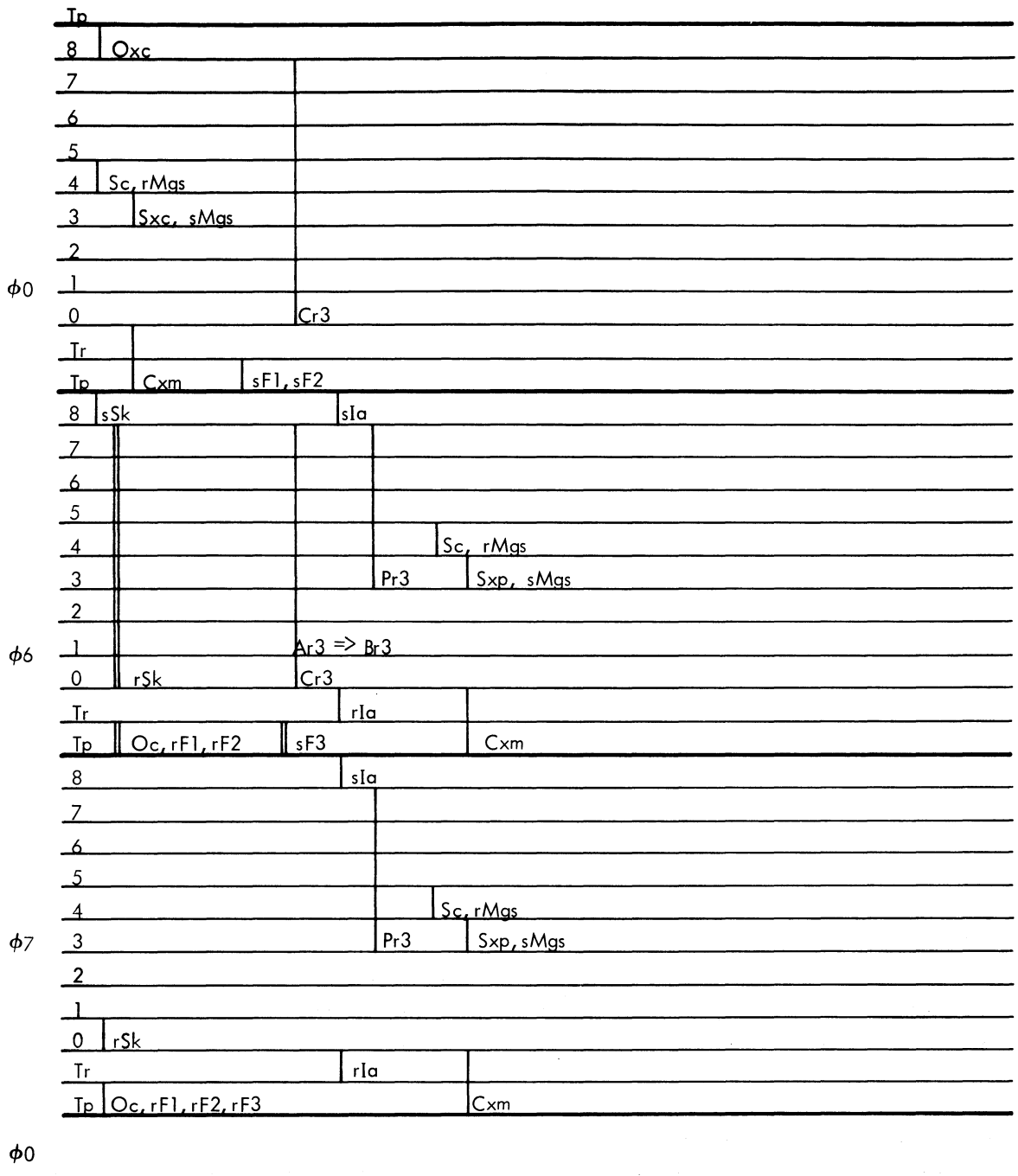


Figure 3-45. SKB Instruction, Timing Diagram

register. The logic then proceeds to phase 6, and operand parity is checked during recirculation of the C-register.

3-358 Skip flip-flop Sk is set at T8 of phase 6. The state of this flip-flop at the end of phase 6 determines whether or not phase 7 is entered.

3-359 While the B- and C- registers are recirculating during phase 6, B21 through B23 are compared with C21 through C23 as follows:

$$B21\ C21 + B22\ C22 + B23\ C23$$

If this expression is true at any time, a pair of ONE's has been detected in corresponding bits of the B- and C-registers, and Sk is reset. The contents of the P-register, which have been increased by one during phase 6 by carry flip-flop Ia, are placed in the S-register. This P+1 instruction is placed in the C-register at Tp. A Tp End Sk gate resets F1 and F2, and the P+1 instruction is executed at the following phase 0.

3-360 If the OR expression in paragraph 3-359 remains false throughout the recirculation of the B- and C-registers, a pair of ONE's is not present in corresponding bit positions, and Sk remains set. Resetting of F1 and F2 is inhibited, and F3 is set, placing the phase counter in phase 7. The instruction in the C-register is disregarded, the contents of the P-register are again increased by one, and this P+2 address is placed in the S-register. The instruction contained in the P+2 location is placed in the C-register at Tp, ready for execution at the following phase 0.

3-361 Skip if Memory Negative (SKN, Code 53)

Ø7

Ø0 - Ø6 ------ Ø0 (See figure 3-46)

3-362 The SKN instruction checks the state of bit 0 of the contents of the effective memory location and skips the next instruction in sequence if M0 contains a ONE. If M0 contains a ZERO, the next instruction in sequence is executed.

3-363 The original instruction parity is checked while Cr3 is active during phase 0. The contents of the effective address is transferred from the C-register to the S-register at T3 and at Tp the contents of the effective memory location are placed in the C-register. Phase 6 is entered and operand parity is checked during recirculation of the C-register.

3-364 Carry flip-flop Ia is set at T8 of phase 6 so that the contents of the P-register may be increased by one to make available the next instruction in memory sequence. The P+1 address is placed in the S-register at T3, but is disregarded if a skip is to be performed. At Tr of phase 6, the opcode, timing, and phase are gated with C0 to set the skip flip-flop if C0 contains a ONE.

3-365 A true signal from Sk drives the Tp End Sk expression false, causing F3 to be set and inhibiting the resetting of F1 and F2. The phase counter is therefore placed in phase 7, and the skip function is carried out. Carry flip-flop Ia is set, so that the contents of the P-register are again increased by one. This P+2 address is placed in the S-register and at Tp the contents of the P+2 memory location are placed in the C-register, ready for execution

3-336 If C0 contains a ZERO at Ø6 Tp, the O-register is cleared, F1 and F2 are reset, and the next instruction in sequence, in the C-register, is executed at the following phase 0.

3-367 Subtract Memory From A (SUB, Code 54)
Ø0 - Ø6 - Ø0 (See figure 3-47)

3-368 The SUB instruction subtracts the contents of the effective memory location from the contents of the A-register and places the result in the A-register. The overflow indicator is set if a carry is propagated out of the most significant bit. The carry from bit 0 of the addition is placed in bit position 0 of the index register.

3-369 The original instruction parity is checked from T7 to T0 of phase 0 during Cr3. The address from the contents of the memory location is transferred from the C-register to the S-register at T3, and at Tp the operand is loaded into the C-register. At Tr of phase 0, the full adder carry flip-flop, Cz, is set in order to obtain the two's complement of the contents of the C-register in the following phase. The one's complement of the contents of the C-register is to be placed in the adder during phase 6; the carry adds a one to the one's complement to form the two's complement.)

3-370 At T8 of phase 6, decoding of phase, timing, and opcode produces a Ck signal, which complements the C-register by resetting flip-flops C0-23 if they are set and setting them if they are reset. The outputs of the three least significant flip-flops of the A- and C-registers are presented to the serial adder under the control of Cr3 and Ar3 and the outputs of the adder are placed in the A-register. The A-register now contains the sum of the contents of the A-register and the two's complement of the contents of the effective memory location.

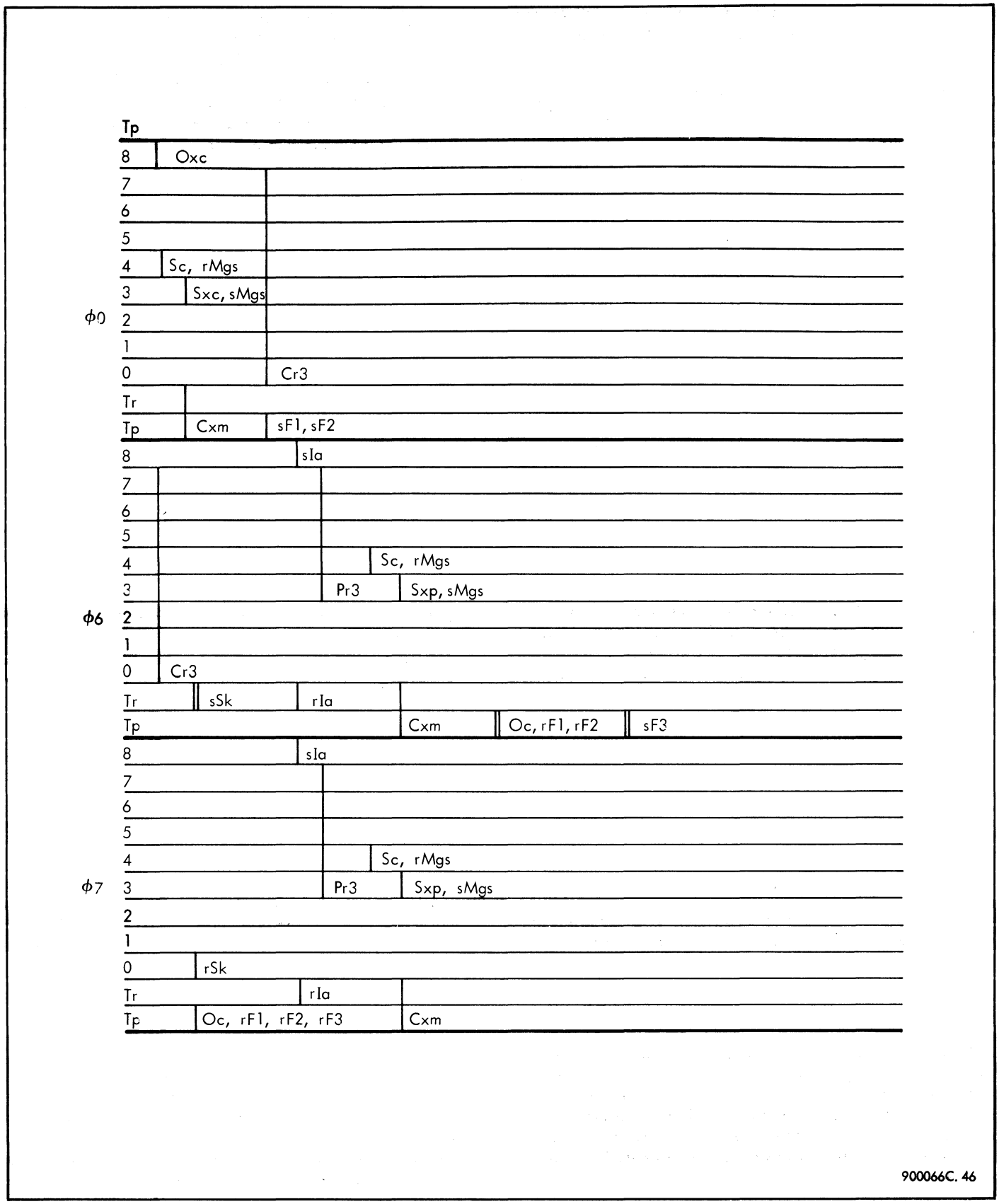


Figure 3-46. SKN Instruction, Timing Diagram

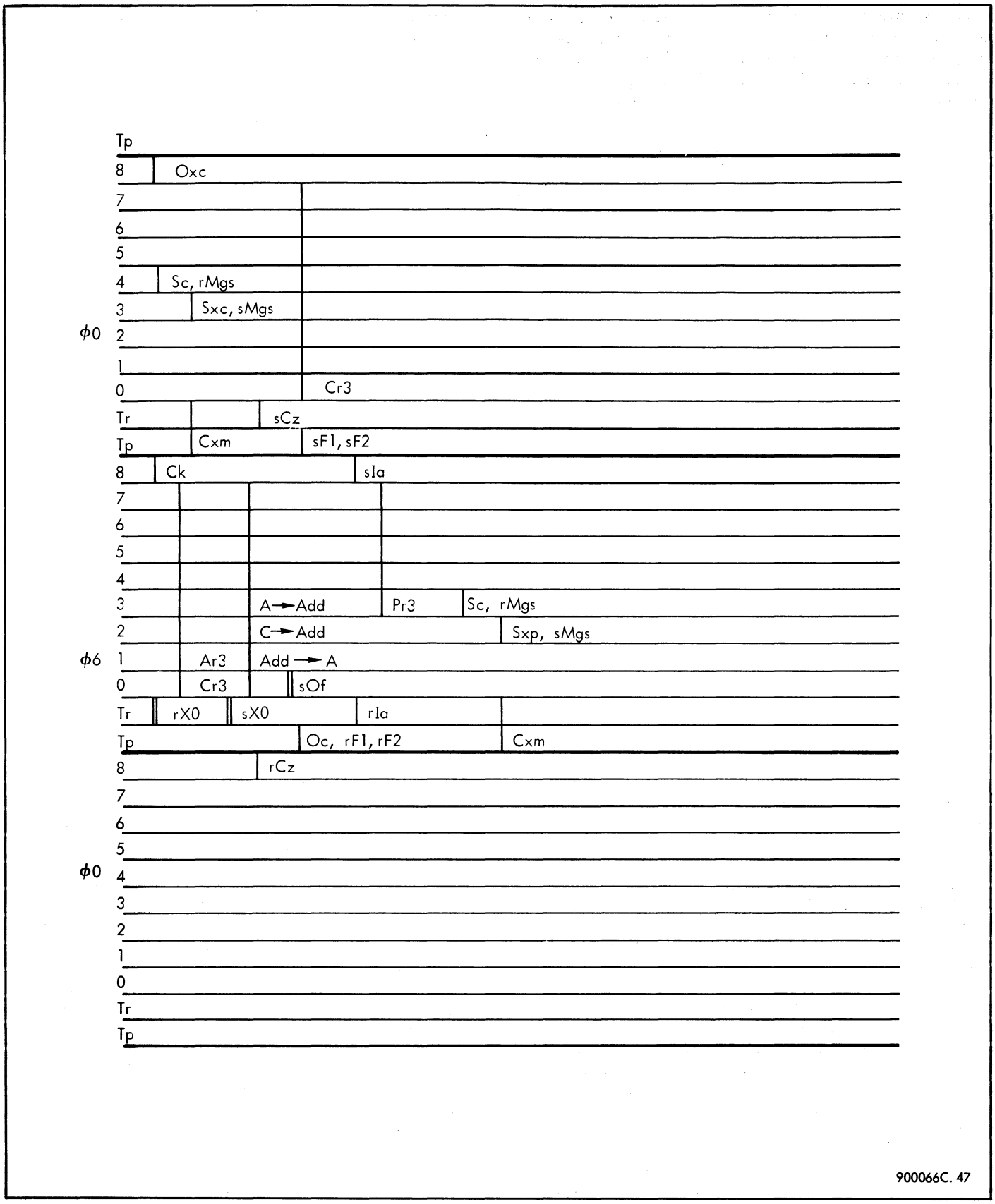


Figure 3-47. SUB Instruction, Timing Diagram

3-371 An overflow out of the most significant bit occurs in subtraction if the sign of the subtrahend after complementing is the same as the sign of the minuend, and the sign of the result is opposite. The logic of this comparison is mechanized in the Ofe signal, which sets overflow flip-flop Of at T0:

$$\overline{Ofe} = Add\ 1\ Xz1\ Yz1 + \overline{Add\ 1}\ \overline{Xz1}\ \overline{Yz1} + Xz1 \oplus Yz1$$

where

$Xz1 = \overline{A21}$ (minuend sign at T0),

$Yz1 = \overline{C21}$ (inverted subtrahend sign at T0),

and

Add 1 = sign of the result at T0

At Tr time of phase 6, the carry from bit 0 of the addition is placed in bit position 0 of the index register:

$$X1d = \left[(O1\ \overline{O2}\ O3\ O4\ O6)\ Tr\right] Cz$$

where X1d is the first delay element of the index register.

3-372 During phase 6, operand parity is checked. The contents of the program counter are increased by one because carry flip-flop Ia is set. This P+1 address is placed in the S register at T2, and the P+1 instruction is transferred to the C-register at Tp. Phase 0 is entered, and Cz is reset at T8.

3-373 Add Memory to A (ADD, Code 55) Ø0 - Ø6 - Ø0
(See figure 3-48)

3-374 The ADD instruction adds the contents of the effective memory location to the A-register and places the result in the A-register. The overflow indicator is set if an overflow occurs, and the carry from bit 0 is placed in bit position 0 of the index register.

3-375 The original instruction parity is checked from T7 to T0 of phase 0 while Cr3 is active. The operand is read from memory during phase 0, and operand parity is checked during phase 6 while the C-register is circulating under the control of Cr3.

3-376 During phase 6, the three least significant bits of the A- and C-registers are presented to the full adder as addend and augend, and the adder outputs are placed in the A-register. Signals Ar3 and Cr3 keep the two registers recirculating while the addition is taking place. From T7 to T0, carry flip-flop Cz is set whenever a carry into the next stage of addition occurs.

3-377 In addition, an overflow occurs out of the most significant bit is the signs of the addend and augend are the same and the sign of the result is opposite. The overflow logic compares the most significant (sign) bit of the two adder inputs and the sign of the result and generates an Ofe signal in case of any overflow:

$$\overline{Ofe} = Add\ 1\ Xz1\ Yz1 + \overline{Add\ 1}\ \overline{Xz1}\ \overline{Yz1} + Xz1 \oplus Yz1$$

where

$Xz1 = \overline{A21}$ (minuend sign at T0),

$Yz1 = \overline{C21}$ (inverted subtrahend sign at T0),

and

Add 1 = sign of the result at T0

If a carry is present after the last stage of addition, Cz is true at Ø6 Tr, and the carry is placed in bit position 0 of the index register:

$$X1d = \left[(O1\ \overline{O2}\ O3\ O4\ O6)\ Tr\right] Cz$$

3-378 With carry flip-flop Ia set at T8, one is added to the contents of the P-register during recirculation, this P+1 address is placed in the S-register at T3, and the next instruction in sequence is placed in the C-register at Tp of phase 6.

3-379 Subtract With Carry (SUC, Code 56) Ø0 - Ø6 - Ø0
(See figure 3-49)

3-380 The SUC instruction subtracts the contents of the effective memory location and a carry, if present, from bit 0 of the index register from the contents of the A-register and places the result in the A-register. This instruction performs multiple-precision subtractions by subtracting the most significant halves of two numbers after the least significant halves have been subtracted with a SUB instruction. The carry from the SUB instruction is preserved in bit 0 of the index register, and the carry from the SUC instruction is placed in bit position 0 of the index register. The overflow indicator is automatically cleared before a SUC instruction is executed, because overflow resulting from the subtraction of the least significant halves of the numbers is not meaningful. At the end of the SUC instruction, the overflow indicator is set if an overflow has occurred from bit 0.

3-381 The original instruction parity is checked while the C-register is recirculating during phase 0, and the operand which is placed in the C-register at the end of phase 0 is checked for parity during Cr3 of phase 6. At Tr time of phase 0, the full adder carry flip-flop, Cz, is set if bit 0 of the index register contains a ONE. Bit 0 of the index register contains the carry from the subtraction in the SUB instruction, which placed a ONE in Cz in order to form the two's complement from the one's complement of the subtrahend. The carry flip-flop is set as follows:

$$sCz = Xw1\ O4\ O5\ (Tr\ Ø0)$$

since Xw1 at this time contains bit 0 of the index register.

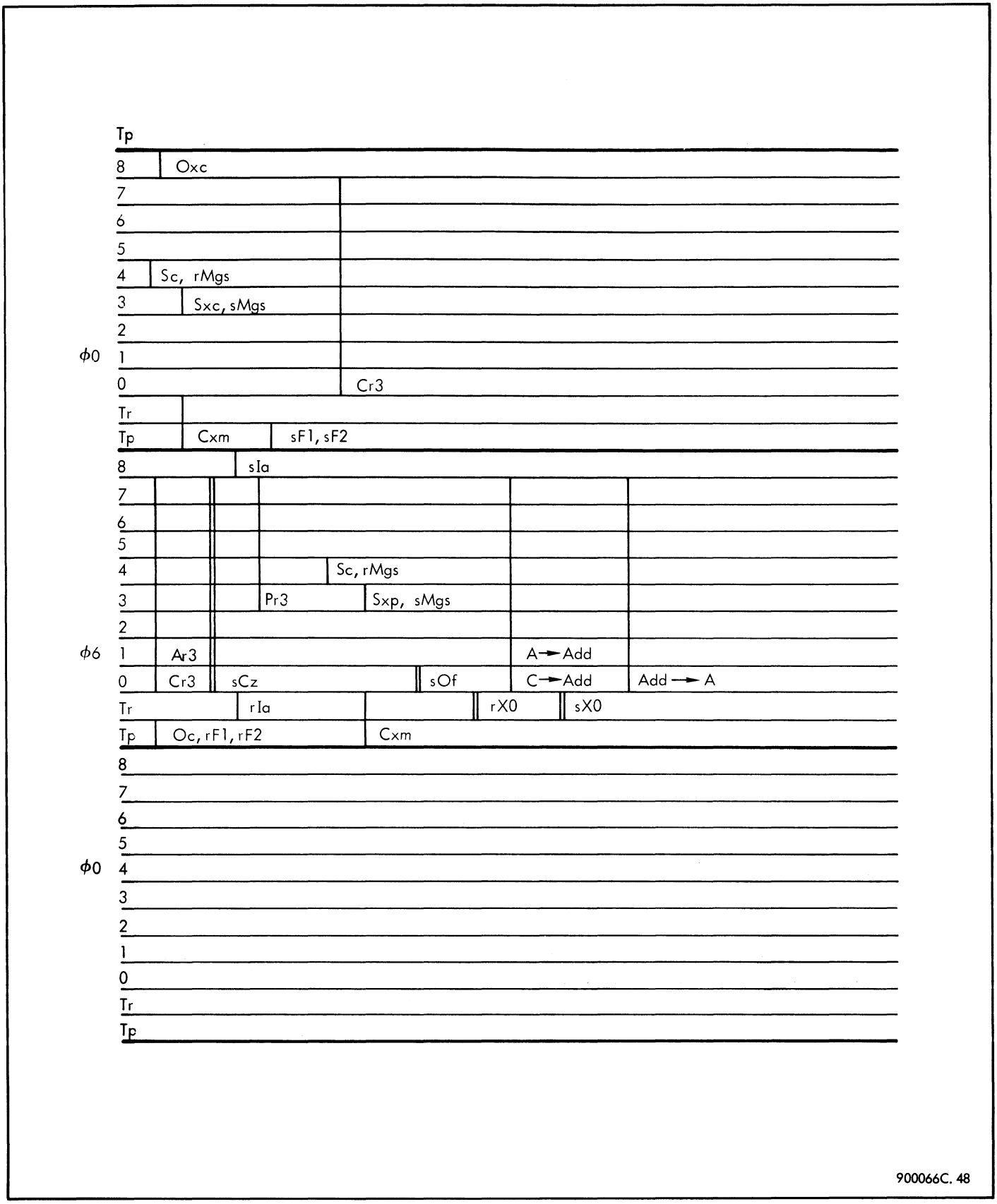


Figure 3-48. ADD Instruction, Timing Diagram

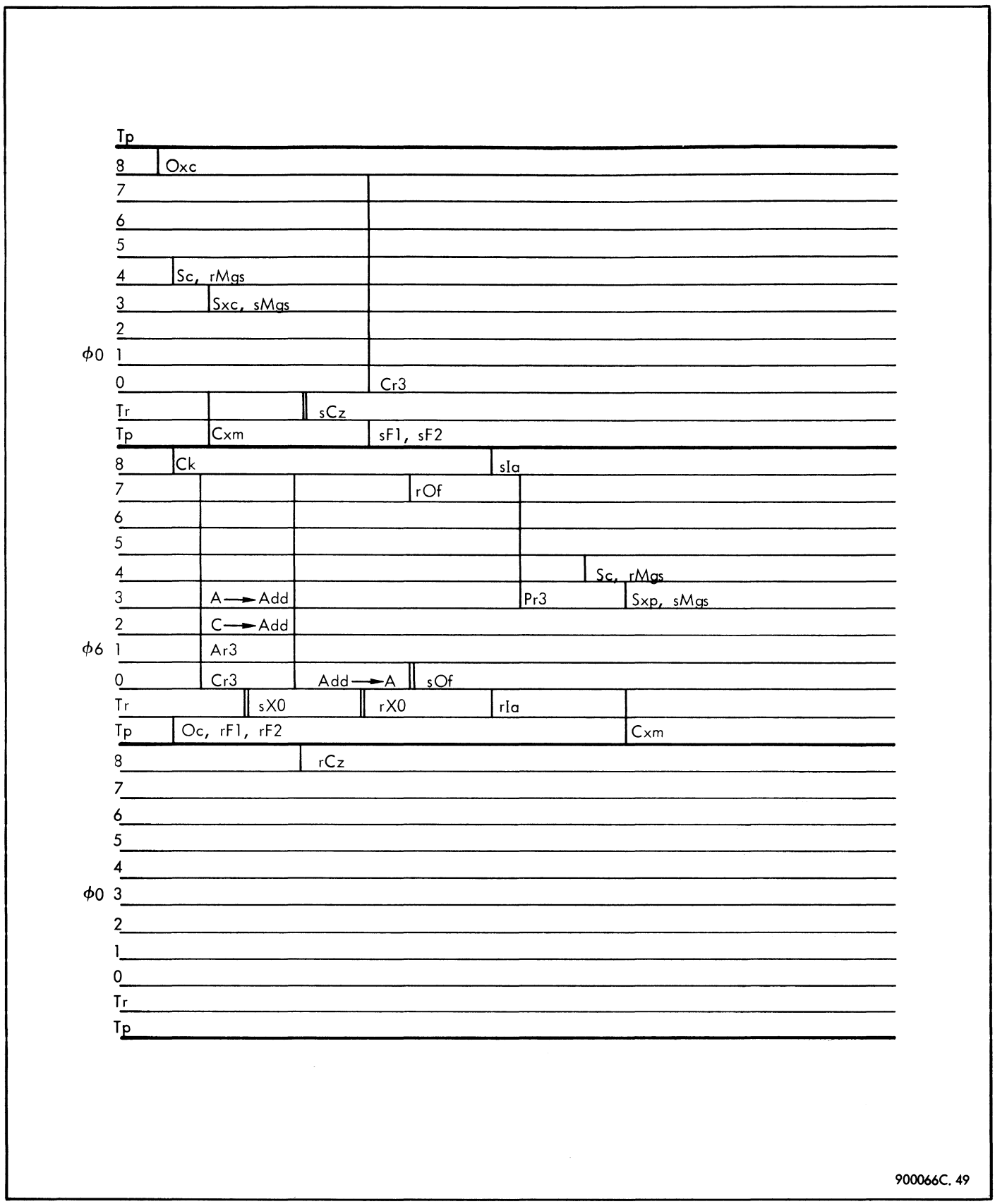


Figure 3-49. SUC Instruction, Timing Diagram

3-382 Phase 6 is entered, and at T8, decoding of opcode with phase and timing produces a Ck signal, which complements the least significant octal digit of the C-register by resetting flip-flops C0 through C23 if they are set and setting them if they are reset.

3-383 The outputs of the three least significant flip-flops of the A- and C-registers are presented to the serial adder under the control of Cr3 and Ar3, and the outputs of the adder are placed in the A-register. The A-register now contains the difference between A and C, obtained by summing the contents of the A-register and the two's complement of the contents of the C-register.

3-384 An overflow out of the most significant bit occurs in subtraction if the sign of the subtrahend after complementing is the same as the sign of the minuend, and the sign of the result is opposite. The logic of this comparison is mechanized in the Ofe signal, which sets overflow flip-flop Of at T0:

$$\overline{Ofe} = Add\ 1\ Xz1\ Yz1 + \overline{Add\ 1}\ \overline{Xz1}\ \overline{Yz1} + Xz1 \oplus Yz1$$

where

$Xz1 = \overline{A21}$ (minuend sign at T0)

$Yz1 = \overline{C21}$ (inverted subtrahend sign at T0)

and

Add 1 = Sign of result at T0

At Tr time of phase 6, the carry from bit 0 of the addition is placed in bit position 0 of the index register:

$$X1d = \left[(O1\ \overline{O2}\ O3\ O4\ O6)\ Tr\right] Cz$$

where X1d is the first delay element of the index register.

3-385 The contents of the P-register are increased by one during phase 6 by Pr3 and carry flip-flop Ia. The P+1 address placed in the S-register at T3 by Sxp, and this next instruction in sequence is transferred from the effective memory location to the C-register at Tp. Carry flip-flop Cz is reset at the following T8.

3-386 Add With Carry (ADC, Code 57) Ø0 - Ø6 - Ø0 (See figure 3-50)

3-387 The ADC instruction adds the contents of the A-register to the contents of the effective memory location and a carry, if present, from bit zero of the index register and places the sum in the A-register. This instruction performs multiple-precision addition by adding the most significant halves of two numbers after the least significant halves have been added with an ADD instruction. The ADC instruction uses the ADD instruction carry, which has been preserved in bit 0 of the index register. The overflow indicator is unconditionally cleared before an ADC instruction is executed, because overflow resulting from the addition of the least significant halves of the numbers is not meaningful. At the end of the ADC instruction, the overflow indicator is set if an overflow has occurred from bit 0.

3-388 The original instruction parity is checked during Cr3 of phase 0. The parity of the operand placed in the C-register at Tp of phase 0 is checked during recirculation of the C-register in phase 6.

3-389 The contents of the A- and C-registers are presented to the inputs of the full adder during phase 6, and a carry from a previous addition is forced into the first stage of addition if bit 0 of the index register contains a ONE. A carry sets full adder carry flip-flop Cz at Tr time:

$$sCz = Xw1\ O4\ O5\ (Tr\ Ø0)$$

since Xw1 at this time contains bit 0 of the index register.

3-390 The outputs of the full adder are presented to the inputs to A0-2. In the last stage of addition, if the signs of the two numbers added are alike but the sign of the result is opposite, an overflow is indicated. The overflow indicator is set by Ofe:

$$\overline{Ofe} = Add\ 1\ XZ1\ Yz1 + \overline{Add\ 1}\ \overline{Xz1}\ \overline{Yz1} + Xz1 \oplus Yz1$$

where

$Xz1 = \overline{A21}$ (minuend sign at T0),

$Yz1 = \overline{C21}$ (inverted subtrahend sign at T0),

and

Add 1 = sign of the result at T0

If a carry is present after the last stage of addition, Cz is true at Ø6 Tr, and the carry is placed in bit position 0 of the index register:

$$X1d = \left[(O1\ \overline{O2}\ O3\ O4\ O6)\ Tr\right] Cz$$

where X1d is the first delay element of the index register.

3-391 A ONE in carry flip-flop Ia forces an increment of one into the P-register; this next instruction address is placed in the S-register at T3; and the contents of the memory location are transferred to the C-register at Tp for execution at the following phase 0.

3-392 Reduce Memory, Skip if Negative (SKR, Code 60) Ø0 - Ø4 - Ø7 - Ø0 (See figure 3-51)

3-393 The SKR instruction reduces the contents of the effective memory location by one, places the result in the same location, and if the result is negative, the next instruction in sequence is skipped. If the result is positive or zero, the next instruction in sequence is executed.

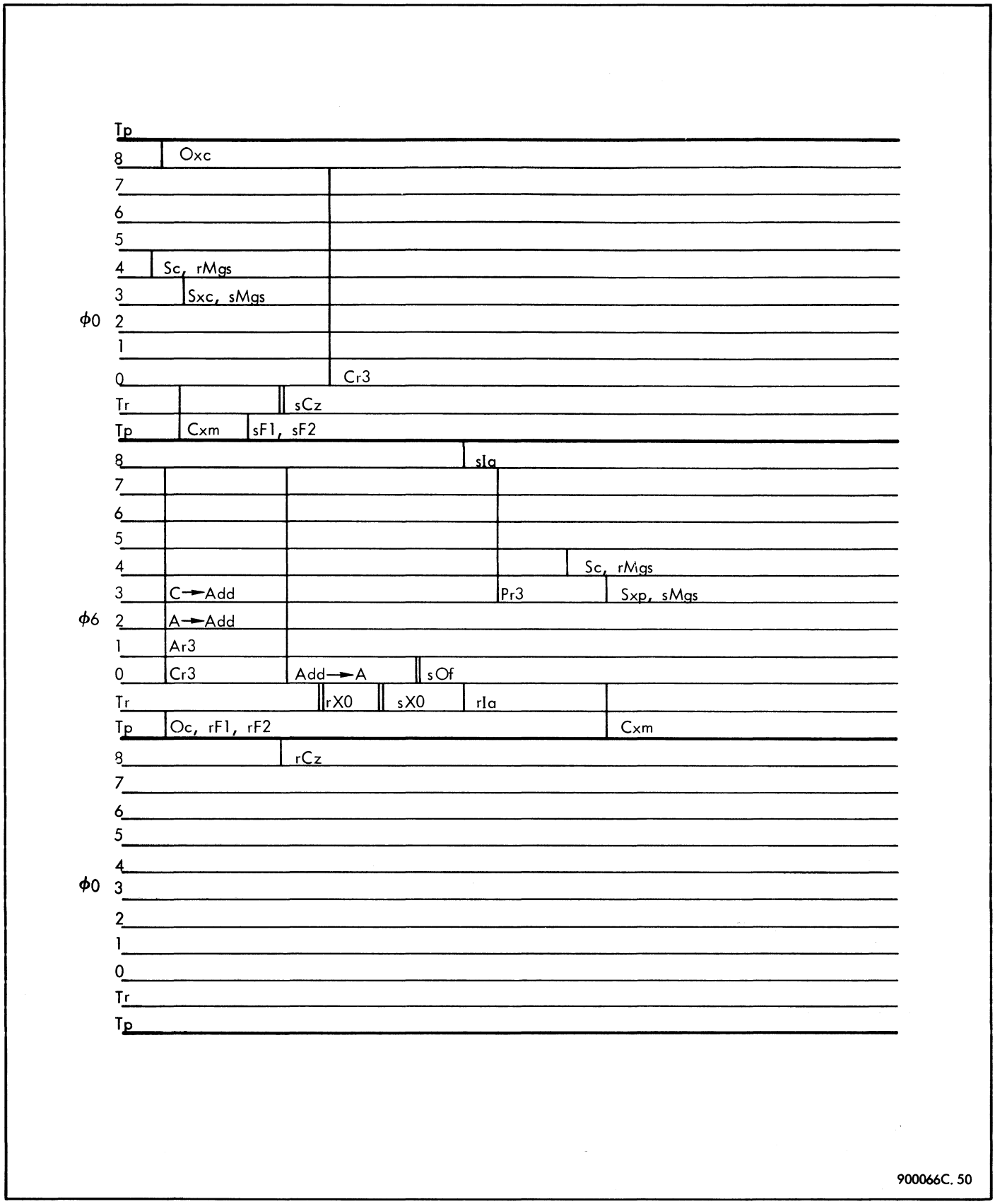


Figure 3-50. ADC Instruction, Timing Diagram

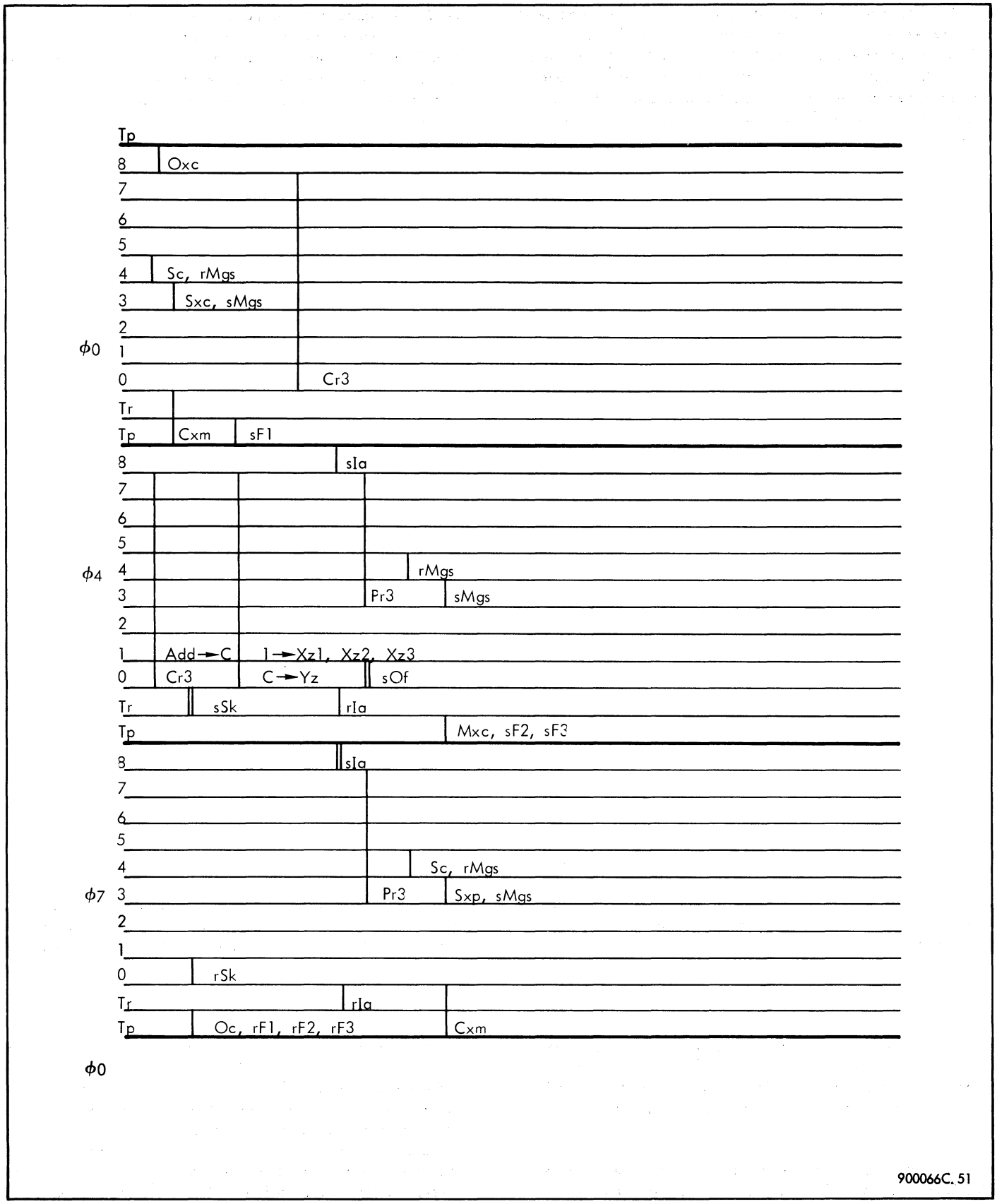


Figure 3-51. SKR Instruction, Timing Diagram

3-394 The original instruction parity is checked in phase 0 during recirculation of the instruction in the C-register. Parity of the operand is checked in phase 4 while the data in the C-register is being shifted into the adder, and parity is generated for the new information in phase 4 while the information is being shifted into the C-register from the adder.

3-395 In phase 0, the opcode is placed in the O-register, the operand address is transferred into the S-register, and the contents of the effective memory location are placed in the C-register. The phase counter then advances to phase 4 when F1 is set.

3-396 From T7 to T0 of phase 4, the outputs of the C-register are presented to the Yz inputs of the full adder and 111 (the two's complement of 1) is placed in Xz1-3. The addition of these two inputs effectively subtracts 1 from the contents of the C-register. The outputs of the adder are shifted into the C-register, and at Tp the contents of the C-register are transferred to memory by Mxc.

3-397 The overflow flip-flop is set at T0 of phase 4 if the initial contents of memory were 40000000. Refer to paragraph 3-390 for the overflow flip-flop equations. Skip flip-flop Sk is set if the contents of memory (now in the C-register) are negative after subtracting one. Since the sign is in bit position 0, the state of C0 is checked:

$$sSk = C0\ Tr\ O1\ \overline{O5}\ \overline{O6}\ \emptyset 4$$

During T8 through T3 of phase 4, the contents of the P-register are increased by one because carry flip-flop Ia is set. The Ia flip-flop is reset at Tr of phase 4 and remains reset if the memory contents minus one are positive. In this case, the phase counter advances to phase 7 and the contents of the P+1 address are placed in the C-register for execution.

3-398 If C0 contains a ONE, Sk is set and causes Ia to be set at T8 of phase 7. When Pr3 is active in phase 7, the contents of the P-register are increased by one, making the P+2 address available for transfer to the S-register. This P+2 instruction is placed in the C-register at Tp of phase 7, ready for execution in the following phase 0. Flip-flops Sk and Ia are reset in phase 7.

3-399 Memory Increment (MIN, Code 61) Ø0 - Ø4 - Ø7 - Ø0 (See figure 3-52)

3-400 The MIN instruction increases the contents of the effective memory location by one and places the resulting sum in the same location. The overflow flip-flop is set if an overflow occurs in the addition of the last octal digit.

3-401 The original instruction parity is checked during Ø0 Cr3. Parity is checked on the operand during phase 4 when the operand is in the C-register and Cr3 is effective.

3-402 The opcode is loaded in the O-register at Ø0 T8, the effective memory address in the instruction is placed in the S-register at T3, and the operand is loaded in the C-register at Tp. Carry flip-flop Cz for the full adder is set at Tr to force a +1 into the adder. The logic then proceeds to phase 4.

3-403 During phase 4, the C-register outputs are presented to the full adder, which increases the contents of C by one because Cz is set. Parity is generated for the memory word while the adder outputs are going into C by setting or resetting C24 whenever an octal going into the C-register an odd number of ONE's. Carry flip-flop Cz is set whenever a carry is needed into the next octal. The overflow flip-flop, Of, is set when the signs of the numbers added are alike and the sign of the result if different. An overflow occurs only when the contents of M are 37777777. The next instruction address is placed in the P-register by setting Ia and thereby increasing the contents of the P-register by one during Pr3 in phase 4. At Tp the new word in the C-register is stored in memory by Mxc. The phase counter then advances to phase 7.

3-404 In phase 7, the next instruction address in the P-register is loaded into the S-register, and this instruction is loaded into the C-register at Tp. The phase counter is reset to phase 0, and the next instruction is ready for execution.

3-405 Exchange Memory and A (XMA, Code 62) Ø0 - Ø4 - Ø7 - Ø0 (See figure 3-53)

3-406 The XMA instruction loads the contents of the effective memory location into the A-register and stores the contents of the A-register in the effective memory location. The original instruction parity is checked during Ø0 Cr3. The opcode in the C-register is transferred to the O-register, and the operand address is loaded into the S-register. The operand is loaded into the C-register at Tp by Cxm, and the parity of the operand is checked during phase 4.

3-407 During phase 4, the next instruction address is placed in the P-register by setting carry flip-flop Ia to increase the contents of the P-register by one during Pr3. The contents of the A-register are shifted into the C-register for transfer to memory, and the memory contents in the C-register are shifted into the A-register. The exchange between A and M having taken place, the contents of the C-register are now transferred to memory by Mxc at Tp. All inputs to the S-register are inhibited during phase 4 so that the contents of the A-register can be stored in the location from which the contents of M were taken. Parity is generated for the new data to be stored as the data is shifted from A into C. The phase counter advances to phase 7. In phase 7, the next instruction in sequence, which is in the P-register, is transferred into the S-register. The instruction is loaded into the C-register at Tp for execution at the following phase 0.

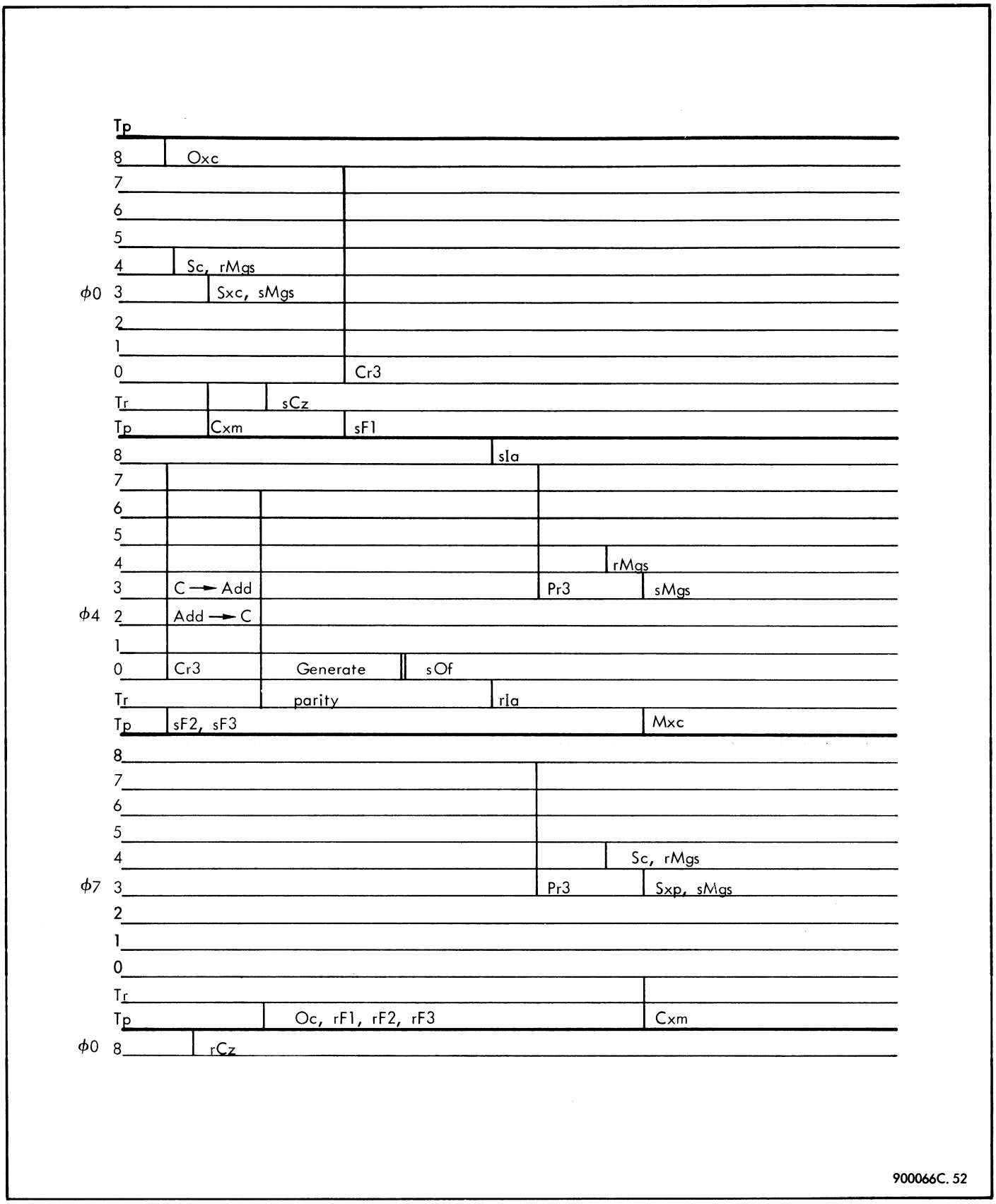


Figure 3-52. MIN Instruction, Timing Diagram

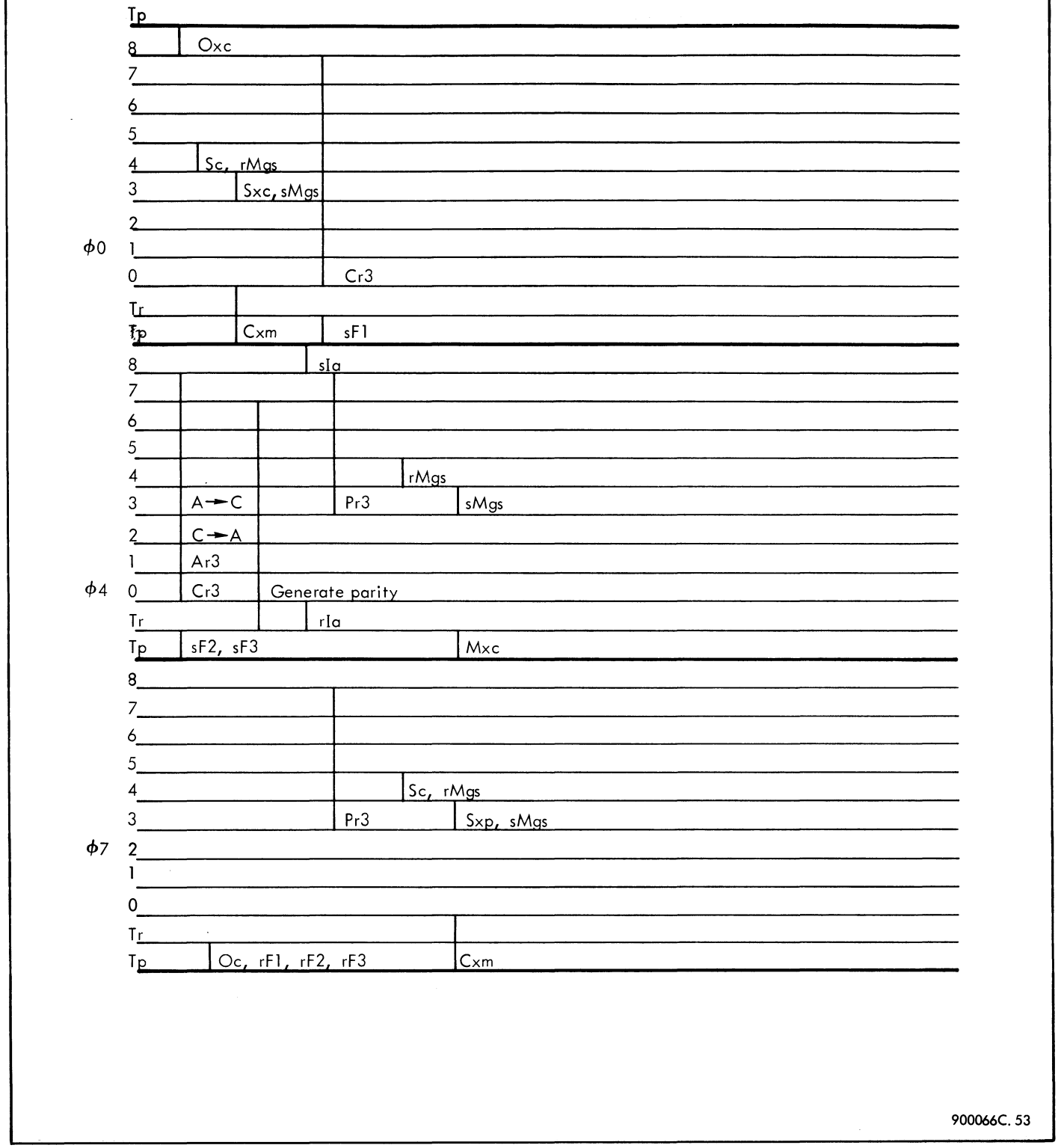


Figure 3-53. XMA Instruction, Timing Diagram

3-408 Add A to Memory (ADM, Code 63) Ø0 - Ø4 - Ø7 - Ø0 (See figure 3-54)

3-409 The ADM instruction adds the contents of the A-register to the contents of the effective memory location and stores the result in the same location.

3-410 The original instruction parity is checked while the C-register is recirculating in phase 0. The operand, which is accessed at Tp of phase 0, is checked for parity in the following phase 4.

3-411 The contents of the A- and C-registers are presented to the full adder in phase 4, and as the adder outputs are shifted into the C-register, parity is generated for the new data word to be stored in memory. The S-register receives no inputs during phase 4; hence, the memory cycle initiated in this phase stores the new data in the same location from which the original memory contents were taken. The contents of the P-register are increased by one by setting carry flip-flop Ia, making the next instruction in sequence available in the P-register for access in the following phase. The overflow flip-flop is set at T0 if the signs of the two numbers added are alike and the sign of the sum is opposite.

3-412 In phase 7 the P-register recirculates without change while the P+1 address is shifted into the S-register for access at Tp.

3-413 Multiply (MUL, Code 64) Ø0 - Ø3 - Ø3 - Ø7 - Ø0 (See figure 3-55)

3-414 The MUL instruction multiplies the contents of the effective memory location by the contents of the A-register and places the product in the A- and B-registers, with the more significant portion in the A-register. The partial product is formed in BA. The sign of the end result is contained in A0; B0 is just another bit of the product, and B23 is zero. The original contents of the B-register do not affect the operation; B is cleared during phase 0.

3-415 The original instruction parity is checked during phase 0 while Cr3 is active. Flip-flop A00 is set if the contents of the A-register are negative, that is, A0 contains a ONE:

$$sA00 = A0\ T7\ Ø0 + \ldots$$

The B-register is cleared by Ar3 and $\overline{Bnr}$:

$$\overline{Bnr} = Ø0\ \overline{O3}\ O4\ \overline{O5}\ \overline{O6} + \ldots$$

Shifting the B-register right with Ar3 loads the register with ZERO's. The operand is accessed at Tp, and K0 is set:

$$sK0 = (Tp\ \overline{Ia}\ Ø0)\ \overline{O3}\ O4\ \overline{O6} + \ldots$$

The opcode bits are decoded and used for advancing the phase counter from phase 0 to phase 3 at Tp.

3-416 The execution of a MUL instruction requires three more cycles after phase 0: phase 3, phase 3, and 2 pulse periods in phase 7. The 24-bit multiplication process requires 24 iterations, controlled by K0, which is set at Tp of phase 0 and reset at T7 of phase 7. The 24 pulse periods between the setting and resetting of K0 provide the timing for the 24 iterations.

3-417 The block diagram in figure 3-56 shows the status of the pertinent logic at Ø0 Tp. The right shift adder is used to form the partial products, since it allows a 24-bit addition in one pulse period. The end result of a right shift add operation is C plus (B plus Bc), the sum being stored in BA. Refer to paragraph 3-109 for a detailed explanation of the operation of the right shift adder.

3-418 In normal multiplication, the multiplicand is shifted to the left one bit position for each bit of the multiplier and added to the partial product when required. The same function is performed in the 930 Computer by shifting the partial product to the right one bit position for each bit of the multiplier, adding the multiplicand to the partial product when required. Whether or not the multiplicand is added is determined by the state of A23, which is the effective bit of the multiplier during any one iteration.

3-419 As the A and B plus Bc registers are shifted right for each iteration, the multiplier is observed one bit at a time by testing A23. If A23 contains a ZERO, it is necessary to simply shift the partial product right without adding the multiplicand. A Br1 signal controls this shift in the B-register.

$$Br1 = (F2\ F3\ \overline{O5}\ \overline{O6}\ K0)\ \overline{A23} + \ldots$$

The A-register shifts right with Ar1, which is true regardless of the state of A23. A right shift one is equivalent to adding ZERO's to the partial product.

3-420 If A23 contains a ONE, it is necessary to add the multiplicand, in the C-register, to the partial product already formed and shift the sum one bit to the right. This shift with addition is performed by Rsa:

$$Rsa = (F2\ F3\ \overline{O5}\ \overline{O6}\ K0)\ A23$$

Since K0 determines the number of iterations, both Br1 and Rsa are qualified by K0.

3-421 At T7 of phase 7, the sign bit of the multiplier is in A23. If A23 contains a ZERO, the multiplier is positive and Ar1 and Br1 shift the A- and B-registers to the right unchanged. If A23 contains a ONE, the multiplier is a negative number. Since a binary point is assumed between bit 0 and bit 1 of the original data word, bit 0 of a negative number represents a minus one, and bits 1 through 23 represent a positive fraction. In the first 23 iterations of multiply, the multiplicand in the C-register has been multiplied by positive one or zero and added to the partial product. The last iteration with a negative multiplier must

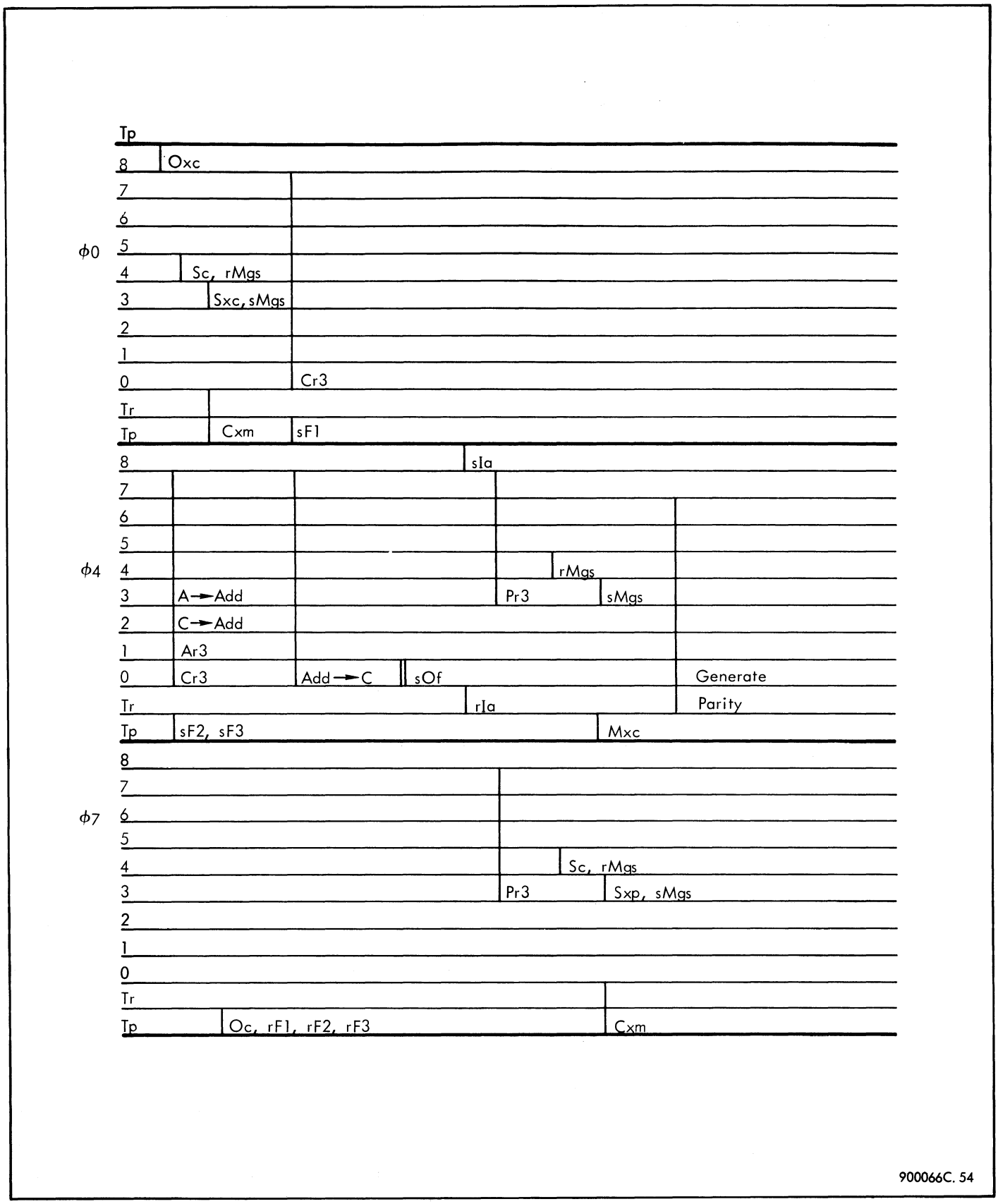


Figure 3-54. ADM Instruction, Timing Diagram

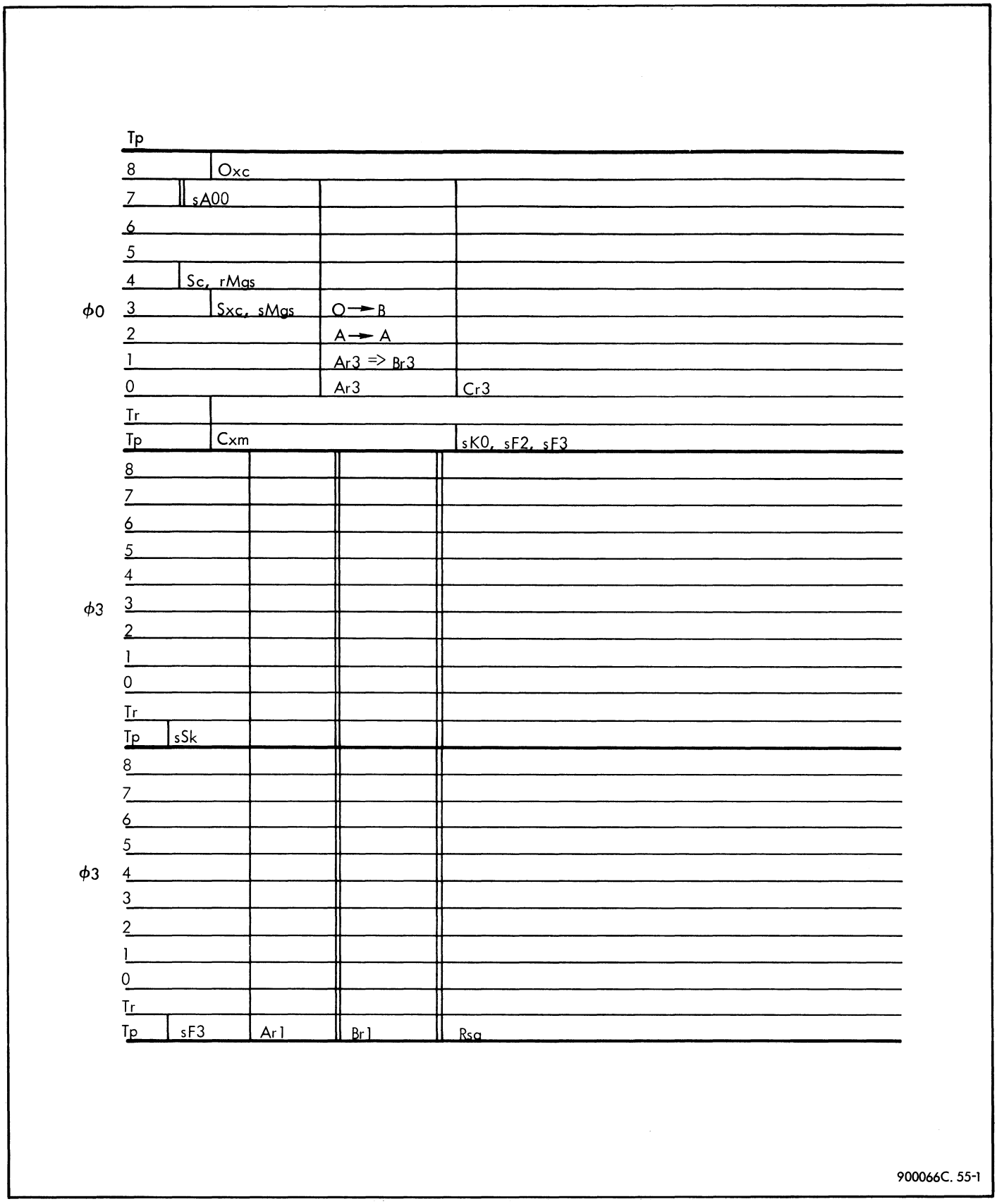


Figure 3-55. MUL Instruction, Timing Diagram (Sheet 1 of 2)

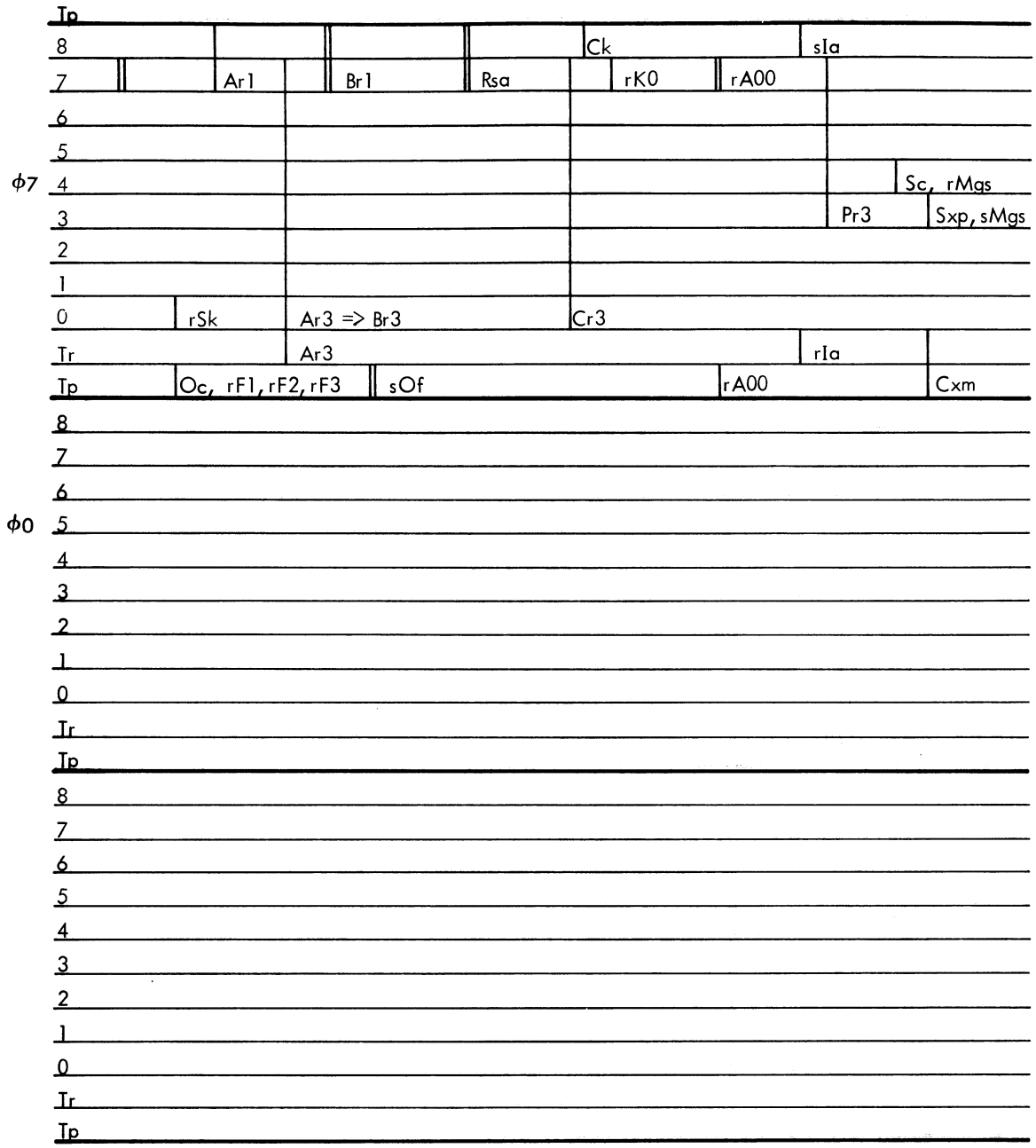


Figure 3-55. MUL Instruction, Timing Diagram (Sheet 2 of 2)

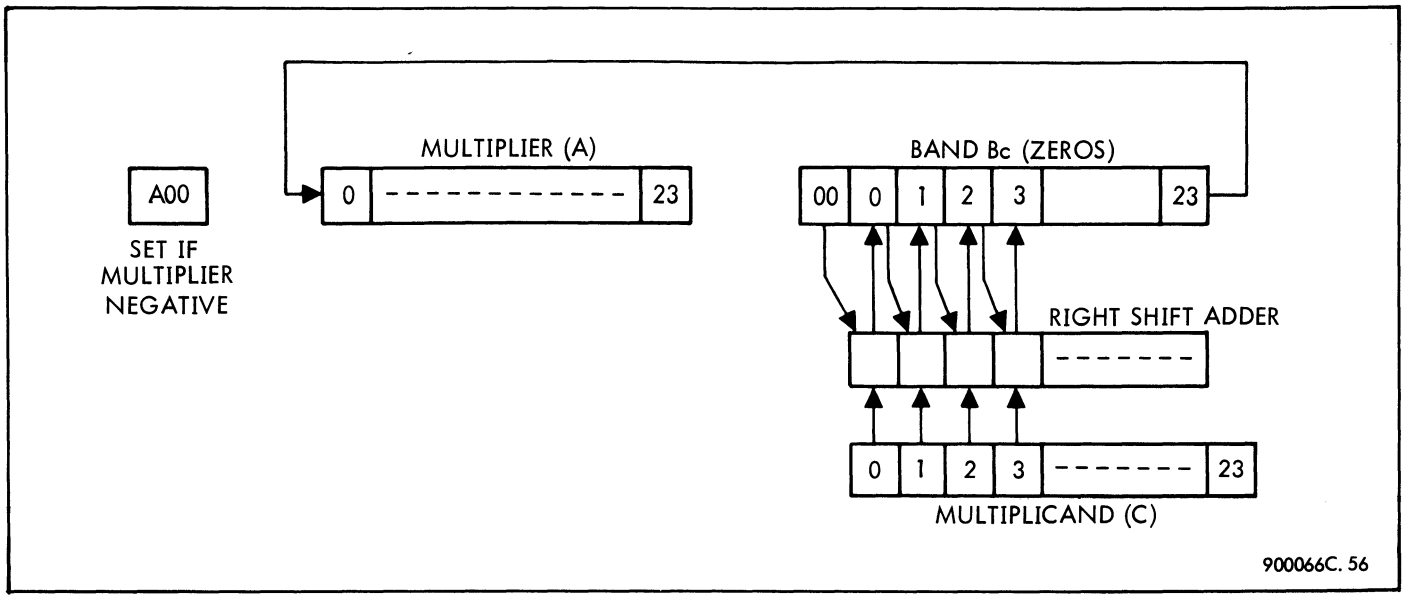


Figure 3-56. Multiplication, Block Diagram

multiply the contents of the C-register by minus one and add the result to the partial product. This is done by taking the two's complement of C and adding one times the two's complemented multiplicand to the partial product. The two's complement is obtained by inverting the C-register with Ck and then setting carry flip-flop Bc23.

$$sCk = \emptyset 7\ T8\ \overline{Ts}\ \overline{O6}$$

$$sBc23 = Rsa\ \emptyset 7\ Q1$$

The Q1 term on sBc23 is actually T7 because during the remainder of Q1, Rsa is reset and sBc23 is disabled.

3-422 If the multiplicand is negative, it is necessary to extend the sign as the partial products are formed. The expression Rsa C0 is used to set B00. The output of B00 is then used as one of the inputs to the most significant position of the right shift adder. If a Br1 is performed, Br1 B00 is used to set B0 in order to extend the sign. Flip-flop B00 is not set by C0 until the first right shift add, because when shifting ZERO's right before the first ONE is observed in the multiplier, the B-register should not be padded with ONE's.

3-423 During right shift add, the partial product is formed in (B plus Bc) and A, and the result must be in AB. The contents of A and (B plus Bc) are exchanged by shifting the registers right with Ar3 during Ø7 (T7 through Tr). The data in A goes directly to B, and B plus Bc goes to A via the full adder.

3-424 Tables 3-13 through 3-15 show the states of the A-, B-, and C-registers at each pulse time during multiplication. In the examples, the 24-bit registers are shortened to 4 bits. Three different combinations of sign bits are used in order to include the steps for both positive and negative multipliers and multiplicands. The results as it stands in BA before exchange of these registers is underlined.

3-425 Overflow will occur if an attempt is made to multiply -1 by -1. The overflow flip-flop, Of, is set by $\emptyset 7\ \overline{O3}\ O4\ \overline{O5}\ \overline{O6}\ Tp\ A0\ A00$. The A0 term implies that the result is negative, since the most significant bit of the result is in A0 at the end of multiplication. The A00 term implies that the multiplier and multiplicand are both negative. As explained in paragraph 3-415, A00 is set if the multiplier is negative. If the multiplicand is positive, C0 contains a ONE after the C-register is complemented, and A00 is reset at T7 time:

$$rA00 = (\emptyset 7\ \overline{O3}\ O4\ \overline{O5}\ \overline{O6})\ T7\ C0$$

If both multiplicand and multiplier are negative and the result is negative, the A- and C-registers both contained -1 at the beginning of multiplication.

3-426 Operand parity is checked in phase 7 while the C-register is recirculating under the control of Cr3. Setting carry flip-flop Ia causes the contents of the P-register to be increased by one during Pr3, thereby making the next instruction in sequence available for transfer to the S-register at T3. The instruction is loaded in the C-register at Tp. The Sk flip-flop is set at Tp of the first phase 3 to enable advancement of the phase counter to phase 7 at the end of the second phase 3. At T0 of phase 7, Sk is reset; at Tp of phase 7, A00 is reset if not previously reset.

3-427 Divide (DIV, Code 65) Ø0 - Ø1 - Ø3 (7 Cycles) - Ø7 - Ø0 (See figure 3-57)

3-428 The DIV instruction divides the contents of the A- and B-registers, treated as a double-precision number, by the contents of the effective memory location and places the quotient in the A-register, with the remainder in the B-register.

Table 3-13. Multiplication (A and C Positive)

Signals	B00	B-Register Contents	A-Register Contents	C-Register Contents
	0 +0	0 0 0 0 0 1 1	0 0 1 1*	0 0 1 1*
Rsa	0 +0	0 0 1 1 0 1 1	0 0 0 1	0 0 1 1
Rsa	0	0 1 0 0	1 0 0 0	0 0 1 1
Br1	0	0 0 1 0	0 1 0 0	0 0 1 1
Br1	0	0 0 0 1	0 0 1 0	0 0 1 1
Ar3 Interchange A and B		0 0 1 0	0 0 0 1	
*Original contents of A-register are 0.011. Original contents of C-register are 0.011. Operation performed is 3/8 x 3/8 = 9/64 (binary 0.001001).				

Table 3-14. Multiplication (A Negative, C Positive)

Signals	B00	B-Register Contents	A-Register Contents	C-Register Contents
	0 +0	0 0 0 0 0 1 1	1 0 1 1*	0 0 1 1*
Rsa	0 +0	0 0 1 1 0 1 1	0 1 0 1	0 0 1 1
Rsa	0	0 1 0 0	1 0 1 0	0 0 1 1
Br1	0 +1	0 0 1 0 1 0 0	0 1 0 1	0 0 1 1
Invert C, Rsa		1 1 0 1 +1 1 1 1 0	0 0 1 0	1 1 0 0
Ar3 Add 1 to B, interchange A and B		0 0 1 0	1 1 1 0	
*Original contents of A-register are 1.011. Original contents of C-register are 0.011. Operation performed is 3/8 x -5/8 = -15/64 (binary 1.1100010).				

Table 3-15. Multiplication (A Positive, C Negative)

Signals	B00	B-Register Contents	A-Register Contents	C-Register Contents
	0 +1	0 0 0 0 0 1 1	0 0 1 1*	1 0 1 1*
Rsa sB00	1 +1	1 0 1 1 0 1 1	0 0 0 1	1 0 1 1
Rsa	1	1 0 0 0	1 0 0 0	1 0 1 1
Br1 sB0	1	1 1 0 0	0 1 0 0	1 0 1 1
Br1 sB0	1	1 1 1 0	0 0 1 0	1 0 1 1
Ar3 Interchange A and B		0 0 1 0	1 1 1 0	1 0 1 1
*Original contents of A-register are 0.011. Original contents of C-register are 1.011. Operation performed is 3/8 x -5/8 = -15/64 (binary 1.1100010).				

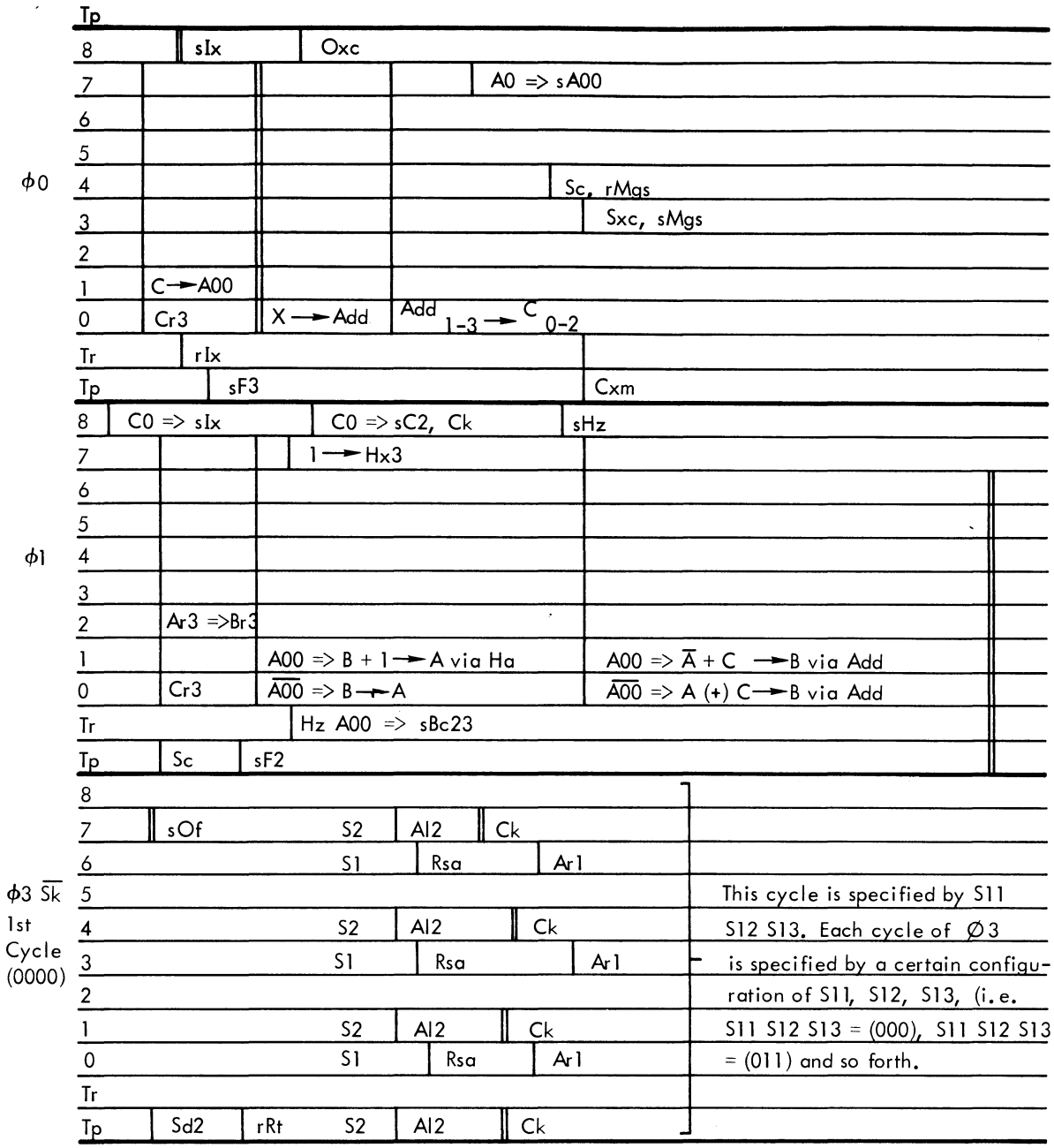
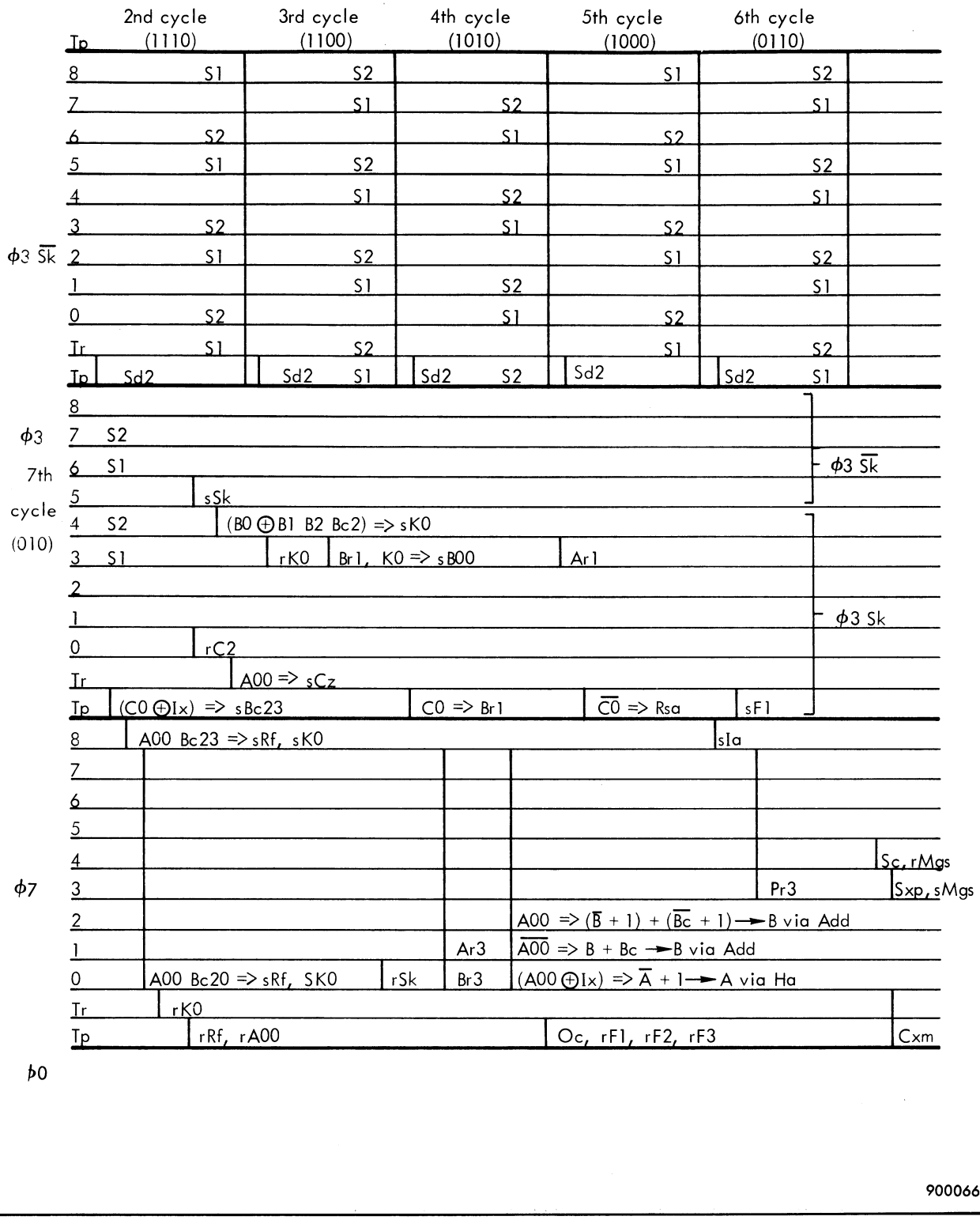


Figure 3-57. DIV Instruction, Timing Diagram (Sheet 1 of 2)



900066C.57-2

Figure 3-57. DIV Instruction, Timing Diagram (Sheet 2 of 2)

3-429 The nonrestoring calculating process is used for division in the 930 Computer. The steps in nonrestoring division are illustrated in table 3-16, where the dividend is +25/64, or binary 0.0110010, and the divisor is +5/8, or binary 0.101. The leftmost bit of the binary number represents the sign: a ZERO for plus and a ONE for minus.

Table 3-16. Example of Nonrestoring Division

Shift-Add Cycle	Calculations	Quotient	Comments
	0.0110010		Dividend
(1)	1.0110000		Add two's complement of divisor
	1.1100010	0	Partial remainder sign minus; q = 0.
	1.1000100		Shift partial remainder left
(2)	0.1010000		Add divisor
	0.0010100	0.1	Partial remainder sign plus; q = 1
	0.0101000		Shift partial remainder left
(3)	1.0110000		Add two's complement of divisor
	1.1011000	0.10	Partial remainder sign minus; q = 0
	1.0110000		Shift partial remainder left
(4)	0.1010000		Add divisor
	0.0000000	0.101	Partial remainder sign plus; q = 1 Remainder zero: division has come out even.

3-430 The divisor is first subtracted from the dividend by adding the two's complement of the divisor. The most significant quotient bit is generated from the sign bit of the difference. This difference is generally known as the partial remainder. If the sign of the partial remainder is minus, the first quotient bit is a ZERO. If the sign is plus, the first quotient bit is a ONE. The partial remainder is next shifted left one bit position. If the partial remainder was minus, the divisor is added to the shifted number. If the partial remainder was plus, the two's complement of the divisor is added. This shifting and adding process continues until the required number of quotient bits for the size of the register have been generated.

3-431 The two's complements of negative dividends and divisors are used in division; therefore, the quotient and remainder are absolute values. During the final phase of divide, the quotient is negated if the numerator and denominator have opposite signs, and the remainder is negated if the numerator is negative. A flow chart of the entire division process is given in figure 3-58.

3-432 As shown in the timing diagram, the operand address is indexed in phase 0 if Ix is set. The Ix flip-flop is set at T8 if the index bit in position 1 of the instruction is true:

$$sIx = \emptyset 0\ T8\ C1\ Go$$

The memory location is increased by the contents of the index register by presenting C and X to the full adder and placing the sum in C. Original instruction parity is checked in phase 0 during Cr3.

3-433 The dividend is in the A- and B-registers, with the most significant half in the A-register. At T7 of phase 0, the dividend sign, in A0, is put in A00, where it remains until the division is complete, at Ø7 Tp End. The operand transferred from memory to the C-register at Ø0 Tp is the divisor.

3-434 At T8 of phase 1, Ix is set if C0 is true, indicating a negative divisor. During phase 1, the numbers in the A-, B-, and C-registers are set up for division and the first subtract operation takes place:

$$|AB| - |C| \longrightarrow BA$$

The contents of the A-register are added to the contents of the C-register and placed in the B-register. The contents of the B-register are placed in the A-register: These transfers are enabled by Ar3 and Br3 and are affected by the sign bits in A and C as follows:

a. Divisor negative (C0 = 1): Flip-flop Ix is set and retains this sign information until reset at Tr of the last division phase. The contents of C, already in two's complement form, are presented to the full adder, which effectively subtracts C from A, and the difference goes to B. This operation is illustrated in the block diagram in figure 3-59.

b. Divisor positive (C0 = 0): In order to subtract C from A in the full adder, the two's complement of C must be taken. First the one's complement of C is obtained by inverting the C-register flip-flops with a Ck signal:

$$Ck = \emptyset 1\ T8\ \overline{C0}\ \overline{Ts} + \ldots$$

Full adder carry flip-flop Cz is set to change the one's complement to the two's complement by adding one:

$$sCz = \emptyset 1\ T8\ \overline{C0} + \ldots$$

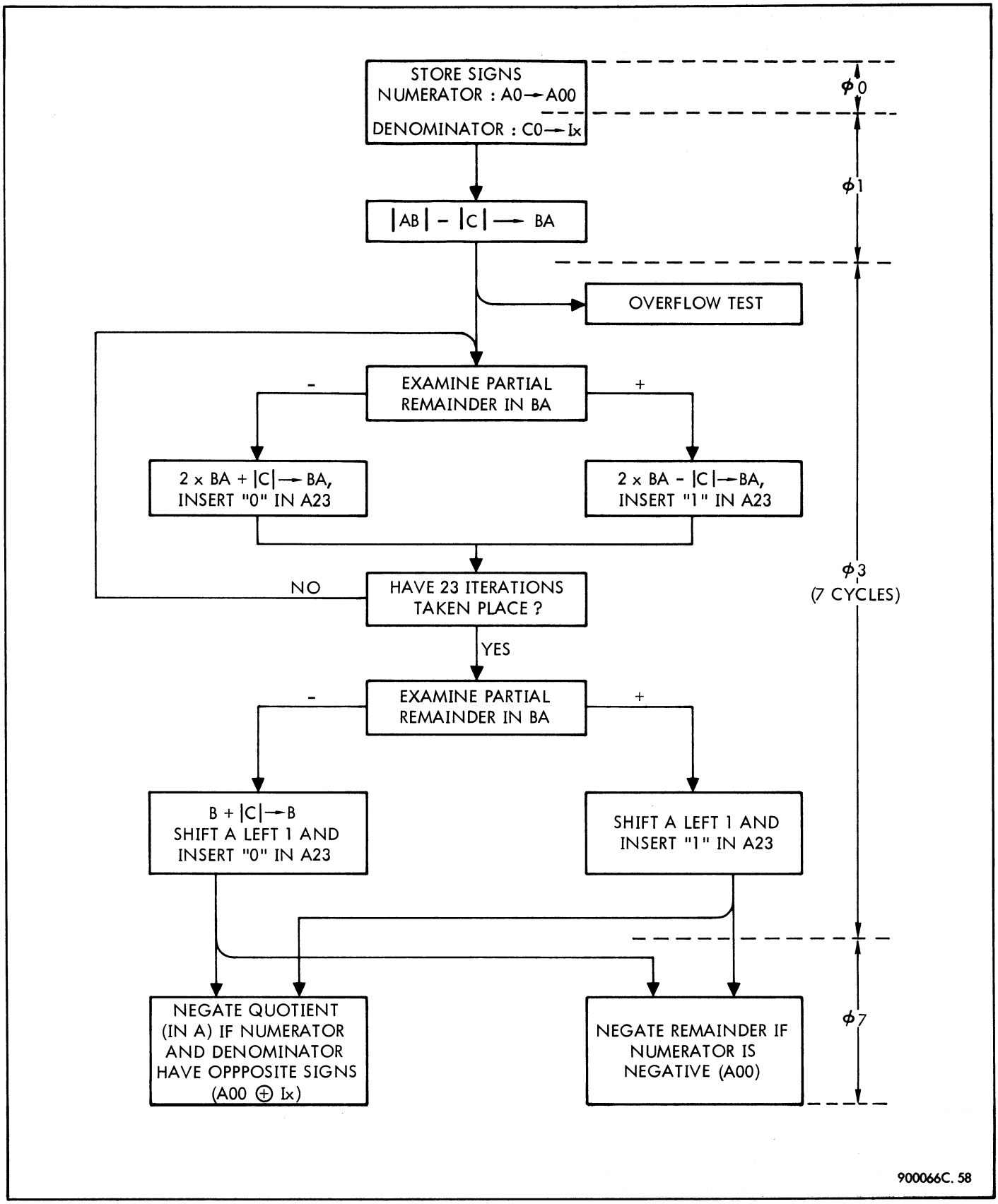


Figure 3-58. Division Flow Chart

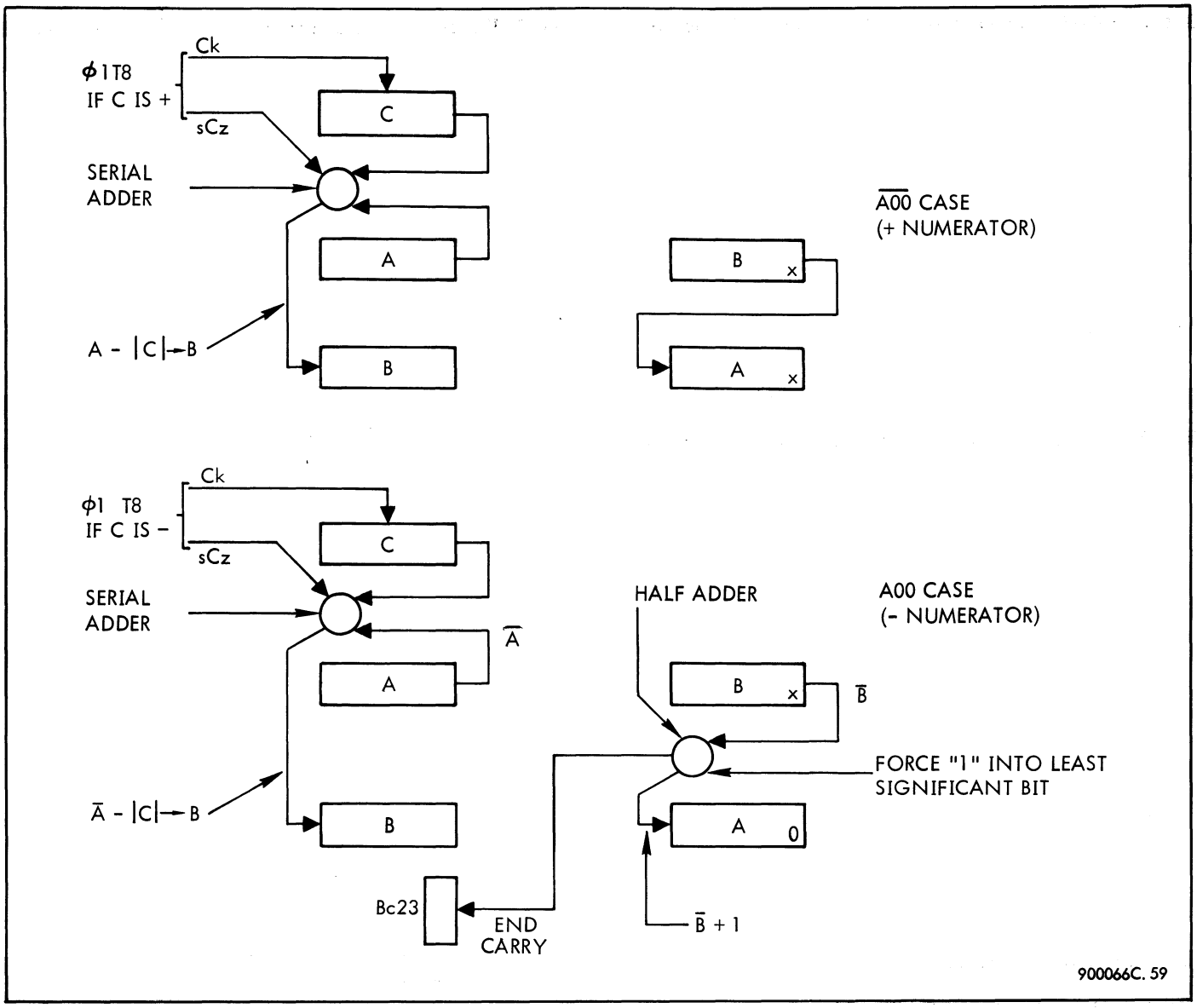


Figure 3-59. First Subtract Operation in Divide, Block Diagram

c. Dividend negative (A00 = 1): The two's complement of the number in AB must be taken before addition takes place. Both numbers are inverted to form the one's complement, and the two's complement is formed by adding +1 to the least significant bit, located in the B-register. The contents of the A-register are inverted by A00 at the inputs to the full adder. The contents of the B-register are inverted by A00 at the inputs to the half adder. Half adder carry flip-flop Hz is unconditionally set at T8 to provide the +1. Because B23 is insignificant and must therefore be ignored, the carry must be forced into B22. This is done by putting a ONE in the Hx3 input of the half adder:

$$Hx3 = \emptyset 1 \ Q2 \ \overline{Q3} + \ldots$$

Regardless of whether B23 contains a ONE or a ZERO, the combination of Hz and Hx3 at the least significant end of the half adder forces a carry into B22. If an end carry is generated by the half adder, the carry is inserted into the most significant part of the sum by setting Bc23. When AB is later shifted left two in preparation for a right shift add, Bc23 is used as an input to B21. A carry-out of the half adder occurs only when the dividend is negative and B0-22 originally contains all ZERO's.

d. Dividend positive (A00 = 0): The contents of B go directly into the A-register and later become part of the add or subtract operations as the dividend is shifted left.

During phase 1 of the Rf flip-flop is set if any ONE's have entered the A- or B-register:

$$sRf = \emptyset 1 \ \overline{(Q3 \ Q6) \ (B0 \ B1 \ B2 \ A0 \ A1 \ A2 \ Bc23)} + \ldots$$

The output of Rf is used in phase 3 to determine whether an overflow has occurred.

3-435 In the first phase 3 the Of flip-flop tests the partial remainder in BA for an overflow condition and 3 1/2 addition or subtraction cycles take place. The equation for the overflow flip-flop is as follows:

$$sOf = (\emptyset 3\ \overline{O5}\ O6)\ T7\ (\overline{S9}\ \overline{S10}\ \overline{S11}\ \overline{S12})$$
$$\overline{(Bz0 \oplus B0)}\ \left[Rf\ \overline{B0} + \overline{(A00 \oplus Ix)}\right]$$
$$+ (\emptyset 3\ \overline{O5}\ O6)\ T7\ (\overline{S9}\ \overline{S10}\ \overline{S11}\ \overline{S12})\ \overline{B0}\ Bz0 + \ldots$$

where S9-12 is a cycle counter and Bz0 indicates that if the Bc flip-flops were added to B there would be a carry into B0. An overflow occurs under the following

a. Quotient ≥ 1: Positive partial remainder $\overline{(B0 \oplus Bz0)}$ indicates that most significant bit of quotient is a ONE, yet the quotient must be positive $\overline{(A00 \oplus Ix)}$.

b. Absolute value of quotient > 1: Partial remainder positive $\overline{(B0 \oplus Bz0)}$ and not equal to zero ($\overline{B0}$ Rf). The term $\overline{B0}$ excludes B0 Bz0, which is a zero partial remainder consisting of all ONE's in B0-23 and Bc23. The Rf flip-flop is set during phase 1 if any ONE's enter BA.

c. $-1/0$: This unique case generates a ZERO in the most significant bit of the quotient and is not included in a. or b. above. It is the only case in which a partial remainder of -1 is generated and is identified by B0 Bz0.

A quotient of -1 is excluded as an overflow case, since -1 is a legitimate quotient.

3-436 The shift and add operations for the division are grouped into seven cycles in phase 3, using S9-13 as a cycle counter. Flip-flop S14 remains reset. The counter counts down by two's as follows:

S9	S10	S11	S12	S13	S14	Phase 3 Cycle	
0	0	0	0	0	0	1	T7 $\overline{S9}$ $\overline{S10}$ $\overline{S11}$ $\overline{S12}$ test overflow
1	1	1	1	1	0	2	
1	1	1	1	0	0	3	
1	1	1	0	1	0	4	
1	1	1	0	0	0	5	
1	1	0	1	1	0	6	
1	1	0	1	0	0	7	T5 $\overline{S11}$ S12 $\overline{S13}$ sets Sk

At each Tp time in phase 3, an Sd2 signal forces the counter down two steps:

$$Sd2 = (\emptyset 3\ \overline{O5}\ O6)\ Tp\ \overline{Sk}$$

3-437 The 24 shift and add operations for the 24 register bits are controlled by the states of S1 S2, which cycle as follows:

S1	S2
0	0
0	1
1	0
0	0

The cycling is terminated by the Sk flip-flop during the seventh cycle time in phase 3. The following equations apply to S1 and S2:

$$sS1 = \emptyset 3\ \overline{O5}\ O6\ S2$$
$$rS1 = \emptyset 3\ \overline{O5}\ O6\ S1$$
$$sS2 = \emptyset 3\ \overline{O5}\ O6\ \overline{Sk}\ S1\ S2$$
$$rS2 = \emptyset 3\ \overline{O5}\ O6\ S2$$

3-438 The timing diagram in figure 3-57 shows what happens at S2 and S1 during each shift and add operation in the first cycle of phase 3. The other cycles are the same, though not shown on the diagram.

3-439 The partial remainder produced by $|AB| - |C| \rightarrow BA$ is processed 23 times according to the flow chart in figure 3-58. The actual mechanization is:

$\overline{S1}\ \overline{S2}$ Pause for sign detection circuit to stabilize and generate Bz0 if the sum of B and Bc produces a carry into B0.

$\overline{S1}$ S2 Shift AB left 2. Invert C is sign is the same as B plug Bc.

S1 $\overline{S2}$ Right shift add 1/2 (B plus Bc) $\pm|C|$.

Shift A right one.

Set Bc23 if C has sign opposite to original C.

Set A23 if C0 = 1 (previous partial remainder was plus).

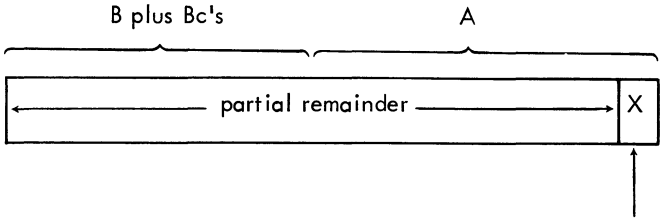
The expression 2 x BA in the flow chart is obtained by shifting left two then right one, a net shift left of one.

3-440 Assuming four-bit A-, B-, and C-registers, the right shift adder operates as follows: (Refer also to figure 3-56).

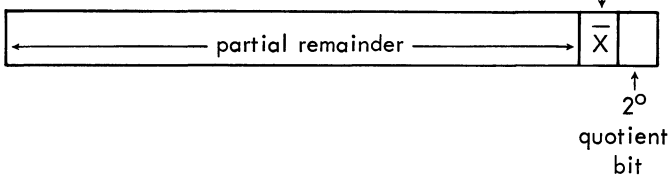
$$\begin{array}{rcccccc} & B00 & B0 & B1 & B2 & B3 & \\ + & C0 & C1 & C2 & C3 & & \\ \hline & B0 & B1 & B2 & B3 & A3 & \end{array}$$

3-441 As the shift and add operations proceed, the contents of BA shift left and the quotient bits fill the vacated least significant end of A. The B-register in the diagrams below consists of B0-23 and the 8 Bc flip-flops. The partial remainder is distributed among them.

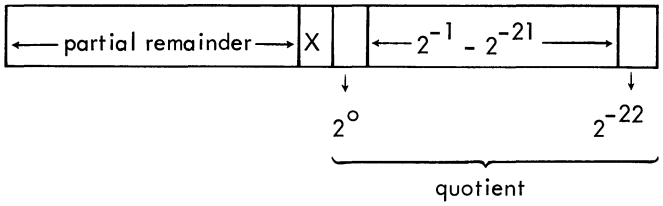
BA before first shift and add (T8, phase 3, first cycle)



BA after first shift and add (T5, Ø3, lst cycle):

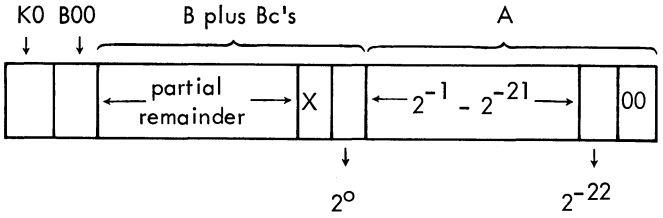


BA after 23rd shift and add (T5, Ø3 7th cycle)

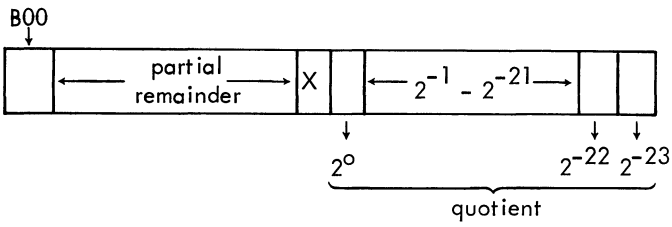


3-442 The Sk signal is turned on after the 23rd shift and add; then the final absolute values of quotient and remainder are obtained:

Ø3 Sk T3 (after BA left 2)



Ø3 Sk T2 (after BA right 1)



3-443 When the B-register is shifted left two at T4 of the seventh cycle, the sign bit in B0 is preserved by setting K0. When the B-register is shifted right one at T3, the sign is transferred from K0 to B00 for later shifting into B0.

3-444 At Tp of phase 3, cycle 7, the A-register remains stationary; the B-register shifts right one if C0 is true, and C is added to B in the right shift adder if C0 is false. The Bc23 flip-flop is set if C0 ⊕ Ix indicates that the original C was negative and that +1 must be added to form the two's complement from the one's complement. The net shifts of the B-register from T4 to Tp amount to zero because one left shift two and two right shift ones have taken place. At the end of the seventh cycle, the quotient is in the A-register and the remainder is in the B-register. Table 3-17 illustrates the steps in the process of dividing 25/64 by 5/8, showing the states of the A-, B-, and C-registers after each shift and add. To make the example brief, the registers are assumed to contain 4 bits instead of 24.

3-445 At T0 phase 3, cycle 7, full adder carry flip-flop Cz is set if the dividend is negative, in preparation for complementing the remainder. Because Ck keeps the active denominator and the partial remainder at opposite signs (Ck = Ø3 $\overline{O5}$ O6 S2 $\overline{Ts}$) ($\overline{C0 \oplus B0 \oplus Bz0}$), $\overline{C0}$ at Ø3 Sk Tp implies a negative partial remainder. If the last partial remainder is negative, (C0 ⊕ Ix) ⟹ sBc23 at Ø3 Sk Tp is used to generate the proper positive form of the denominator to feed Rsa and produce an absolute value remainder at this point. If the last partial remainder is positive, B is simply shifted right one at Q3 Sk Tp. Phase 7 may later negate (complement) the remainder if the original numerator was negative.

3-446 During phase 7, the contents of the Bc register are serially added in the full adder to the contents of the B-register. The outputs of Bc23 are presented to the Yx3 inputs of the adder, and K0 and Rf are presented to the Yz2 and Yz1 inputs, respectively. The outputs of B21 through B23 are presented to the Xz1 through Xz3 inputs of the adder. If the dividend is positive (A00 = 0), a straightforward addition is made, and the sum is put in the B-register. If the dividend is negative (A00 = 1), the one's complement of B21 through B23 is presented to the Xz1 through Xz3 inputs, and carry flip-flop Cz provides the +1 to change the one's complement to the two's complement. The two's complement of the contents of Bc is obtained by setting Rf and K0 as soon as the first ONE appears in the register. When K0 is set, the bit in Bc20 is inverted as it enters Bc23. This logic passes everything up to and including the least significant ONE and inverts all the higher order bits, as explained under right shift three, paragraph 3-127. The sum of the two's complements is placed in the B-register.

3-447 If the numerator and denominator were of opposite sign (A00 ⊕ Ix), the quotient is negated in the half adder. The inverted outputs of A21 -A23 are presented to the half adder, and half adder carry flip-flop Hz, which is set at every T8 time, adds one to the one's complement to form the two's complement.

Table 3-17. Register States in Division

Time	Signals	C-Register Contents	A-Register Contents	B00	B-Register Contents	Bc23	Bz0
Ø0	Original states	0 1 0 1*	0 0 1 1* +1 0 1 1	0	0 0 1 X*	0	0
Ø1 T7-T0	B→A, A + $\overline{C}$→B	1 0 1 0	0 0 1 X	0	1 1 1 0	0	0
Ø3 $\overline{Sk}$ S2 (Cycle 0000)	Aℓ2, Invert C	0 1 0 1	1 X 1 1	1 +0	1 0 0 0 1 0 1	0	0
Ø3 $\overline{Sk}$ S1 (Cycle 0000)	Rsa, Shift A right 1	0 1 0 1	0 1 X 0	0	0 0 0 1	0	0
Ø3 $\overline{Sk}$ S2 (Cycle 0000)	Aℓ2, Invert C	1 0 1 0	X 0 0 0	0 +1	0 1 0 1 0 1 0	0	0
Ø3 $\overline{Sk}$ S1 (Cycle 0000)	Rsa, Shift A right 1	1 0 1 0	1 X 0 1	1	1 1 0 0	1	0
Ø3 $\overline{Sk}$ S2 (Cycle 0100)	Aℓ2, Invert C	0 1 0 1	0 1 1 1	1 +0	0 1 1 X 1 0 1	0	0
Ø3 $\overline{Sk}$ S1 (Cycle 0100)	Rsa, Shift A right 1	0 1 0 1	X 0 1 0	0	0 0 0 0	0	0
Last Iteration Ø3 Sk S2 (Cycle 0100)	Aℓ2, Invert C	1 0 1 0	1 0 0 0	0	0 0 X 0	0	0
Ø3 Sk S1 (Cycle 0100)	Arl, Brl	1 0 1 0	0 1 0 1	0	0 0 0 X	0	0
Ø3 Sk Tp (Cycle 0100)	Brl	1 0 1 0	0 1 0 1	X	0 0 0 0	0	0

*Dividend is 0.011001, or 25/64 in A-register and B-register. Divisor is 0.101, or 5/8 in C register. Operation performed is 25/64 - 5/8 = 5/8 (in A-register and B-register).

3-448 The contents of the P-register are increased by one during phase 7 by setting carry flip-flop Ia, and the address placed in the S-register at T3 is accessed at Tp to obtain the next instruction.

3-449 Shift Series (RSH, RCY, LRSH, Code 66; LSH,NOD, DCY, Code 67) Ø0 - Ø1 - Ø3 - Ø3 - Ø7
(See figure 3-60)

3-450 The shift instructions consist of three right shift instructions (code 66) and three left instructions (code 67). The control bit, which determines whether the shift is to be right or left, is placed in O6 of the opcode register. The type of shift is designated by bits 10, 11, and 12 of the shift instruction as shown in figure 3-61. Configurations in bit positions 8, 10, 11, and 12, other than those listed in figure 3-61, are illegal and need not be considered. The number of bit positions to be shifted is indicated in bits 15 through 23 of the instruction. If the count exceeds decimal 48, a count of 48 is substituted before shifting begins. In a Normalize instruction, the shift count limits the number of positions of shift permitted.

3-451 The functions of the six shift instructions are given in table 3-18.

3-452 General Logic for All Shift Operations. If bit 9 contains a ONE, a complete phase 0 cycle will be used for indirect addressing, as explained in paragraph 3-242. If indexing is required in the indirect addressed instruction, bits 10 through 23 of the index register are added to C10 through C23 as described under indexing, paragraph 3-240. A second phase 0 cycle is started, and is terminated at T5. If indirect addressing is not required, the first phase 0 lasts only through T5. The nine shift count bits may be indexed during T7 through T5 of this final phase 0.

3-453 As the C-register starts circulating in phase 0, the control bits in positions 10, 11, and 12 leave their original positions. When the shifting stops at phase 1 T2, the control bits are in C4, C5, and C6.

3-454 Phase 1 starts and lasts three pulse periods, T4 through T2. No indexing takes place during this period, because the C-register recirculates instead of receiving the adder output. Since the control bits originally in C10 through C12 do not reach C21 through C23 until T4, the nine-bit indexing does not modify the control bits.

3-455 At phase 1 T3, the shift count is transferred to S6 through S14 with Sxc. The bits transferred to S1 through S5

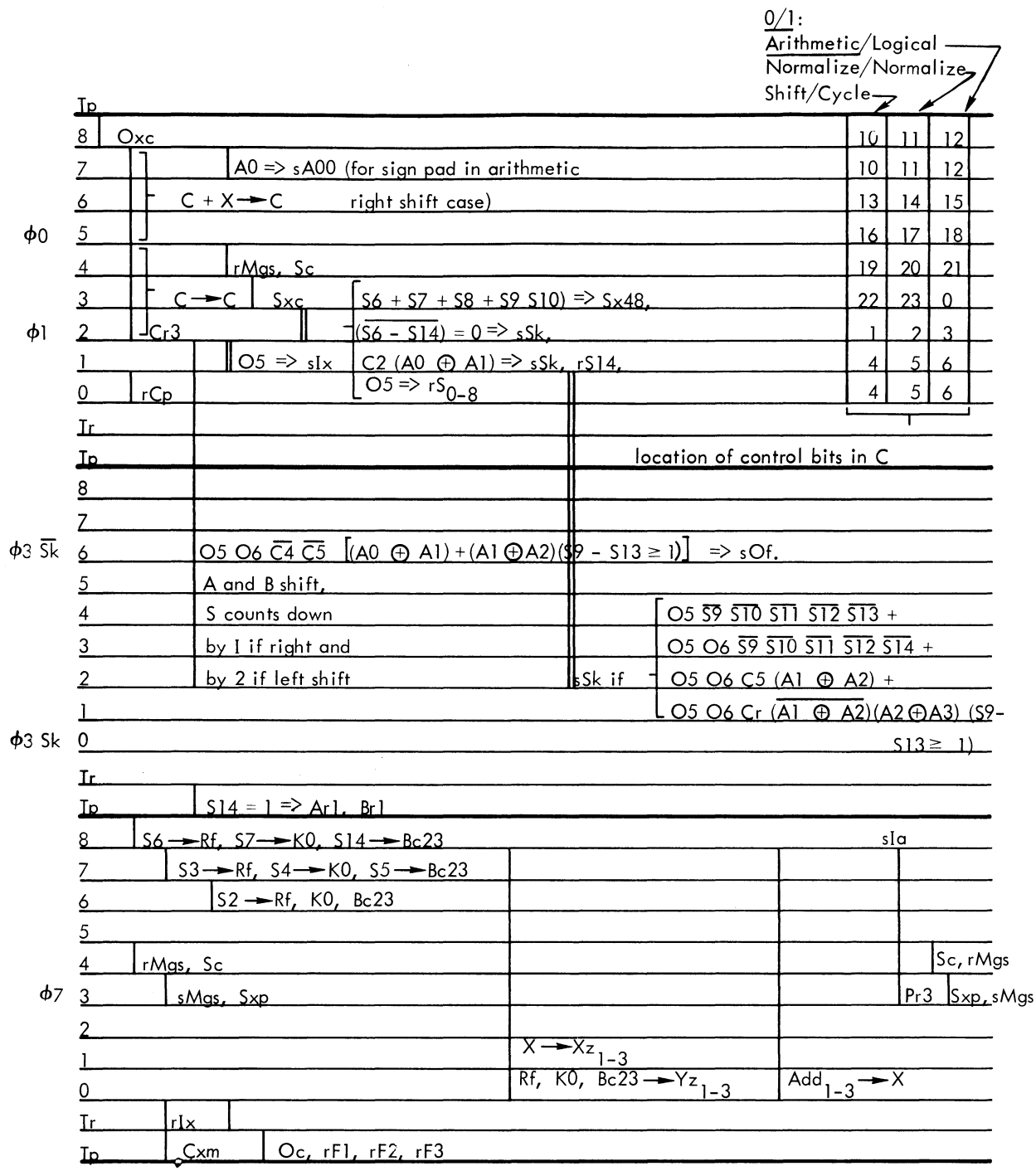


Figure 3-60. Shift Series, Timing Diagram

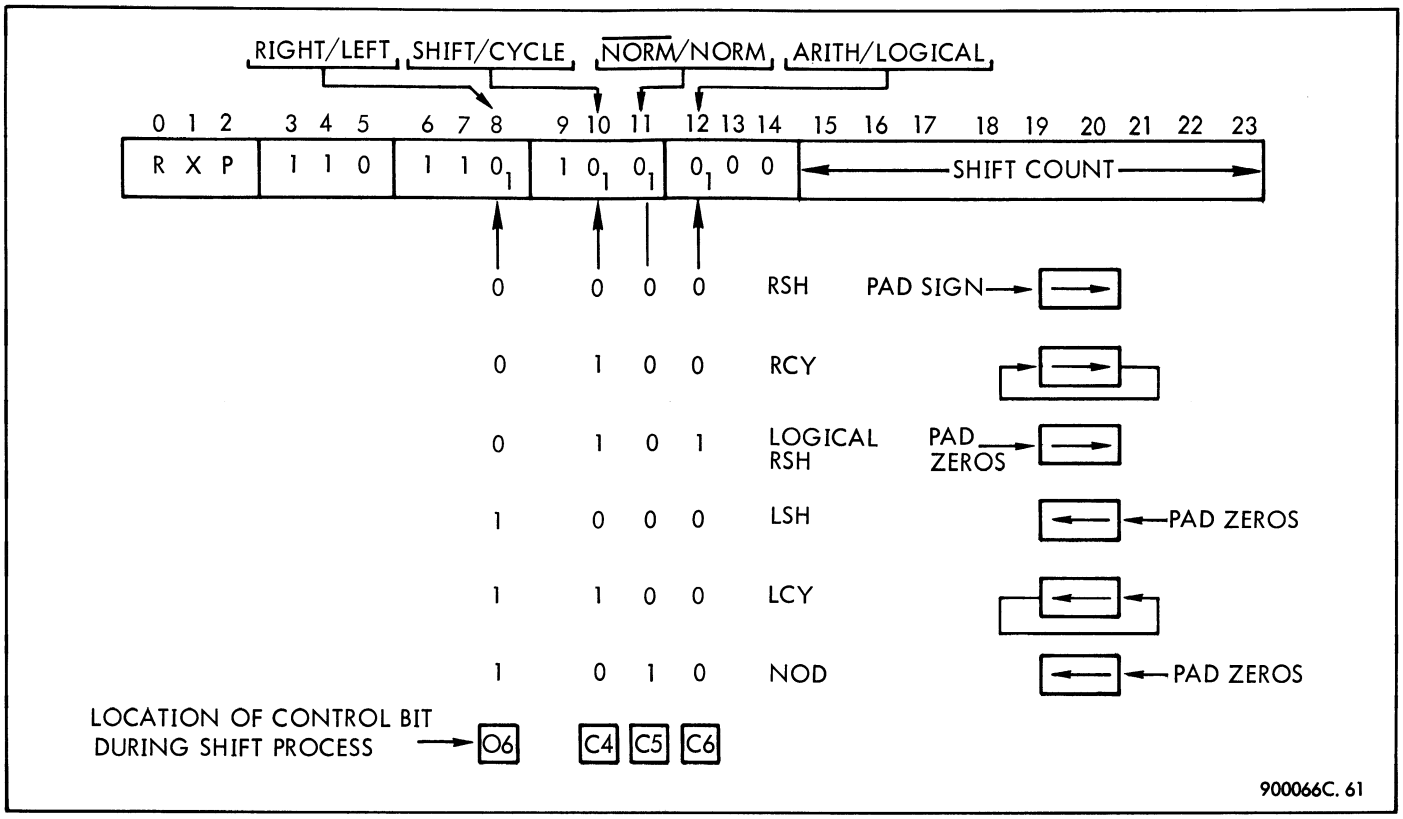


Figure 3-61. Shift Series Codes and Functions

Table 3-18. Shift Instructions

Instruction	Mnemonic	Code	Function
Right Shift AB	RSH	06600XXX	Shifts contents of AB right. Sign bit in A copied into vacated positions of A. B sign bit shifts with B. A23 shifts into B0; bits shifting past B23 are lost.
Right Cycle AB	RCY	06620XXX	Shifts contents of AB right. Sign bits in A and B shift with other bits. A23 shifts into B0; B23 shifts into A0. The register cycles onto itself and loses no bits.
Logical Right Shift AB	LRSH	06624XXX	Shifts contents of AB right. Sign bits of A and B shift with rest of number. Vacated bits in A fill with ZERO's. A23 shifts into B0; bits from B23 are lost.
Left Shift AB	LSH	06700XXX	Shifts contents of AB left. Overflow indicator is set when bit different from original sign enters sign position. B0 shifts into A23; Bits shifting out of A0 are lost. Vacated bits on right end of B are filled with ZERO's.
Normalize and Decrement Index	NOD	06710XXX	Shifts AB left until position 1 in A contains bit not equal to bit in sign position or until number of shifts specified in bits 15 through 23 occur. Count of number of places shifted is subtracted from index register and result placed in index register. If shifting exceeds 48 places, 46 is subtracted from index register. Vacated positions on right end of B are filled with ZERO's. Bits 15 through 23 contain upper limit of shifts.
Left Cycle AB	LCY	06720XXX	Shifts contents of AB left. Sign bits in A and B shift with other B0 shifts into A23; A0 shifts into B23. The register cycles onto itself and loses no bits.

can be neglected. At T2 the S-register is examined and modified to accommodate the various shifting options: The state of the S-register at phase 1 T2 and at phase 3 T1 is shown in figure 3-62. Memory cycle request signal Mgs is not true until phase 7 T3; therefore, no memory access occurs until that time and the S-register is available for control purposes. Starting with phase 3 T1, the S-register may be considered as two independent counters.

3-456 The Sk flip-flop is used to control the number of shifts. Phase 3 $\overline{Sk}$ enables shifting, and shifting stops when Sk is set.

3-457 Parity checking of the C-register is cancelled at phase 3 T0, since the Cr3 pulses stop at the end of phase 1 after only six pulse periods. Parity check would be on only six octal positions instead of eight, and would therefore be invalid.

3-458 Right Shift Logic. The A- and B-registers shift right one bit position at a time when Ar1 and Br1 are true. These shift signals are qualified by Ø3 $\overline{Sk}$ O5 $\overline{O6}$. The only difference between the three right shift operations is the input to A0, as indicated in figure 3-63.

3-459 The shift count for a right shift instruction is contained in S9 through S14, which counts down by one each pulse period. The timing diagram shows a right shift with a shift count of decimal 11. During the 11 pulse periods of Ø3 $\overline{Sk}$, S9 through S14 counts down and the A- and B-registers shift right one bit position at a time. At T2 of the first full cycle of phase 3, when S9 through S14 contains 000001, a set term is applied to Sk:

$$sSk = Ø3\ \overline{Sk}\ C5\ \overline{S9}\ \overline{S10}\ \overline{S11}\ \overline{S12}\ \overline{S13}$$

During this final pulse period, S9 through S14 counts down to zero and the AB register makes its final shift.

3-460 Had the original shift count exceeded decimal 48, only 48 positions would have been executed. In right cycle, the AB register would not change. Logical right shift would have cleared AB, while right shift would have filled AB with the original sign bit of A.

3-461 Phase 7 is set by Tp Sk, and the usual End cycle functions are performed; that is, adding one to the P-register and obtaining the next instruction.

3-462 Left Shift, Not Normalize. The A- and B-registers shift left two bit positions at a time whenever Aℓ2 is true. The Aℓ2 signal is qualified by Ø3 $\overline{Sk}$ O5 O6. Flip-flops B21 and B22 receive ZERO's except in left cycle, in which they receive the contents of A0 and A1 (see figure 3-64). In the left shift case, $\overline{C4}$ $\overline{C5}$ allow the overflow flip-flop to set if a shift changes the sign bit.

3-463 Left shifting by two's will take care of left shift instructions in which the shift count is an even number. If the shift count is odd, AB will overshift by one, after which a single right shift one bit position will take place at Ø3 Ø3 Sk Tp. For example, if a shift count of decimal 21 is specified, 11 left shifts by two will take place, followed by one right shift by one. The bit in A1 shifts left into A00. The bit in A00 shifts right into A0 at Ø3 Sk Tp.

3-464 As AB shifts left by two's, S9 through S14 counts down by two's. Flip-flop S14 is not affected by the down count, and retains even odd information. When the contents of S9 through S14 are equal to or less than two (00000X or 0000X0), a set term is applied to Sk. During this final pulse period of Ø3 $\overline{Sk}$, AB either shifts exactly into place (even shift count) or overshifts by one (odd shift count). In the odd shift case, S14 enables the compensatory right shift one bit position at Tp Sk:

$$Ar1,\ Br1 = Ø3\ O5\ O6\ Tp\ Sk\ S14 + \ldots$$

The timing shown on the timing diagram could apply to a left shift count of decimal 21 or 22.

3-465 Overflow term sOf, which indicates that a shift will change the sign bit and make the number invalid, is interpreted as follows:

$$sOf = Ø3\ O5\ O6\ \overline{C4}\ \overline{C5}\ \overline{Sk}\ [(A0 \oplus A1) + (S > 1)\ (A1 \oplus A2)]$$

where

Ø3 O5 O6	indicates	left shift	
$\overline{C4}$	indicates	not cycle	
$\overline{C5}$	indicates	not normalize	
$\overline{Sk}$	indicates	S > 0 (number will shift one or more places)	
A0 ⊕ A1	indicates	number is already normalized	number will shift left at least one place beyond register capacity
S > 1	indicates	number will shift two or more places	
A1 ⊕ A2		and would be normalized if shifted only one place	

3-466 Normalize. The Normalize instruction is a variation of left shift in which termination of the shift is controlled not only by the shift count, which sets a limit, but also by the contents of AB. The AB register is shifted left until A0 is not equal to A1; that is, (A0 ⊕ A1) is true. The number of positions of shift required is subtracted from the index register. The logic is somewhat complicated by the fact that AB shifts left by two's rather than by one's.

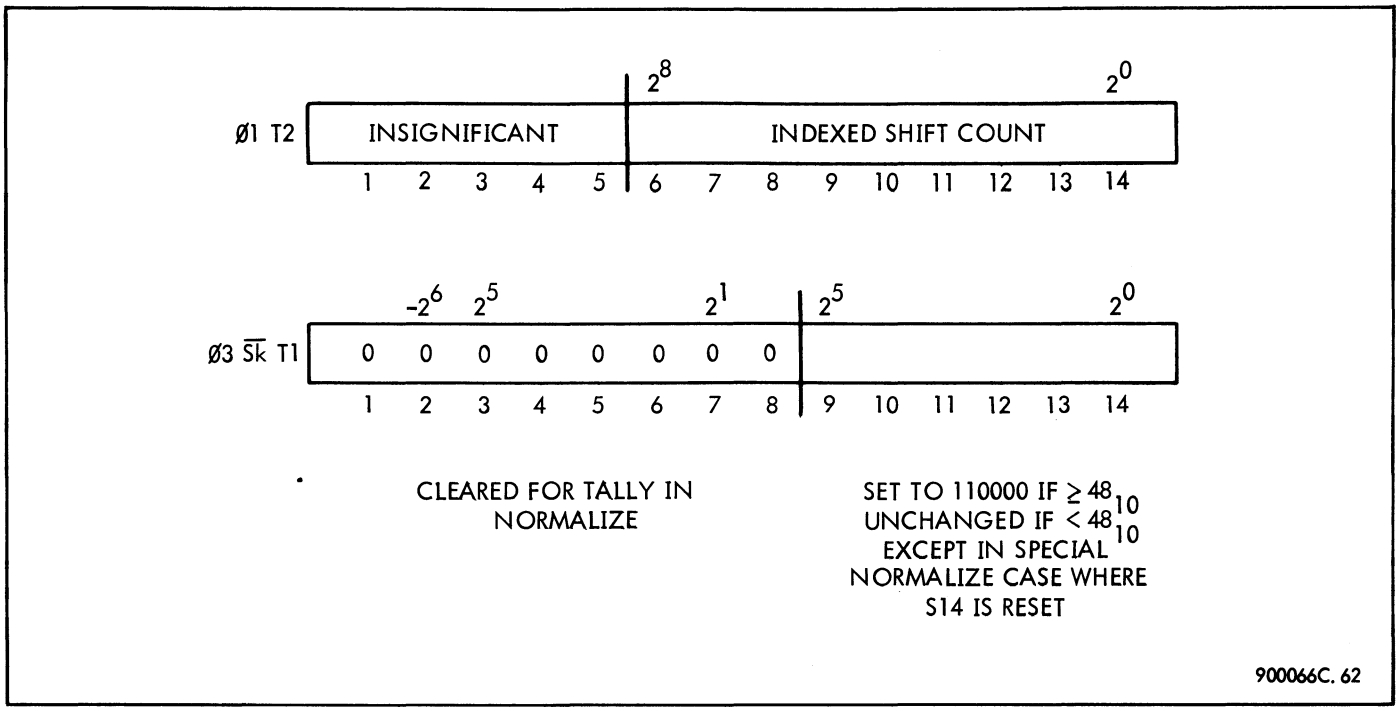


Figure 3-62. S-Register States

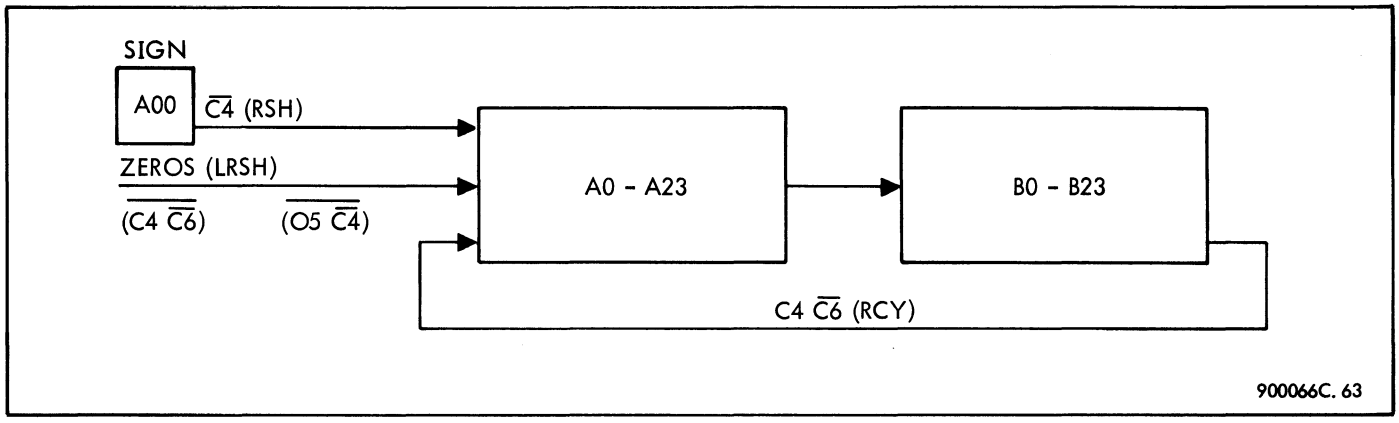


Figure 3-63. Right Shift Options

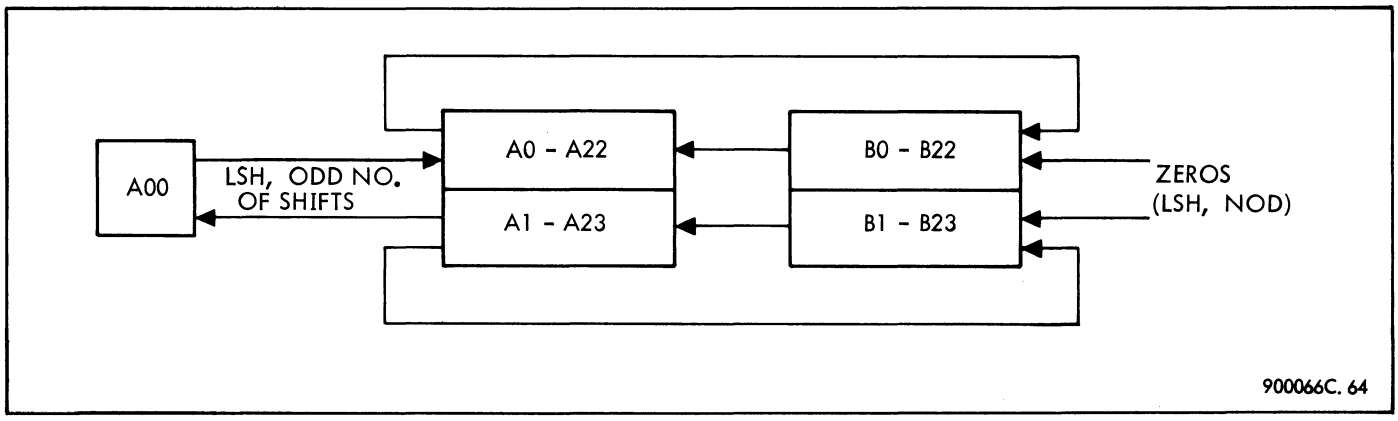


Figure 3-64. Left Shift Options

3-467 At Ø1 T2, after Sxc has taken place, A0 ⊕ A1 is examined to determine whether AB is already normalized. If AB is normalized, Sk is set and no shifting takes place:

$$sSk = Ø1\ O5\ T2\ C2\ (A0 \oplus A1)$$

where C2 is the location of the original bit 11, which specifies normalize or not normalize. Starting with T1, the normalize bit is frozen in C5. The expression which sets Sk also resets S14 to prevent a shift right by one at Ø3 Sk Tp and a false tally during phase 7. Bits 2 through 7 of the S-register count the number of bit positions of shift required to normalize AB. The bits are organized to handle the tally as shown in figure 3-65. The tally obtained from S2 through S7 is greater than -46 and less than zero and is subtracted from the X-register. The result is placed in the X-register during phase 7.

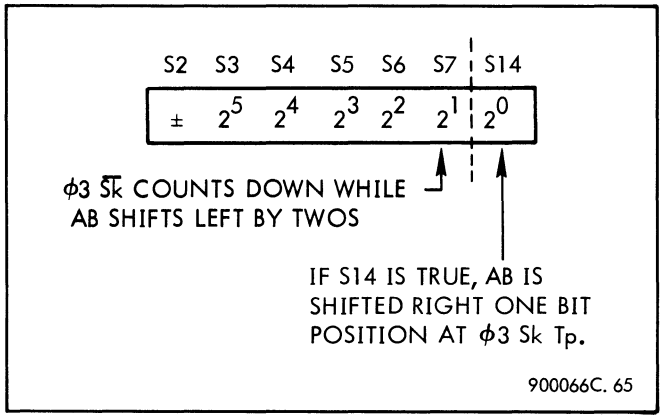


Figure 3-65. Organization of S-Register for Normalize Shift Count

3-468 Assuming that AB was not initially normalized, Ø3 S̄k is entered and AB shifts left by two's. Shifting is terminated by setting Sk with expressions which reduce to the following:

$$(S9\text{-}14 \leqslant 2) + C5\left[(A1 \oplus A2) + (A2 \oplus A3)\ \overline{(A1 \oplus A2)}\ (S9\text{-}14 > 1)\right]$$

where

(S9-14 ⩽ 2)	is	usual shift count logic for left shift cases (limit)
C5	indicates	normalize only
A1 ⊕ A2	indicates	one more position to go Sk ⟹ sS14
(A2 ⊕ A3) $\overline{(A1 \oplus A2)}$	indicates	two more positions to go Sk ⟹ rS14

The A1 ⊕ A2 case above will result in an overshift similar to that in an ordinary left shift case where the shift count is odd. Flip-flop S14 is set to ensure a right shift by one at Ø3 Sk Tp.

3-469 The A2 ⊕ A3 case calls for two positions of shift exactly provided (A1 ⊕ A2) + (S9-14) = 1) do not specify a net shift of one instead. Flip-flop S14 is therefore forced off by:

$$(A2 \oplus A3)\ \overline{(A1 \oplus A2)}\ (S9\text{-}14 > 1)\ \overline{Sk}$$

where

A2 ⊕ A3	indicates	shift two more places (maximum)
$\overline{A1 \oplus A2}$	indicates	exclude shift one to normalize
(S9-14 > 1)	indicates	exclude shift one to reach limit
$\overline{Sk}$	indicates	prevent S14 change after shift left ends

When shifting is completed, S14 registers whether the number of positions shifted was odd or even and is used as the 2^0 bit of the tally. Since S2 through S7 counted down by Ø3 S̄k as AB shifted left by two's, S7 represents the 2^1 bit position of the tally.

3-470 During phase 7, the tally is injected into the Yz inputs of the full adder via the Rf, K0, and Bc23 flip-flops as shown in figure 3-66. The S-register bits are gated into Rf, K0, and Bc23 at T8, T7, and T6, respectively, as shown in the timing diagram. From T7 to T0, Rf, K0, and Bc23 are added to X in the full adder. The sign bit is extended to the most significant bit because Rf, K0, and Bc23, which contain the sign bit from T6 on, are presented continually to the Yz inputs of the adder from T5 through T0. The Xz inputs to the adder are fed by the X-register by forcing Ix on with Ø3 O5. The Xz input equations are:

$$Xz1\text{-}3 = Ix\ Xn1\text{-}3\ Ø7\ O5 + \ldots$$

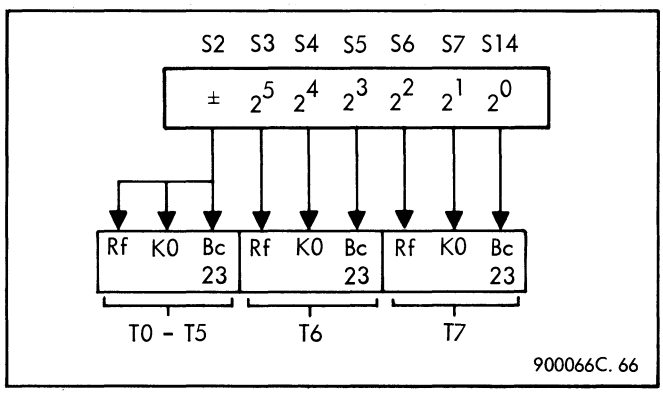


Figure 3-66. Tally Transfer from S-Register

3-471 After T6, when the last bit of S-register information is directed into the adder, the S-register is free for its normal memory addressing function, which begins with Sc at T4.

3-472 The adder feeds the X-register during phase 7 of Normalize, but not during any other shift instruction:

$$sXw1 = Add\ 1\ (F1 + \overline{O6})\ (\overline{O3}\ O4\ O5\ O6\ C5\ \emptyset7) + \ldots$$

The C5 term distinguishes Normalize from the other shifts because this is the original bit 11 of the instruction in the C-register. In the 1.925-microsecond 930 Computers (certain models with serial numbers under 20), $\overline{C5}$ instead of C5 does the enabling. In these models, the C-register is inverted at $\emptyset7$ T8 for the benefit of Multiply and C5 reverses its significance. In the 1.75-microsecond models, the C inversion is inhibited by O6, hence the C5 bit remains unchanged during phase 3 and phase 7.

3-473 Overflow testing does not take place automatically when the negative tally is added to the X-register. This testing is done by the Record Exponent Overflow (REO) instruction, which is an EOM:

$$sOf = (Eom\ C10\ \overline{C11})\ C20\ T4\ (Xn3\ \overline{Xw1} + \overline{Xn3}\ Xw1)$$

3-474 Skip if A Equals Memory on B Mask (SKM, Code 70)

$\emptyset0 - \emptyset6$ (skip $\emptyset7$) $\emptyset0$ (See figure 3-67)

3-475 The SKM instruction compares selected bits of the contents of the A-register with the corresponding bits in the contents of the effective memory location. If the selected bits in A and the memory location are identical, the computer skips the next instruction in sequence and executes the following instruction. If the selected bits are not identical, the next instruction in sequence is executed. The bits to be compared are selected by placing ONE's in the corresponding bit positions of the B-register and ZERO's in the remaining bit positions of B.

3-476 The original instruction parity is checked during phase 0 when Cr3 is active. A memory cycle is initiated during phase 0 to place the operand in the C-register, and operand parity is checked while the C-register is recirculating in phase 6.

3-477 The Sk flip-flop is unconditionally set at $\emptyset6$ T8. If at any time during phase 6 T7 through T0 the selected bits in the A- and C-registers are not identical, Sk is reset and a skip is not performed. The equation for resetting Sk is as follows:

$$rSk = B21\ (A21 \oplus C21) + B22\ (A22 \oplus C22) + B23\ (A23 \oplus C23) + \ldots$$

The A-, B-, and C-registers are examined three bits at a time from T7 through T0 while the A-, B-, and C-registers are recirculating.

3-478 If Sk contains a ZERO at $\emptyset6$ Tp, implying that at least one pair of selected bits in A and C are not identical, sF3 is inhibited and Oc, rF1 and rF2 are enabled. This takes the phase counter to phase 0, ending the SKM instruction. The P+1 instruction which has been placed in the P-register during phase 6 is placed in the C-register at $\emptyset6$ Tp.

3-479 If Sk remains set at $\emptyset6$ Tp, implying that all of the selected bits in the A- and C-registers are identical, F3 is set and F1 and F2 remain set. The phase counter is now in phase 7, and the P-register is increased by one to make the P+2 address available for access. When the logic goes to phase 7 to access the P+2 instruction, the P+1 instruction accessed during phase 6 is disregarded.

3-480 Load Index (LDX, Code 71) $\emptyset0 - \emptyset6 - \emptyset0$ (See figure 3-68)

3-481 The LDX instruction loads the entire 24-bit contents of the effective memory location into the index register.

3-482 The original instruction parity is checked while the C-register is recirculating in phase 0. The operand is transferred from memory to the C-register at $\emptyset0$ Tp, and operand parity is checked in phase 6 during Cr3. Flip-flops F1 and F2 are set to advance the phase counter to phase 6.

3-483 The X-register normally recirculates. When executing the LDX instruction, recirculation of old data must be inhibited via Xnr and the contents of the C-register must be shifted into the X-register:

$$Xnr = \emptyset6\ O1\ O2\ O3\ \overline{O4}\ \overline{O5}\ O6 + \ldots$$

The contents of C are transferred directly to X an octal digit at a time during $\emptyset6$ Cr3:

$$sXw1\text{-}3 = \overline{Ts}\ \emptyset6\ O1\ O2\ O3\ \overline{O4}\ \overline{O5}\ O6\ C21\text{-}23$$

As each new octal is loaded into X, the contents of X are shifted right to make room for the next octal. Setting carry flip-flop Ia at $\emptyset6$ T8 enables the contents of the P-register to be increased by one so that the next instruction in sequence may be placed in the C-register at Tp.

3-484 Skip if A and Memory Do Not Compare Ones

(SKS, Code 72) $\emptyset0 - \emptyset6$ (skip $\emptyset7$) $\emptyset0$ (See figure 3-69)

3-485 The SKA instruction compares the contents of the A-register, bit by bit, with the contents of the effective memory location. If A and the effective memory location do not contain any one pair of ONE's in corresponding bit positions, the next instruction in sequence is skipped and the following instruction is executed. If A and the effective location contain at least one pair of one bits in corresponding bit positions, the next instruction in sequence is executed. The instruction logically AND's corresponding bits in A and memory.

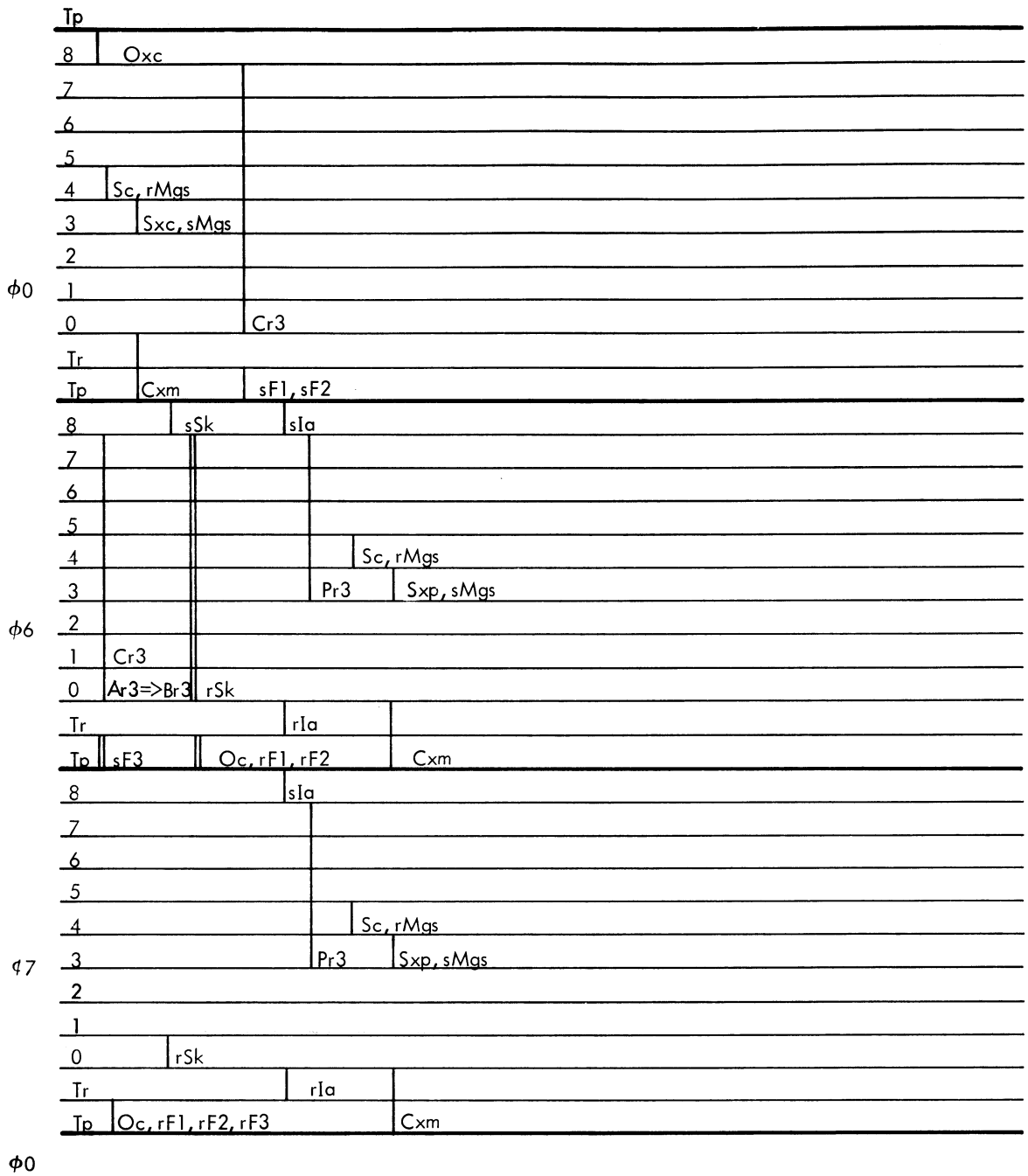
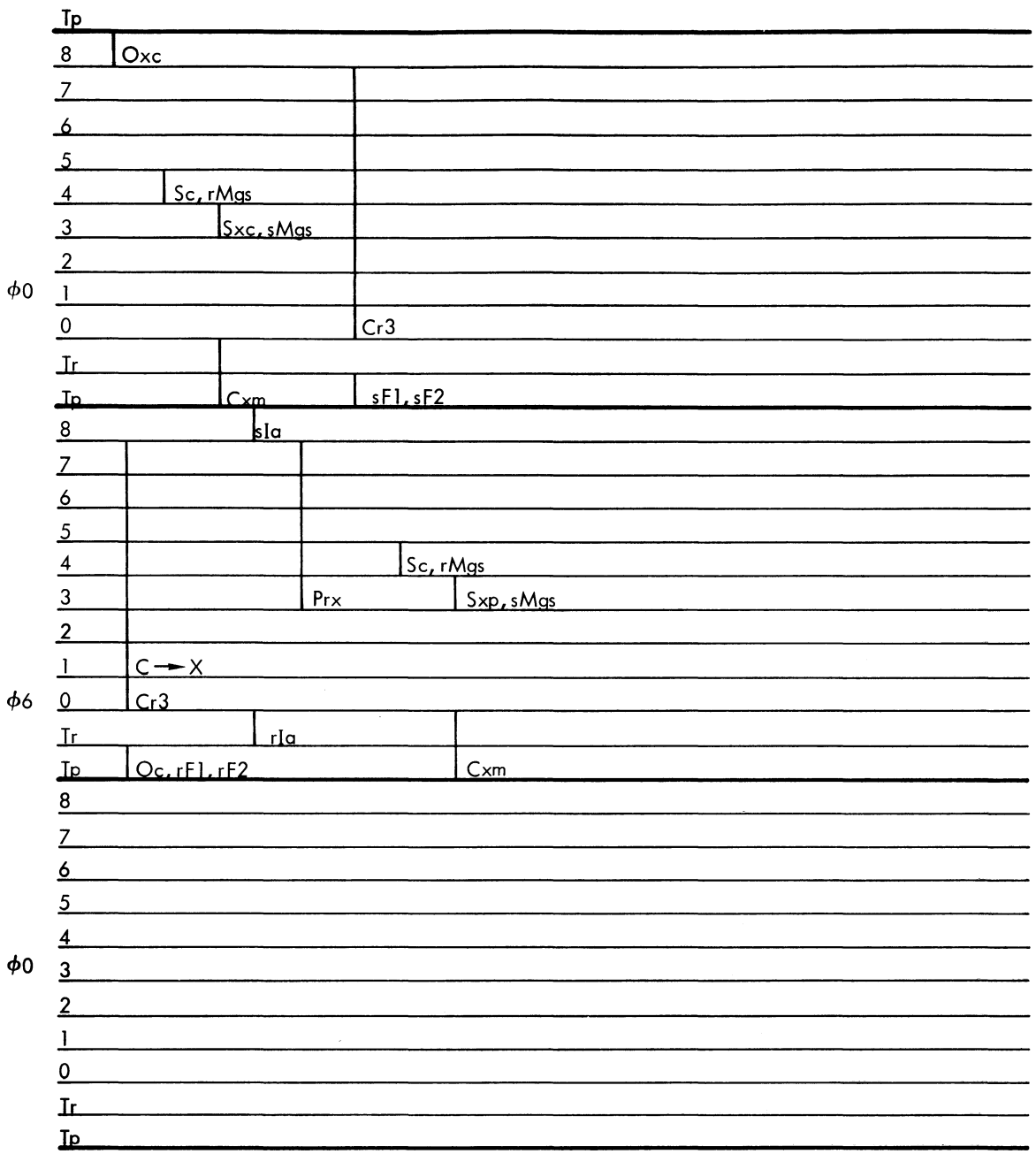


Figure 3-67. SKM Instruction, Timing Diagram



900066C. 68

Figure 3-68. LDX Instruction, Timing Diagram

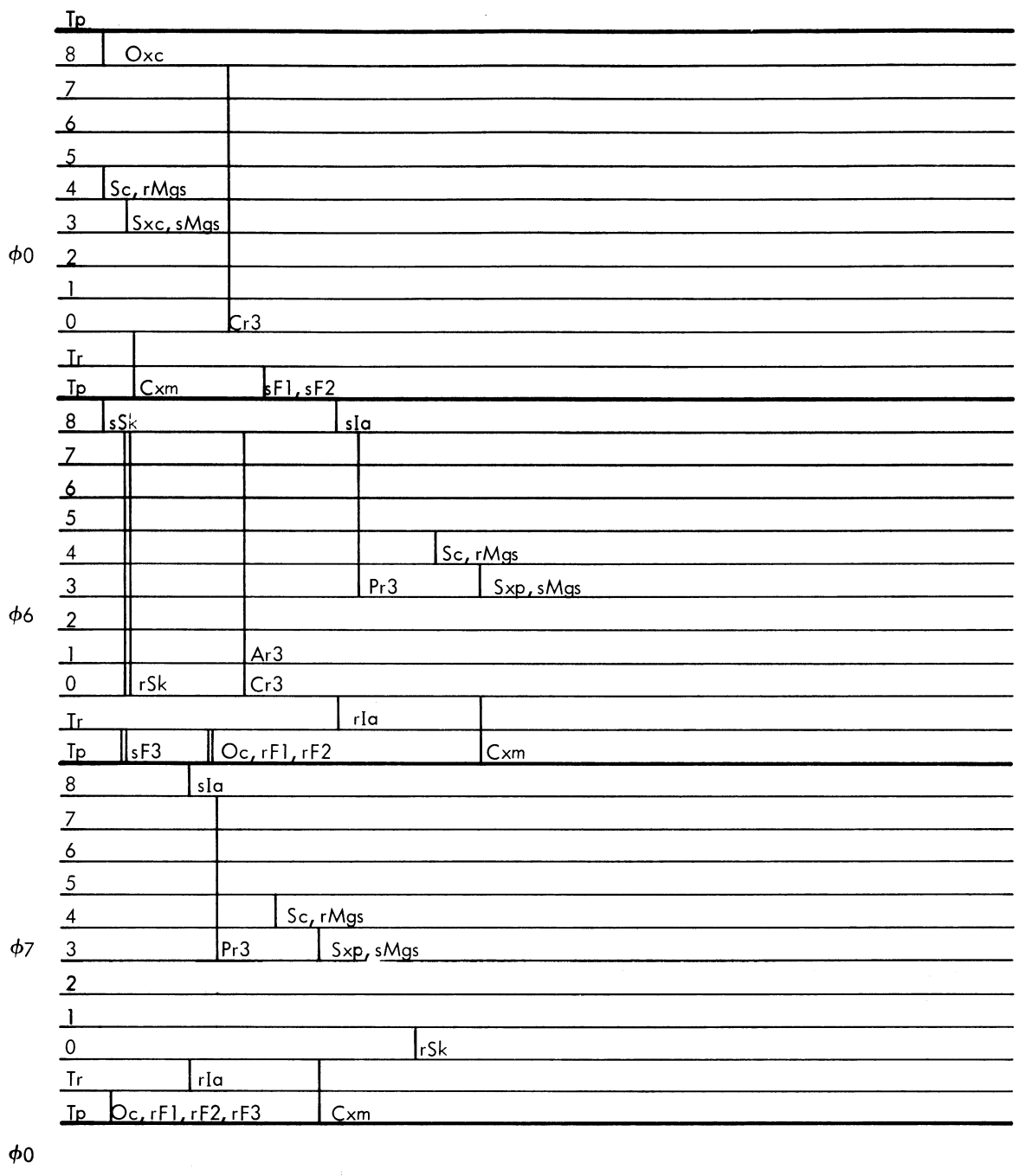


Figure 3-69. SKA Instruction, Timing Diagram

3-486 The original instruction parity is checked during phase 0 while the C-register is recirculating. A memory cycle is started in phase 0 to place the operand in the C-register, and operand parity is checked in phase 6 while Cr3 is active.

3-487 The Sk flip-flop is unconditionally set at Ø6 T8. If at any time during phase 6 T7 through T0, a corresponding pair of ONE's is observed in the A- and C-registers, Sk is reset and a skip is not performed. The equation for resetting Sk is as follows:

$$rSk = A21\ C21 + A22\ C33 + A23\ C23 + \ldots$$

The A- and C-registers are examined three bits at a time during Ar3 and Cr3.

3-488 If Sk contains a ZERO at Ø6 Tp, implying that at least one corresponding pair of ONE's has been found in A and C, sF3 is inhibited and Oc, RF1 and rF2 are enabled. This takes the phase counter to phase 0, ending the SKA instruction. The P+1 instruction which has been placed in the P-register during phase 6 is transferred to the C-register at Ø6 Tp.

3-489 If Sk remains set at Ø6 Tp, implying that no corresponding pair of ONE's has been found in A and C, F3 is set and F1 and F2 remain set. The phase counter is now in phase 7, and the P-register is increased by one to make the P+2 memory address available for access. When the logic goes to phase 7 to access the P+2 instruction, the P+1 instruction accessed during phase 6 is disregarded.

3-490 Skip if A Greater Than Memory (SKG, Code 73)

Ø0 - Ø6 ⟨Ø7⟩ Ø0 (See figure 3-70)

3-491 The SKG instruction algebraically compares the contents of the A-register with the contents of the effective memory location. If the contents of A are greater than the contents of the effective memory location, the next instruction in sequence is skipped and the following instruction is executed. If the contents of A are less than or equal to the contents of the effective location, the next instruction in sequence is executed.

3-492 The original instruction parity is checked during phase 0 while the C-register is recirculating. A memory cycle is started in phase 0 to place the operand in the C-register, and operand parity is checked in phase 6 while Cr3 is active. Full adder carry flip-flop Cz is set in phase 0 in order to add one to the one's complement of the contents of the A-register. The result is the two's complement of A. The one's complement of A is obtained by presenting the reset outputs of the A-register flip-flops to the full adder during phase 6.

3-493 Adding the two's complement of the contents of the A-register to the contents of the C-register in effect subtracts A from the contents of the effective memory location, now in C. The result is placed in the C-register. If A is greater than C, the result is negative and Sk is set at Ø6 Tr. The set term for Sk is qualified by C0 ⊕ Ix; C0 contains the sign bit of the result of subtraction. The Ix flip-flop is set at Ø6 T0 (the last stage of addition) if Ofe is true:

$$Ofe = Add\ 1\ \overline{Xz1}\ \overline{Yz1} + \overline{Add\ 1}\ Xz1\ Yz1$$

3-494 The exclusive OR term C0 ⊕ Ix determines whether the bit in C0 means a true positive or negative number or whether the subtraction has caused an overflow into the sign bit. If no overflow has taken place, Ix is false and Sk is set or not set according to the state of C0. The following examples illustrate this case:

(C)	(A)		(C)	(A)	
+8	+10	A > C	+10	+8	A < C
01000		C0 = 1	01010		C0 = 0
10110		Ix = 0	11000		Ix = 0
11110		Sk = 1	100010		Sk = 0
Negative result			Positive result		

3-495 An overflow into the sign bit (Ofe) sets Ix. In this case, the sign bit has moved out of the register; a ONE in C0 represents a positive number, and a ZERO in C0 represents a negative number:

(C)	(A)		(C)	(A)	
-10	+8	A > C	+10	-8	A < C
10110		C0 = 0	01010		C0 = 1
11000		Ix = 1	01000		Ix = 1
101110		Sk = 1	10010		Sk = 0
Negative result			Positive result		

3-496 If a positive result of subtraction leaves Sk reset at Ø6 Tr, the phase counter is taken to phase 0, ending the SKG instruction. The P+1 instruction which has been placed in the P-register during phase 6 is transferred to the C-register at Ø6 Tp.

3-497 If a negative result of subtraction causes Sk to be set at Ø6 Tr, F3 is set, F1 and F2 remain set, and Oc is inhibited. The phase counter is now in phase 7, and the P-register is increased by one to make the P+2 memory address available for access. When the logic goes to phase 7 to access the P+2 instruction, the P+1 instruction accessed during phase 6 is disregarded.

3-498 Difference Exponents and Skip (SKD, Code 74)

Ø0 - Ø6 ⟨Ø7⟩ Ø0 (See figure 3-71)

3-499 The SKD instruction subtracts the contents of bits 15 through 23 of the effective memory location from bits 15 through 23 of the B-register. The absolute magnitude of the difference is stored in bits 15 through 23 of the index register. The contents of bits 0 through 14 of the index

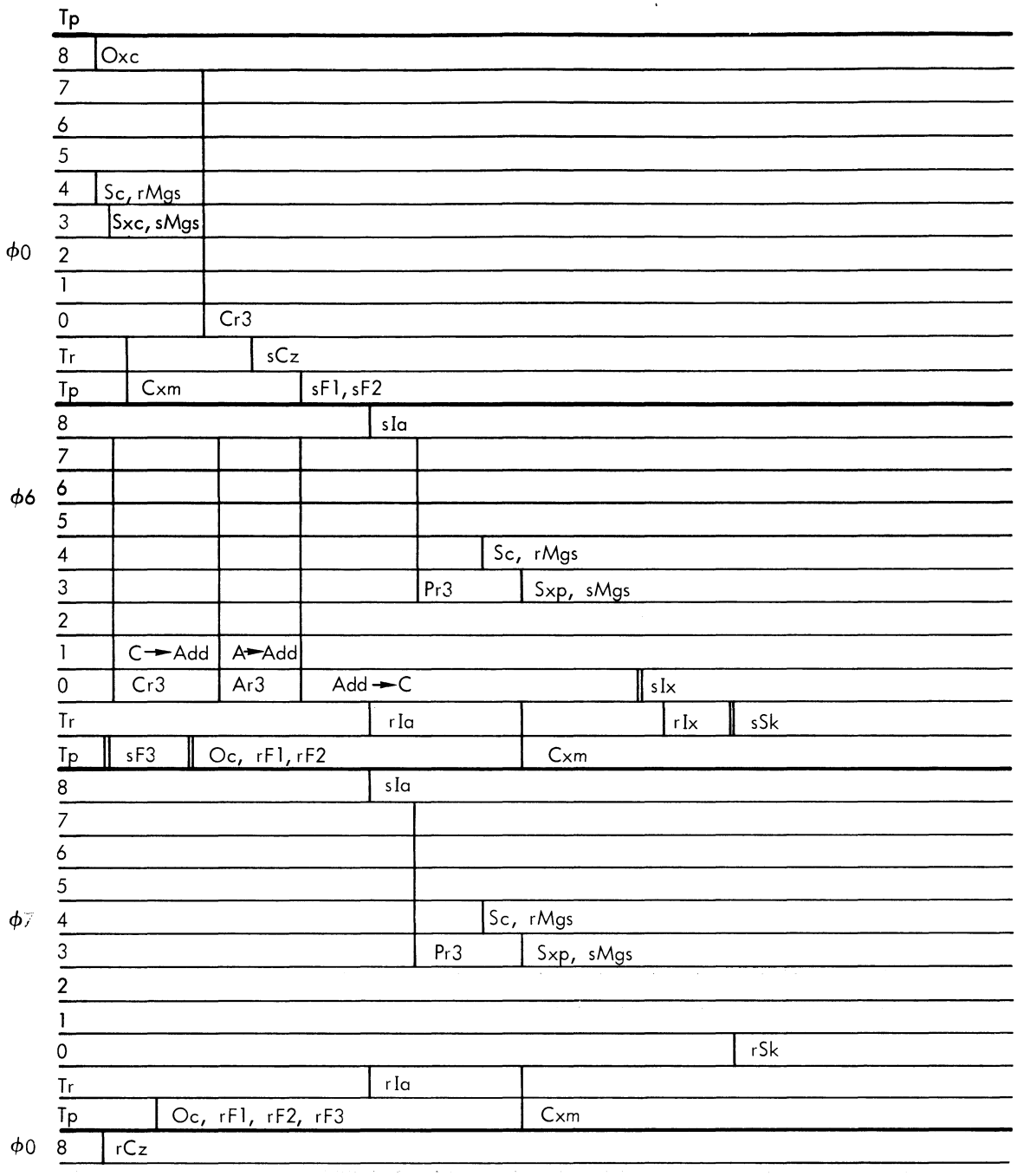


Figure 3-70. SKG Instruction, Timing Diagram

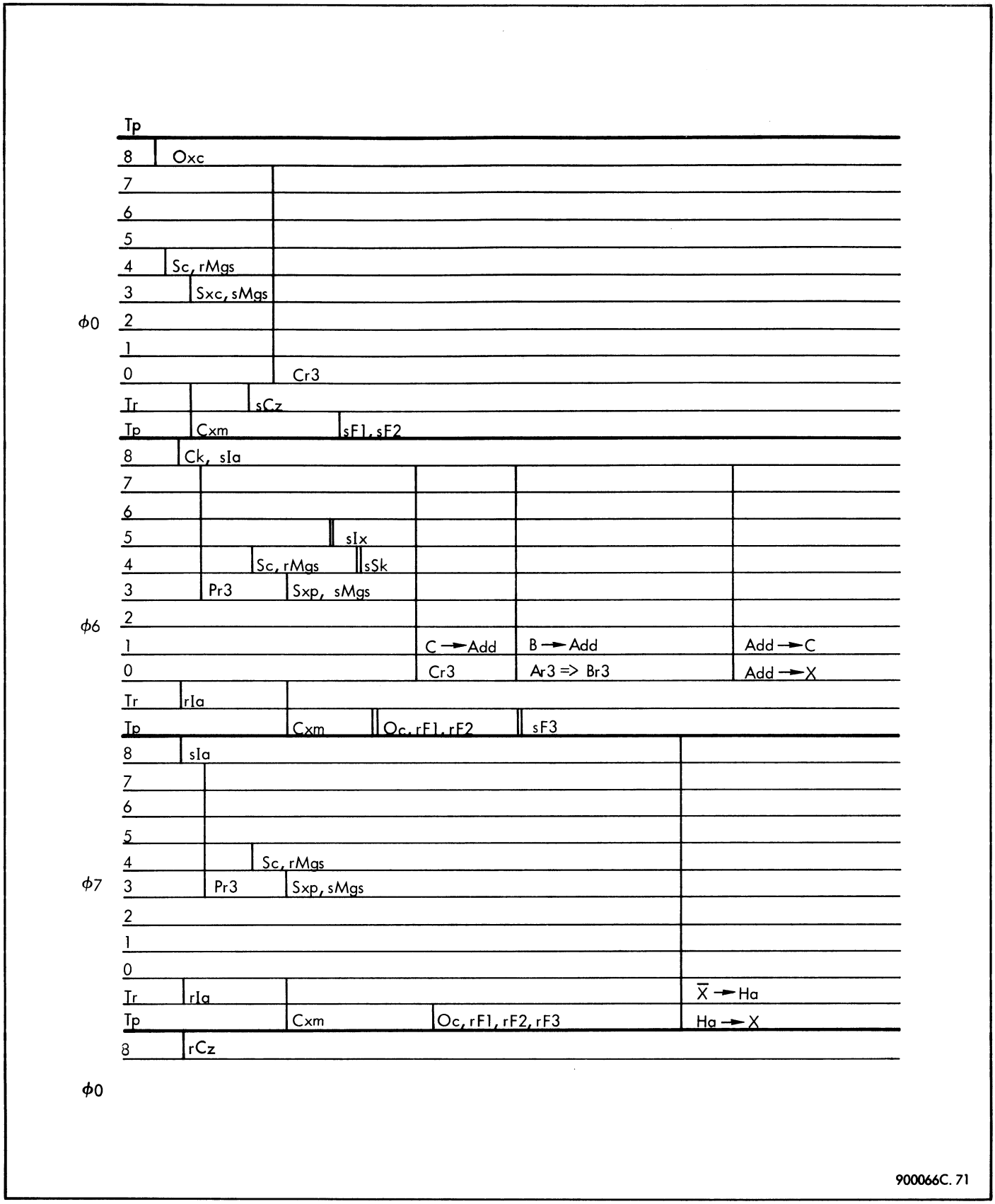


Figure 3-71. SKD Instruction, Timing Diagram

register are destroyed. If the 9-bit contents of M are less than or equal to B, the next instruction in sequence is executed. If the contents of M are greater than B, the next instruction in sequence is skipped and the following instruction is executed.

3-500 The original instruction parity is checked during phase 0 while Cr3 is active. A memory cycle is started in phase 0 to place the operand in the C-register, and operand parity is checked in phase 6 during Cr3. Full adder carry flip-flop Cz is set in phase 0 in order to add one to the one's complement of the C-register. The result is the two's complement of C. The one's complement of C is obtained by generating Ck at Ø6 T8, thereby inverting the C-register flip-flops. The inverted outputs of the C-register and the B-register outputs are presented to the full adder. The result is placed in the C-register and the index register.

3-501 Adding the two's complement of the contents of the C-register to the contents of the B-register in effect subtracts the contents of the effective memory location from B. If C is greater than B, the result is negative and Sk is set at Ø6 T4. The set term for Sk is qualified by C0 $\oplus$ Ix and is explained in detail in paragraphs 3-490 through 3-497 in the description of the SKG instruction. The C0 flip-flop contains the most significant (sign) bit of the result, since the adder outputs are fed into C as well as X.

3-502 If a positive result of subtraction indicating that B is greater than C, leaves Sk reset at Ø6 T4, the phase counter is taken to phase 0. This ends the SKD instruction. The P+1 instruction address has been placed in the P-register during phase 6 via Pr3 and carry flip-flop Ia. Its contents are transferred to the C-register at Ø6 Tp. The number in the X-register, being positive, is the required value.

3-503 If the result of subtraction is negative, C is greater than B and Sk is set at Ø6 T4. The F3 flip-flop is set, F1 and F2 remain set, and Cc is inhibited. The phase counter is now in phase 7, and the P-register is increased by one with carry flip-flop Ia and Pr3 to make the P+2 memory address available for access. When the logic goes to phase 7 to access the P+2 instruction, the P+1 instruction accessed during phase 6 is disregarded.

3-504 Since the absolute value of the result of subtraction is required in X, and since the X-register contains a negative number in phase 7, the two's complement of the number must be obtained. This is done by presenting the inverted outputs of the X-register to the half adder and adding one with half adder carry flip-flop Hz. The carry flip-flop is set at Ø7 T8. The outputs of the half adder are shifted into the X-register.

3-505 Load B (LDB, Code 75) Ø0 - Ø6 - Ø0
(See figure 3-72)

3-506 The LDB instruction loads the contents of the effective memory location into the B-register.

3-507 The original operand parity is checked during Ø0 Cr3. The operand is placed in the C-register at Ø0 Tp, and operand parity is checked during phase 6 while Cr3 is true. While the Ar3 signal shifts the B-register right three bits at a time in phase 6 and the C-register recirculates, the contents of the C-register are transferred to the B-register. Setting carry flip-flop Ia at Ø6 Tp enables the contents of the P-register to be increased by one during Pr3. A memory cycle is started in phase 6 to access the next instruction in sequence.

3-508 Load A (LDA, Code 76) Ø0 - Ø6 - Ø0
(See figure 3-73)

3-509 The LDA instruction loads the contents of the effective memory location into the A-register.

3-510 The original operand parity is checked during Ø0 Cr3. The operand is placed in the C-register at Ø0 Tp, and operand parity is checked during phase 6 while Cr3 is true. While the Ar3 signal shifts the A-register right three bits at a time and Cr3 recirculates the C-register, the contents of the C-register are transferred to the A-register. Setting carry flip-flop Ia at Ø6 Tp enables the contents of the P-register to be increased by one during Pr3. A memory cycle is started in phase 6 to access the next instruction in sequence.

3-511 Copy Effective Address into Index Register (EAX, Code 77) Ø0 - Ø6 - Ø0 (See figure 3-74)

3-512 The EAX instruction copies the address of the effective memory location into the index register.

3-513 The original instruction parity is checked in phase 0 during Cr3. Decoding of the instruction in the C-register sets Eax at T8, and Eax is then used to enable the transfer of effective address from the C-register to the X-register. The outputs of the C-register are presented to the Yz inputs of the full adder, and the adder outputs are enabled into the X-register as follows:

$$Xwc = Eax\ \overline{Ju}$$

$$sXw1 = Add\ 1\ Xwc\ (F1 + \overline{Q6})$$

Bits 2 and 3 of the full adder are gated into the second and third stages of the index register by Xwc and $\overline{Ts}$.

3-514 If the instruction is to be indexed, bit 1 of the C-register is set and the output is used to set Ix. When Ix is true, the X-register outputs are gated into the Xz inputs of the full adder and C is added to X before the sum is placed in the index register. Since the contents of the effective memory location are not needed, the information loaded into the C-register from memory at Ø0 Tp is disregarded.

3-515 The phase counter advances to phase 6, and a normal end cycle takes place. Carry flip-flop Ia is set so that one is added to the contents of the P-register during Pr3. The next instruction in sequence is placed in the C-register at Ø6 Tp, and the phase counter goes to phase 0.

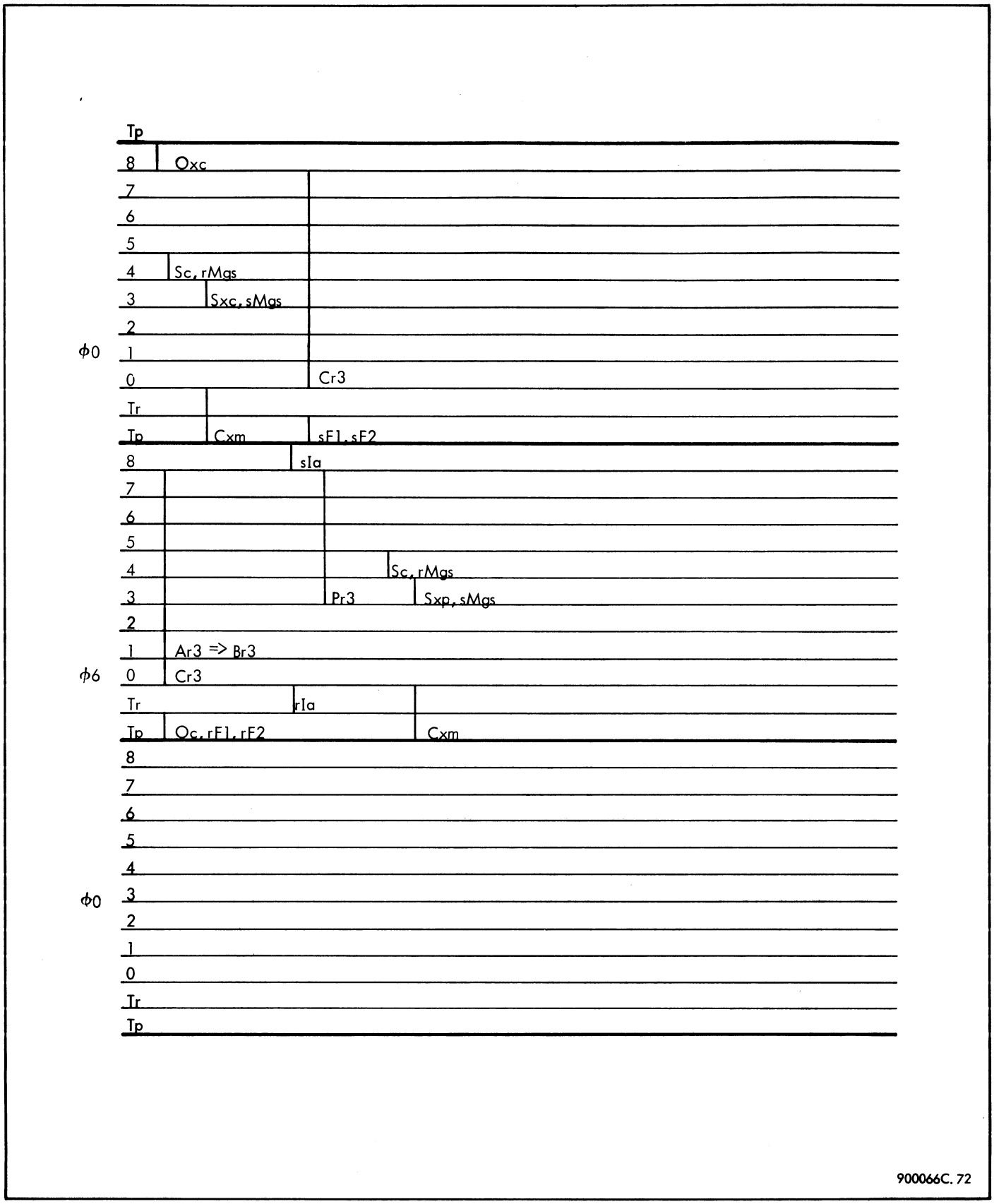


Figure 3-72. LDB Instruction, Timing Diagram

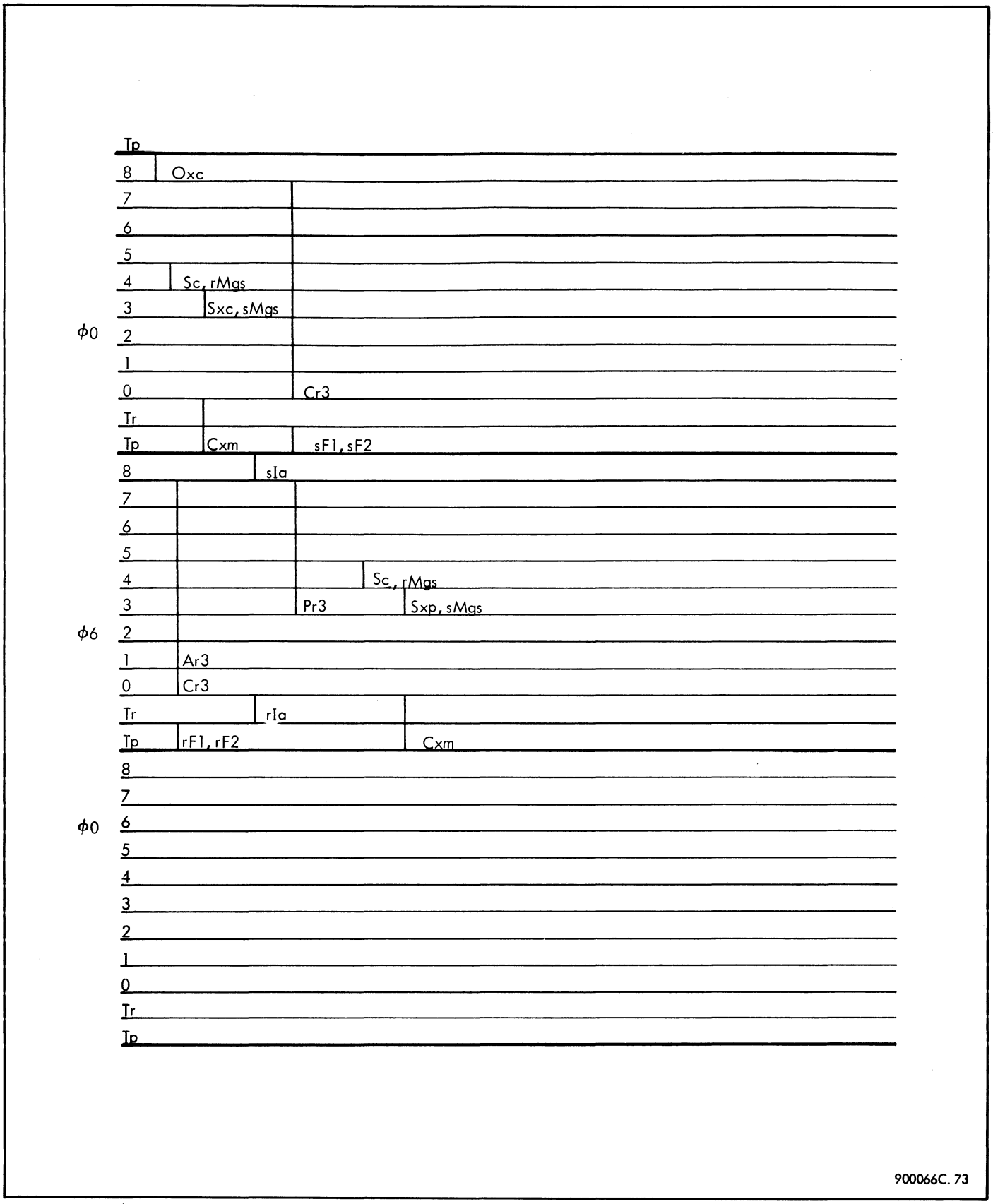


Figure 3-73. LDA Instruction, Timing Diagram

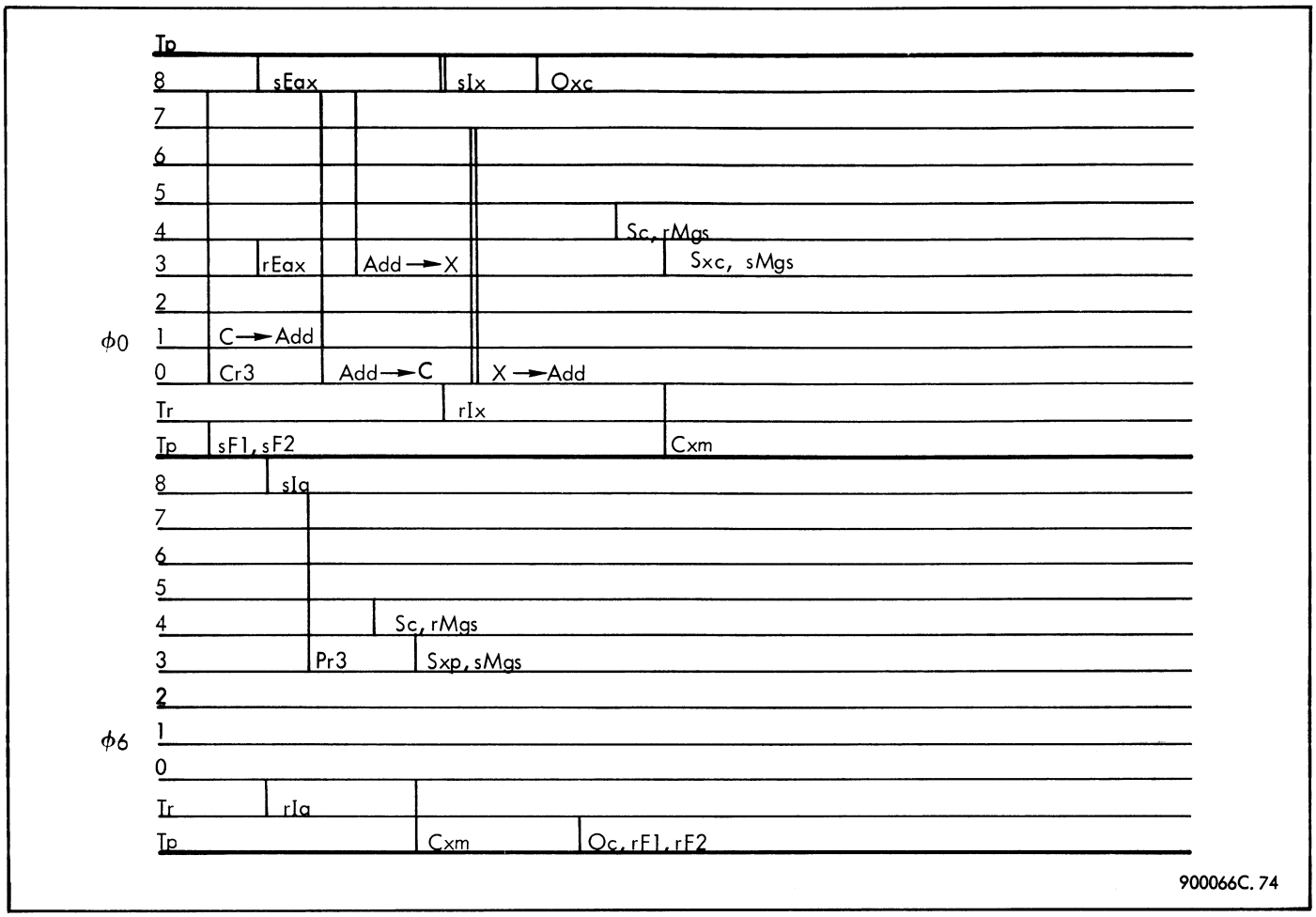


Figure 3-74. EAX Instruction, Timing Diagram

3-516 MEMORY OPERATION

3-517 The 930 Computer memory system is a parallel-transfer, random access, four-wire coincident current, magnetic core memory. The memory is expandable from 4,096 to 32,768 words. Memory banks are available in capacities of 4,076; 8,192; and 16,384 words. Two 16K memories may be used to expand the capacity to 32,768 words. The details of magnetic core storage theory are given in the Troubleshooting Manual Computer Memories for Computer Models 930/9300, SDS 900865.

3-518 The 930 Computer can have up to two memory doors. Both memories share the same access path; hence, only one memory can be addressed in any one memory cycle. If both memories also contain the optional Multiple Access to Memory path, memory overlap can be performed by reading from one memory with the central processor and simultaneously reading from the second memory with the DACC, DMC, or MIC. The direct access channels have priority over the central processor; therefore, if the direct access channel and the CPU address the same memory, the clock to much of the CPU is stopped until the direct access channel has finished with that memory.

3-519 The memory contains all of the circuits necessary for receiving addresses, establishing priority of access paths, and decoding the address to select a single address location in the core stack. The memory does not have an internal address register, and the incoming address information must be held stable during the memory cycle. The memory contains an internal M-register to store data being written into or read from memory. Circuits are included for receiving and transmitting data and for sensing and inhibiting each bit in the core stack.

3-520 A block diagram of the memory system is shown in figure 3-75.

3-521 BASIC OPERATION

3-522 The basic unit in a 4K, 8K, and 16K memory stack is a 16,384-core frame. Each core frame is threaded with 128 X lines and 128 Y lines as shown in figure 3-76. The core frame is divided into four mats, or quadrants, each containing 4,096 cores. Each mat has its individual sense winding and inhibit winding. The sense windings are on the diagonal of the core frame. The inhibit windings run parallel to the X lines in quadrants 0 and 2 and parallel to the Y lines in quadrants 1 and 3.

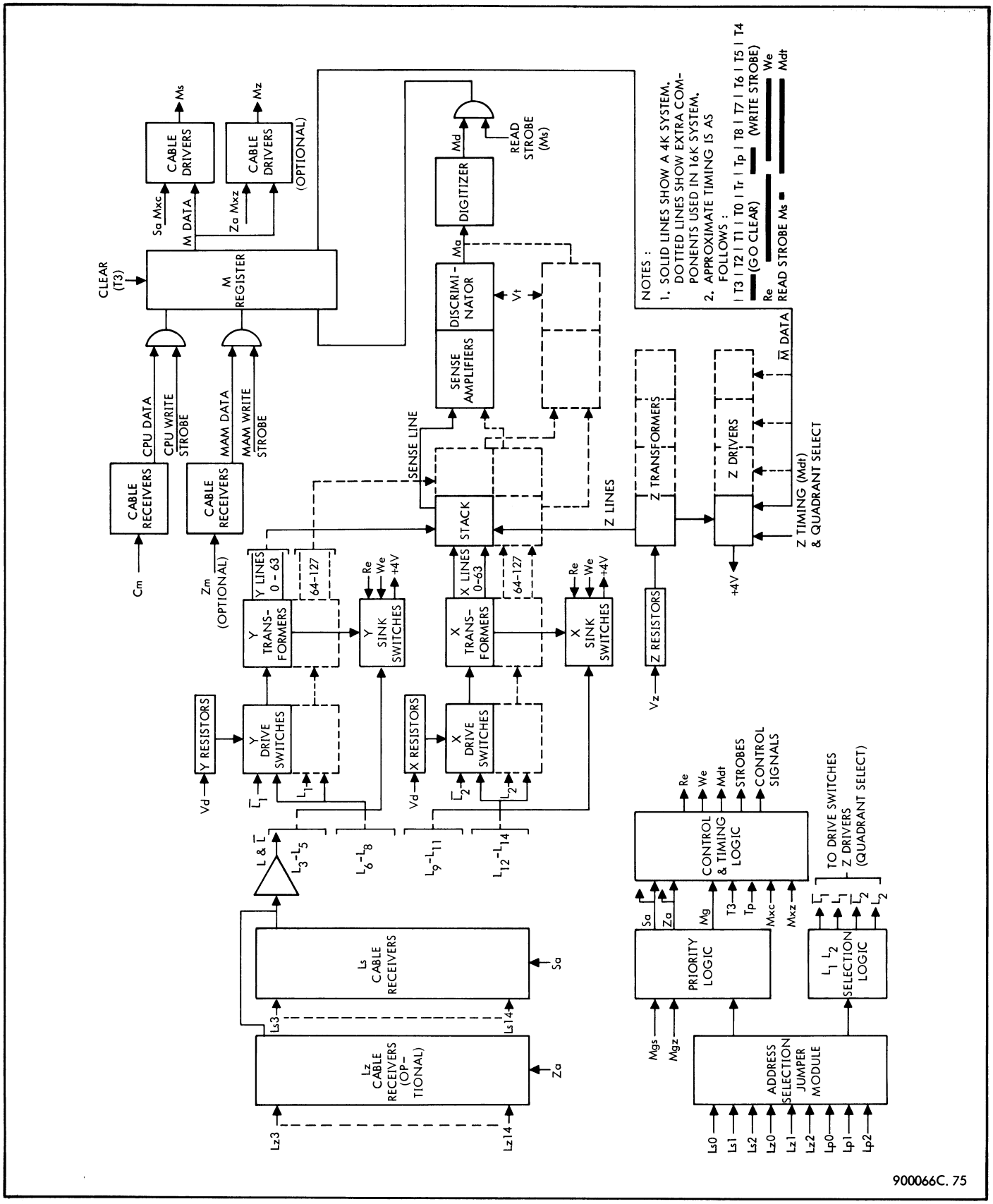


Figure 3-75. 930 Computer Memory System, Block Diagram

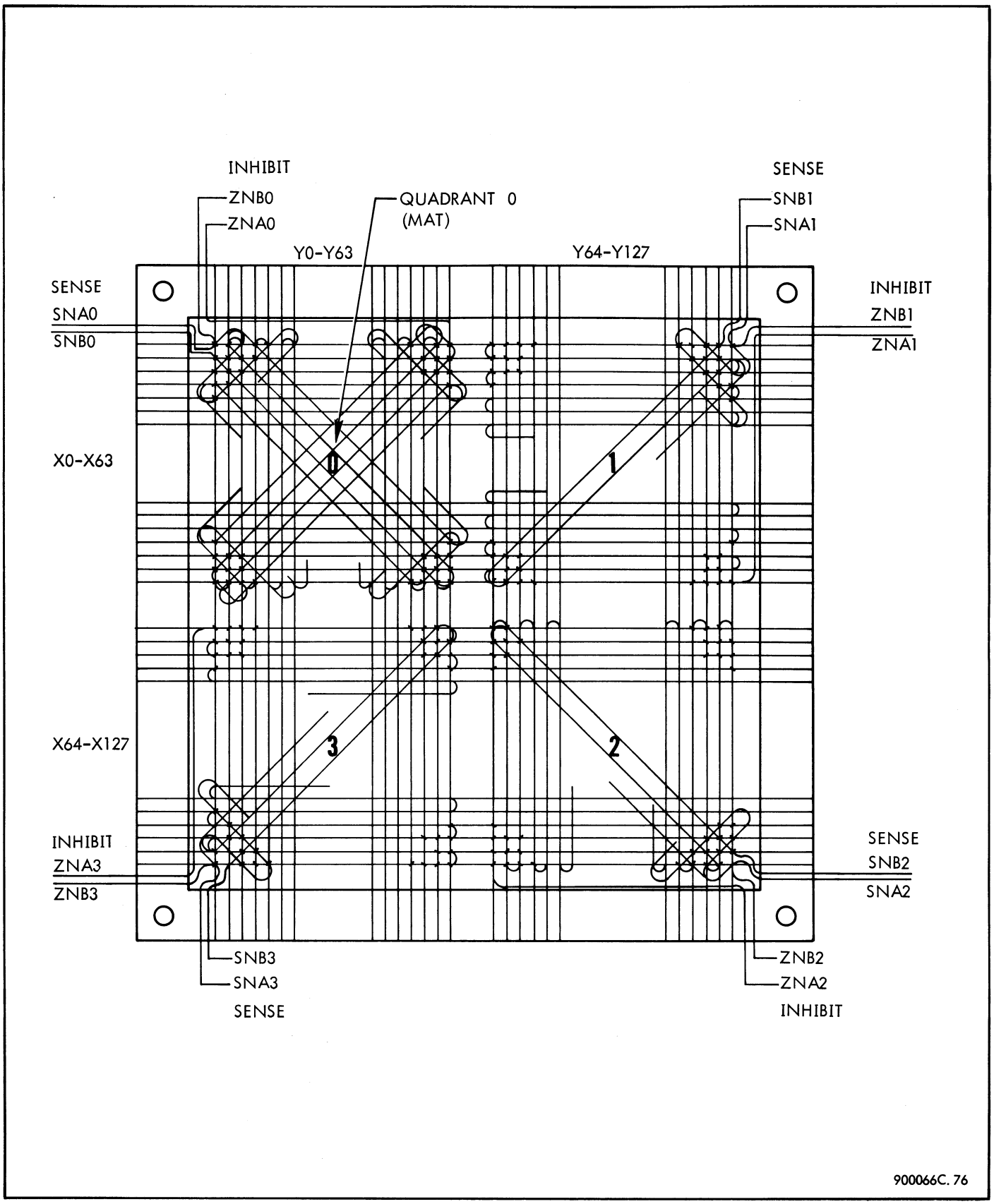


Figure 3-76. Memory Core Frame

3-523 The 4K memory stack contains seven core frames, or 28 quadrants. All of the cores in one quadrant represent the same bit in each of 4,096 words. For example, quadrant 1 of frame 0 contains bit 13 of 4,096 24-bit words. The bits are physically arranged as indicated in figure 3-77. The 28 quadrants represent 23 bits, a parity bit, and 3 spares. The X and Y lines are threaded through one side of the stack, and are then folded back to traverse the other side. On the top of the stack, as represented in the figure, is a connector frame; on the bottom is an end frame, used to fold back the X and Y lines without transposing them.

3-524 The 8K memory stack contains 13 core frames, or 52 quadrants. Each bit of the memory word occupies two adjacent quadrants to make up 8,192 words. The bits are arranged as shown in figure 3-78 so that the stack is comparable to two 4K memories mounted side by side. Of the 52 quadrants, 48 represent two 24-bit words, 2 quadrants represent parity bits, and 2 quadrants on the 13th frame are spares. The Y lines are folded back in order to traverse the entire stack. The frames are terminated with a connector frame and an end frame, as in the 4K memory. The end frame is used to fold back the Y lines and also to transpose them to reduce pickup between wires.

3-525 The 16K stack contains 25 core frames, or 100 quadrants. Each bit of a memory word occupies four quadrants in one core frame to make up 16,384 words. The bits are arranged consecutively through the stack as shown in figure 3-79. Of the 100 quadrants, 96 quadrants represent four 24-bit words and 4 quadrants represent the 4 parity bits. The frames are terminated with a connector frame and a transposition frame. The transposition frame transposes the X and Y wires to reduce pickup between wires.

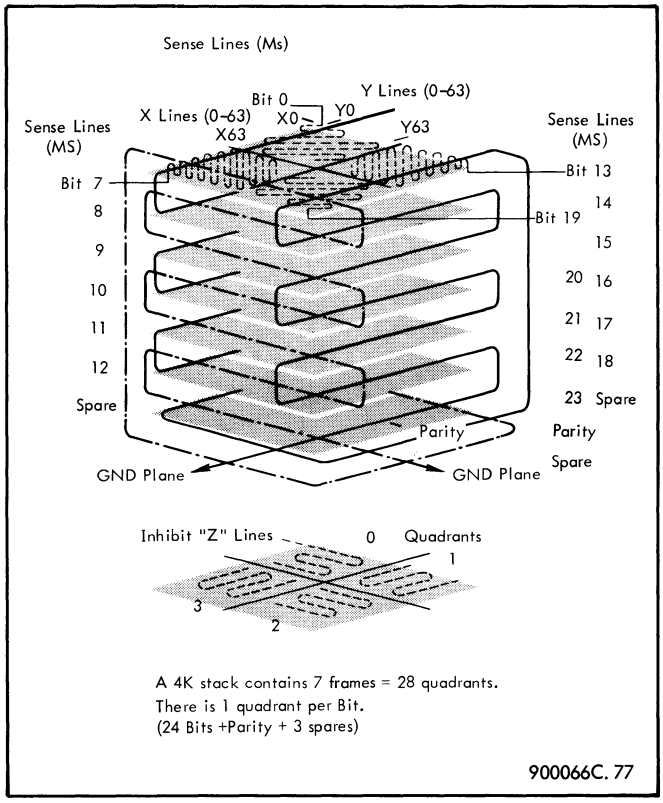


Figure 3-77. 4K Stack Layout

3-526 MEMORY CYCLES

3-527 Two types of memory cycles are used in the 930 Computer: a read-restore cycle and a clear-write cycle. A read-restore cycle reads data from core memory and restores the same data to the location from which it was read. A clear-write cycle changes the data in memory by clearing the memory location and entering new data. The duration of a memory cycle is 1.75 microseconds, or 11 computer clock pulses of 159 nanoseconds each. Before a memory cycle begins, the dc flip-flops of the M-register are reset by the leading (positive-going) edge of T3.

3-528 The memory cycle is started by setting the Mgs flip-flop for central processor access or Mgz for direct access:

$$sMgs = T3\ \overline{St}\ \overline{(\overline{F1}\ \overline{F3}\ \overline{Tsm})}\ \overline{(\emptyset 5\ Bc23)}$$

$$rMgs = T4$$

$$sMgz = Zrq\ T3\ \overline{St}$$

$$rMgz = T4$$

where Zrq is the request signal from the direct access channel.

3-529 Read-Restore Cycle

3-530 The read portion of the read-restore cycle is started at the trailing edge of $\overline{T3}$, when one-shot Du1 is set. This produces a delay of 80 nanoseconds to allow for propagation time through the priority logic and to allow the selected X and Y drive switches to turn on and saturate completely. The Du1 signal in turn triggers the Du2 one shot:

$$Du1 = \overline{T3} \text{ (about 80 nsec)}$$

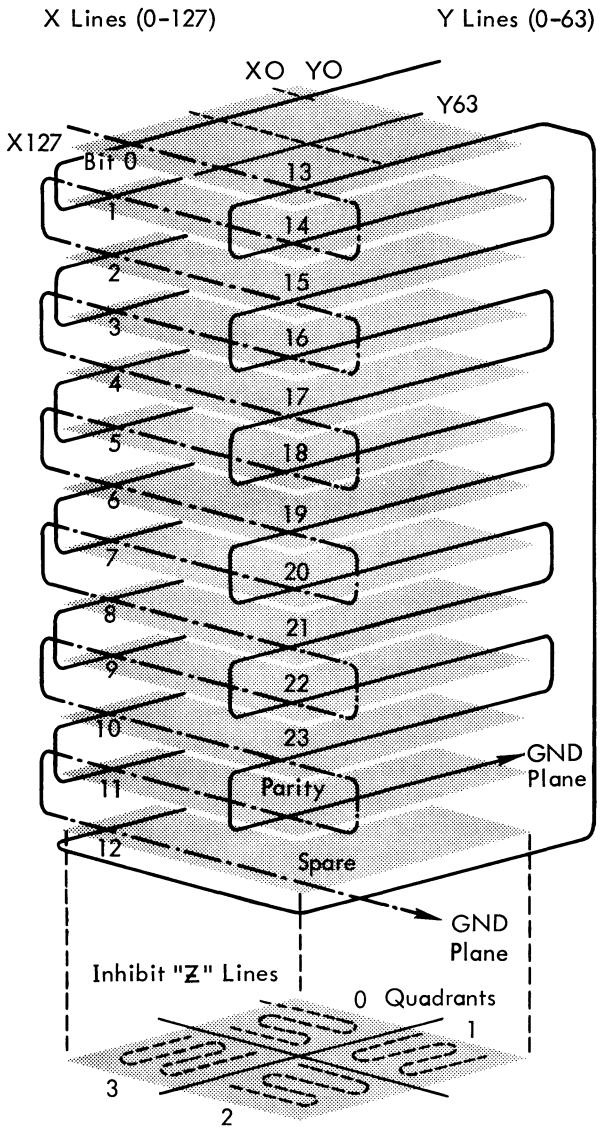
$$Du2 = Du1 \text{ (about 620 nsec)}$$

A memory cycle timing diagram is given in figure 3-80.

3-531 In order to generate a read signal, Mg must be true, indicating that the memory has been requested to perform a cycle on the address presented on the Ls or Lz lines. Read enable comes true with the trailing edge of Du1:

$$Re = Mg\ Du2\ \overline{Du1}$$

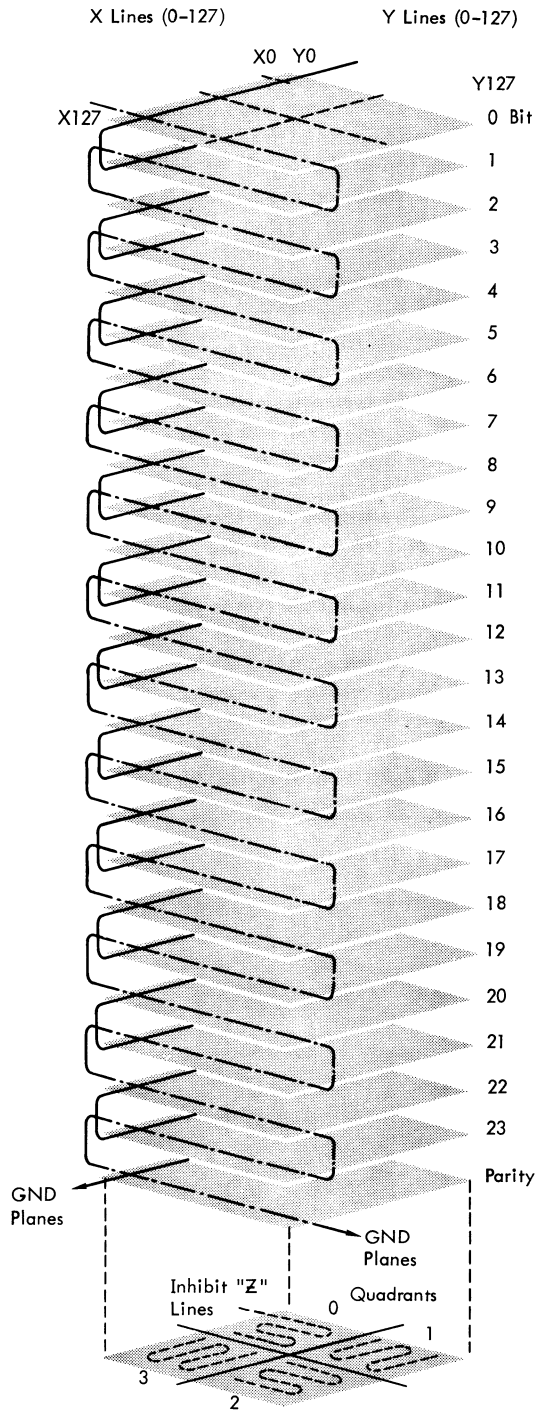
Read enable turns on the selected X and Y read sink switches. Together with the drive switches selected by the addressing logic, the sink switches cause read drive half currents to flow in one X line and one Y line in the core stack. Full current is applied to each of the 25 cores in the selected memory location for approximately 540 nanoseconds, tending to switch all the cores in the word to ZERO. If a particular core was in the ZERO state before the read half currents were applied, it does not change state, but produces a small ZERO voltage signal. If the core was in the ONE state before the read half currents



An 8K stack contains 13 frames = 52 quadrants. There are 2 quardrants per Bit (24 Bits + Parity Bit + 2 spare quadrants.

900066C. 78

Figure 3-78. 8K Stack Layout



A 16K stack contains 25 frames + an empty transposition plane. There are 4 quadrants per bit.

900066C. 79

Figure 3-79. 16K Stack Layout

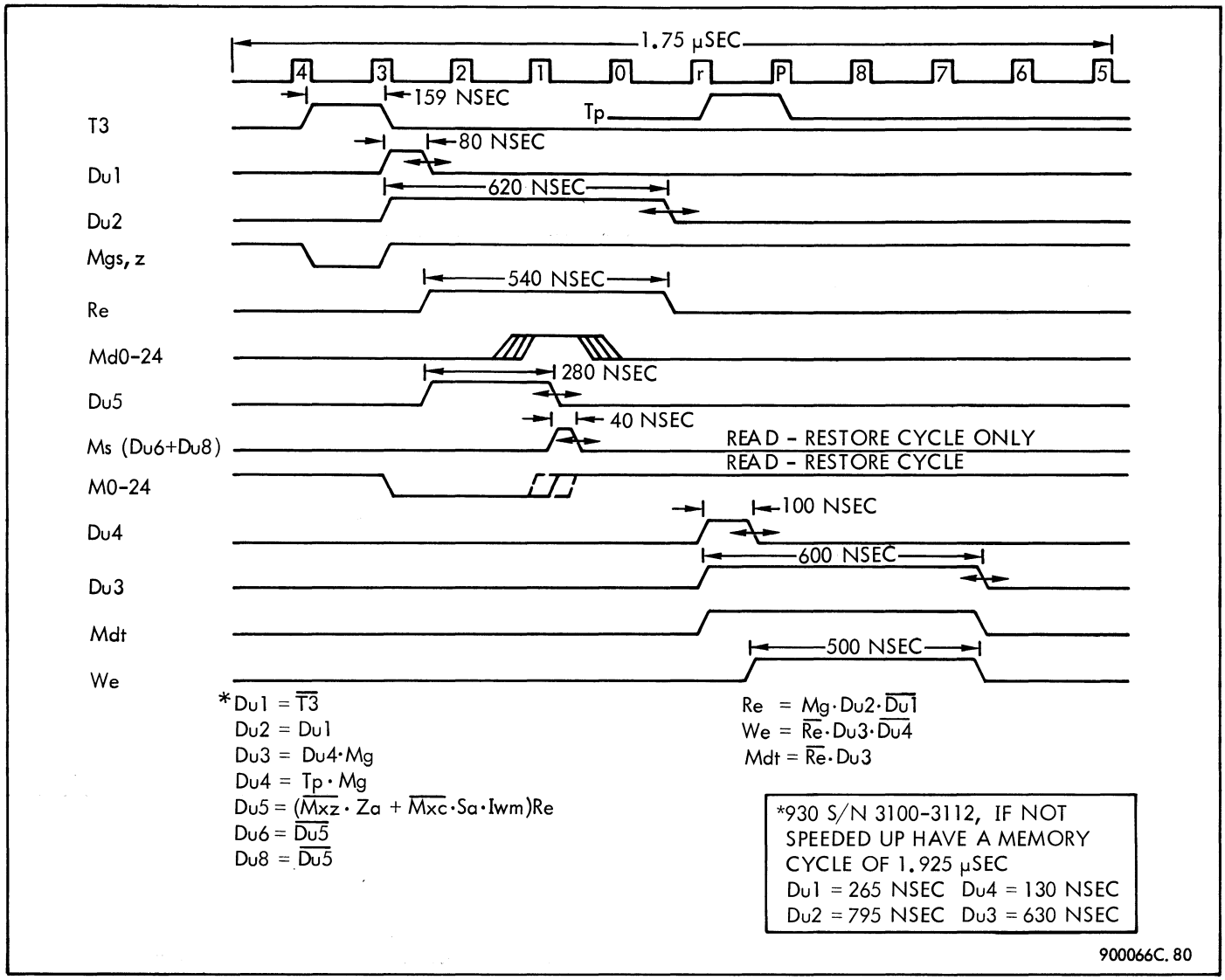


Figure 3-80. Memory Cycle Timing

were applied, it switches, producing a ONE voltage signal, which is amplified and detected by the sense amplifier. At the end of the read phase, all the cores in the selected word are in the ZERO state.

3-532 The leading edge of Re sets strobe timing one-shot Du5. The output of Du5 is gated with $\overline{Mxz}$ Za (direct access read cycle access), $\overline{Mxc}$ Sa (central processor read cycle access), or Iwm (computer attempts to write in a locked out block, so that data read out from cores must be restored). This inverting gate generates $\overline{Du5}$ so that the trailing edge of Du5 can set strobe width one-shots Du6 and Du8. These, in turn, produce memory strobe signals Ms1 through Ms4, which strobe the amplified and digitized core signals Md0 through Md24 into the M-register flip-flops.

3-533 In some memories the Ms1 through Ms4 strobes occur simultaneously, as illustrated in figure 3-81. The equations are as follows:

Du5 = about 280-nsec pulse beginning at positive-going edge of Re

Du6 = 40-nsec pulse beginning at positive-going edge of Du5

Du8 = 40-nsec pulse beginning at positive-going edge of Du5

Ms1 = Du6

Ms2 = Du6

Ms3 = Du8

Ms4 = Du8

3-534 In later memories, the strobes are staggered by the fixed circuit propagation delay of the Du6 one-shot, as shown in figure 3-82. This stagger improves the matching of the strobes to the time when the core signals for different bits appear on the sense windings. Bit 0 occurs 50 to 60

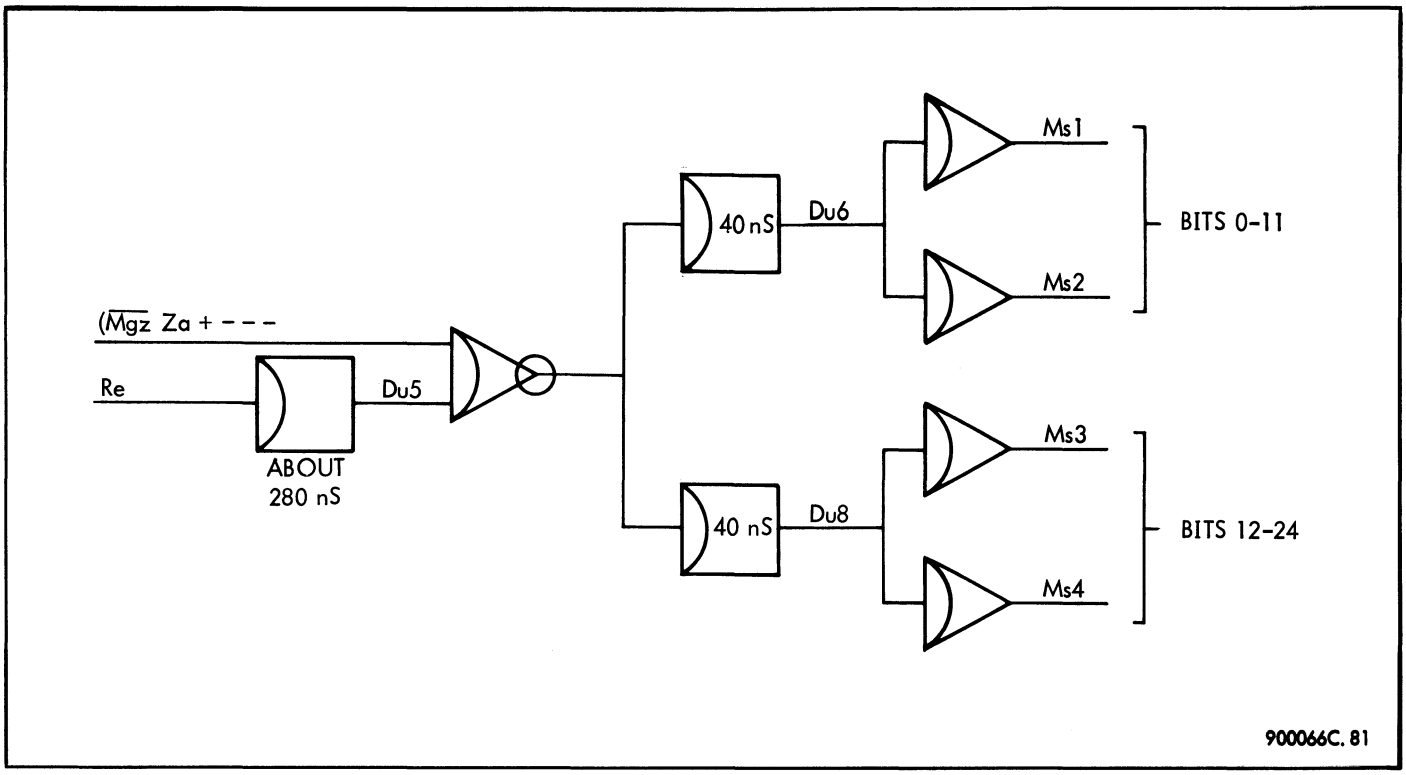


Figure 3-81. Simultaneous Strobes

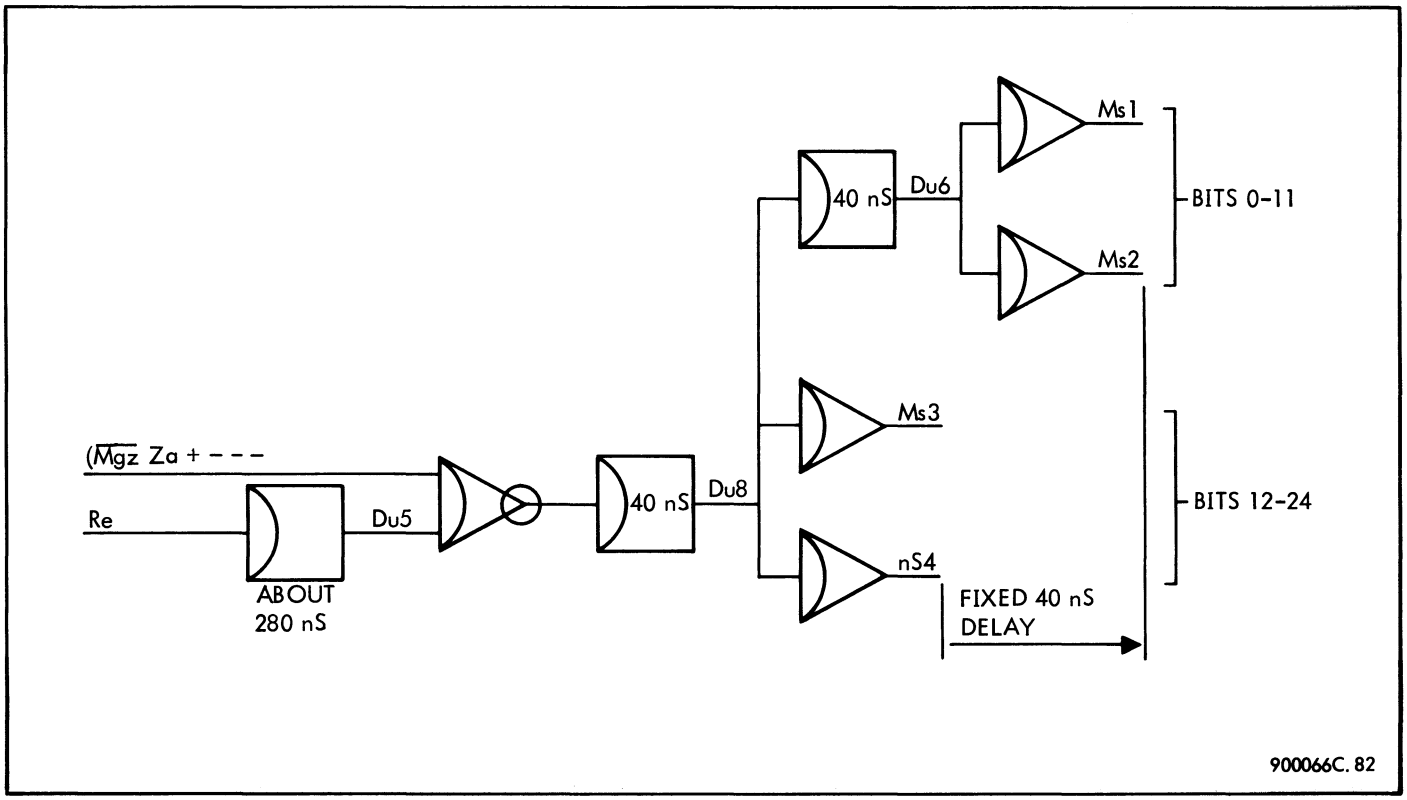


Figure 3-82. Staggered Strobes

nanoseconds later than bit 24 in an 8K or 16K stack and 30 nanoseconds latter in a 4K stack. With this configuration,

Du6 = 40-nsec pulse beginning at positive-going edge of Du8

3-535 For an S, or central processor, access, the M-register outputs are gated onto the central processor data bus (Ms data) by Sa $\overline{Mxc}$; therefore, data is transmitted to the central processor as soon as the M flip-flops are set by the data strobed from the sense amplifier outputs. Similarly, for a Z, or direct, access, the outputs are gated onto the MAM data bus (Mz data) by Za $\overline{Mxz}$:

$$sM0\text{-}24 = Md0\text{-}24$$

$$rM0\text{-}24 = T3$$

$$M0\text{-}24s = (Sa\ \overline{Mxc})\ M0\text{-}24$$

$$M0\text{-}24z = (Za\ \overline{Mxz})\ M0\text{-}24$$

3-536 Since the read cycle destroys the contents of the accessed location, it is necessary to restore the data. This is done by writing the M-register data back into the memory location from which it was taken.

3-537 The write phase begins when the leading edge of Tp sets the Du4 one-shot, provided that the Mg enabling signal is true. The Du4 signal sets Du3, which generates memory digit timing signal Mdt if Re is false. The Mdt signal turns on the inhibit drivers of the bits which, under the control of the data in the M-register, are to be rewritten as ZERO's. The false outputs of the M-register flip-flops are applied to the respective inhibit drive inputs so that a ZERO bit in the M-register causes inhibit current to oppose the drive line current and hold the core in the ZERO state. The 100-nanosecond delay in Du4 ensures that the inhibit currents are fully established in the stack before the X and Y write drive half currents are turned on. After this delay is complete, the trailing (positive-going) edge of $\overline{Du4}$ produces write enable signal We. The We signal is applied to the selection gates of the write sink switches. At this time, full current is passed through the selected cores, driving them to the ONE state. The write phase is complete when We and Mdt are turned off by the trailing edge of Du3. The Mdt signal is about 600 nanoseconds long, and We is about 500 nanoseconds long.

Du6 = 600-nsec pulse beginning at positive-going edge of Du4 Mg

Du4 = 100-nsec pulse beginning at positive-going edge of Tp Mg

We = $\overline{Re}$ Du3 $\overline{Du4}$

Mdt = $\overline{Re}$ Du3

3-538 Clear-Write Cycle

3-539 The clear-write cycle is used when storing words in memory from the computer or direct access channel registers. A write cycle is produced if Mxc is true during a central processor access or Mxz is true during a MAM, or direct access. The M-register is reset to the ZERO state at the end of the previous cycle by the leading edge of T3. The write cycle begins in the same way as the read cycle, with Re generated from Du1 to turn on the read drive half currents and Du5 set. However, during the clear portion of the write cycle, ($\overline{Mxz}$ Za + $\overline{Mxc}$ Sa + Iwm) is false, and memory strobes Ms1 through Ms4 are not generated. The core data is therefore cleared, and the M-register remains in the ZERO state.

3-540 At the leading edge of Tp, new data from the central processor (Cm) or direct access channel bus (Zm) is strobed into the M-register. The Cm signals become C signals in the memory logic.

$$sM0\text{-}24 = C0\text{-}24\ (Sa\ Mxc\ \overline{Iwm}\ Tp) + Zm0\text{-}24\ (Za\ Mxz\ \overline{Iwm}\ Tp)$$

If the system contains two memories, only the memory selected by the address will strobe data into its M-register. The data in the M-register is written into memory in the same manner as in the restore portion of the read-restore cycle.

3-541 ADDRESSING

3-542 The 930 Computer memory can be addressed by the standard central processor (S) path and by the optional MAM direct access (Z) path. These paths are both capable of reading from and writing into the memory. Each path consists of a two-cable bus system.

3-543 Central Processor Addressing

3-544 The S path is used by the central processor to read and write operands and instructions. It is also used for interlace by the Time Multiplexed Communication Channel (TMCC). The address from which data is to be read is transmitted to the memory on the Ls lines, which are controlled by the S-register for a central processor access and by the Ir lines for a TMCC access:

$$Ls3\text{-}14 = \overline{Tsm}\ S3\text{-}14 + Tsm\ Ir3\text{-}14$$

Ls0, Ls1, and Ls2 are similar, but the S term is derived from S1, S2, and the memory extension registers.

3-545 The fifteen /$\overline{Ls}$/ address bits, together with /Ls0/, /Ls1/, /Ls2/, /$\overline{Mgs}$/, /$\overline{Mxc}$/, /$\overline{T3}$/, and /$\overline{Tp}$/ are transmitted on the S address cable. If the system contains two memories, this cable connects to both memories.

3-546 Direct Access

3-547 The optional Z path is used by a DACC, DMC, or MIC to address memory directly. This access path resembles the S access path, except that Mgz and Mxz are used instead of Mgs and Mxc. The timing depends upon the device addressing memory via the Multiple Access to Memory (MAM) path. The address to be accessed is transmitted to the memory on the Lz lines, which are controlled by the MAM Jz register, physically located in the central processor. The Jz register holds information from the Iz lines from the input/output unit for the duration of the memory cycle:

$$sJz0\text{-}14 = T3\ Iz0\text{-}14$$

$$rJz0\text{-}14 = T4$$

The fifteen /$\overline{Lz}$/ lines, together with /Lz0/, /Lz1/, /Lz2/, /$\overline{Mgz}$/, and /$\overline{Mxz}$/ are transmitted from the central processor on the Z address cable.

3-548 Priority Logic

3-549 The first stage of the priority logic detects a request for memory access. The inputs to this logic from the S access path are /$\overline{Ls0j}$/, /$\overline{Ls1j}$/, /$\overline{Ls2j}$/, and /$\overline{Mgs}$/. The small j indicates that these address terms are brought from the central processor interface into the memory logic through a jumper module. The terms are selected by the jumper module to provide the priority logic inputs with the unique address combination required for that particular memory. Each 4K, 8K, or 16K memory has its own jumper module for priority addressing. Since the interface address terms and control term represent the reset sides of the respective flip-flops in the central processor, an inverting gate is used to generate priority term Sb. When Mgs comes true, if the requested address is located in that particular memory, the equation for the request for S access signal, Sb, is as follows:

$$Sb = Mgs\ Ls0j\ Ls1j\ Ls2j$$

Similarly, Zb is generated for a Z access:

$$Zb = Mgz\ Lz0j\ Lz1j\ Lz2j$$

3-550 If only a Z access is requested or if simultaneous Z and S access are requested, Zb generates Za (Z access). If an S access and not a Z access is requested, Sb generates Sa (S access). In other words, a Z access takes priority over an S access:

$$Za = Zb$$

$$Sa = Sb\ \overline{Zb}$$

If simultaneous accesses are requested, a memory interference signal, Mit, is transmitted to the computer and input/output equipment:

$$Mit = Za\ Sb$$

The Mit signal is used to set the time-share flip-flop, Ts, in the central processor:

$$sTs = Mit\ Tr + \ldots$$

The Sb and Za terms are used to produce memory go signal Mg, which enables the memory cycle to proceed:

$$Mg = Za + Sb$$

3-551 For an S access, Sa gates the external Ls1k, Ls2k, and Ls3 through Ls14 address onto the internal L1 through L14 lines. The small k indicates that these terms are brought from the central processor interface into the memory logic through a jumper module. These address terms are selected by the jumper module to place the proper address on the internal address lines for the size of memory being addressed. Similarly, Za gates Lz1k, Lz2k, and Lz3 through Lz14 onto the L lines:

$$L1 = Lz1k\ Za + Ls1k\ Sa$$

$$L2 = Lz2k\ Za + Ls2k\ Sa$$

$$L3\text{-}14 = Lz3\text{-}14\ Za + Ls3\text{-}14\ Sa$$

3-552 Jumper Modules. The 15 address lines of the S or Z access paths can address $32,768_{10}$ memory locations. The 12 least significant address lines select an address within a 4K block. The three most significant address lines select blocks of 4K addresses as shown in table 3-19. A special ZB65 jumper module is used for each 4K, 8K, or 16K memory. The dash number of the jumper module codes the address selection function of the module. The -10 series selects 4K blocks of addresses; the -20 series selects 8K blocks; and the -30 series selects 16K blocks. The last digit is a code representing the first address in the selected address block. The jumpers cannot select overlapping blocks of addresses. For example, the 16K jumper modules can select only the 0 through 16K addresses or the 16K through 32K addresses, and do not allow for positioning a 16K memory in an address range such as 8K through 24K.

3-553 The first function of a jumper module is to select the address terms used by the priority logic described in paragraph 3-549. In order to supply an Sb access request for an particular memory stack, the $\overline{Ls0\text{-}2j}$ inputs from the jumper module to the priority logic must all be false. The jumper module converts the $\overline{Ls0}$ through Ls2 inputs from the computer to make the $\overline{Ls0\text{-}2j}$ lines false when that memory is addressed.

3-554 The ZB65 through ZB22 jumper module for the 8K through 16K address range will be used as an example of memory selection. This block of addresses (20,000 through $37,777_8$) is the only block which has $\overline{Ls0}$ Ls1 on the address lines for every location; therefore, an Sb signal derived from $\overline{Ls0}$ Ls1 will select the 8K through 16K memory only. The jumper module connects /Ls0/ to /$\overline{Ls0j}$/ and /$\overline{Ls1}$/ to /$\overline{Ls1j}$/, so that when /$\overline{Ls0}$/ /Ls1/ is true, /$\overline{Ls0j}$/ and /$\overline{Ls1j}$/

Table 3-19. Address Selection Jumper Modules

ADDRESS RANGE	ADDRESS COMBINATION TO UNIQUELY SELECT: a 4K Memory	an 8K Memory	a 16K Memory	JUMPER MODULE 4K	8K	16K
0- 4K	$\overline{Ls0}$ $\overline{Ls1}$ $\overline{Ls2}$	$\overline{Ls0}$ $\overline{Ls1}$	$\overline{Ls0}$	ZB65-10	ZB65-20	ZB65-30
4- 8K	$\overline{Ls0}$ $\overline{Ls1}$ Ls2			ZB65-11		
8-12K	$\overline{Ls0}$ Ls1 $\overline{Ls2}$	$\overline{Ls0}$ Ls1		ZB65-12	ZB65-22	
12-16K	$\overline{Ls0}$ Ls1 Ls2			ZB65-13		
16-20K	Ls0 $\overline{Ls1}$ $\overline{Ls2}$	Ls0 $\overline{Ls1}$	Ls0	ZB65-14	ZB65-24	ZB65-34
20-24K	Ls0 $\overline{Ls1}$ Ls2			ZB65-15		
24-28K	Ls0 Ls1 $\overline{Ls2}$	Ls0 Ls1		ZB65-16	ZB65-26	
28-32K	Ls0 Ls1 Ls2			ZB65-17		

are both false. Since the state of Ls2 is immaterial in the selection of the 8K through 16K memory, the /$\overline{Ls2j}$/ is grounded to make this line always false. With the above jumper connections, $\overline{Ls0}$ Ls1 on the external computer address will place false signals on /$\overline{Ls0j}$/ /$\overline{Ls1j}$/, and /$\overline{Ls2j}$/; and when /$\overline{Mgs}$/ is false, an Sb signal is generated to access the 8K through 16K memory.

3-555 The second function of the jumper module is to convert the external address terms, which select the associated memory stack, into address terms suitable for the internal memory logic. This is necessary because within each memory stack the addresses begin at location zero. In the memory for the 8K through 16K range, for example, location zero within the memory is addressed by $20,000_8$ (8192_{10}) on the external address lines. The $\overline{Ls0}$ Ls1 $\overline{Ls2}$ code on the external lines must therefore be converted to $\overline{L1}$ $\overline{L2}$ on the internal memory lines. Location 10000_8 (4096_{10}) within the memory is addressed by 30000_8 ($12,888_{10}$) on the external address lines. The $\overline{Ls0}$ Ls1 Ls2 code on the external lines must therefore be converted to $\overline{L1}$ L2 on the internal memory lines. This conversion takes place in the ZB65 through ZB22 jumper module. Since Ls0 is used only in the priority logic and is beyond the 16K internal memory addressing range, this signal is not used in internal memory address conversion.

3-556 The jumper module outputs, which generate L1 and L2 in the internal memory address logic, are /$\overline{Ls1k}$/ and /$\overline{Ls2k}$/. In the ZB65-22 example used above, L1 is above the range of an 8K memory; therefore, /$\overline{Ls1k}$/ is jumpered to a +4v via a resistor to make L1 always false. Since L2 is the same whether in the 16K range or the 8K range, a ONE in L2 on the external lines is also a ONE on the internal address lines. In the ZB65 through ZB22, therefore, /$\overline{Ls2}$/ is jumpered to /$\overline{Ls2k}$/. The Lz direct address lines are jumpered in the same manner to provide internal address signals when a direct access is being made. The complete equations for the ZB65-22 jumper module are as follows:

/$\overline{Ls0j}$/ = /Ls0/	/$\overline{Lz0j}$/ = /Lz0/
/$\overline{Ls1j}$/ = /$\overline{Ls1}$/	/$\overline{Lz1j}$/ = /$\overline{Lz1}$/
/$\overline{Ls2j}$/ = Ground	/$\overline{Lz2j}$/ = Ground
/$\overline{Ls1k}$/ = +4v	/$\overline{Lz1k}$/ = +4v
/$\overline{Ls2k}$/ = /$\overline{Ls2}$/	/$\overline{Lz2k}$/ = /$\overline{Lz2}$/

The equations for all jumper modules are included in the 930 Computer logic equations in section VII.

3-557 Address Decoding

3-558 The 14 address lines L1 through L14, are decoded within the memory to select up to 4,096, 8,192, or 16,384 individual locations, depending upon the size of the memory. The 12 least significant lines, L3 through L14, are decoded to select addresses within the 4,098-word blocks, and L1 and L2 are used to select on of several of these 4,096-word blocks.

3-559 Drive Line Selection. The 4,096 locations in each 4K block of addresses are selected by the application of two coincident currents to the core matrix. One of these currents flows in one of a set of 64 lines known as the X drive lines, and the other current flows in one a similar set of 64 Y drive lines. Each of the 64 X lines is selected by a smaller 8 x 8 decoding matrix of transformers as shown in figure 3-83. A similar matrix selects one of the 64 Y drive lines.

3-560 Current is caused to flow in the primary of a transformer by connecting the center tap to a positive current source and one of the other terminals of the primary winding to a negative current sink. The circuit that connects the center tap to the positive current source (a resistor to

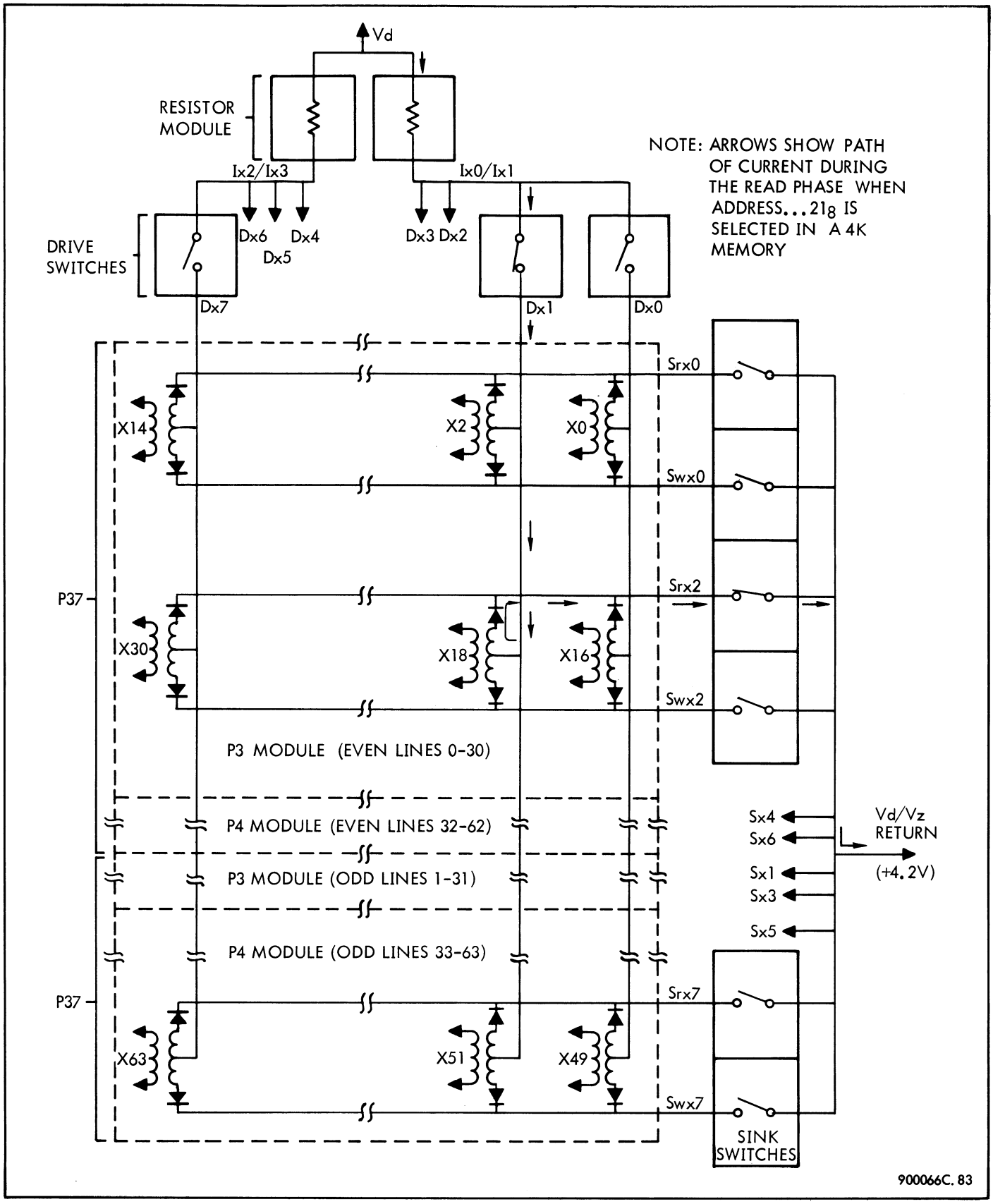


Figure 3-83. X Transformer Decoding Matrix

the positive Vd supply) is known as a drive switch. The circuit which connects the other terminal to the negative current sink (the Vd/Vz return) is known as a sink switch. To obtain one direction of current in the stack during the read phase and the opposite direction of current during the write phase, two separate sink switches are used. Figure 3-84 illustrates the X selection system for a single drive line.

3-561 Eight drive switches and eight sink switches are used to select one of 64 transformers. For one set of X lines, the drive switches are known as Dx0 through Dx7, and the sink switches are known as Srx0 through Srx7 and Swx0 through Swx7. An Sr switch is for read current, and an Sw switch is for write current.

3-562 Address lines L12, L13, and L14, the three least significant bits of the address word, as decoded by octals to energize the X drive lines. The equations for a 4K memory are as follows:

$$Dx0 = \overline{L2}\ \overline{L12}\ \overline{L13}\ \overline{L14}\ MD$$

$$Dx1 = \overline{L2}\ \overline{L12}\ \overline{L13}\ L14\ MD$$

$$\vdots$$

$$Dx7 = \overline{L2}\ L12\ L13\ L14\ MD$$

Where MD is the memory disable, Cr power fail safe signal from the computer. For an 8K or a 16K memory, eight additional octal signals, Dx10-Dx17, are derived from the same combinations of L12-L14, using L2 instead of $\overline{L2}$.

3-563 The X sink switch signals are decoded in octals from L9 L10 L11 of the address lines in the same manner as the X drive switches. Two typical equations follow:

$$Srx0 = Re\ \overline{L9}\ \overline{L10}\ \overline{L11}$$

$$Swx0 = We\ \overline{L9}\ \overline{L10}\ \overline{L11}$$

where Re and We are read enable and write enable, respectively.

3-564 The Y drive lines are decoded in a similar manner, using L6 through L8 for the Y drive switches and L3 through L5 for the Y sink switches. The octal decoding system for both X and Y lines in a 4K memory is shown in table 3-20. A block diagram of the overall decoding system from the 12 address lines is shown in figure 3-85.

3-565 The circuits used to select a single X drive line are shown in figure 3-86, using X18 as an example. The selection of one X line is accomplished as follows: When unselected, at T3 time, for example, the output of Dx1 is at about +3.5v due to resistor Ra and diode Da. The waveform of the Dx1 output voltage is shown in figure 3-87. When not selected, the sink switch outputs are at Vd because of the effect of resistors Rb and Rc.

3-566 At about the trailing edge of T3, address lines $\overline{L2}$, $\overline{L12}$, $\overline{L13}$, and L14 come true, selecting Dx1. Transistors Qa saturate, connecting the ZB67 resistors to the Dx1 drive switch output, which charges to Vd (region A in figure 3-87). When Re comes true, Srx2 is selected, and transistors Qb saturate. This connects the Srx2 sink switch output to +4.2v. The Dx1 output rapidly discharges to +4.2v (region B), turning diodes Db on and diodes Dc off. A current path is now created from Vd through the ZB67 resistor module, through the read primary winding of the transformer to the Vd/Vz return at +4.2v. The current flowing through the primary induces a secondary current, which flows through the stack in the read direction (see figure 3-87). The waveform is stepped because the initial current and voltage is partly determined by the ac impedance of the drive line (about 100 ohms). The stable voltage (region C) and current is partly determined by the dc resistance (8 or 16 ohms) of the drive lines. The steady value of the current is determined by Vd and the resistance of the stack and the ZB67 resistor module.

3-567 At about the trailing edge of Tr, Re goes false and the Dx1 output recharges to Vd (region D). At about the trailing edge of Tp, We comes true, selecting Swx2. This causes the Dx1 output to discharge rapidly to about +6v and causes current to flow through the stack in the write direction. At about the trailing edge of T6, We goes false and the Dx1 output recharges to Vd. At about the trailing edge of T4, all addresses are reset and the Dx1 output discharges to +3.5v (region E).

3-568 Because of the transformer turns ratio, the primary current (about 500 milliamperes), is twice the required secondary current (about 250 milliamperes). Conversely, the secondary voltage is twice the primary voltage (Vd), producing faster current rise times than would otherwise be available. Because of the heavy primary current, all the circuits use two diodes (Db, Dc) or two transistors (Qa, Qb, Qc) in parallel. The drive circuits are all referenced to +4.2v. The currents are returned to +4.2v at the sink switches, and the cable shields are connected to +4.2v on the ZB67 resistor modules and the QB53 sink switch modules. The shields are continuously connected by links on the P3/P4 transformer module and the QB54 drive switch module, but are not directly connected to +4.2v on these modules. The Vd/Vz return is directly connected to +4.2v by a link between TB2A3 and TB2A6, and similarly, TB1A3 and TB1A6. The Vd and Vz signals should always be measured relative to +4.2v. The transformer secondaries are completely floating; therefore, the stack drive lines and ground planes are floating and are not connected to +4.2v or to ground.

3-569 Unlike the drive and sink switches, the drive lines are not selected with a direct numerical relationship to the addresses. Address... 00_8 (Srx0, Dx0) selects drive line X0, but address ... 01_8 (Srx0, Dx1) selects drive line X2 and so on until the first eight even X drive lines have been selected, as indicated in table 3-21. Address ... 10_8 (Srx1, Dx0) selects X1 and address ... 11_8 (Srx1, Dx1)

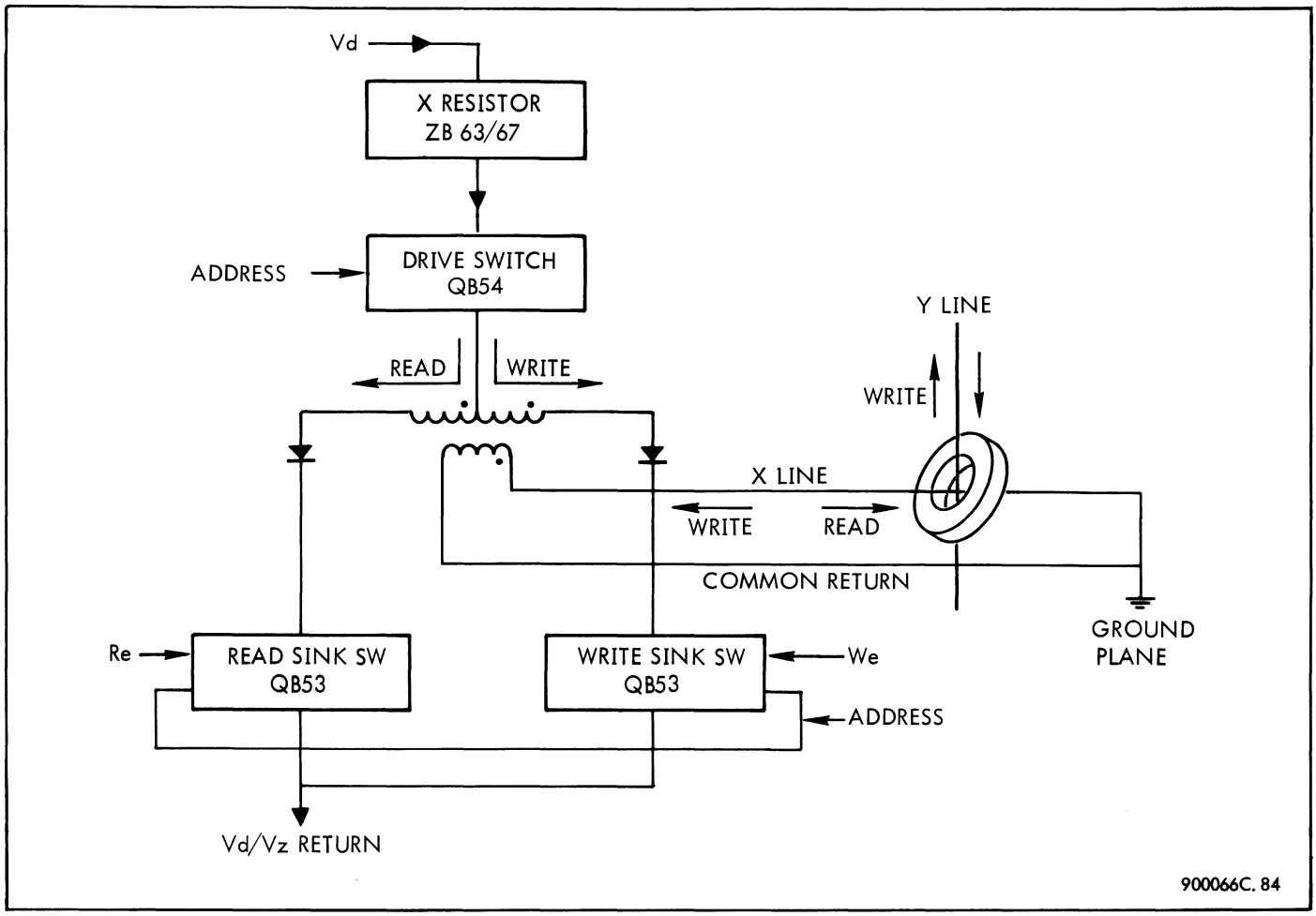


Figure 3-84. Single-Core Drive Line Selection

Table 3-20. Octal Decoding for 4,096 Words

Instruction Word	12	13	14	15	16	17	18	19	20	21	22	23
S-Register	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
L Lines	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14
Decoding	Y Sink Switches Sry0-7, Swy0-7			Y Drive Switches* Dy0-7			X Sink Switches Srx0-7, Swx0-7			X Drive Switches* Dx0-7		
	Y Lines Y0-Y63						X Lines X0-X63					

*The drive switches listed are those used to select the lowest 4K block in a memory.

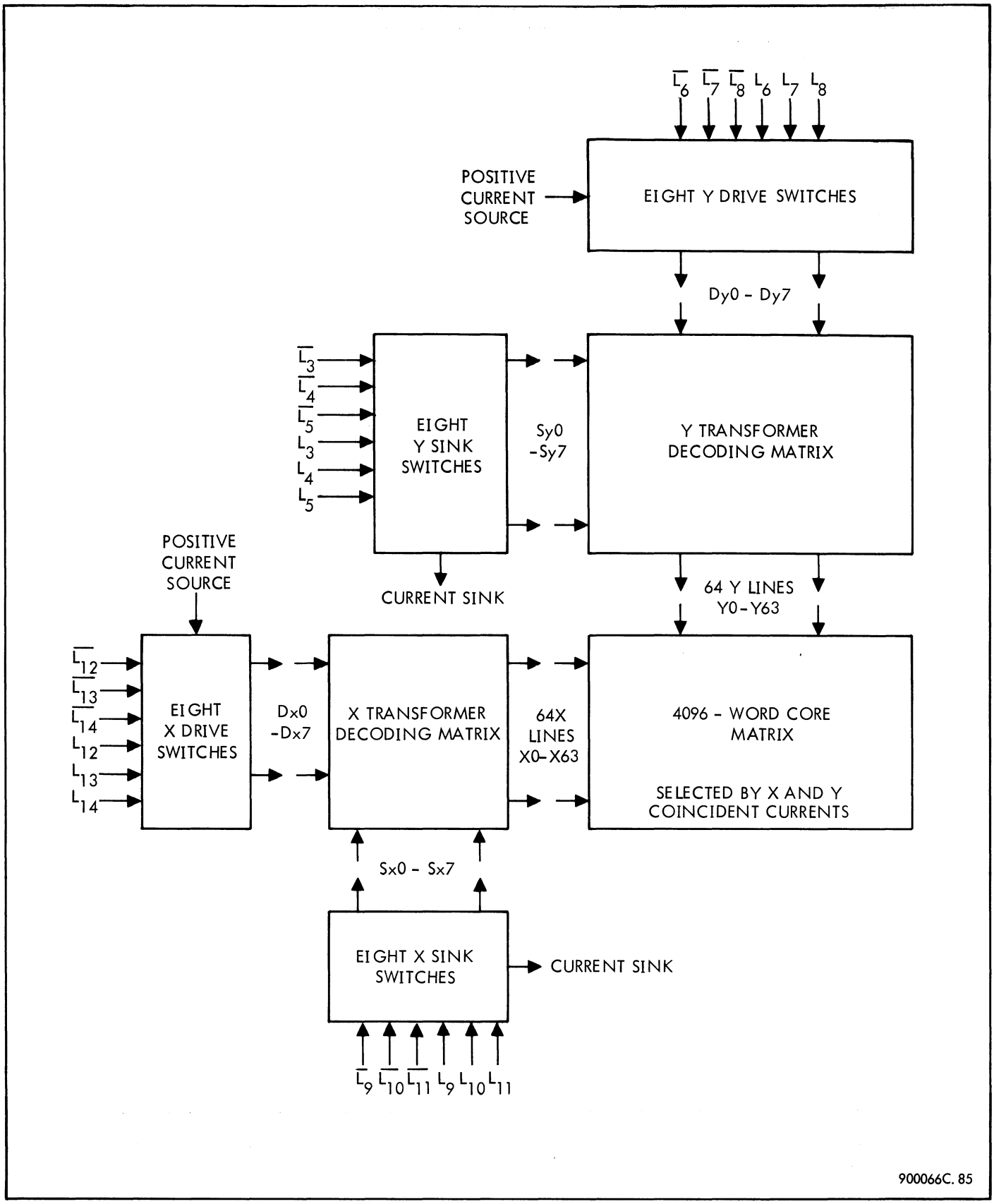


Figure 3-85. Address Decoding for 4,096 Words

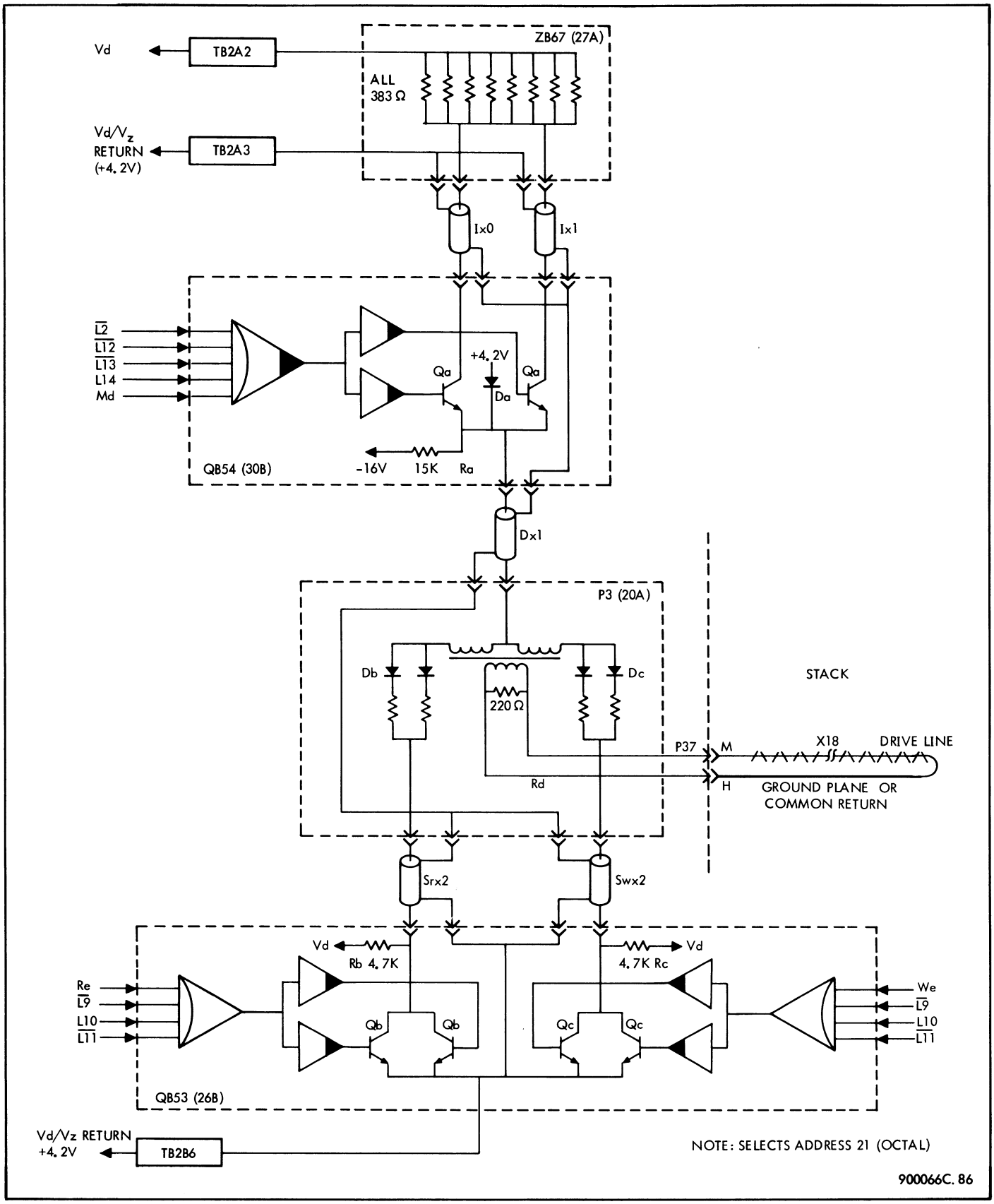


Figure 3-86. Selection Circuits for One X Drive Line

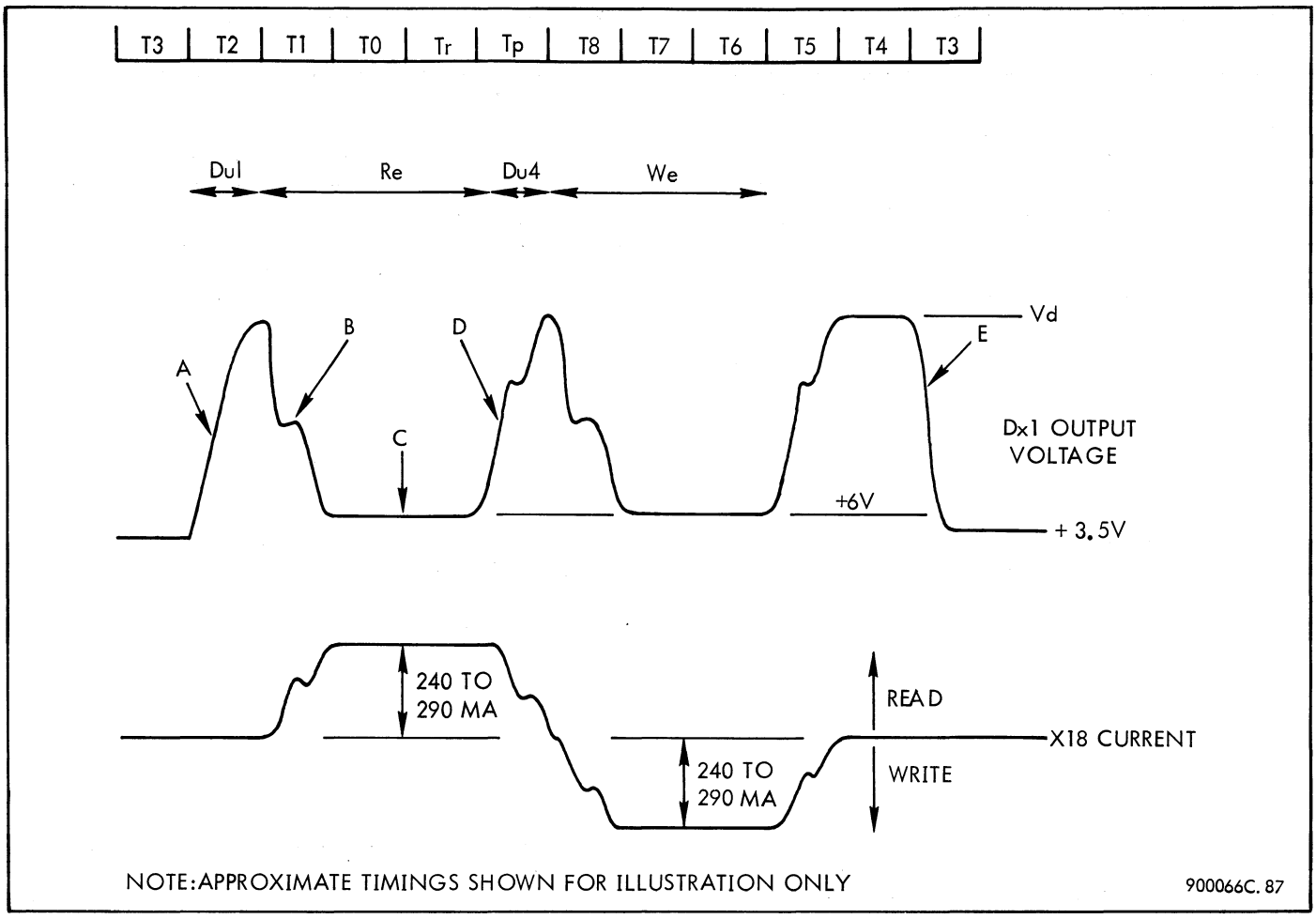


Figure 3-87. Selection Voltage and Current Waveforms

Table 3-21. Address-Drive Line Relationship

Octal Address	Drive Line	Octal Address	Drive Line	Octal Address	Drive Line	Octal Address	Drive Line
--00	X0	--20	X16	--40	X32	--60	X48
--01	X2	--21	X18	--41	X34	--61	X50
--02	X4	--22	X20	--42	X36	--62	X52
--03	X6	--23	X22	--43	X38	--63	X54
--04	X8	--24	X24	--44	X40	--64	X56
--05	X10	--25	X26	--45	X42	--65	X58
--06	X12	--26	X28	--46	X44	--66	X60
--07	X14	--27	X30	--47	X46	--67	X62
--10	X1	--30	X17	--50	X33	--70	X49
--11	X3	--31	X19	--51	X35	--71	X51
--12	X5	--32	X21	--52	X37	--72	X53
--13	X7	--33	X23	--53	X39	--73	X55
--14	X9	--34	X25	--54	X41	--74	X57
--15	X11	--35	X27	--55	X43	--75	X59
--16	X13	--36	X29	--56	X45	--76	X61
--17	X15	--37	X31	--57	X47	--77	X63

selects X3 and so on until the first eight odd X drive lines have been selected. The second group of eight even X drive lines are next selected as the addresses increase and so on until 64 drive lines have been selected. The four 8 x 8 transformer matrices used for the selection of 128 X drive lines are shown on the X transformer module logic diagrams. For memories using only 64 X drive lines, only two of these 8 x 8 matrices are used and Dx10 through Dx17 are not generated.

3-570 The Y drive lines are selected in a similar manner with Dy0 through Dy7, Sry0 through Sry17 and Swy0 through Swy17.

3-571 When the addresses are written to in a single checkerboard pattern, the drive lines, and hence the stack, are written to in a double checkerboard pattern. The double checkerboard (worst) pattern is written in the stack by placing a ONE in all locations where the least significant X address bit, L14, and the least significant Y address bit, L8, are the same (both even or both odd). Similarly, a ZERO is written in all locations where L14 and L8 are not the same (see figure 3-88).

3-572 Stack Addressing

3-573 The 4K memory stack has 64 X lines and 64 Y lines, each of which thread through 28 mats of 4,096 cores each. Refer to figure 3-77 for stack layout and wiring. Each X or Y line threads 28 x 64 cores and has a resistance of about 8 ohms. Correct drive current for both X and Y lines is obtained from Resistor ZB67 modules. The 64 X lines are selected by a resistor module, eight drive switches, an 8 x 8 transformer matrix, and eight read and eight write sink switches. The 64 Y lines are selected in a similar manner. Because only $4,096_{10}$ words are addressed, only 12 internal address bits are required for address selection, and L1 and L2 remain false.

3-574 The 8K memory stack has 128 X lines and 64 Y lines. The X lines thread through 26 mats of 4,096 cores each. Refer to figure 3-78 for stack layout and wiring. Each X line threads through 26 x 64 cores and has a resistance of about 8 ohms. Correct drive current for the X lines is obtained from Resistor ZB67 modules. The Y lines thread through 52 mats of 4,096 cores each. Each Y line threads 52 x 64 cores and has a resistance of approximately 16 ohms. Correct drive current for the Y lines is obtained from resistor ZB63 modules. The 128 X lines are selected by a resistor module, 16 drive switches, an 8 x 16 transformer matrix, and eight read and eight write sink switches. Twice as many drive switches and transformer modules are needed to select 128 lines than are used for 64 lines, but the sink switches and resistor module are shared, and so are not duplicated. The 64 Y lines are selected in the same way as in the 4K memory. Because $8,192_{10}$ words are addressed, 13 internal address bits are required, and L1 remains false. The $\overline{L2}$ signal selects the low eight X drive switches, Dx0 through Dx7, to energize the 64 low X lines, X0 through X63. The L2 signal selects the high eight X drive switches, Dx10 through Dx17 to energize the high 64 X lines, X64 through X127.

3-575 The 16K memory stack has 128 X lines and 128 Y lines, each of which thread through 50 mats of 4,096 cores each (two mats per frame). Refer to figure 3-79 for stack layout and wiring. Each X or Y line threads 50 x 64 cores and has a resistance of approximately 16 ohms. Correct

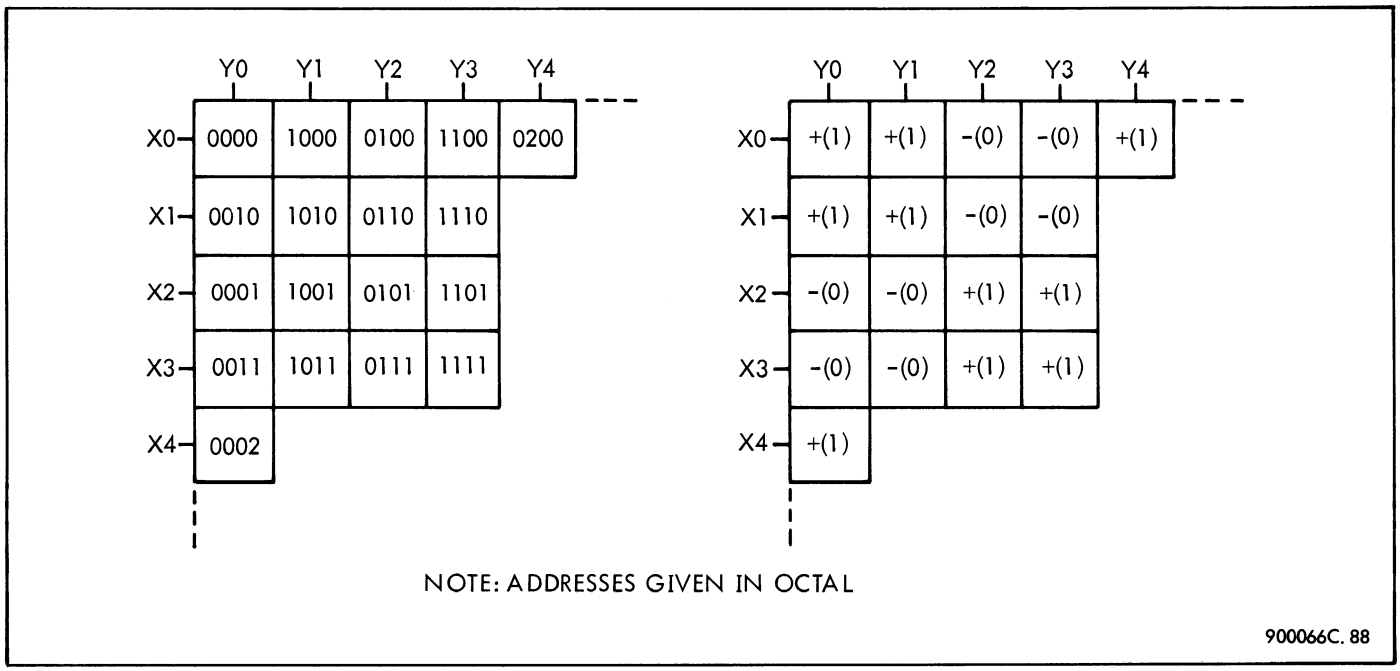


Figure 3-88. Relationship of Address and Stack Location

drive current for the X and Y lines is obtained from Resistor ZB63 modules. The 128 X lines are selected by a resistor module, 16 drive switches, an 8 x 16 transformer matrix, and eight read and eight write sink switches. The 128 Y lines are selected in a similar manner. Fourteen internal address bits are required to select the $16,384_{10}$ words. An address in the first 4K block is selected by $\overline{L1}$ (low Y drive switches, $\overline{Dy0}$ through Dy7 and low Y lines, Y0 through Y63) and $\overline{L2}$ (low X drive switches, Dx0 through Dx7, and low X lines, X0 through $\overline{X63}$). An address in the second 4K block is selected by $\overline{L1}$ (Dy0 through Dy7 and Y0 through Y63) and L2 (Dx10 through Dx17 and X64 through X127). An address in the third 4K block is selected by L1 (Dy10 through Dy17 and Y64 through Y127) and $\overline{L2}$ (Dx0 through Dx7, X0 through X63). An address in the fourth 4K block is selected by L1 (Dy10 through Dy17 and Y64 through Y127) and L2 (Dx10 through Dx17 and X64 through X127). The decoding for the 4K, 8K, and 16K stacks is shown in table 3-22.

3-576 INHIBITING SYSTEM

3-577 During the write phase, an inhibit current is applied through the inhibit winding for each bit which is to store a ZERO. The Z drivers for the appropriate quadrant and bits are turned on by timing pulse Mdt. The Z driver acts as a current switch and closes a current path through the primary of the Z transformer. The primary current induces the required inhibit current to flow in the transformer secondary and through the inhibit winding for the appropriate quadrant. The value of the primary current is established by the Z resistor to the Vz supply. A schematic for one bit in the 16K inhibiting system is shown in figure 3-89.

3-578 The Z transformers have a 2:1 turns ratio, and the primary current (about 500 milliamperes) is twice the required secondary current. This produces faster current rise times than would otherwise be available. The Z circuits are all referenced to +4.2 volts. The currents are returned to +4.2 volts at the AB56 Z drivers, and the cable shields are connected to +4.2 volts on the Resistor ZB64 modules and on the Z drivers. The shields are continuously connected via junctions adjacent to the Z transformer modules. These junctions are shown on the logic diagrams as location 33A. The shields are not connected to the P1 transformer module. The transformer secondaries, and hence the inhibit windings, are completely floating from the logic voltages. In early stacks, one end of the inhibit winding is connected to the ground planes, but in later stacks the inhibit windings are floated from the ground planes.

Table 3-22. Octal Decoding, 16,384 Words

Stack	4K Block	L3 L4 L5	L6 L7 L8	L9 L10 L11	L12 L13 L14
		Y Sink Switches	Y Drive Switches	X Sink Switches	X Drive Switches
4K	1st $\overline{L1}$ $\overline{L2}$	Sry0-7, Swy0-7	Dy0-7	Srx0-7, Swx0-7	Dx0-7
		Y0-Y63		X0-X63	
8K	2nd $\overline{L1}$ L2	Sry0-7, Swy0-7	Dy0-7	Srx0-7, Swx0-7	Dx10-17
		Y0-Y63		X64-X127	
16K	3rd L1 $\overline{L2}$	Sry0-7, Swy0-7	Dy10-17	Srx0-7, Swx0-7	Dx0-7
		Y64-Y127		X0-X63	
	4th L1 L2	Sry0-7, Swy0-7	Dy10-17	Srx0-7, Swx0-7	Dx10-17
		Y64-Y127		X64-X127	

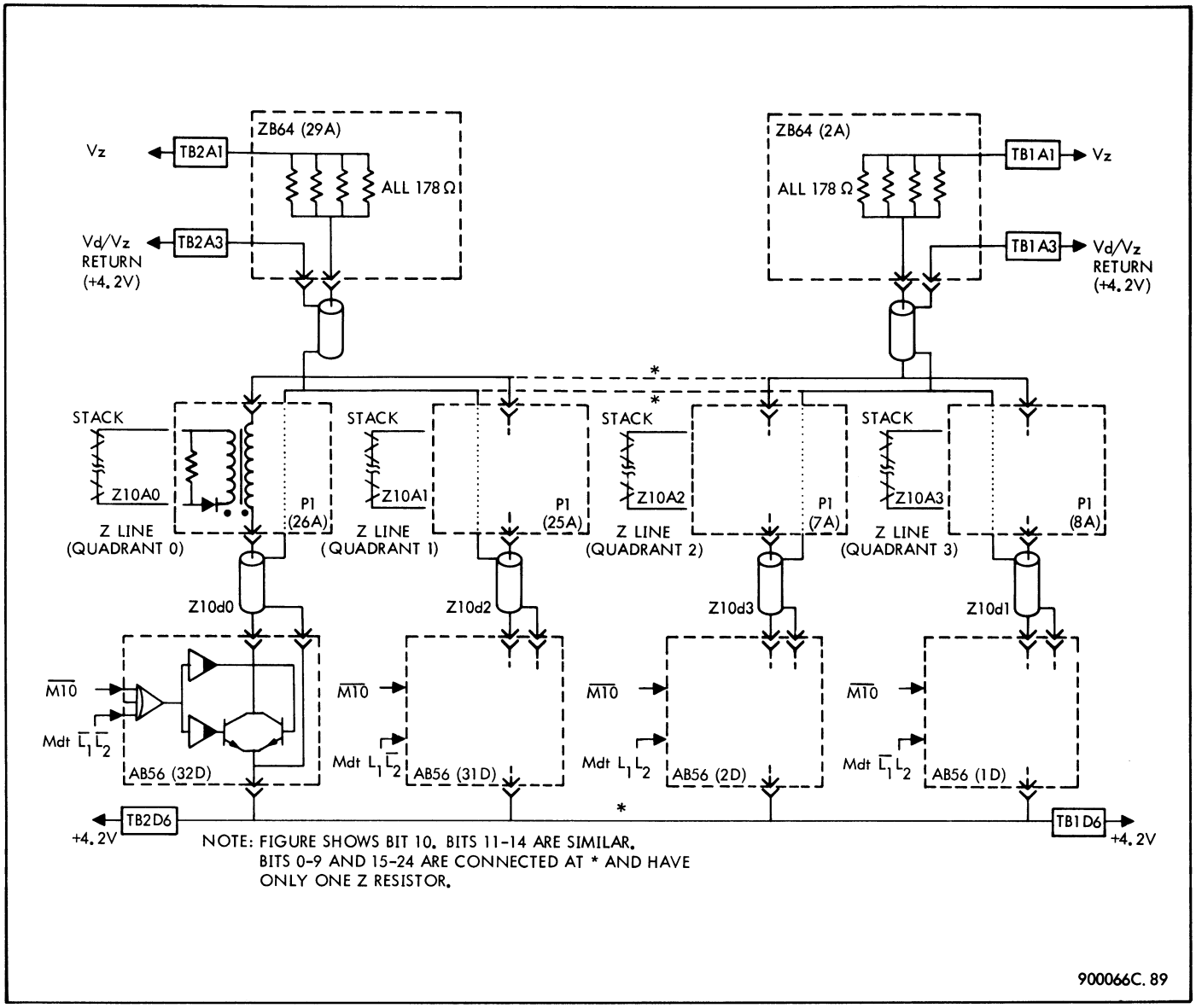


Figure 3-89. Inhibiting System, 16K Memory

3-579 Each driver can drive one quadrant of 4,096 cores; hence, each bit of a 16K memory system requires 4 Z drivers and 4 Z transformers. For bits 0 through 9 and 15 through 24, the Z resistor is shared, and there is only one per bit. Because of layout considerations, there are two Z resistors for each of bits 10 through 14 (see figure 3-89).

3-580 Inhibit (Z) Lines

3-581 Each core mat in a memory frame represents one bit in one memory word. The location of the bits is determined by the inhibit lines, each of which threads all the cores in one core mat. The memory is divided by the inhibit logic into blocks and quadrants. A block is a group of 4,096 words of memory, selected by the state of L1 and L2 as follows:

Block 0 = $\overline{L1}\ \overline{L2}$, locations 0000_8 - 7777_8 (4K, 8K, and 16K memories)

Block 1 = $\overline{L1}$ L2, locations 10000_8 - 17777_8 (8K and 16K memories)

Block 2 = L1 $\overline{L2}$, locations 20000_8 - 27777_8 (16K memory only)

Block 3 = L1 L2, locations 30000_8 - 37777_8 (16K memory only)

3-582 The arrangement of blocks in each memory is different. The 4K memory, containing only 4,096 words, is considered as block 0. The 8K memory contains block 0 and block 1, located physically side by side. The 16K memory

contains four blocks, also located adjacent to each other. The positions of the blocks in the memories is shown in diagram form in figures 3-77 through 3-79 and figures 3-90 through 3-92.

3-583 Each memory is divided into four quadrants. Quadrant 0 is the upper left core mat in each frame. The remaining quadrants are numbered clockwise as follows:

0	1
3	2

The quadrants in all core frames in all three memories correspond; however, in none of the memories do blocks and quadrants correspond.

3-584 The location of each bit in memory by block and quadrant can be determined by the logic terms used in the memory Z driver and Z transformer logic diagrams. One hundred Z driver signals (4 blocks, 25 bits per word) are generated from the outputs of the M-register and the two most significant bits on the address lines. Table 3-23 gives the Z driver equations and the meaning of the logic terms.

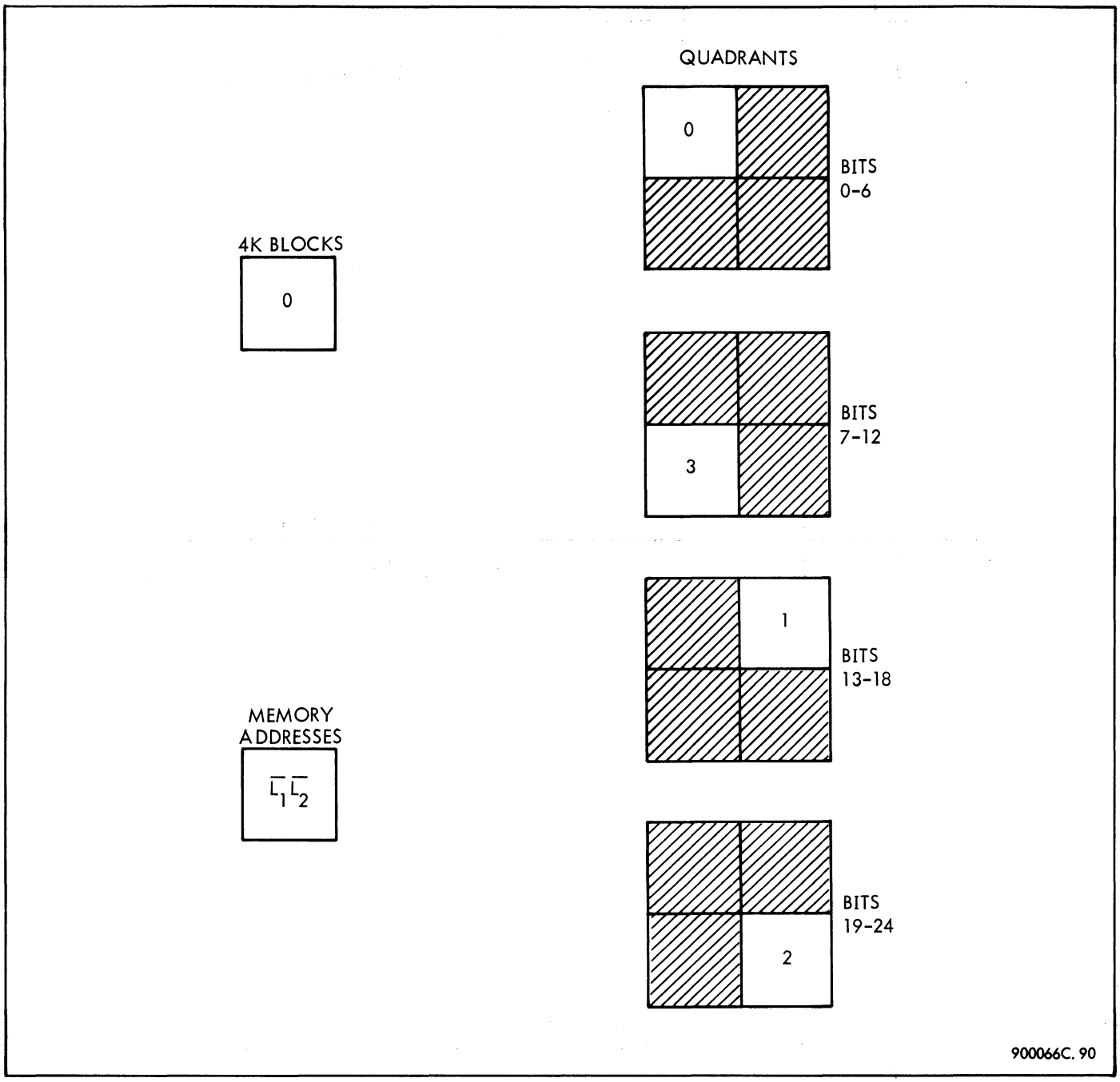


Figure 3-90. Blocks and Quadrants, 4K Memory

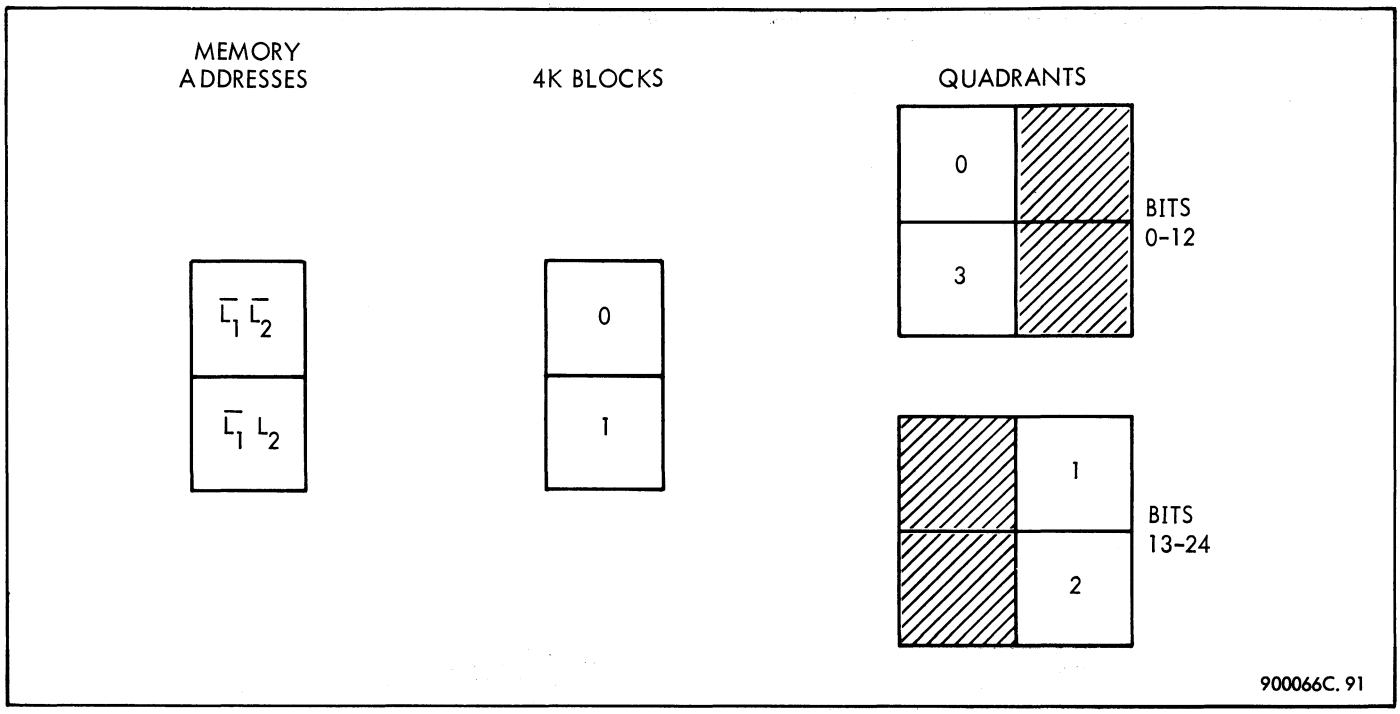


Figure 3-91. Blocks and Quadrants, 8K Memory

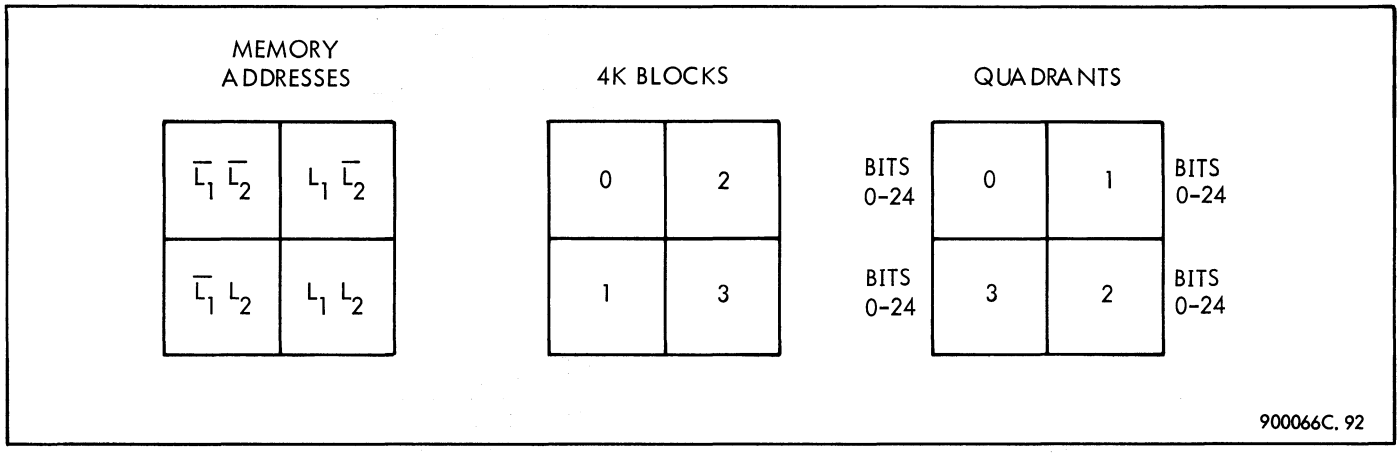


Figure 3-92. Blocks and Quadrants, 16K Memory

Table 3-23. Inhibit (Z) Drivers

INHIBIT DRIVER TERM				EQUATION
Inhibit	Bit No.	Driver	Block No.	
Z	0-24	d	0	M0-24 $\overline{L1}$ $\overline{L2}$ Mdt
Z	0-24	d	1	M0-24 $\overline{L1}$ L2 Mdt
Z	0-24	d	2	M0-24 L1 $\overline{L2}$ Mdt
Z	0-24	d	3	M0-24 L1 L2 Mdt

3-585 Each memory contains a set of transformer modules to provide current through the inhibit lines. The inhibit lines are selected in the transformer modules according to the physical location of the bits in memory. The 4K memory, with one block, uses only one Z driver for each M-register bit; the 8K memory, with two blocks, uses two Z drivers for each bit; and the 16K memory, with four blocks, uses four Z drivers for each bit.

3-586 Table 3-24 shows the way in which the inhibit lines are selected and gives the meaning of the logic terms. Bit 19 is used as an example, because this is the first bit that occupies a different frame in each of the three memories. See figures 3-71 through 3-79 for bit locations in the stack layouts.

3-587 The relationship between inhibit terms, quadrant numbers, block numbers, and bits is illustrated for frame 0 and frame 1 of the 4K, 8K, and 16K memories in figures 3-93 through 3-95.

3-588 SENSING SYSTEM

3-589 The basic sensing system consists of Sense Amplifier HB53 modules, each of which consists of two sense amplifiers and two digitizer circuits. Each sense amplifier has two inputs and provides a matched termination for the sense lines, amplification of about 165 at ONE signal frequency, amplitude discrimination relative to the external threshold voltage, and rectification. A simplified schematic of the sensing system is shown in figure 3-96.

3-590 The differential core ONE or ZERO signal is generated in the memory stack and transmitted along the sense winding to the input of the sense amplifier. The two-terminal input matches the sense winding impedance. The core signal is amplified in two ac amplifier stages. The A2 output, of ±5v amplitude, is bipolar because the sense winding passes through alternate cores in opposite directions. The A2 output drives the output transformer primary. The bipolar signal is rectified into a negative-going signal, and only the part of the signal that exceeds the threshold voltage (a ONE signal) passes to the digitizer stage as the memory analog, Ma, signal. The digitizer stage converts the negative-going rectified sense amplifier output signal into a digitized positive-going signal memory digital signal, Md, to be strobed into the M-register input gates. The M-register flip-flop is set at the leading (positive-going) edge of the strobe signal is Md is true. Figure 3-97 shows the waveforms at various points in the sensing circuit, and figure 3-98 illustrates signal discrimination.

3-591 The sensing system for a 16K memory is illustrated in figure 3-99. The sense winding for quadrant 0 enters one sense amplifier at the first input, and the sense winding for the diagonally opposite quadrant (quadrant 2) enters the same sense amplifier at the second input. Quadrants 1 and 3 are connected to the inputs of a second sense amplifier.

3-592 During the critical read phase, if a core is selected in quadrant 0, half-select current flows in line Xn and Yn. In quadrant 0 the currents select the addressed core. The current induced in the sense winding for quadrant 0 flows through the input transformer primary in the sense amplifier. The analog outputs of both sense amplifiers enter an OR circuit at the input of the digitizer stage. In the 8K and 4K memory systems, there is only one sense amplifier for each quadrant.

3-593 Each core mat in memory is threaded with one individual sense winding to read one bit in a memory word. The arrangement of sense windings in blocks and quadrants corresponds to that of the inhibit windings. Since only one memory word is read at one time, the corresponding bits in all memory blocks are read simultaneously. Only the bit selected will produce a read signal. Each core mat in a 4K memory, which has only one block, is read separately. The sense lines for corresponding bits in 8K and 16K memories are connected. The sense windings, bit numbers, and logic signals for frame 0 of the 4K, 8K, and 16K memory are shown in figure 3-100.

3-594 MEMORY WRITE LOCKOUT

3-595 The monitor or executive program approach to programming involves holding in memory the executive program that controls the selection and loading of other programs on the demand of the computer operator. Other programs may require retaining in memory various standard constants for use by several programs in memory at one time. In either case, it is desirable to inhibit accidental replacement of essential data by programming or operator error.

3-596 Two optional memory write lockout features are available for the 930 Computer to protect certain portions of memory from being inadvertently destroyed. The Manual Memory Write Lockout Feature Model 92060 provides manual toggle switches for selection of memory areas to be protected. The Programmed Memory Write Lockout Feature Model 92061 offers selection of memory areas by means of EOM and POT instructions. Each memory stack may be provided with one of these options if desired.

3-597 When a write lockout feature is installed, any attempt to write into a protected address is handled by the computer as follows:

a. The offending instruction is completed normally, except that the portion of the storage cycle which would alter memory is inhibited, and memory remains unchanged. Write requests from the input/output channels are similarly ignored. All memory locations may be read.

b. An internal interrupt to location 035 takes place. The interrupt routine may then determine the cause of the error and take appropriate action.

3-598 The lower 2K of memory (locations 0000 through 3777_8) can be protected in blocks of 512_{10} words; above the first 2K, memory can be protected only in groups of 2048_{10} words.

Table 3-24. Inhibit Lines for Bit 19

INHIBIT LINE TERM				Z DRIVER	MEMORY
Inhibit	Frame	Terminals	Quadrant		
Z	0	A, B	2	Z19d0	4K
Z	6	A, B	1	Z19d0	8K
Z	6	A, B	2	Z19d1	
Z	19	A, B	0	Z19d0	16K
Z	19	A, B	3	Z19d1	
Z	19	A, B	1	Z19d2	
Z	19	A, B	2	Z19d3	

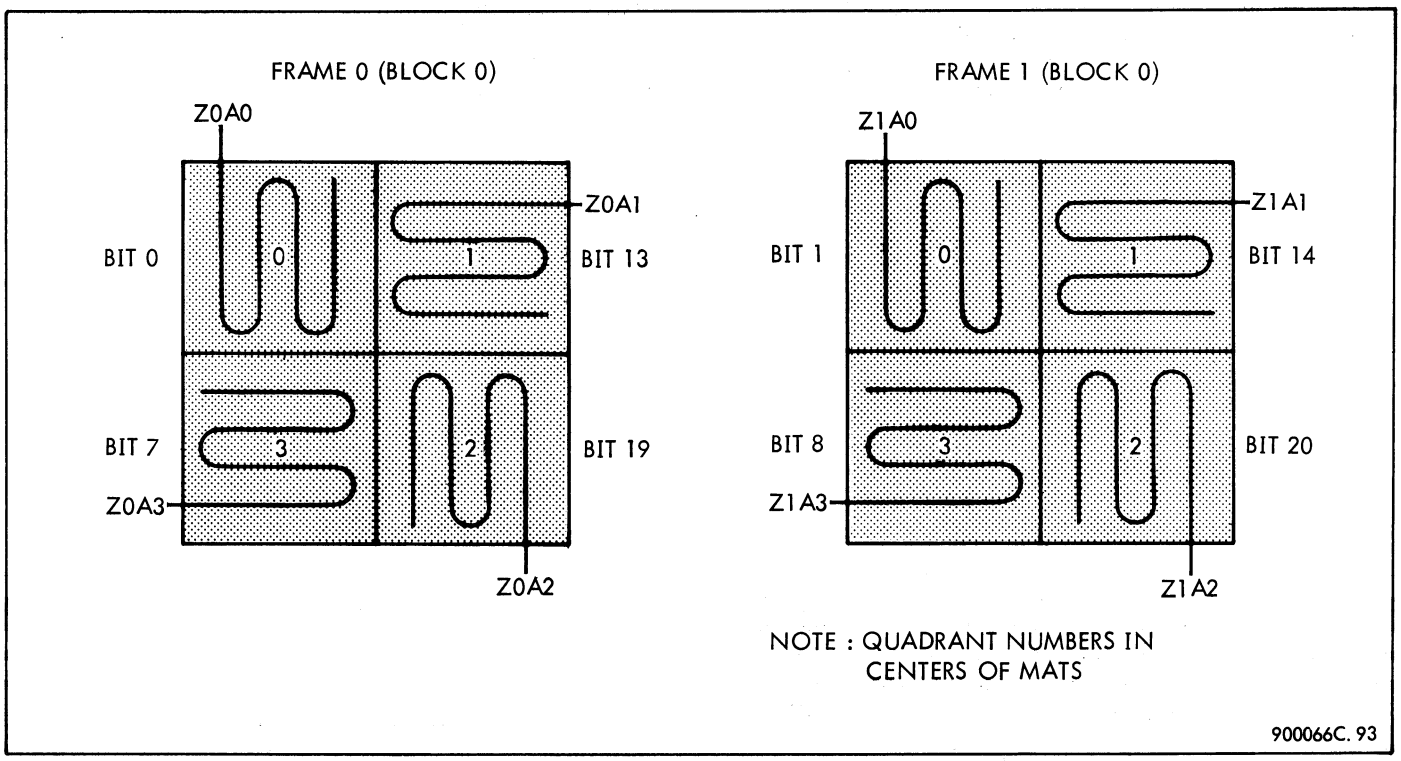
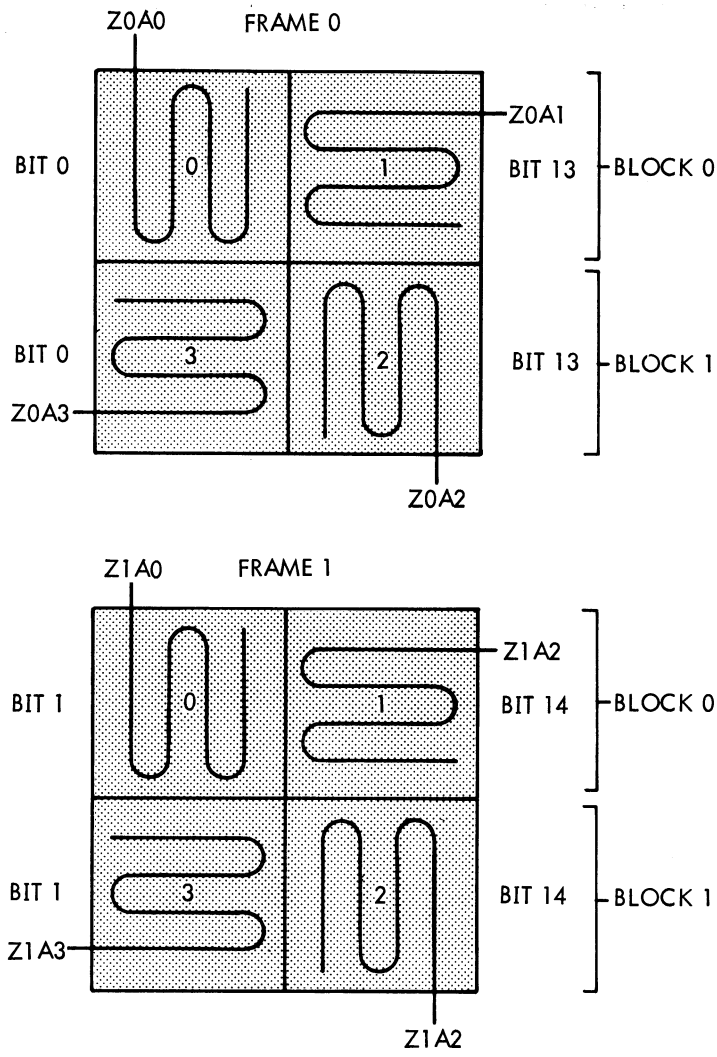


Figure 3-93. Inhibit Lines, 4K Memory



NOTE: QUADRANT NUMBERS IN CENTERS OF MATS

900066C. 94

Figure 3-94. Inhibit Lines, 8K Memory

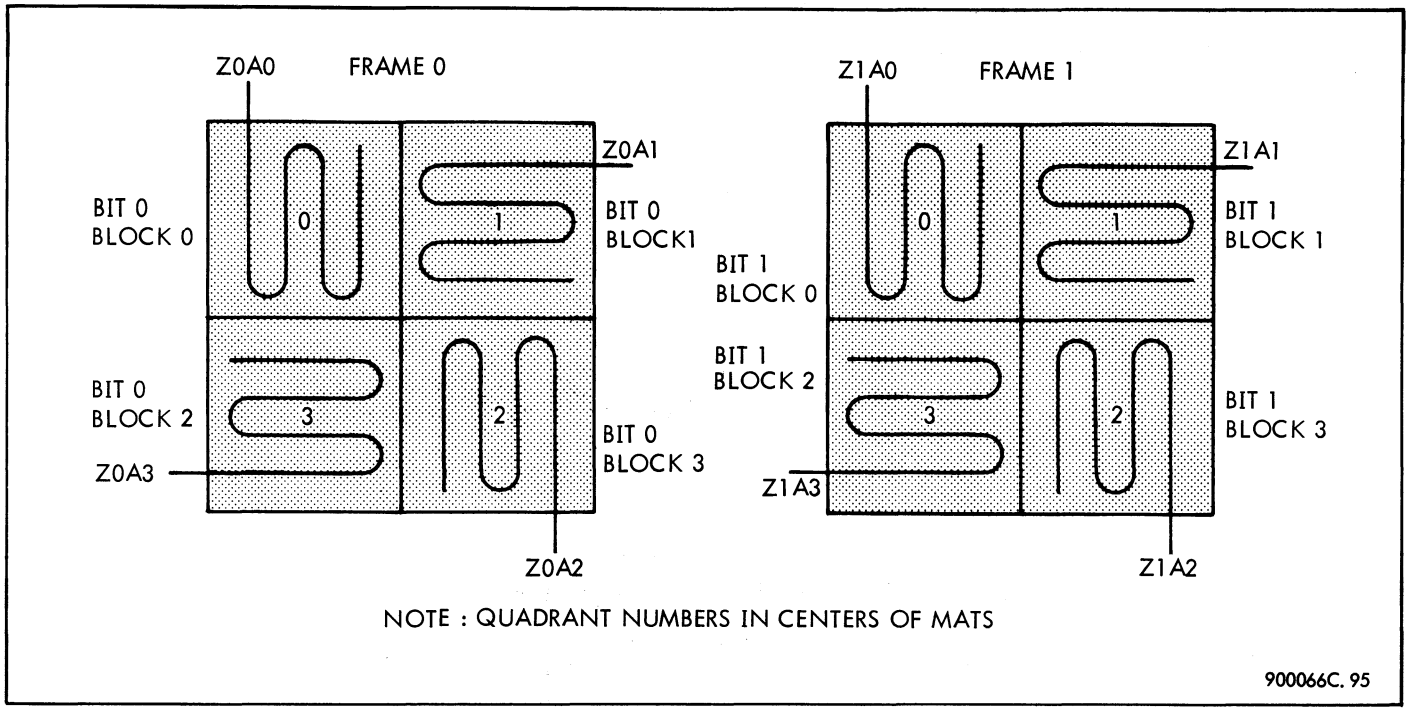


Figure 3-95. Inhibit Lines, 16K Memory

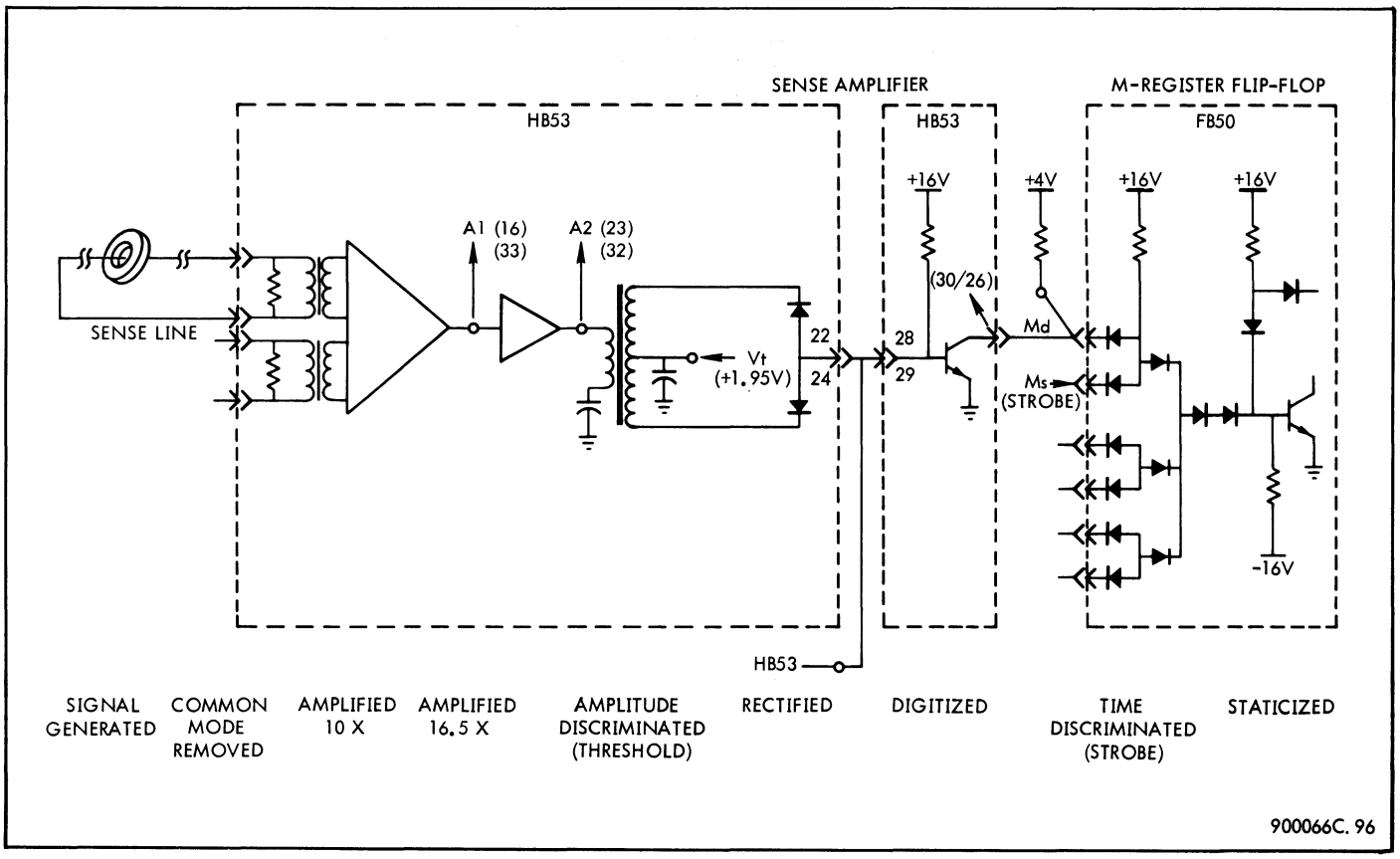


Figure 3-96. Sensing System

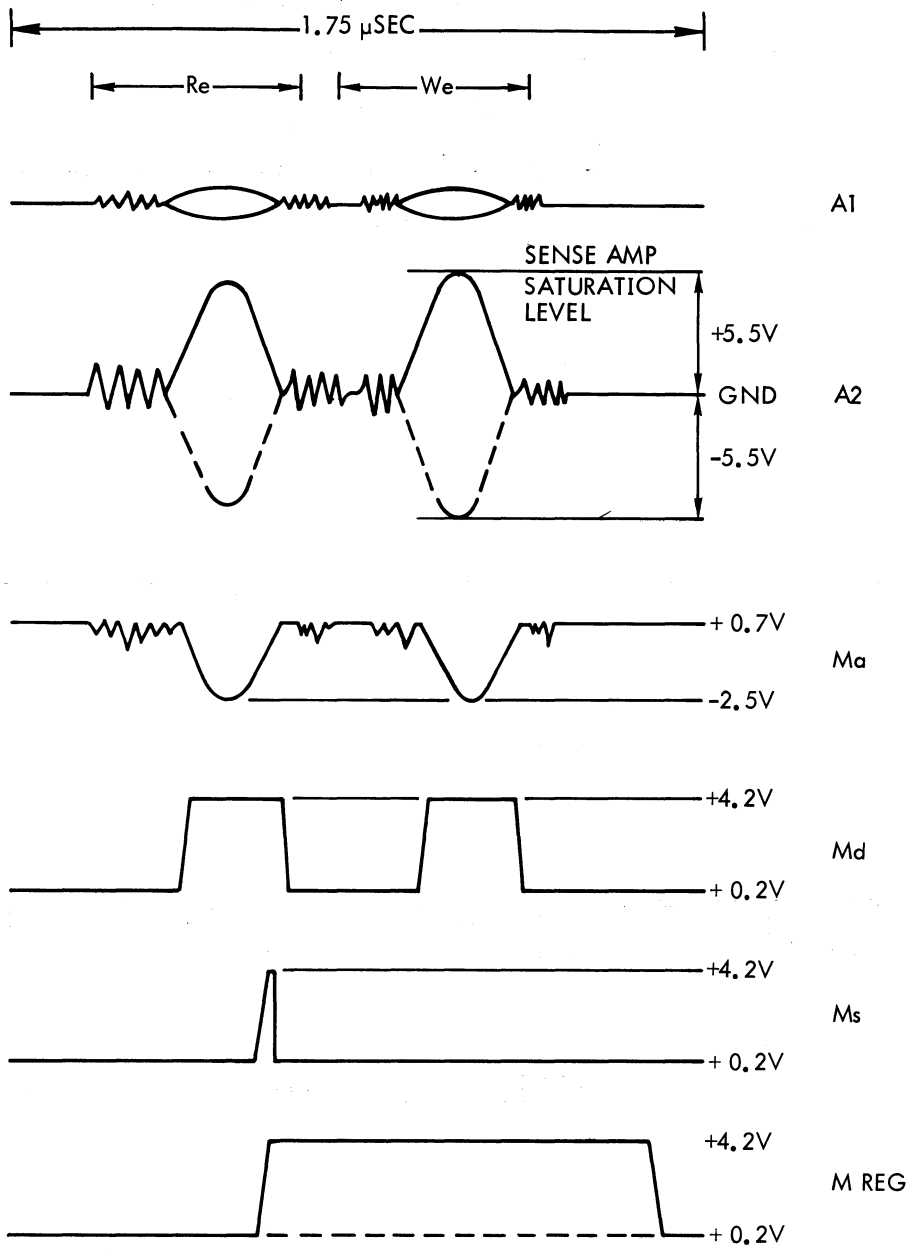


Figure 3-97. Sensing System Waveforms

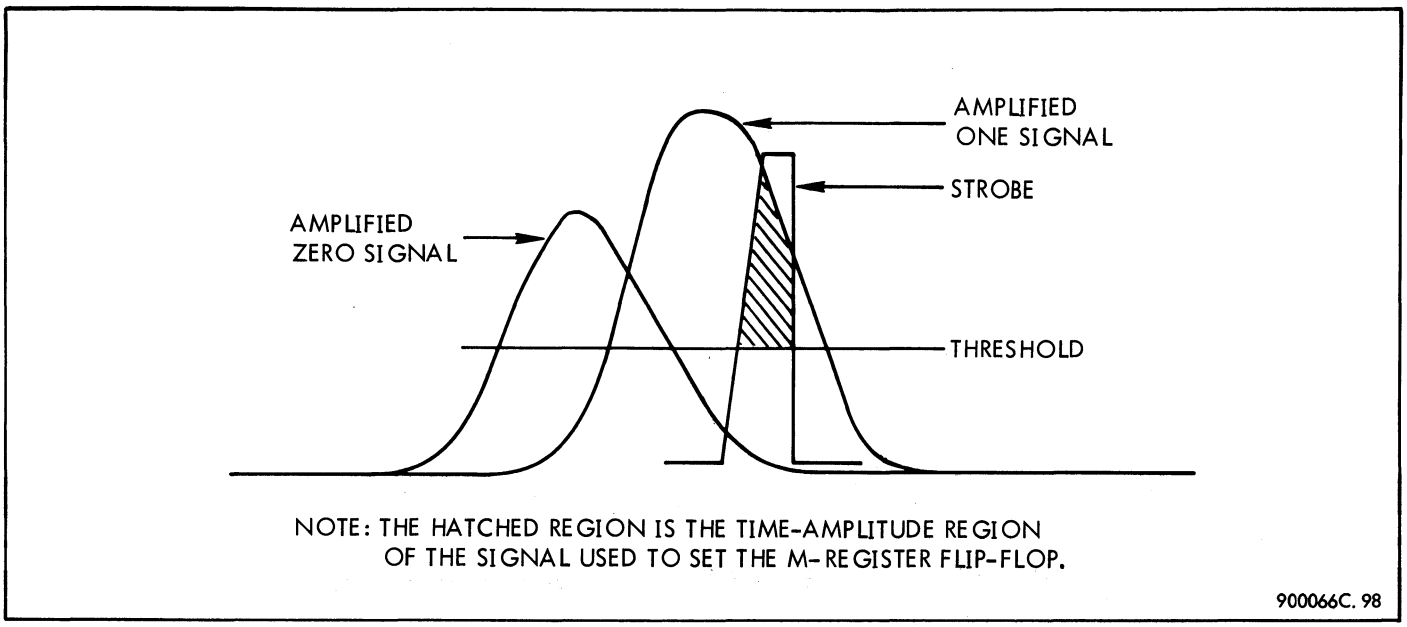


Figure 3-98. Read Signal Discrimination

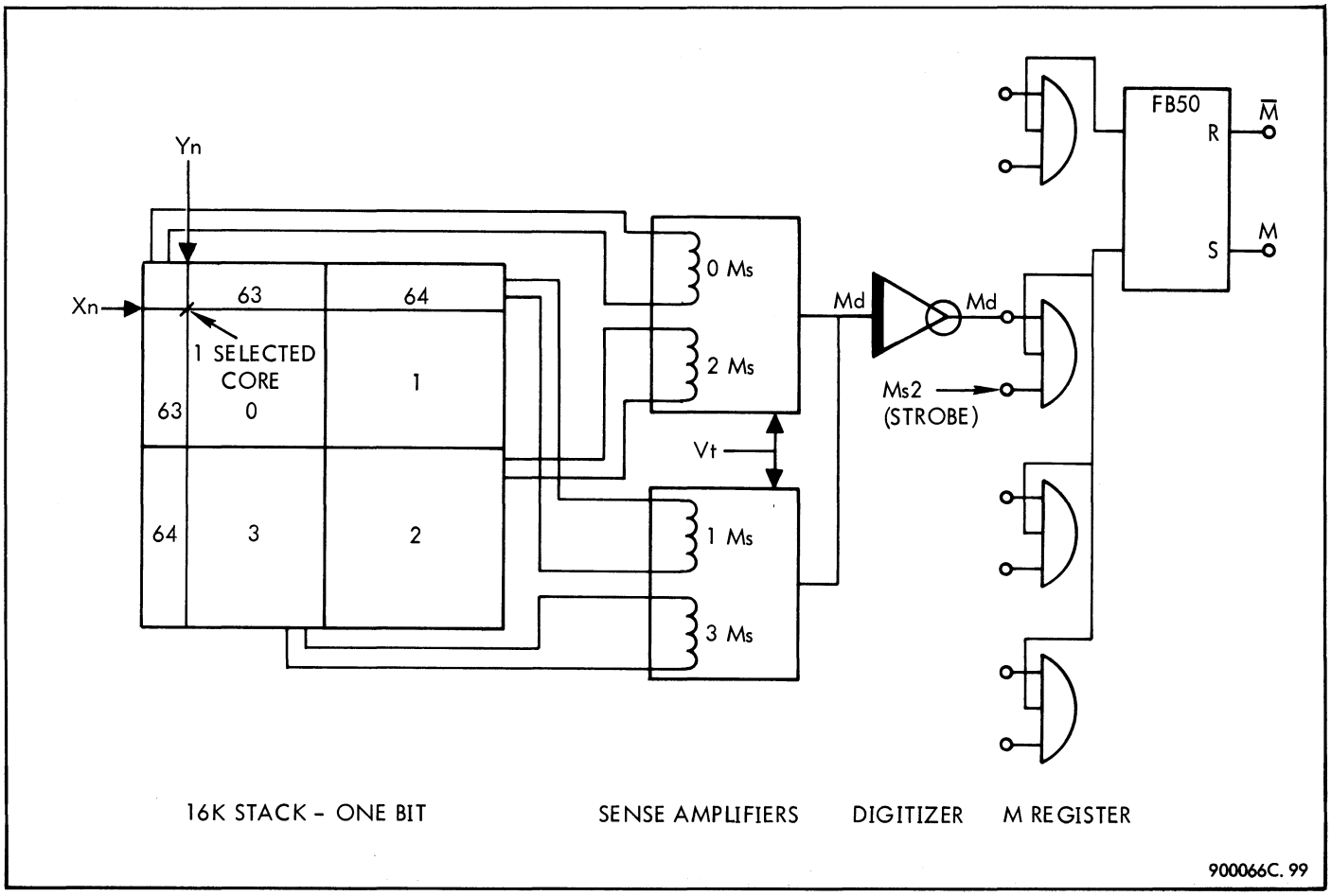


Figure 3-99. 16K Sensing System

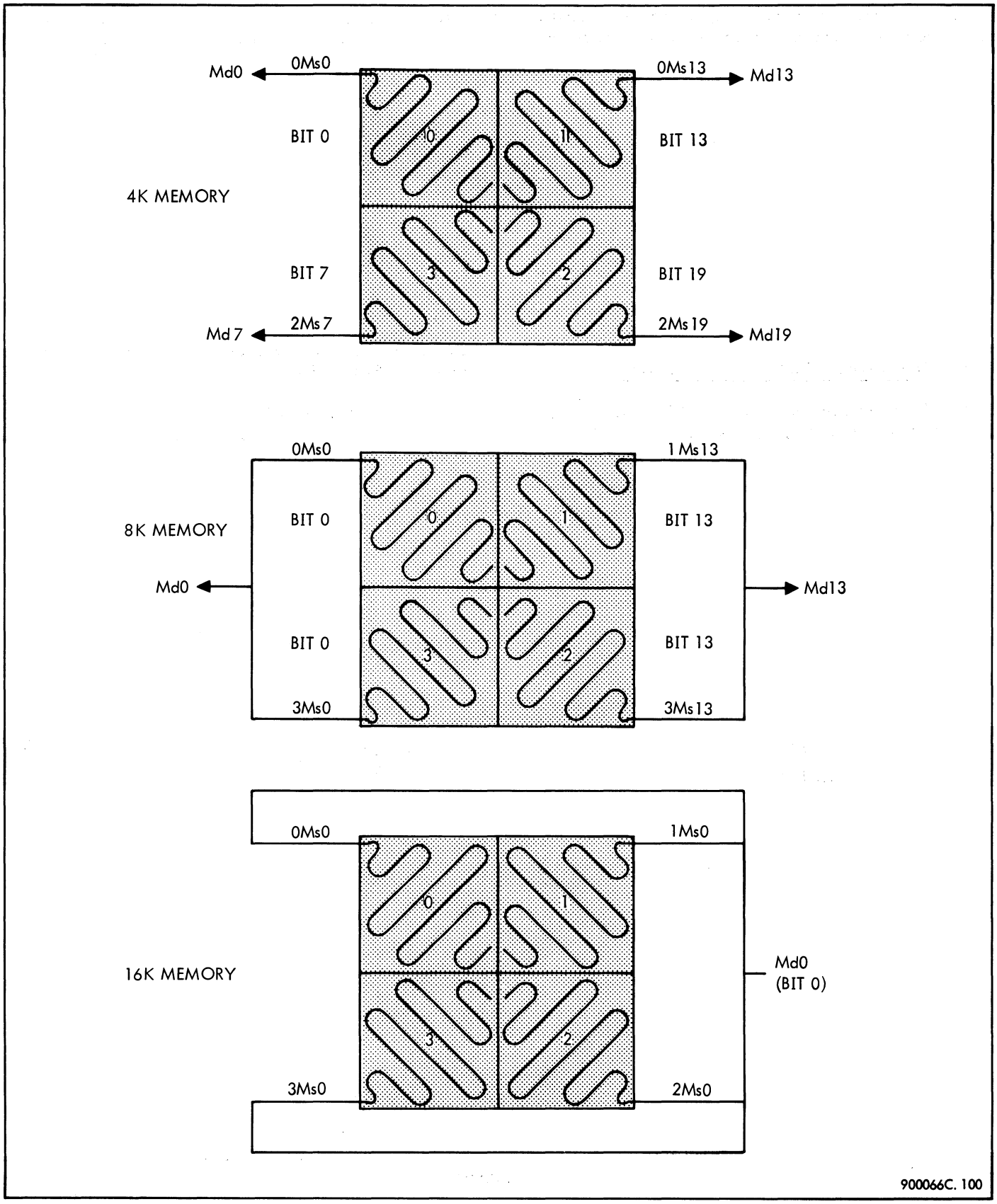


Figure 3-100. Sense Windings, Frame 0 for All Memories

3-599 The memory write lockout option consists primarily of a toggle switch or flip-flop register, a set of comparison gates, and a priority interrupt channel. One Priority Interrupt SK61 module containing the write lockout interrupt circuit is inserted into the interrupt chassis. The remainder of the lockout option is added to the main memory chassis. For the manual interrupt, four Switch Flops SX51 and two NAND No. 2 IB56 modules are needed. The programmed write lockout requires four Flip-Flops FB50, two NAND No. 2 modules, and a Jumper ZB65 module. An additional cable plug module, P907, is required for both options, except that this module is not necessary for the manual write lockout if the memory is already fitted in the MAM option. Figure 3-101 is a simplified logic diagram of the two least significant bits of the write lockout register, showing both manual and programmed operation.

3-600 Programmed Write Lockout Model 92061

3-601 In the programmed write lockout feature, memory protection is controlled by a pair of EOM and POT instructions:

EOM

0	0	0	02	2	0	0	4	0

0 1 2 3 8 9 11 12 14 15 17 18 20 21 23

POT

		0	13		Address Field

0 1 2 3 8 9 10 23

Bits 5 through 23 of the contents of the effective address of the POT instruction determine which block or blocks of memory are to be locked out as shown in table 3-25. The EOM instruction generates Rℓc in the main frame:

$$R\ell c = (EOM\ C10\ \overline{C11}\ C18 + Kmc)\ Q2$$

The Rℓc signal clears the write lockout register, Rℓ00 through Rℓ03 and Rℓ1 through Rℓ7. Flip-flop Ed, also set by Rℓc, alerts the interlock circuitry.

3-602 The lockout register set signal, Rℓs, is generated from the POT instruction:

$$R\ell s = Ek\ (POT\ Q2)$$

The Rℓs signal is combined with bits 5 through 23 of the contents of the POT instruction effective address to set the respective Rℓ flip-flops:

$$sR\ell 00 = R\ell s\ C23j$$
$$sR\ell 01 = R\ell s\ C22j$$
$$sR\ell 02 = R\ell s\ C21j$$
$$sR\ell 03 = R\ell s\ C20j$$
$$sR\ell 1 = R\ell s\ C19j$$
$$sR\ell 2 = R\ell s\ C18j$$
$$\vdots \qquad \vdots$$
$$sR\ell 7 = R\ell s\ C13j$$

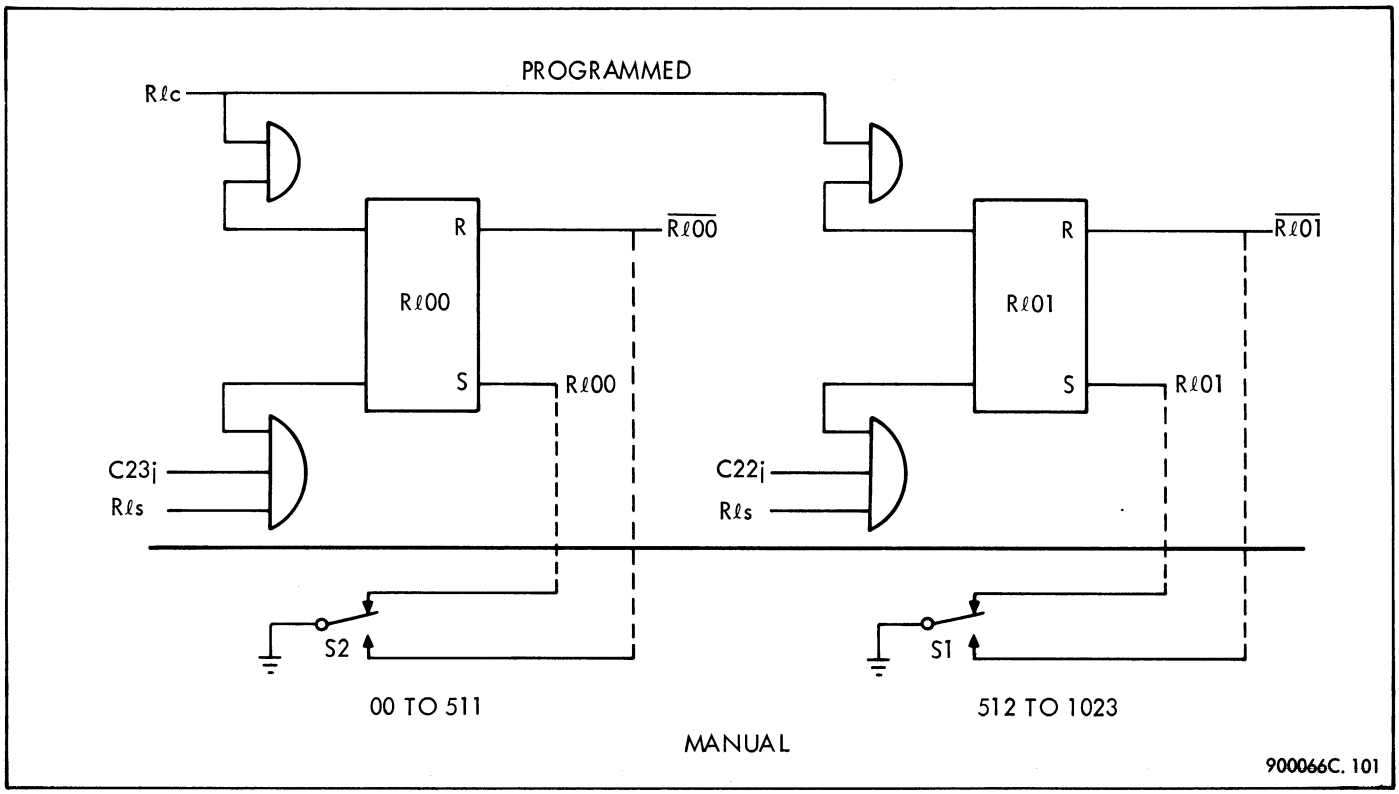


Figure 3-101. Write Lockout Register (Two Bits)

Table 3-25. Write Lockout Memory Blocks

Addresses (Decimal)	Addresses (Octal)	SX51 Module Location	Switch No.	Bit Number (Programmed Lockout)
0-512	000-777	7F	S2	C23
512-1023	1000-1777	7F	S1	C22
1023-1535	2000-2777	6F	S3	C21
1536-2047	3000-3777	6F	S2	C20
2048-4095	4000-7777	6F	S1	C19
4096-6143	10000-13777	4F	S3	C18
6144-8191	14000-17777	4F	S2	C17
8192-10239	20000-23777	4F	S1	C16
10240-12287	24000-27777	3F	S3	C15
12288-14335	30000-33777	3F	S2	C14
14336-16383	34000-37777	3F	S1	C13
16384-18431	40000-43777			C12
18432-20479	44000-47777			C11
20480-22527	50000-53777			C10
22528-24575	54000-57777			C9
24576-26623	60000-63777			C8
26624-28671	64000-67777			C7
28672-30719	70000-73777			C6
30720-32767	74000-77777			C5

510_{10}-word blocks

2048_{10}-word blocks

Signals C23j through C13j are selected from C23 through C5 by the lockout jumper module. The jumper module used is determined by the address of the first word in the memory door. For example, if the memory section locked out is in the second memory door and the first address in the memory is 4096, bit C18, which selects block 4096 through 6143, must be interpreted as C23j, or block 0 through 4095, for that particular memory door. If the programmed memory lockout is installed in the first memory in a system, a ZB65 through ZB50 jumper module is used. This jumper module connects each C bit in the range C12 through C23 with its corresponding j term. If the lockout is installed in the second, third, or fourth memory in a system, jumper modules are used as shown in table 3-26.

Table 3-26. ZB65 Jumper Modules

First Address in Memory Door	Jumper ZB65 Module
4K	ZB65-51
8K	ZB65-52
12K	ZB65-53
16K	ZB65-54
20K	ZB65-55
24K	ZB65-56
28K	ZB65-57

The equations for the jumper modules are given in the logic equations.

3-603 In each memory cycle, address lines L1 through L14 are compared with the contents of the lockout register (Rℓ flip-flops), and if the program accesses a locked-out location, the register lockout match signal, Rℓm is generated:

$$R\ell m = R\ell 00\ \overline{L1}\ \overline{L2}\ \overline{L3}\ \overline{L4}\ \overline{L5} + R\ell 01\ \overline{L1}\ \overline{L2}\ \overline{L3}\ \overline{L4}\ L5 + R\ell 02\ \overline{L1}\ \overline{L2}\ \overline{L3}\ L4\ \overline{L5} + R\ell 03\ \overline{L1}\ \overline{L2}\ \overline{L3}\ L4\ L5 + R\ell 1\ \overline{L1}\ \overline{L2}\ L3 + R\ell 2\ \overline{L1}\ L2\ \overline{L3} + R\ell 3\ \overline{L1}\ L2\ L3 + R\ell 4\ L1\ \overline{L2}\ \overline{L3} + R\ell 5\ L1\ \overline{L2}\ L3 + R\ell 6\ L1\ L2\ \overline{L3} + R\ell 7\ L1\ L2\ L3$$

3-604 If the memory cycle is not a read-restore cycle, but new information is to be written into the location. Rℓm is used along with Mxz or Mxc to set the lockout flip-flop, Iwm:

$$sIwm = R\ell m\ Za\ Mxz\ Re + R\ell m\ Sa\ Mxc\ Re$$

Setting Iwm allows reading out of the addressed word by generating Ms, the memory strobe generated from $\overline{Du5}$:

$$\overline{Du5} = Du5\ (\overline{Mxz}\ Za + \overline{Mxc}\ Sa + Iwm)$$

However, Iwm inhibits setting the memory register from the computer or the input/output unit by inhibiting the two strobes. For bit 0 of the register the equation is as follows:

$$sM0 = (Sa\ Mxc\ \overline{Iwm}\ Tp)\ C0 + Za\ Mxz\ \overline{Iwm}\ Tp)\ Z0$$

Since M is unchanged, the original information is returned to the locked-out location.

3-605 When the lockout register has been set up, it can be changed in one of two ways:

a. Depressing the memory CLEAR switches on the control console:

$$R\ell c = (. . . + Kmc)\ Q2$$

Depressing the memory CLEAR switches sets the Ek flip-flop (with Kmc); therefore, the START pushbutton should be pressed to reset Ek. Otherwise, an Rt signal will be generated and the lockout register will respond to the next POT instruction.

b. Using another set of EOM, POT instructions to change the register.

3-606 Manual Write Lockout Model 92060

3-607 The toggle switches for the manual write lockout option are located on the front of the SX51 modules in locations 3F, 4F, 6F, and 7F in the memory door. Any combination of switches may be used to lock out any desired block of memory locations. The switch identifications and the address block locked out by each switch are shown in figure 3-102. Table 3-25 contains the same information as well as the octal notation for each address block. Switch S3 in location 7F should be left in the up position at all times, because the outputs of this switch are connected to the Ek flip-flop, which when set generates a ready signal Rt, for a POT instruction.

3-608 The switch outputs substitute for the lockout register flip-flop outputs in that they provide the Rℓ signals for

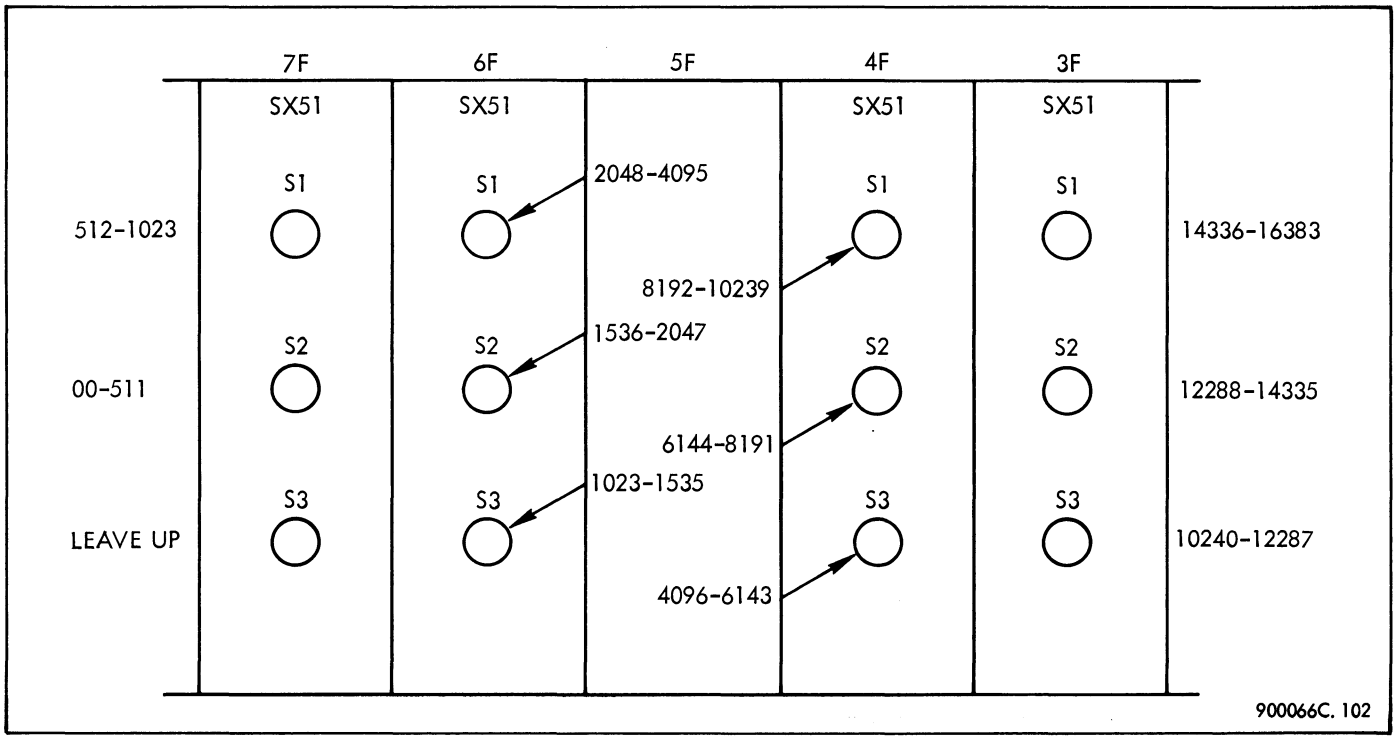


Figure 3-102. Write Lockout Switches

comparison with the address lines. The remainder of the lockout operation is the same as programmed lockout.

3-609 Lockout Interrupt

3-610 A special interrupt, IL, is provided to inform the programmer that an access of a locked-out location has taken place. This interrupt is sent to the basic interrupt chassis, where, when enabled, the signal interrupts the central processor to location 358.

$$sIL = Iwm\ Tp$$

$$rIL = \overline{Iwm}\ Tp$$

In resetting Iwm, the circuit characteristics of the DC Flip-Flop FB50 are used. The reset term, $\overline{Iwm}\ Tp$, is inverted and connected to the set gate output so that at Tp time when Iwm is false, $\overline{Iwm}\ Tp$ at the set gate output goes false and resets the flip-flop.

3-611 MEMORY HEATING SYSTEM

3-612 The basic memory stack, including frames and ground planes, is mounted inside an oven. The oven, which is also used to mount the connectors, provides protection for the stack and assists in providing thermal insulation. The stack heaters, together with the diode boards used for temperature sensing and the thermostats used as warning temperature indicators, are also mounted inside the oven. See figures 3-103 and 3-104 for the heater components for the 4K and 8K stacks and for a 16K stack.

3-613 The temperature of the memory stack is controlled to approximately ±2°C by a closed-loop feedback amplifier in the memory heater Power Supply PX21. Eight series-connected diodes are located inside the memory enclosure to sense the memory temperature and provide a feedback signal for controlling the memory temperature. The heater-regulator amplifier is a differential-input, single-ended, output-type amplifier with a bridge circuit to drive the inputs. The eight sensing diodes form one leg of the bridge. The amplifier output drives the heater, located inside the memory enclosure, through a power inverter. Two thermal switches, an under-temperature switch, and an over-temperature switch are also located inside the memory enclosure. The over-temperature switch is a protective device to remove heater power if the memory temperature exceeds a certain level. The under-temperature switch is used to generate an enable signal, Tem, when the memory

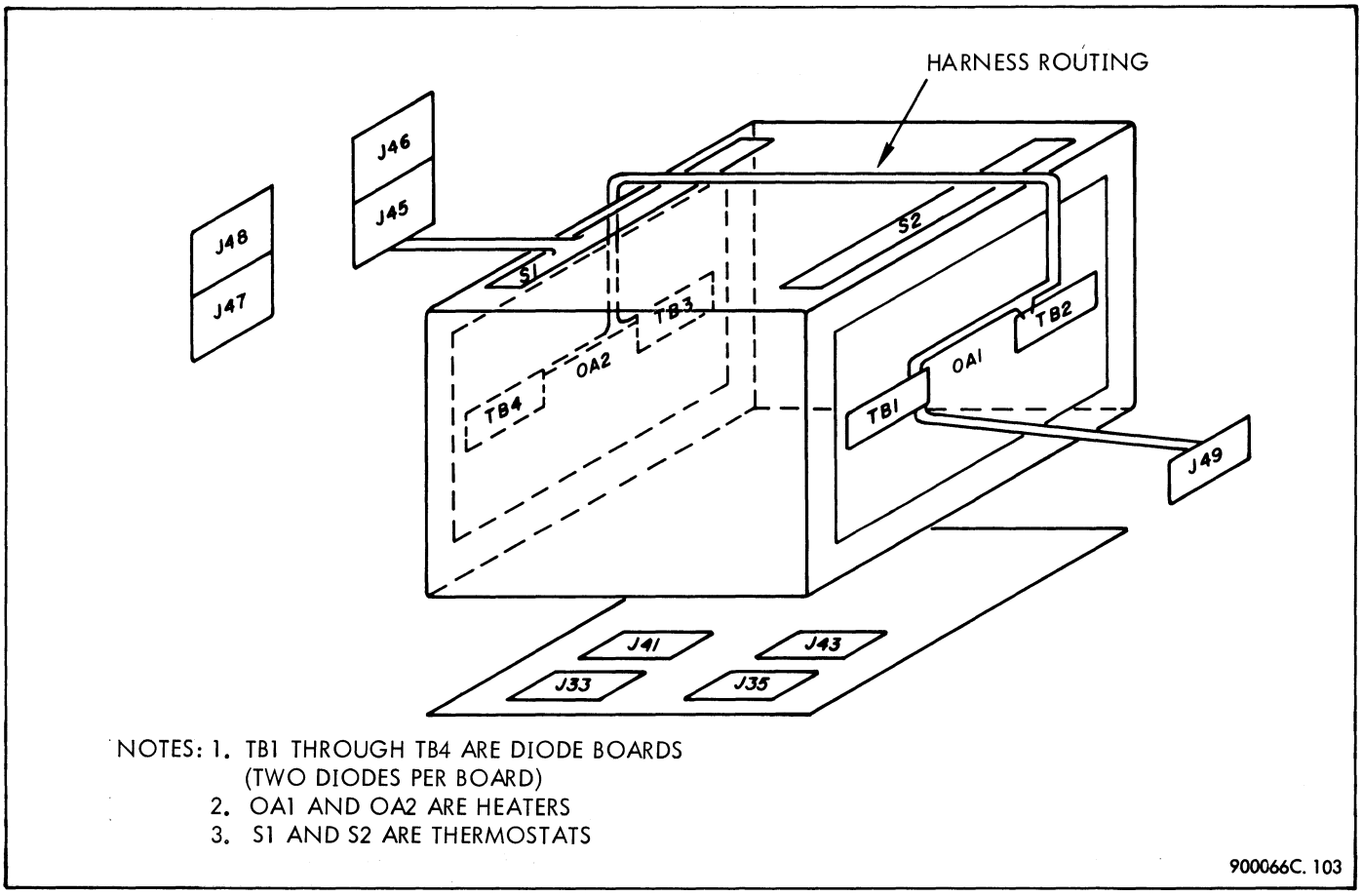
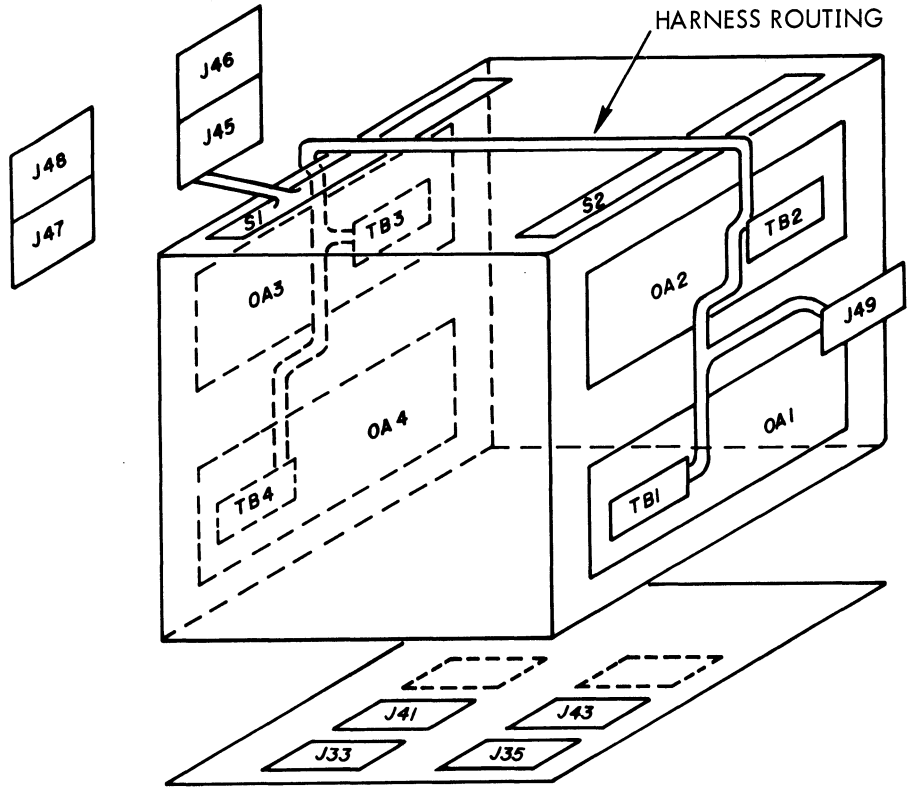


Figure 3-103. Heater Components, 4K and 8K Stacks



NOTES: 1. TB1 THROUGH TB4 ARE DIODE BOARDS (TWO DIODES PER BOARD)
2. OA1 AND OA2 ARE HEATERS
3. S1 AND S2 ARE THERMOSTATS

900066C. 104

Figure 3-104. Heater Components, 16K Stack

temperature is up to its operating point. A $\overline{\text{Tem}}$ signal inhibits central processor operation. Neither of the thermostat switches is used for regulator cycling during normal operation. The overall temperature control system is shown in figure 3-105.

3-614 The 4K and 8K stacks contains two heater elements, and the 16K stack contains four heater elements (see figures 3-103 and 3-104). Each heater element has a resistance of 37 ohms at 25°C and produces 40 watts of heat. The total heater power in the 4K and 8K stack is 80 watts, and in the 16K stack is 160 watts. In addition, up to 10 watts of heating may be produced by X, Y, and inhibit currents in the stack wire resistance. The range of the regulator system is ±2°C to ±3°C about the average stack temperature; therefore, the stack temperature ranges from about 50°C to 55°C.

3-615 POWER DISTRIBUTION

3-616 Three-phase, 208-volt power is wired to a circuit breaker on the ac power distribution panel. From the circuit breaker it is wired to a contactor, and from the contactor it is wired to the inputs of a PX25 ac line regulator. The line regulator output consists of three one-phase, 120-volt, 1 kva lines as shown in figure 3-106. All ac regulated and unregulated power is wired to ac plug molds, and the power supplies are in turn connected to the plug molds. The power supplies contained in the 930 Computer central processor and memory are identified in table 3-27. The memories in serial numbers 3101 through 3112 operate from Harrison Laboratories power supplies.

3-617 Regulated and unregulated ac power is distributed to each input/output rack for use by Power Supplies PX22 and PX23 and the blower in the input/output equipment.

3-618 Each memory system contains its own PX20 and PX21 power supplies to provide drive voltage Vd, inhibit voltage Vz, threshold voltage Vt, and heater voltage. The memory power supplies also provide temperature regulation for the memory stack, as described in the Maintenance section.

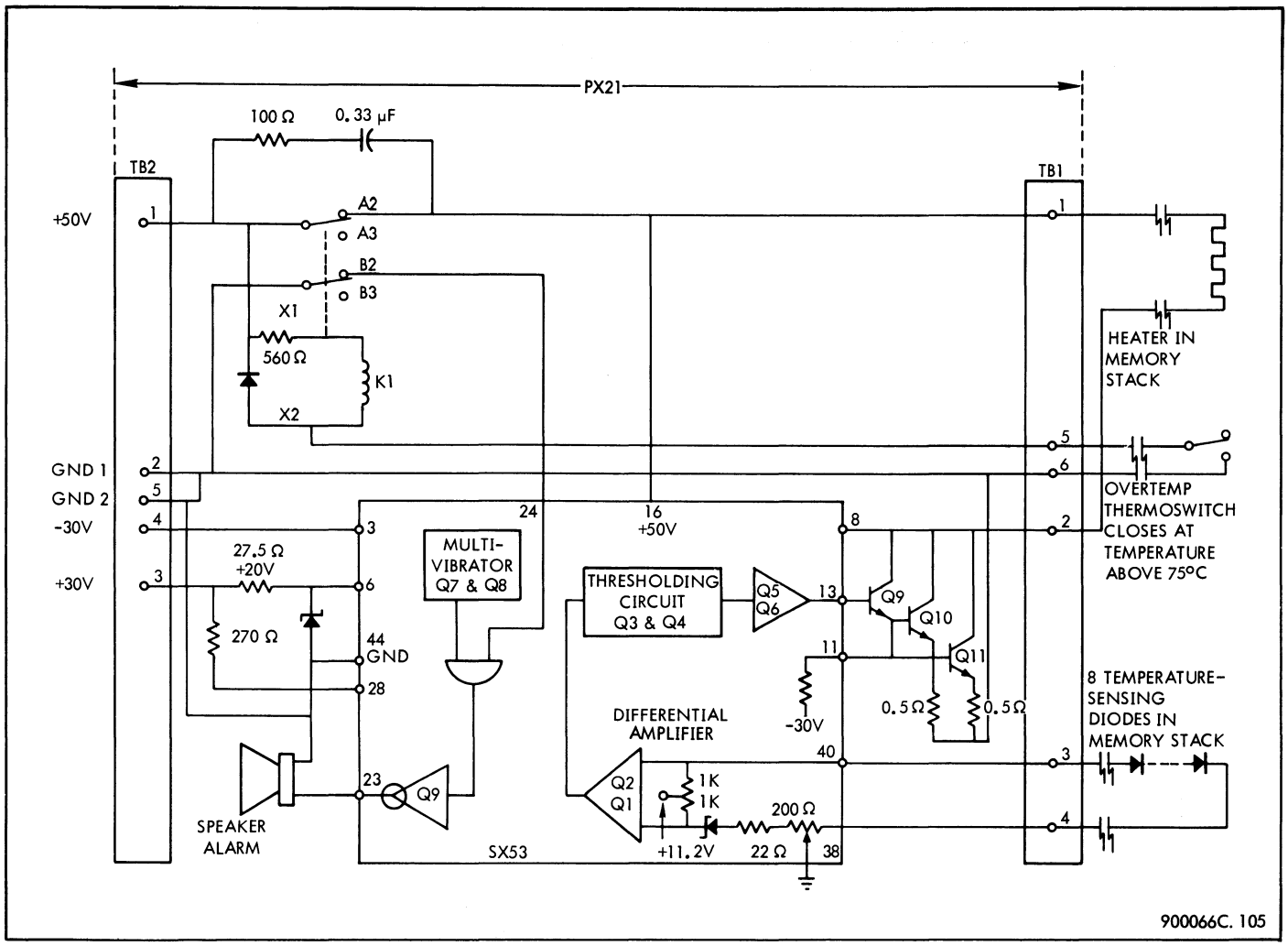


Figure 3-105. Memory Heater Regulator

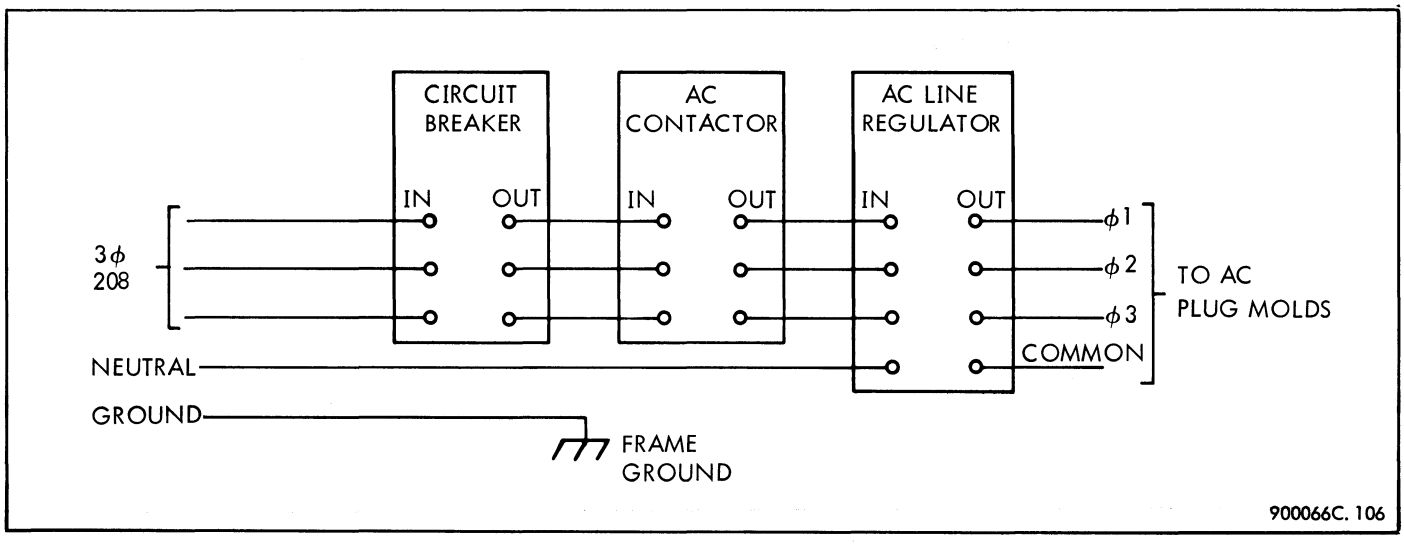


Figure 3-106. AC Power Distribution

Table 3-27. Power Supplies

Power Supply	Location	Voltage
PX18	CPU	+4
PX19	CPU	+16, -16
PX22	CPU	+4, +16, -16
PX23	CPU	+8, +25, -25, +50, 1 vac
PX20	Memory	Delivers rough regulated voltages to PX21
PX21	Memory	+24.1 through +34.0 (switch controlled) 0.9 through 3.0 (switch controlled)

3-619 The PX20 is a complete dc power supply, which requires both a regulated ac input for the Vd, Vt, and Vz supplies, and an unregulated ac input for the heater supplies.

3-620 Power Supply PX21 stabilizes the dc PX20 voltages and provides the Vd, Vt, and Vz voltages. The Vz output, which can be controlled by 100 switch positions, has an output capability of 7 amperes. The typical 16K memory load is about 5.5 amperes when writing all ZERO's. The Vd output, also controlled by 100 switch positions, has an output capability of 3 amperes. The typical 16K memory load is about 2 amperes. The Vd and Vz voltage-current relationships are shown in figure 3-107. The Vt output is adjustable by 10 switch positions. The threshold circuit requires +16 volts supplied from memory. A separate ground return is used for the threshold supply to minimize noise.

3-621 The details of power supply operation are described in the individual power supply manuals listed in the List of Related Publications in the front of this manual.

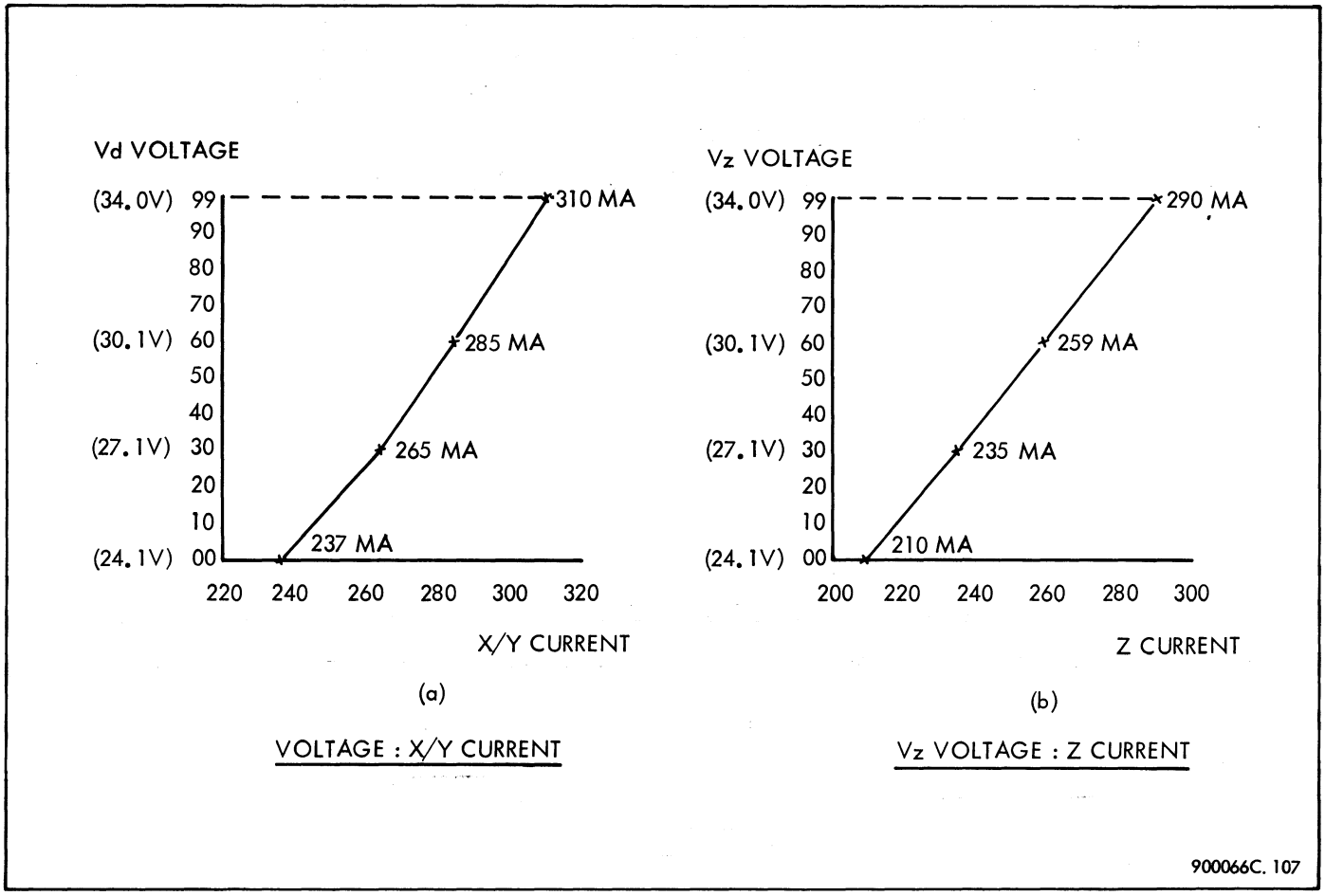


Figure 3-107. Drive Voltage-Current Relationships

Table 3-28. Central Processor Terms

Term	Description
Add1, 2, 3	Outputs of 3-bit serial adder
Anr	Signal indicating that A-register should not recirculate
Ar1	Control signal that shifts A right one bit position
Ar3	Control signal that shifts A and B right three bit positions
Al2	Control signal that shifts A and B left two bit positions
A00	Flip-flop that momentarily extends sign of the A-register. Also used, along with B00, to remember which register is displayed when computer is in IDLE, and to count cycles during SKS instruction
A0 thru A23	Twenty-four A-register flip-flops
Bc2, Bc5 thru Bc23	Eight "carry" flip-flops in B-register
Bk2 thru 22	Carry signals in right shift parallel adder
Bnr	Control signal indicating that B-register should not recirculate
Br1	Control signal that shifts B-register right 1 bit position
Bz0	Sign detector circuit which indicates that if Bc flip-flops were to be added into B, a carry would be propagated into B0
B00	Flip-flop that extends B-register
B0 thru B23	Twenty-four B-register flip-flops
Cd0 thru Cd23	Input data lines read by C-register during PIN instruction
Cen	Signal that gates C-register information onto cable drivers
Ci0 thru Ci23	C-register cable drivers
Ck	Control signal that inverts C-register bits in parallel, producing a one's complement

Table 3-28. Central Processor Terms (Cont.)

Term	Description
Cm0 thru Cm24	Cable drivers that normally send C-register contents to memory and I/O
Cp	Flip-flop that checks memory parity
Cpi	Flip-flop set when input/output parity error occurs with HALT CONTINUE switch in HALT position
Cpr	Flip-flop that generates parity for the C-register three bits at a time as the data is sent serially to the time multiplexed I/O channels
Cpu Parity	Signal to interrupt circuits indicating that an internal memory parity error has occurred
Cr3	Control signal that causes C to shift right three bit positions
Cxi	Control signal that reads Cd input lines into C
Cxm	Control signal that causes parallel transfer of information from M to C
Cz	Carry flip-flop in serial adder
C0 thru C23	Twenty-four C-register flip-flops
C1x	Signal from C1 in C-register used by I/O channel to distinguish between (W or Y) and (C or D) channels
C0i thru C23j	Signals that light REGISTER DISPLAY indicators
C21r thru C23r	Signals used for serial transfer from C-register to buffer register
C24	Flip-flop that generates memory parity
Eax	Control flip-flop set by EAX instruction. Also used for Programmed Operator function
Em1i thru Em2i	Signals that light MEMORY EXTENSION indicators
En	Interrupt enable flip-flop
(En)	INTERRUPT ENABLED switch

Table 3-28. Central Processor Terms (Cont.)

Term	Description
End	Signal true during last cycle of each instruction
Eod	Signal true during EOD instructions
Eom	Signal true during EOM instructions
Ex	Flip-flop that controls exchanges in conjunction with register switch
E2m0 thru E2m2	Memory extension flip-flops
F1, F2, F3	Phase counter flip-flops
Go	Flip-flop that allows computation
Ha1, 2, 3	Half-adder output signals
Ht	Flip-flop that halts computer
Hx1, 2, 3	Half-adder input signals
Hz	Half-adder carry flip-flop
Ia	Indirect address flip-flop. Also used as a carry to add one to contents of P-register
Ib	Interrupt subroutine exit signal
Ie	Interrupt subroutine entry signal
Ij	Signal from interrupt logic calling for a single instruction or special interrupt
IL	Write lockout signal from memory
Inr	Interrupt interlock flip-flop
Int	Interrupt flip-flop
I off	Power off signal from interrupt circuit
I/O Parity	Signal to interrupt circuits indicating that a parity error has been detected in the input/output equipment
Ir	Interrupt signal, enable necessary
Ir0 thru Ir14	Interlace address signals from buffer
Is	Interrupt signal, enable unnecessary

Table 3-28. Central Processor Terms (Cont.)

Term	Description
Ix	Index flip-flop which controls indexing. Time-shared in several other operations
Iz0 thru Iz14	Address lines sent from direct access I/O, used to set Jz register
Ju	Flip-flop that controls branching and programmed operator function
Jz0 thru Jz14	Register that accepts and synchronizes memory address for direct memory access I/O channels
(Ka) (Kb), (Kc), (Kx)	Register switch positions A, B, C, and X
(Kb1) thru (Kb4)	Breakpoint switches
(Kc0) thru (Kc23)	C-register set pushbuttons
(Kcc0) thru (Kcc2)	I/O DISPLAY SELECT switch
(Kcc1x)	Switch-controlled signal used to distinguish between the (W or Y) and (C or D) channels
(Kcm)	MEMORY CLEAR switch
(Kcr)	Switch that clears C-register
(Kfc)	Cards switch
(Kfd)	Drum switch
(Kfm)	Magnetic tape switch
(Kfp)	Paper tape switch
(Kf1), (Kf2)	Signals actuated by load or fill switches
(Kg)	RUN switch
(Kp)	Parity HALT CONTINUE switch
(Kr)	HOLD switch
(Ks)	STEP switch
K0	Control flip-flop used in multiplication, division, and shifting
Ls0 thru Ls14	Cable drivers that send memory address from computer to memory

Table 3-28. Central Processor Terms (Cont.)

Term	Description
Lz0 thru Lz14	Cable drivers that send memory address from Jz register to memory for direct access I/O channels
Md	Memory disable
Mgs	Flip-flop that requests memory to cycle for Ls address
Mgz	Flip-flop that requests memory to cycle for Lz address
Mit	Signal from memory indicating that Ls and Lz addresses are actively addressing the same memory module
Mxc	Signal sent to memory when new information is to be written into memory
M0 thru M24	Memory register
N5 thru N14	Interrupt address lines
Oc	Signal that clears 0 register
Of	Overflow flip-flop
Ofe	Signal from serial adder indicating overflow condition
Ofi	Signal that lights OVERFLOW indicator
Oxc	Signal that transfers opcode from C-register to O-register
O1 thru O6	Six operation code register flip-flops
Pin	A signal sent to external equipment during PIN instruction
Pio	Signal indicating that parity interrupt option is present and HALT switch is on
Pot	Signal sent to external equipment during POT instruction
Pr3	Control signal that causes P-register to shift right three bit positions
P0 thru P14	Fifteen-bit P-register

Table 3-28. Central Processor Terms (Cont.)

Term	Description
P1i thru P14i	Signals that light P-REGISTER indicators
Q1 thru Q6	Pulse counter, or cycle counter
Rad0 thru 23	Outputs of right-shift parallel adder
Rc	Flip-flop true during register change instruction
Rde	Signal from buffer denoting status of error detector
Rdei	Signal that lights ERROR indicator
Rd9 thru Rd14	Signals from buffer denoting unit address register contents
Rd9i thru Rd14i	Signals that light INPUT OUTPUT indicators
Rf	Flip-flop that receives ready signal during I/O instructions also used in division and normalize
Rℓc	Signal to memory during EOM instruction
Rmp	Parity error signal from I/O channel
Rn1, 2, 3	Three flip-flops that receive serial information from time-multiplexed I/O channels
Rsa	Control signal that transfers right shift adder into B-register in parallel
Rt	Read signal that releases central processor from parallel input or output instruction
Rti	Signal sent to external equipment indicating that PIN instruction has just terminated
Rwy1 thru Rwy3	Serial data signals received from I/O channel during WIM instructions
Rx	Signal sent to time-multiplexed channels during programmed I/O instruction

Table 3-28. Central Processor Terms (Cont.)

Term	Description
Rr1 thru 3	Octal of data transmitted by the buffer
Sc	Control signal that clears S
Sd2	Signal that causes S-register to count down by two's
Sk	Skip flip-flop
Ski	Output signal indicating that a skip occurred during a single instruction interrupt
Skrz	Skip gate signal from I/O channel
Skss	Strobe signal sent to external equipment during SKS instructions
Sks	Signal that controls the Skip If Signal Not Set Instruction
(St)	Start switch
Sxc	Control signal that causes C to S parallel transfer
Sxn	Control signal that loads S-register with information on N lines from interrupt logic
Sxp	Control signal that causes P to S parallel transfer
Sx48	Signal that forces decimal 48 into S-register
S1 thru S14	Fourteen-bit S-register
Tem	Signal indicating memory stack temperature is up to operation level
Ti	Timing signal used in interrupt circuits
Tp	Timing pulse decoded from Q counter
Tpc	Timing pulse that synchronizes pulse counter in I/O channel with pulse counter in central processor
Tr	Timing pulse decoded from Q counter

Table 3-28. Central Processor Terms (Cont.)

Term	Description
Trq	Request signal sent from time-multiplexed I/O channels
Ts	Flip-flop set during time-share interlace operation. Stops clock to many circuits
Tsm	Flip-flop that counts cycles during interlace with multiplexed channels and switches address lines to memory
Tsr	Flip-flop true during time-share interlace operation with multiplex channel
T8 thru T0	Timing pulses decoded from Q counter
Wf	Signal indicating whether W buffer register is full or empty
W0	Halt interlock signal from W buffer
W9	Signal from W buffer unit address register defining whether process is input or output
Xnr	Signal which stops X-register from recirculating
Xn1 thru Xn3	X-register outputs
Xw1	Write flip-flop on the most significant bit of three X-register recirculation loops
Xz1, 2, 3	Addend input to serial adder
X1d	Delay element usually fed by Xw1
Yf	Signal indicating whether Y-buffer register is full or empty
Yz1, 2, 3	Augend input to serial adder
Y0	Halt interlock signal from Y buffer
Y9	Signal from Y buffer unit address register defining whether process is input or output
Zrq	Memory request signal sent from direct memory I/O channels. Sets Mgz
Ø0 thru Ø7	Phase signals decoded from phase counter

Table 3-29. Memory Terms

Term	Description
C0 thru C24	Incoming data from central processor C-register
C13j thru C23j	Input signals to lockout register. Signals selected from C5 thru C23 by lockout jumper module in programmed lockout option
Du1	One-shot delay that ensures that X and Y drive switches are fully turned on before X and Y read sink switches are selected by Re signal
Du2	One-shot delay that sets duration of X and Y read currents by turning off Re signal
Du3	One-shot delay that determines duration of Z (inhibit) current. Also sets the duration of X and Y write currents by turning off We signal
Du4	One-shot delay that ensures that inhibit current is fully established in stack before X and Y write currents are turned on by We signal
Du5	One-shot delay that sets timing of memory strobes Ms1 through Ms4 to coincide with core output data from sense amplifiers
Du6, Du8	Parallel one-shot delays that determine memory strobe pulse widths
Dx0 thru Dx7	X drive switches used to drive 4K, 8K, and 16K stacks
Dx10 thru Dx17	X drive switches used to drive 8K and 16K stacks only
Dy0 thru Dy7	Y drive switches used to drive 4K, 8K, and 16K stacks
Dy10 thru Dy17	Y drive switches used to drive 16K stack only
Ek	Enable flip-flop in memory that allows the following parallel output (POT) from computer to set up memory lockout register. Set by special signal from computer
Iℓ	Flip-flop that copies Iwm flip-flop at Tp time. Signals computer interrupt system that an attempt to write into a locked out portion of memory has been made
Iwm	Flip-flop set whenever an attempt to write in a locked out portion of memory is made. Inhibits modification of M-register and set Iℓ
L1 thru L14	Addresses used internally in memory. Generated from external Ls and Lz addresses
Ls0 thru Ls14	Incoming addresses from CPU S-register
Lz0 thru Lz14	Incoming addresses from CPU Jz register (direct I/O access to memory options only)
Ls0j thru Ls2j	Address terms, used for determining priority, derived by jumper module from Ls0 through Ls2
Lz0j thru Lz2j	Address terms, used for determining priority, derived by jumper module from Lz0 through Lz2
Ls1k, Ls2k	Address terms used to generate L1 and L2 address terms in memory. Derived by jumper module from Ls1 and Ls2
Lz1k, Lz2k	Address terms used to generate L1 and L2 address terms in memory. Derived by jumper module from Lz1 and Lz2
M0 thru M24	M-register

Table 3-29. Memory Terms (Cont.)

Term	Description
M0s thru M24s	Output of M-register controlled by the Ls address lines
M0z thru M24z	Output of M-register controlled by Lz address lines (direct I/O access to memory options only)
Md	Memory disable term for power fail-safe option
Md0 thru Md24	Digitized sense amplifier output before strobing into M-register
Mdt	Memory digit timing for inhibit drivers
Mg	Memory go signal generated from Mgs, and Mgz
Mgs	Signal that requests memory to perform a cycle on address presented on Ls lines
Mgz	Signal that requests memory to perform a cycle on address presented on Lz lines
Mit	Signal to computer and I/O channels indicating that memory module was addressed by both Z address lines and CPU memory address lines. Informs CPU that it will not be able to cycle for CPU since DACC (or DMC, etc.) is using memory and has higher priority
Ms1 thru Ms4	Signals that strobe sensed core data into M-register
Mxc	Signal that instructs memory to perform a clear write operation to store data presented on Cm lines
Mxz	Signal that instructs memory to perform a clear write operation to store data presented on Zm lines
(Pot Q2)	Compound signal generated in CPU, true during parallel output (POT) instruction. Used in memory only if memory lockout logic was previously enabled
Re	Read enable timing signal to the read current sink switches
Rℓ00 thru Rℓ03	Memory lockout register terms that can lock out first four 512-word blocks of memory
Rℓ1 thru Rℓ7	Memory lockout register terms that can lock out 2K word blocks of memory from 2K upwards
Rℓc	Signal that clears lockout register
Rℓm	Register lockout match signal that indicates that memory address is within a locked area of memory as indicated by memory lockout register
Rℓs	Signal that sets Cj terms into lockout register
Rt	Ready signal to CPU to release CPU from a parallel input or output instruction
Sa	Signal generated by priority logic, instructing memory to perform a cycle on address presented on Ls lines
Sb	A priority term used in generating Sa and Pa
Srx0 thru Srx7	X read current sink switches
Sry0 thru Sry7	Y read current sink switches

Table 3-29. Memory Terms (Cont.)

Term	Description
(St Q2)	A signal from the CPU that is activated by the RESET button on the console
Swx0 thru Swx7	X write current sink switches
Swy0 thru Swy7	Y write current sink switches
T3	Timing period pulse that initiates read half cycle memory timing
Tp	Timing period pulse that initiates write half cycle memory timing
Tem	Enabling signal generated by the under-temperature thermostat. Indicates memory stack is up to temperature
We	Write enable timing signal to write current sink switches
Z0d0 thru Z24d0	Output signals from Z drivers for address block 0 ($\overline{L1}$ $\overline{L2}$)
Z0d1 thru Z24d1	Output signals from Z drivers for address block 1 ($\overline{L1}$ L2)
Z0d2 thru Z24d2	Output signals from Z drivers for address block 2 (L1 $\overline{L2}$)
Z0d3 thru Z24d3	Output signals from Z drivers for address block 3 (L1 L2)
Za	Signal generated by priority logic, instructing memory to perform a cycle on address presented on Lz lines
Zb	Priority term used in generating Za and Sa
Zm0 thru Zm24	Incoming data from direct I/O access to memory data lines

SECTION IV
INSTALLATION AND MAINTENANCE

4-1 INSTALLATION

4-2 CONTROL CONSOLE

4-3 The control console assembly is mounted on a rectangular table 60 inches long, 22 inches wide, and 30 inches high. Overall height of the control console assembly is 44 inches. The assembly can be located in any position with relation to the central processor unit, within the limits of cable length. Cables between central processor unit and control console are 22 feet long from console cable trough entry. Control console cables enter the central processor unit through cable entry location C, shown in figure 4-1. Figure 4-2 is a plan view of the control console.

4-4 FORCED-AIR COOLING

4-5 Forced-air cooling is not a requirement, but provision is made for air cooling of the central processor cabinet from below (see figure 4-1). No provision is made for forced-air cooling of power supply or input/output cabinets.

4-6 On the bottom of the central processor cabinet are raised plenums. Covers on the plenums must first be removed to expose the bottom plates, which may then be removed. When forced-air cooling is used, it is necessary to cover the air vents in the rear doors of the central processor cabinet. Air normally enters each cabinet through these vents.

4-7 Local conditions will determine whether the fans and blowers inside the cabinets are required when air conditioning is used. If the force of cold air is sufficient, it may be desirable to remove the internal blowers. The internal blowers should be removed only when the forced air system is extremely reliable, since lack of air flow may cause electrical failures.

4-8 SYSTEM ARRANGEMENT

4-9 The arrangement of parts in the computer system is shown in figure 4-3. The parts contained in the lettered locations in the figure are identified in tables 4-1 and 4-2.

4-10 INPUT POWER WIRING

4-11 Figure 4-4 shows the details of the power cable entry box, located on the outside of the power supply cabinet below the rear door. Figure 4-5 is a schematic diagram of the input power wiring required to service the 930 Computer.

4-12 PRE-INSTALLATION CONSIDERATIONS

4-13 Raised flooring is required for cabling between 930 Computer cabinets, optional features and peripheral equipment. Cable troughs or covers for on-deck routing can be substituted if unavoidable, but are not recommended. Space for a documentation library should be provided in the computer room or nearby. Most of the documentation is used in maintenance and troubleshooting. A maintenance area should be available with space for test equipment, workbenches and spare parts. A large blackboard and other working aids are important.

4-14 INSTALLATION PROCEDURE

4-15 To install the 930 Computer, use the following procedure:

a. Bolt I/O and power supply cabinets to the central processor unit as shown in figure 4-3.

b. Connect power wires to computer central processor as indicated by tags on cable wire ends. Power wires are secured in power supply cabinet.

c. Install cables between units as listed in table 4-3. Refer to the 930 Logic Diagrams, SDS 900592.

d. Connect control console power cable wires to central processor as indicated in table 4-4. Cable is secured to control console and terminated with solderless lugs.

e. To determine power consumption and heat dissipation of a particular 930 Computer, add power consumption of computer and options using table 4-5. To determine total power consumption and heat dissipation for system, use information from applicable peripheral equipment installation data sheets. Three kva of power are delivered by Voltage Regulators PX25. If additional regulators are required, install in locations shown in tables 4-1 and 4-2.

4-16 Parity Interrupt Installation

4-17 To install the Parity Interrupt Option Model 92070, place a NAND flip-flop FB54 module in location 31E of the central processor and a Priority Interrupt SK61 module in location 28H of the basic interrupt chassis.

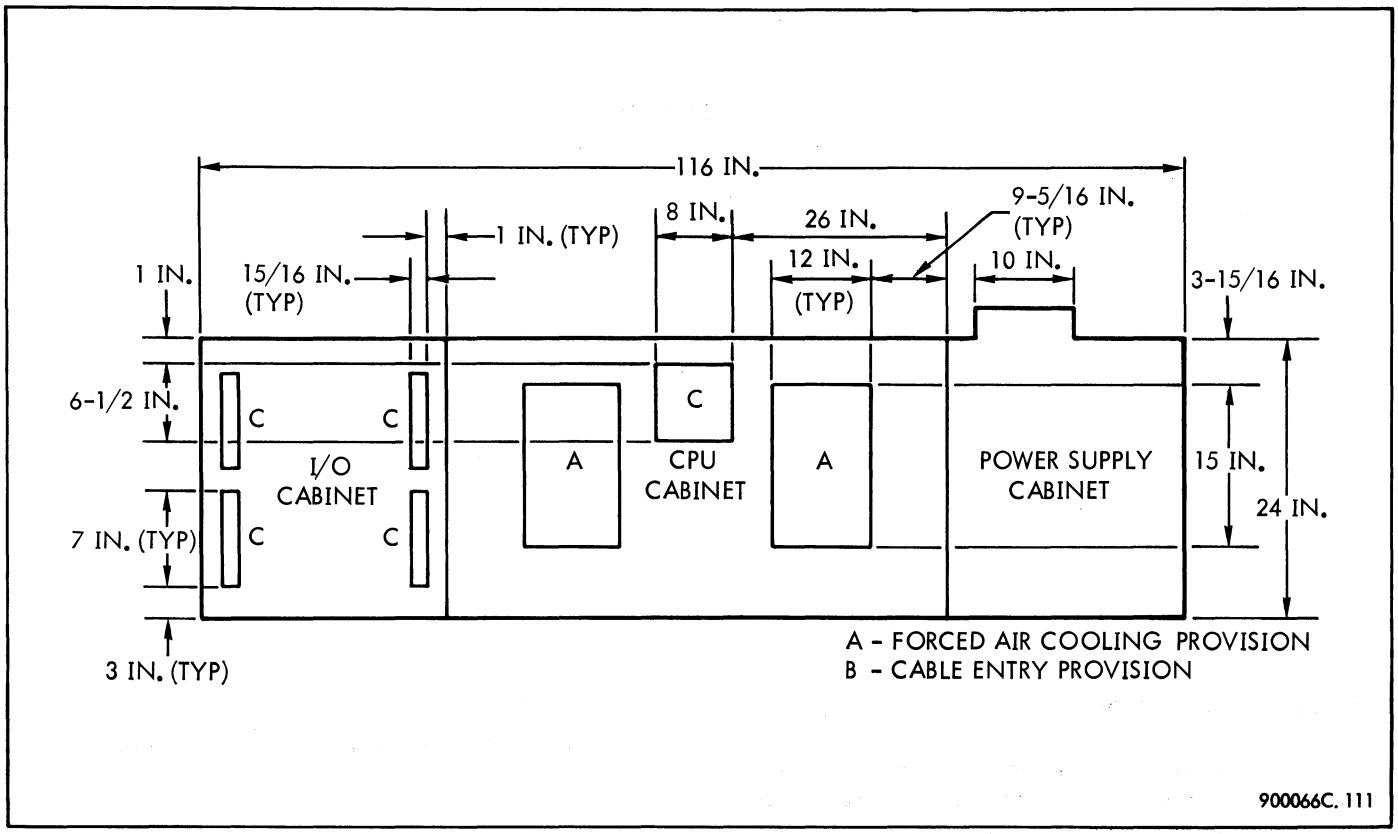


Figure 4-1. Cabinet Base, Plan View

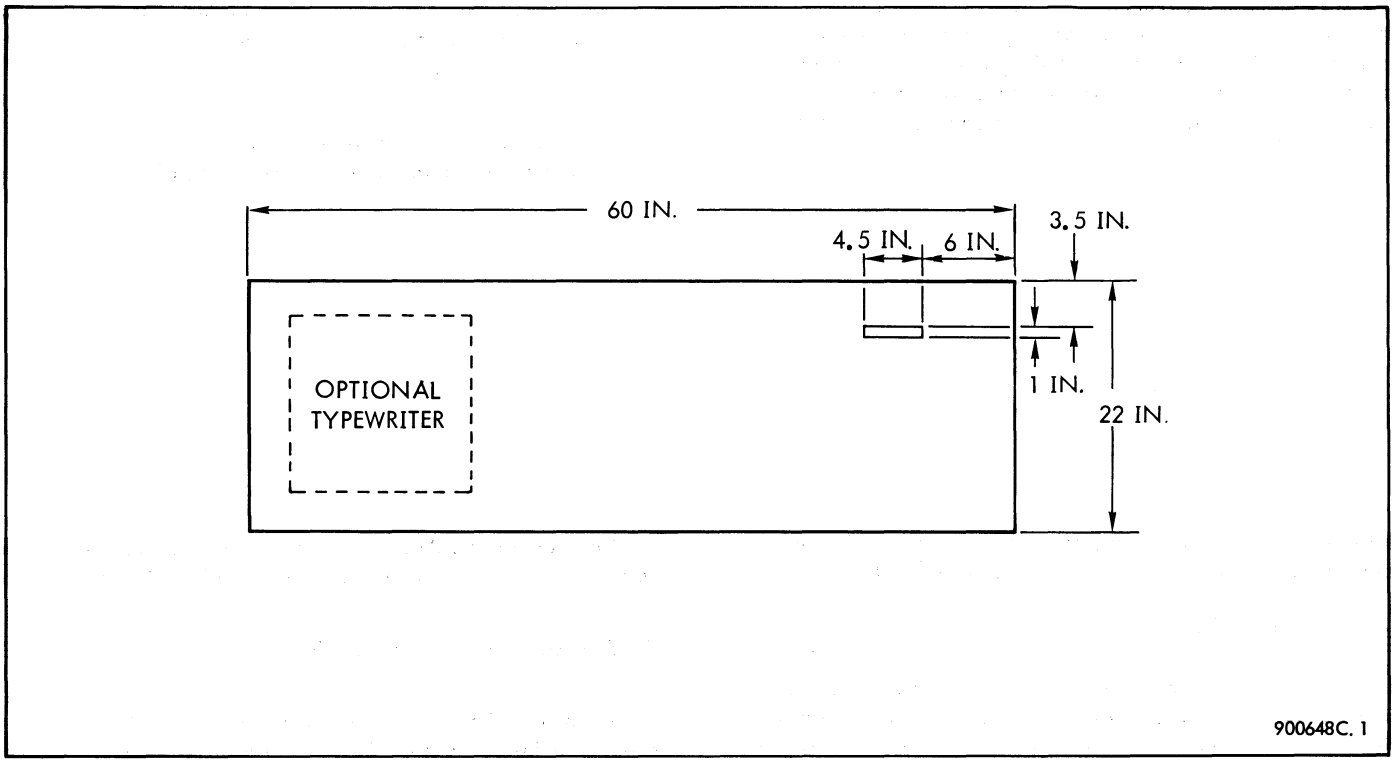


Figure 4-2. Console, Plan View

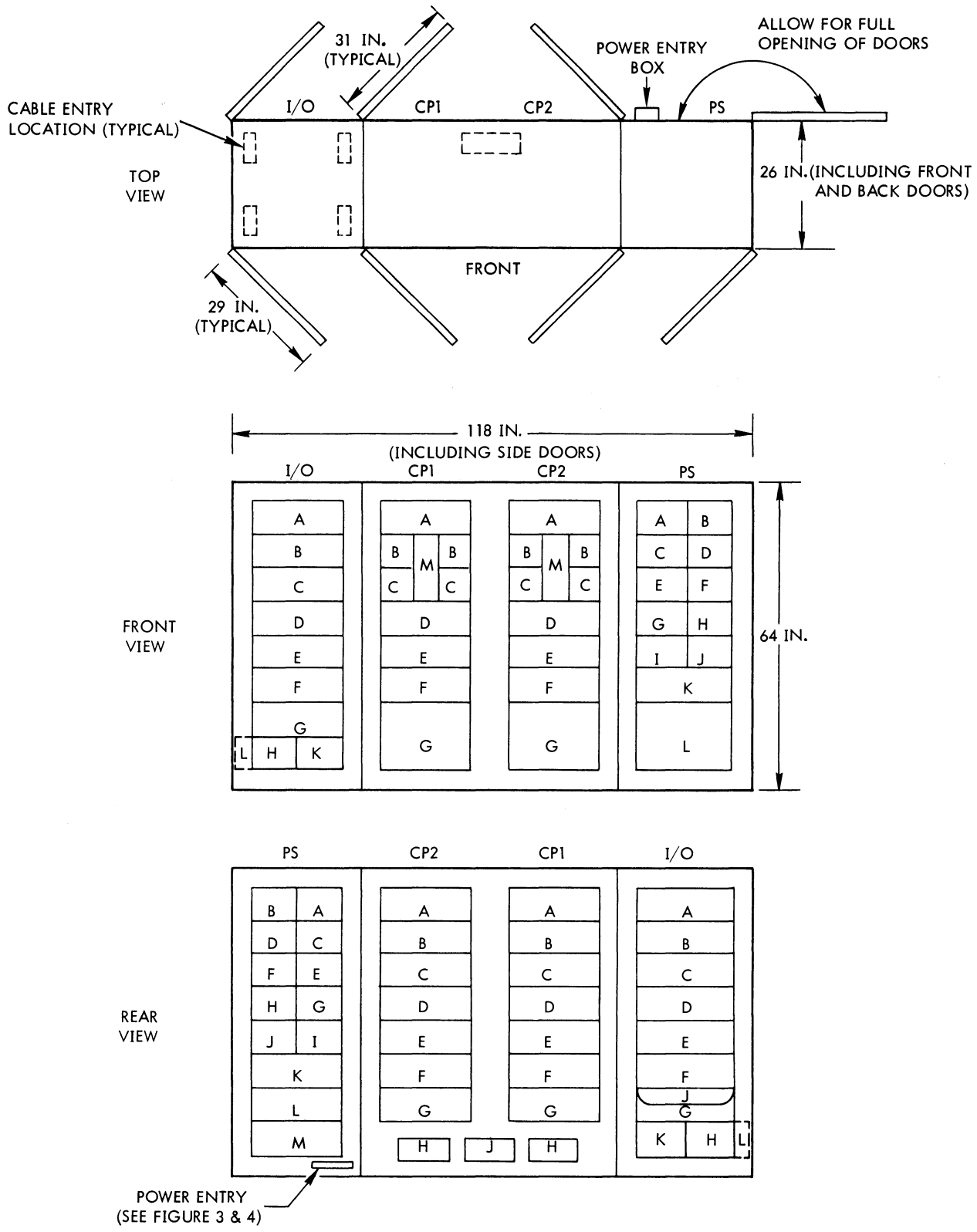


Figure 4-3. Physical Layout

Table 4-1. Equipment Location, Front View

Cabinet	Location	Contents
I/O	A thru F	Couplers for optional peripheral equipment; row F contains the Basic Interrupt (19-inch rack mounting)
	G	Blower unit
	H	Power Supply PX23
	K	Power Supply PX22
	L	Power Supplies PX14, 15, 16, or 19 for peripheral options
	(Last I/O cabinet only)	
CP1	A thru F	Memory bank No. 1
	G	Vacant
	M	Core memory, 4K standard, 8K or 16K optional
CP2	A thru F	Memory bank No. 2, optional
	G	Vacant
	M	4K, 8K, or 16K core memory for optional memory bank No. 2
PS	A } PS8	Power Supply PX20, furnished with second memory
	B } PS8	Power Supply PX21, furnished with second memory
	C } PS7	Power Supply PX20, furnished with first memory
	D } PS7	Power Supply PX21, furnished with first memory
	E - PS5	Power Supply PX18, for use with optional equipment
	F - PS6	Power Supply PX19, for use with optional equipment
	G - PS3	Power Supply PX18, standard
	H - PS4	Power Supply PX19, standard
	I - PS1	Power Supply PX19, standard
	J - PS2	Power Supply PX19, standard
	K - PS11	Voltage Regulator PX25, for use with optional equipment
	L	Power distribution panel

Table 4-2. Equipment Location, Rear View

Cabinet	Location	Contents
I/O	A, B	TMCC Channel B (Y buffer) optional, 6, 12, or 24 bits
	C thru F	TMCC Channel A (W buffer) 6-, 12-, or 24-bit selection
	F	I/O connectors (in addition to W buffer)
	G, H, K, L	Same as for I/O front
	J	Cable entry for on-deck cable routing (not recommended)
CP1	A	Vacant
	B thru F	Standard 930 Computer logic
	G	Vacant
	H	Air intake blower
	J	Same as for I/O
CP2	A thru H	Same as for CP1
PS	A thru K	Same as for PS front
	L - PS10	Voltage Regulator PX25, standard
	M - PS9	Voltage Regulator PX25, standard

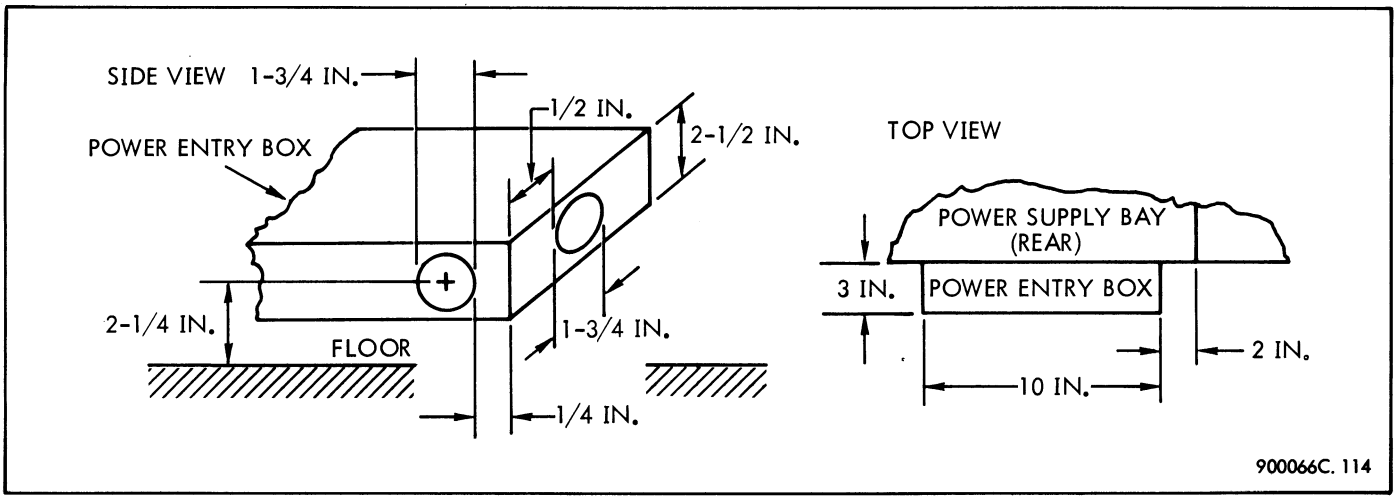


Figure 4-4. Power Entry Box

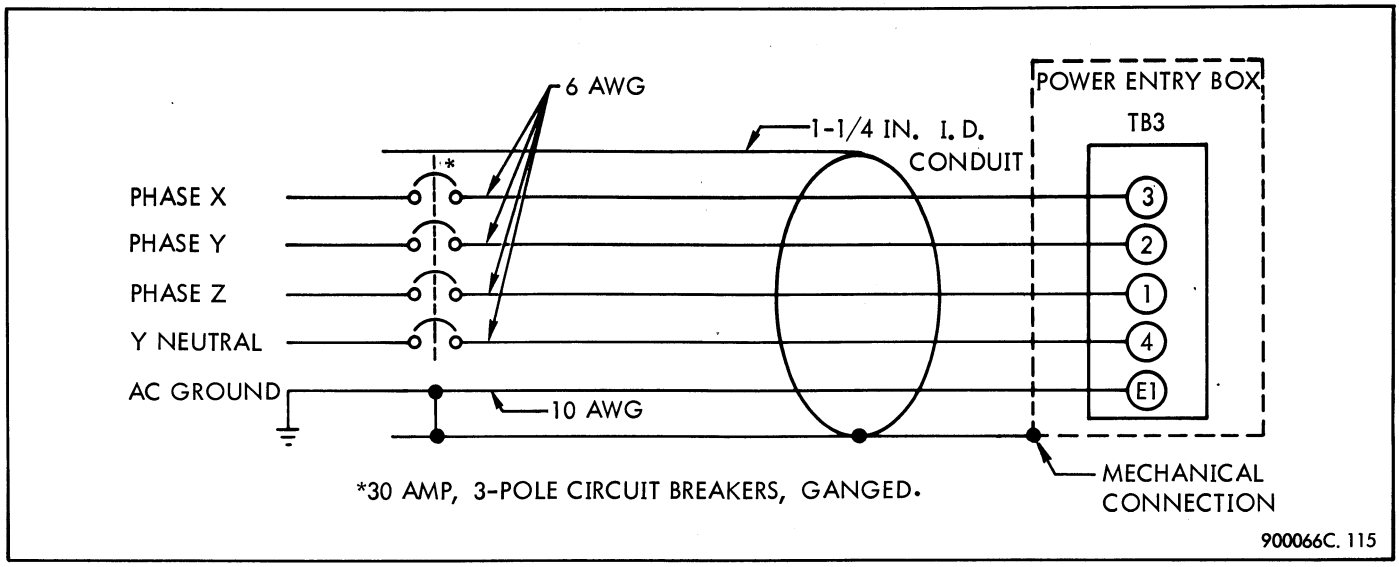


Figure 4-5. Service, Schematic Diagram

Table 4-3. Cable Installation

Plug Module	From Location	To Location
P901	Central processor 61F	Control console J1
P902	Central processor 62F	Control console J2
P903	Central processor 63F	Control console J3
P904	Central processor 64F	Control console J4
P905	Central processor 29F	First memory chassis 17F
P905	Central processor 30F	Second memory chassis 17F
P906	Central processor 33F	First memory chassis 10F
P906	Central processor 34F	Second memory chassis 10F
P907*	Central processor 35F	First memory chassis 13F
P907*	Central processor 36F	Second memory chassis 13F
P908	Central processor 7F	Permanently wired to basic interrupt chassis
P910	Central processor 2F	P909 TMCC 28E
P912	Central processor 1F	P911 TMCC 30E

Table 4-3. Cable Installation (Cont.)

Plug Module	From Location	To Location
P914	Central processor 3F	P913 TMCC 15E
P916	Central processor 5F	P915 TMCC 23E
P918	Basic interrupt chassis 44H	P917 TMCC 2F
P940	Central processor 12F	P939 DACC 8E
P941	First memory chassis 24F	P942 second memory chassis 25F
P942	Central processor 15F	P941 DACC 26E
P945	Basic interrupt chassis 45H	Permanently wired to interrupt control system chassis
AC	Power control console	Convenience strip central processor
I/O 3 phase supply, regulated and unregulated	5-wire power cables, I/O cabinet	5-pin power connector, ac power junction box, power supply cabinet

*Used only with MAM (DACC, DMC, MIC) or memory lock-out options. Requires Termination Module ZB52 in location 14F of respective memory.

Table 4-4. Control Console Connection

Wire No.	Control Console	Central Processor	Signal
1	TB3-1	TB3A-1	Run time meter
2	TB3-5	TB3A-8	110 vac } Power switch
3	TB3-6	TB3A-9	110 vac } Power switch
4	TB3-7	TB3A-6	1 vac
5	TB3-8	TB3A-7	1 vac, +4v bias
6	TB3-9	TB3A-5	+50 vdc

Table 4-5. Power Consumption and Heat Dissipation

COMPUTER CONFIGURATION OR OPTION	AC POWER (KVA)			BTU PER HOUR
		Internal		
	Line Input	Unregulated	Regulated	
Single memory bank and one TMCC I/O channel (W buffer)				
4K	4.77	2.00	2.49	11,950
8K	4.80	2.00	2.52	12,050
16K	4.85	2.00	2.56	12,200
Second memory bank				
4K	2.27	0.90	1.23	5,400
8K	2.31	0.90	1.27	5,550
16K	2.34	0.90	1.30	5,700
Second TMCC I/O channel (Y buffer)	0.28	None	0.25	950
Third TMCC I/O channel (C channel), with cabinet and required power supplies	0.66	0.20	0.41	2,200
Third and fourth TMCC I/O channel (C and D channel), with cabinet and required power supply	0.94	0.20	0.67	3,200
DACC, with cabinet and required power supply	0.82	0.20	0.55	2,750
MIC, without cabinet or power supplies	0.18	None	0.16	610

4-18 Addressing Modification Installation

4-19 To install the Addressing Modification Model 91903 feature, proceed as follows, referring to the logic diagrams in figure 3-10:

a. Remove and add wires as shown in table 4-6.

Table 4-6. Addressing Modification Wiring Changes

SIGNAL	REMOVE		ADD	
	From	To	From	To
C15	47C7	51C7	47C7	62D39
			62D39	51C7
$\overline{C15}$	16D5	17C4	16D5	62D27
			62D27	17C4
C12	26E7	24D42	26E7	62D20
			62D20	24D42
Eod C10 $\overline{C11}$	56C36	30D34	56C36	52C37
			52C37	30D34
Sxc	62E23	64E23	62E23	62D13
			62D13	64E23
S1	51D10	55C12	51D10	52C24
			52C24	55C12
S2	51D16	55C15	51D16	52C22
			52C22	55C15

Table 4-6. Addressing Modification Wiring Changes (Cont.)

SIGNAL	REMOVE		ADD	
	From	To	From	To
Sc +Ø1 05 T2	42E5	51C36	42E5	62D2
			62D2	51C36
St	52D5	55C22	52D5	62D26
			62D26	55C22
T4	53C41	40C41	53C41	52C38
			52C38	40C41
Cg24	45E49	50C41	45E49	62D43
			62D43	56D43
$\overline{Ls0A}$			52C28	51D9
$\overline{Ls1A}$	45D36	16E15	45D36	52C20
			52C20	16E15
$\overline{Ls2A}$	46C39	55C17	46C39	52C15
			52C15	55C17
Eod C10 $\overline{C11}$ T4			52C36	62D11
			62D11	60D39
Gnd No. 1			62D1	62D3
			62D3	62D4
			62D4	62D5

Table 4-6. Addressing Modification Wiring Changes (Cont.)

SIGNAL	REMOVE		ADD	
	From	To	From	To
Gnd No. 1 (Cont.)			62D5	62D18
			62D18	62D19
			62D19	62D21
Gnd No. 2			62D44	62D42
			62D42	62D41
			62D41	62D40
			62D40	62D25
			62D25	62D24
$\overline{\text{Shell}}$			62D38	61D23
			61D23	51D12
			51D12	51D18
			51D18	55C7
			55C7	55C39
Shell			62D23	52C25
			52C25	52C39
Sho			62D9	52C26
			52C26	52C40

b. With needle-nose pliers, remove polarizing pins in locations 52C and 62D. Pins are located beside pins 2, 24, and 46.

c. With needle-nose pliers or polarizing pin inserting tool, insert polarizing pins beside pins 30 and 44 in location 52C and beside pins 28 and 42 in location 62D.

d. Insert IB57 module in location 52C and FB51 module in location 62D.

e. Mount nameplate, part No. 101109, adjacent to 930 Computer nameplate and mark with 0.12 high black characters as follows:

SDS	32K Direct Address Kit Model No. 91903

4-20 Memory Write Lockout Installation

4-21 Manual Memory Write Lockout. To install the Manual Memory Write Lockout Model 92060 feature, proceed as follows:

a. Install modules in associated memory door and basic interrupt chassis as shown in table 4-7.

Table 4-7. Module Installation for Manual Write Lockout

Chassis	Location	Module
Memory Door	1F	IB56
Memory Door	2F	IB56
Memory Door	3F	SX51
Memory Door	4F	SX51
Memory Door	6F	SX51
Memory Door	7F	SX51
Basic Interrupt	29H	SK61

b. If a MAM 92990 option has been installed in memory, return direct memory address cable, terminated on both ends with P907, to stock. If a MAM has not been installed, install one end of direct address cable in location 13F of memory. Install other end of cable in central processor or memory according to system arrangement as shown in table 4-8.

Table 4-8. Installation of Direct Address Cable (P907) for Manual Write Lockout

FROM		TO	
Chassis	Location	Chassis	Location
First memory	13F	Central processor	35F
Second memory	13F	Central processor	36F
Third memory	13F	First memory	14F
Fourth memory	13F	Second memory	14F

4-22 Programmed Memory Write Lockout. To install the Programmed Memory Write Lockout Model 92061 feature, proceed as follows:

a. Install modules in associated memory door and basic interrupt chassis as shown in table 4-9.

Table 4-9. Module Installation for Programmed Write Lockout

Chassis	Location	Module
Memory Door	1F	IB56
Memory Door	2F	IB56
Memory Door	3F	FB50
Memory Door	4F	FB50
Memory Door	5F	ZB65*
Memory Door	6F	FB50
Memory Door	7F	FB50
Basic Interrupt	29H	SK61

*The dash number of the ZB65 module in location 5F depends on the memory in which the module is installed. If the programmed write lockout is installed in the first memory, use ZB65-50. If the write lockout is to be installed in the second, third, or fourth memory, determine the address of the first word in the memory and install the ZB65 shown in table 4-10.

Table 4-10. Jumper Module ZB65 Installation

First Address	ZB65
4096	ZB65-51
8192	ZB65-52
12288	ZB65-53
16384	ZB65-54
20480	ZB65-55
24576	ZB65-56
28672	ZB65-57

b. If a MAM 92990 option has been installed in memory, return direct memory address cable, terminated on both ends with P907, to stock. If a MAM has not been installed, install one end of direct address cable in location 13F of memory. Install other end of cable in central processor or memory according to system arrangement as shown in table 4-11.

Table 4-11. Installation of Direct Address Cable (P907) for Programmed Write Lockout

FROM		TO	
Chassis	Location	Chassis	Location
First memory	13F	Central processor	35F
Second memory	13F	Central processor	36F
Third memory	13F	First memory	40F
Fourth memory	13F	Second memory	14F

4-23 Memory Expansion

4-24 Kits are available for field expansion of 930 Computer memories. Each kit consists of a material list and installation instructions. Drawing numbers for these kits and for supplementary information are shown in table 4-12.

Table 4-12. Memory Expansion Kits

Power Supply	Drawing Number	Title
Harrison Laboratories	117008	Modification Procedure, 38v to 50v Heater Conversion
Harrison Laboratories	123593	Assembly, Memory Expansion Kit, 4K to 8K
Harrison Laboratories	123594	Assembly, Memory Expansion Kit, 8K to 16K
Harrison Laboratories	123595	Assembly, Memory Exapnsion Kit, 4K to 16K
SDS	123596	Assembly, Memory Expansion Kit, 4K to 8K
SDS	123597	Assembly, Memory Expansion Kit, 8K to 16K
SDS	123598	Assembly, Memory Expansion Kit, 4K to 16K

4-25 CHECKOUT

4-26 Before operating the computer, the following checks should be made:

a. Check timing of memory one-shots Du1, Du2, Du3, Du4, Du5, Du6, and Du8, using procedure given in paragraph 4-39. Adjust if necessary.

b. Check memory threshold voltage Vt, drive voltage Vd, and inhibit voltage Vz as directed in paragraphs 4-41 through 4-47. Adjust if necessary.

c. Execute Instruction Diagnostic, catalog No. 304002. If failure occurs, refer to SDS 930 Computer Examiner Diagnostic System Technical Manual, SDS 900097.

d. Execute Memory Diagnostic, catalog No. 304001. If failure occurs, refer to diagnostic manual specified in step c.

4-27 PREVENTIVE MAINTENANCE

4-28 Preventive maintenance on the memory consists of periodically executing the Instruction Diagnostic, catalog No. 304002 and the Memory Diagnostic, catalog No. 304001. In case of failure, refer to the SDS 930 Computer Examiner Diagnostic System Technical Manual, SDS 900097.

Note

The memory should be cleared after power turn-on and before loading the diagnostic. Failure to clear memory may result in parity errors from words disturbed by power turn-on.

4-29 CORRECTIVE MAINTENANCE

4-30 Maintenance instructions in this section are given for the memory only, since there are no moving parts or adjustments in the central processor. For power supply maintenance, refer to the appropriate power supply manuals.

4-31 SPECIAL TEST EQUIPMENT

4-32 Special test equipment required for maintenance of the 930 Computer is listed in table 4-13. These items or equivalent test equipment should be used.

Table 4-13. Special Test Equipment

Name	Manufacturer	Manufacturer's Number
Oscilloscope	Tektronix	585
Preamplifier	Tektronix	82
Meter	Simpson	267

4-33 ADJUSTMENTS

4-34 Because of the long heating time, stack temperature should be adjusted first. One-shot periods and voltage levels may then be checked and adjusted. Since the adjustment procedures for memories with SDS power supplies are different from the procedures for memories with Harrison Laboratories or PX10 power supplies, complete procedures are given for both cases.

4-35 Temperature Adjustment

4-36 It is desirable to operate the memory at the lowest possible temperature which can be safely controlled. The lowest safe temperature is approximately 5°C higher than the undertemperature thermostat, which may be set for 45°C or 55°C. Figure 4-6 illustrates the ideal stack operating temperature. Refer to paragraph 3-611 for the theory of operation of the heating system.

4-37 Memories With SDS Power Supplies. Temperature adjustment on memories with SDS power supplies should be made as follows:

a. If the memory contains more than one memory bank, disconnect P45 and P49 from the banks being adjusted.

CAUTION

Prior to disconnecting P49, be sure that all power to the heater board is off. Failure to shut off power to the heater results in blown output transistors.

b. On SX53 module located in J3 at rear of Power Supply PX20 for memory being adjusted, turn potentiometer fully clockwise. This ensures that stack will heat when heater power is applied to memory.

c. Place HEATER POWER switch on front of Power Supply PX20 in the ON position.

d. During first 2 minutes of warmup, measure voltage at TB1B-E-1 (negative probe on VOM) with respect to TB1B-E-2 (positive probe). Terminal board TB1B is located on side of memory door and is second terminal board from top, on level with B section of memory chassis. A voltage of +5.6 (±0.4) volts indicates that temperature-sensing diodes are operating properly. This voltage will drop to +5.0 (±0.4) volts when stack temperature reaches 55°C to 60°C.

e. With PX20/21 meter select switch in A/10 position, observe meter. Meter should indicate current flow.

f. Measure voltage at pin 10F41 in memory door, and observe this voltage continuously until it rises to +4 volts. Voltage rise indicates that stack has reached correct temperature and that undertemperature thermostat switch has opened. Within 30 seconds of time voltage rise, turn potentiometer on SX53 module counterclockwise until heater current stops flowing, then turn potentiometer two full turns counterclockwise.

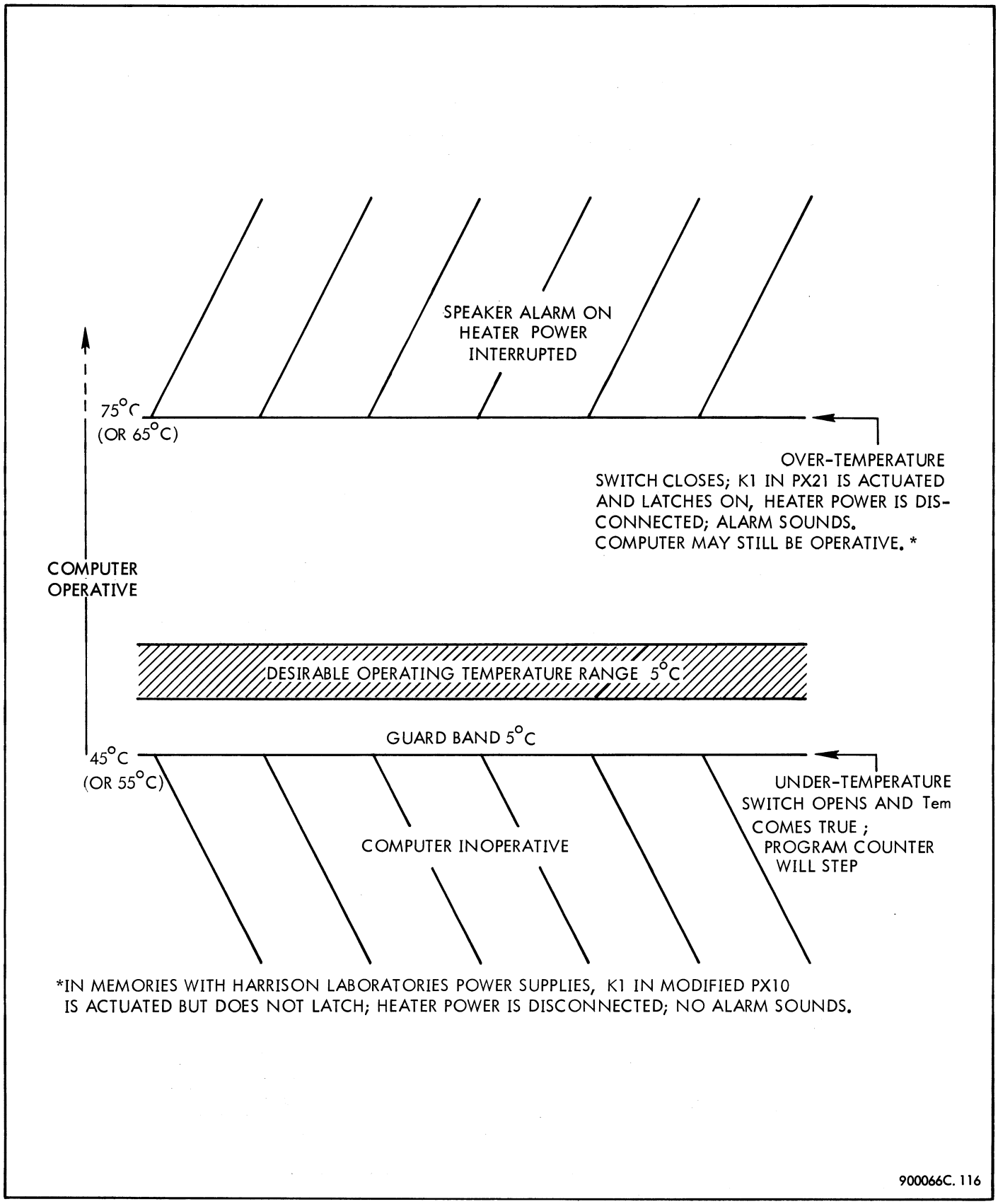


Figure 4-6. Ideal Stack Operating Temperature

Note

The 30-second time limit after opening of the thermostat switch must be strictly observed.

4-38 Memories With Harrison Laboratories Power Supplies. The following adjustment procedure applies to memories with Harrison Laboratories power supplies:

a. If the memory contains more than one memory bank, disconnect P45 and P49 from the banks being adjusted.

b. On TB1 heater regulator board on side of stack, turn potentiometer fully clockwise. This ensures that stack will heat when heater power is applied to memory.

c. Turn heater power on.

d. Place HEATER POWER switch on front of Power Supply PX20 in the ON position.

e. With PX10 meter select switch in straight up position (usually marked A), observe meter. Meter should indicate current flow.

f. Measure voltage at pin 10F41 in memory door, and observe this voltage continuously until it rises to +4 volts. Voltage rise indicates that stack has reached correct temperature and that undertemperature thermostat switch has opened. Within 30 seconds of time voltage rise, turn potentiometer on SX53 module counterclockwise until heater current stops flowing, then turn potentiometer two full turns counterclockwise.

g. Turn TBL heater-regulator potentiometer counterclockwise one-fourth turn, lowering temperature approximately 1-1/4°C, and wait about 30 minutes, occasionally checking voltage at 10F41 to see whether the undertemperature switch has closed. Closing of switch is indicated by a drop in voltage at 10F41 to 0 volts. Program counter will not step when undertemperature switch is closed.

h. If switch does not close, turn potentiometer another one-fourth turn counterclockwise. Repeat approximately every 30 minutes until switch closes, then turn potentiometer clockwise exactly one turn.

4-39 Adjustment of One-Shots

4-40 The periods of the OB50 one-shots for memories with a 1.925-microsecond cycle and a 1.75-microsecond memory cycle should be adjusted with an oscilloscope as shown in table 4-14. The potentiometers to be adjusted are identified by output pin in figure 4-7. Set the oscilloscope sync at internal and measure the delay at 50 percent amplitude of the output pulse at the pin listed in the order given in the table.

Table 4-14. OB50 One-Shot Adjustments

DELAY	LOCATION	OUTPUT PIN	PERIOD (nsec)	
			1.925-μsec cycle	1.75-μsec cycle
Du1	20E	33	265	80
Du2	20E	26	795	620
Du4	21E	33	130	100
Du3	21E	26	630	600
Du5	20E	19	280 to 340*	
Du6	20E	12	40	40
Du8	21E	12	40	40

*Du5 controls the strobe pulse. The preliminary settings are:

With Sense Amplifiers HB53, 280 nsec for 4K stack, 310 nsec for 8K or 16K stack

With Sense Amplifiers HB53-2, 300 nsec for 4K stack, 340 nsec for 8K or 16K stack

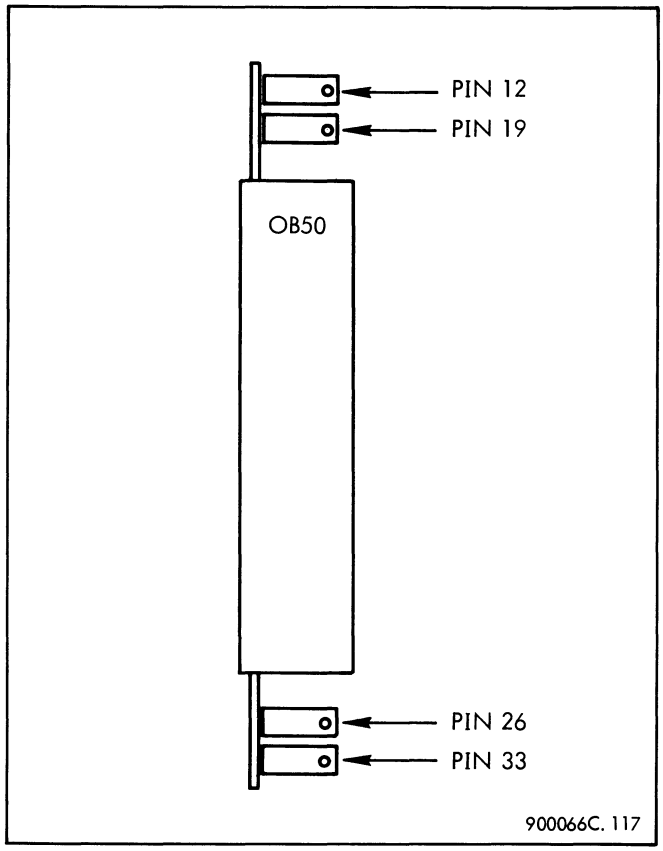


Figure 4-7. Identification of OB50 Potentiometers by Output

4-41 Threshold Voltage Adjustment

4-42 The threshold voltage adjustment procedure is the same for memories with SDS power supplies and memories with Harrison Laboratories power supplies. To make the adjustment, proceed as follows:

a. Using voltmeter, measure threshold voltage Vt at pin 5D20 on memory door with reference to ground.

b. With THRESHOLD screw on front of appropriate power supply, adjust threshold setting to obtain threshold voltage at pin 5D20 as follows:

Sense Amplifiers HB53 – +1.95 (±0.15) volts (about setting number 5)

Sense Amplifiers HB53-2 – +1.70 (±0.15) volts (about setting number 4)

Note

To obtain the necessary sensitivity in memories with Harrison Laboratories power supplies, it may be necessary to remove the wire link between wafers A and B at position 9 of the threshold switch and replace with a 100-ohm resistor.

c. Record final Vt voltage on Schmoo Chart SDS-Ma-488-1 in Adjustment Settings block.

4-43 Drive Line and Inhibit Voltage Adjustments

4-44 The preliminary settings of XY drive line voltage Vd and inhibit voltage Vz are obtained as follows:

a. With voltmeter, measure Vd voltage by placing positive probe on pin 10 in location 6A and negative probe on pin 45 of any module. Module power bus is connected to pin 45 of each module and is +4.0-volt reference point.

b. Using a screwdriver, turn XY DRIVE switches on Power Supply PX20 to 99 if this is possible without allowing Vd voltage to go above +34.1 volts.

Note

Setting switches at 99 may partially destroy memory contents.

Considering the numbers in the drive switch windows as a decimal figure, the number on the right represents the units and the number on the left represents the tens. When the right-hand switch has reached nine, it should be turned back to zero before changing the setting of the left-hand switch. If Vd cannot be set to +34.0 (±0.1) volts with drive adjust switches, make fine adjustment with Vd potentiometer in power supply. In Power Supply PX20-PX21, this potentiometer is R11-2, front potentiometer on top of SX52 module in J2 at rear of power supply, and may be reached with offset screwdriver.

c. For safety, temporarily set XY DRIVE switches at some value well below 99.

d. With voltmeter, measure Vz voltage by placing positive probe on pin 10 of location 5A and negative probe on pin 45 of any module.

CAUTION

Both XY DRIVE and Z DRIVE switches should never be at 99 at the same time. This could damage memory.

e. Using screwdriver, turn Z DRIVE control switches on Power Supply PX20 to 99 if this is possible without allowing Vz voltage to go above +34.1 volts. Switches operate same as XY DRIVE switches in step b. If Vz cannot be set to +34.0 (±0.1) volts with Z DRIVE switches, make fine adjustment with Vz potentiometer in power supply. In PX20-PX21, this potentiometer is R11-1, rear potentiometer on top of SX52 module in J2 at rear of power supply, and may be reached with offset screwdriver.

f. On memory with stack operating temperature of 50°C to 55°C, set Z DRIVE switches at 65. On memory with stack operating temperature of 60°C to 65°C, set Z DRIVE switches at 55. Record final Vz switch setting in Adjustment Settings block on schmoo chart.

g. On memory with Harrison Laboratories power supply, set Z DRIVE switches to 30. It may be necessary to adjust Vd by about ±10 settings of right-hand switch to enable memory diagnostic to run. On memory with SDS power supplies, store LDA 200 instruction in location 100. Insert BRU 100 in control console C-register. Place appropriate control console HOLD switch in up position. Place RUN IDLE STEP switch in RUN position. Increase and decrease Vd by varying XY DRIVE control switches, and note high and low switch settings at which parity errors indicate that program fails. Place switches at setting midway between high and low values. Record switch setting after Vd in Adjustment Settings block on schmoo chart.

h. If system contains second memory, repeat above procedure for second memory door. It may be necessary to adjust XY DRIVE switches by 10 or 20 settings of the right-hand switch to enable memory diagnostic to run.

4-45 Final Memory Setup

4-46 After preliminary adjustments have been made the final memory setup is conducted by running the memory diagnostic and plotting a schmoo chart of memory failure points for various settings of Vd and Du5 with Vz, Vt, and

stack temperature at preselected values. The purpose of the schmoo chart is to select a memory working point which will allow sufficient margins to enable error-free operation. A typical schmoo curve is shown in figure 4-8.

4-47 Proceed with final memory setup as follows:

a. Successfully execute Memory Diagnostic, catalog No. 304001. If failure occurs, refer to 930 Examiner Diagnostic Technical Manual, SDS 900097.

b. With oscilloscope on internal sync, place signal probe on pin 19 of location 20E, which is Du5 output. With XY DRIVE switches at setting obtained in paragraph 4-44 step e, decrease delay of Du5 by turning potentiometer for output pin 19 (figure 4-7) counterclockwise until failure is indicated by occurrence of parity errors. Record pulse width of Du5 for this early failure point on schmoo chart (see figure 4-8).

c. Turn potentiometer in step b clockwise until failure is indicated by parity errors. Record width of Du5 for this late failure on schmoo chart.

d. Plot early and late failure points for other settings of Vd by varying Du5 with XY DRIVE switches at 10 and 20 points above the setting in step b and 10 and 20 points below setting in step b until strobe range is reduced to 10 or 20 nanoseconds.

e. Within the schmoo curve, set Vd to the lowest setting divisible by five and adjust Du5 to 10 nanoseconds later than the early failure point.

f. Clear memory and attempt to reload and run the memory diagnostic. Increase Du5 pulse width by 10-nanosecond steps until the diagnostic runs successfully. Record this point on the schmoo chart.

g. Increase Du5 pulse width by increments of 10, repeating the above loading test at each step. A record of points at which the diagnostic runs successfully produces a failure line which runs inside the first schmoo curve as shown in figure 4-8.

h. Determine the range of Vd from the latest setting of Du5 on the schmoo curve to the setting of Du5 at the center point on the inside failure curve. This range in figure 4-8 is 30 to 60. Find the midpoint in this Vd range and select the final Vd value as the nearest setting above this midpoint which is divisible by five (45 on the schmoo chart).

i. Draw a straight line from the early Du5 failure point on the inside curve at the final setting of Vd plus 10 to the late failure point of Du5 on the schmoo curve at the final setting of Vd minus 10.

j. Draw a straight line from the early Du5 failure point on the inside curve at the final setting of Vd minus ten to the late failure point of Du5 on the schmoo curve at the final setting of Vd plus 10.

k. Select the final setting of Du5 five nanoseconds later than the setting of Du5 at the intersection of the two lines.

l. Mark on the schmoo chart the final operating point, which is at the intersection of the final Du5 value and the final Vd setting. The composite of the original schmoo curve and the inside failure curve should include the following four points:

1. Vd, Vt, and Vz at final settings and Du5 at final setting plus 30 nanoseconds with HB53-2 sense amplifiers (plus 20 nanoseconds with HB53 sense amplifiers).

2. Vd, Vt, and Vz at final settings and Du5 at final setting minus 40 nanoseconds with HB53-2 sense amplifiers (minus 30 nanoseconds with HB53 sense amplifiers).

3. Du5, Vt, and Vz at final settings and Vd at final setting plus 18.

4. Du5, Vt, and Vz at final settings and Vd at final setting minus 18.

m. Run the memory diagnostic at the following additional points:

1. Du5, Vd, and Vt at final settings and Vz at final setting plus 25.

2. Du5, Vd, and Vt at final settings and Vz at final setting minus 25.

The diagnostic should run successfully.

Note

It is possible to obtain a larger schmoo by moving Vz up or down ten divisions or Vt up or down one division from the settings found in the preliminary memory setup, paragraph 4-44. The use of these deviations should be justified by plotting both schmoos on the same chart. If the settings obtained in the above procedures fall outside the schmoo curve, a possible cause may be incorrect temperature adjustment or a faulty memory stack.

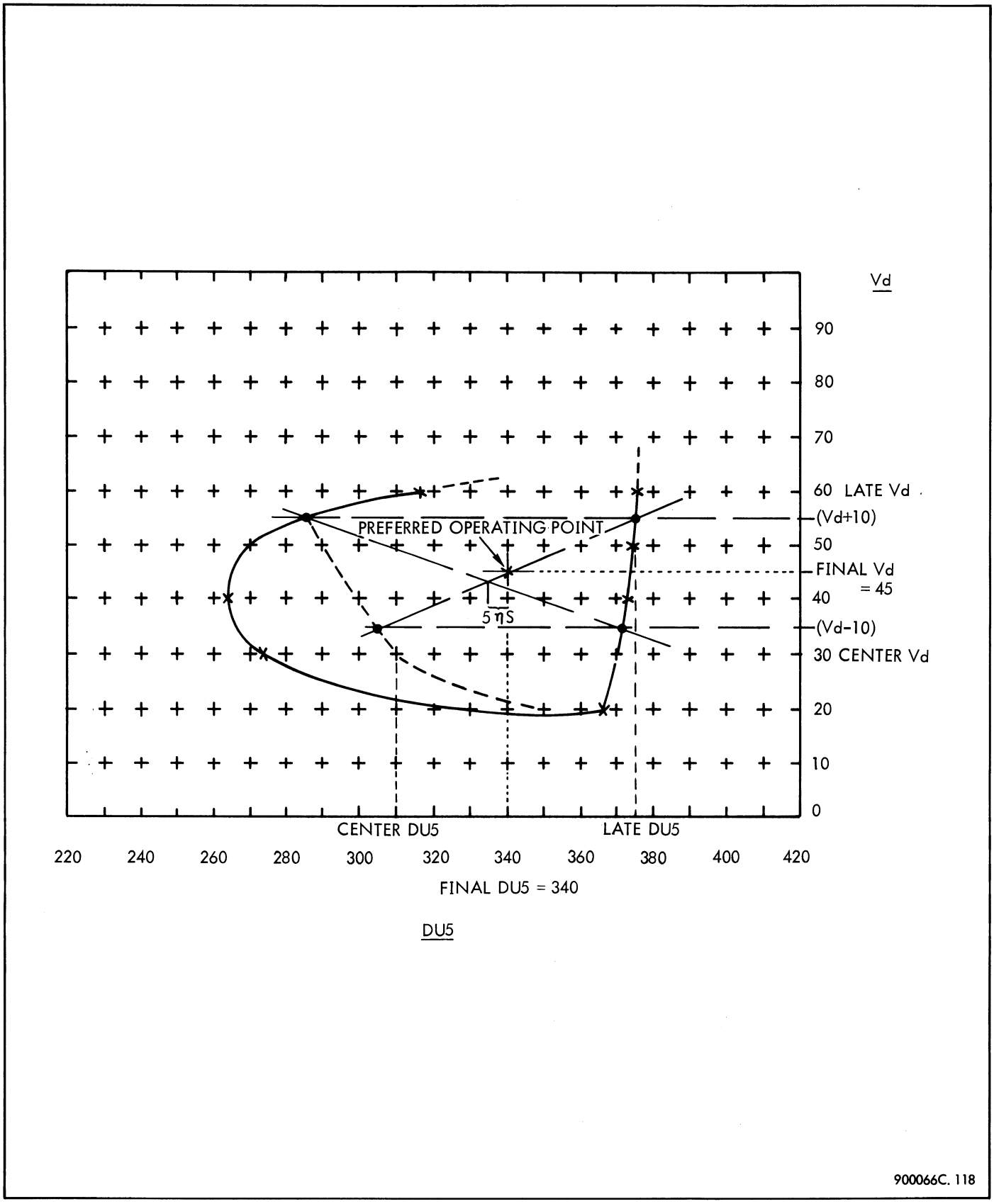


Figure 4-8. Typical Schmoo Curve

SECTION V
PERFORMANCE TESTING AND TROUBLE ANALYSIS

5-1 TEST PROGRAMS

5-2 The following programs are available for performance testing and trouble analysis on the 930 Computer:

Program	Catalog Number
Memory Diagnostic	304001
Instruction Diagnostic	304002
P & S Register Test	304003
SDS 930 Examiner Diagnostic System (Includes preceding three programs)	304004

Refer to the 930 Computer Examiner Diagnostic Manual, SDS 900097, for an explanation of these programs.

5-3 The 930 Computer logic diagrams, general reference drawings, and equations, as well as the theory of operation section in this manual, will be helpful in performing trouble analysis.

5-4 MEMORY TROUBLESHOOTING

5-5 In all cases of memory problems, the cause can be localized logically before leaving the control console, and the location and particular pattern failing can be determined. Compiling and analyzing the data obtained usually produces a certain pattern of failing locations or specific bit failures. Refer to the flow diagram in figure 5-1.

5-6 In the case of addressing problems in a 16K bank, more than one 4K group of locations is usually affected. Specific bit failures can normally be localized to a particular 4K group, since there are separate inhibit drivers and sense lines for each 4K bank. See figures 5-2 through 5-10.

5-7 If an entire bank is failing randomly, the failures can normally be attributed to temperature, incorrect timing adjustment, the M-register, incorrect adjustment of XY currents or threshold voltage, or a missing voltage. Refer to figure 5-11.

Note

Turn off power before removing modules, connectors, etc., from the memory door.

5-8 For further troubleshooting information, the following manuals may be consulted:

925/930/9300 Internal Memory Stack Troubleshooting Guide, SDS 900689

930/9300 Memories Troubleshooting Manual, SDS 900865

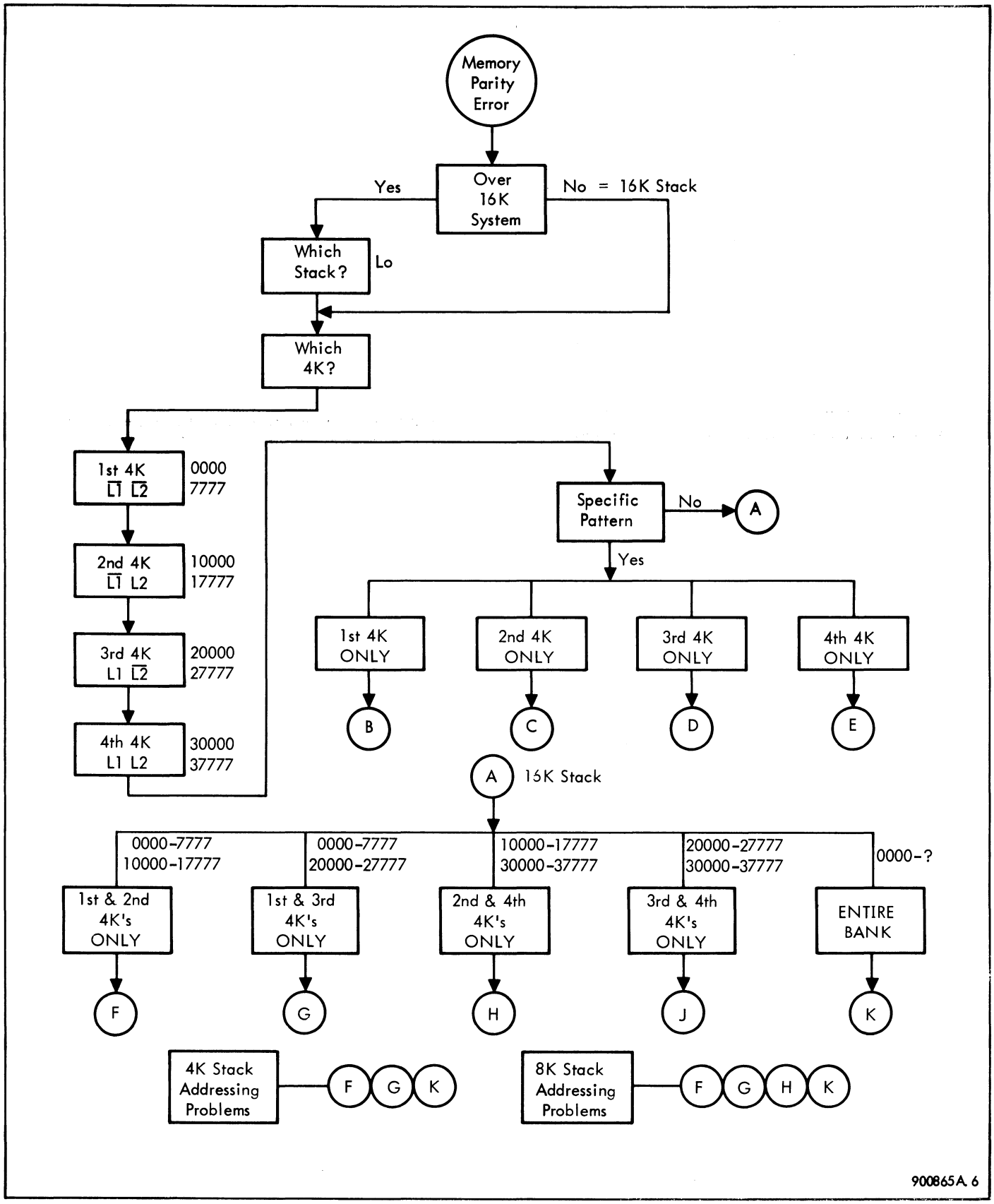
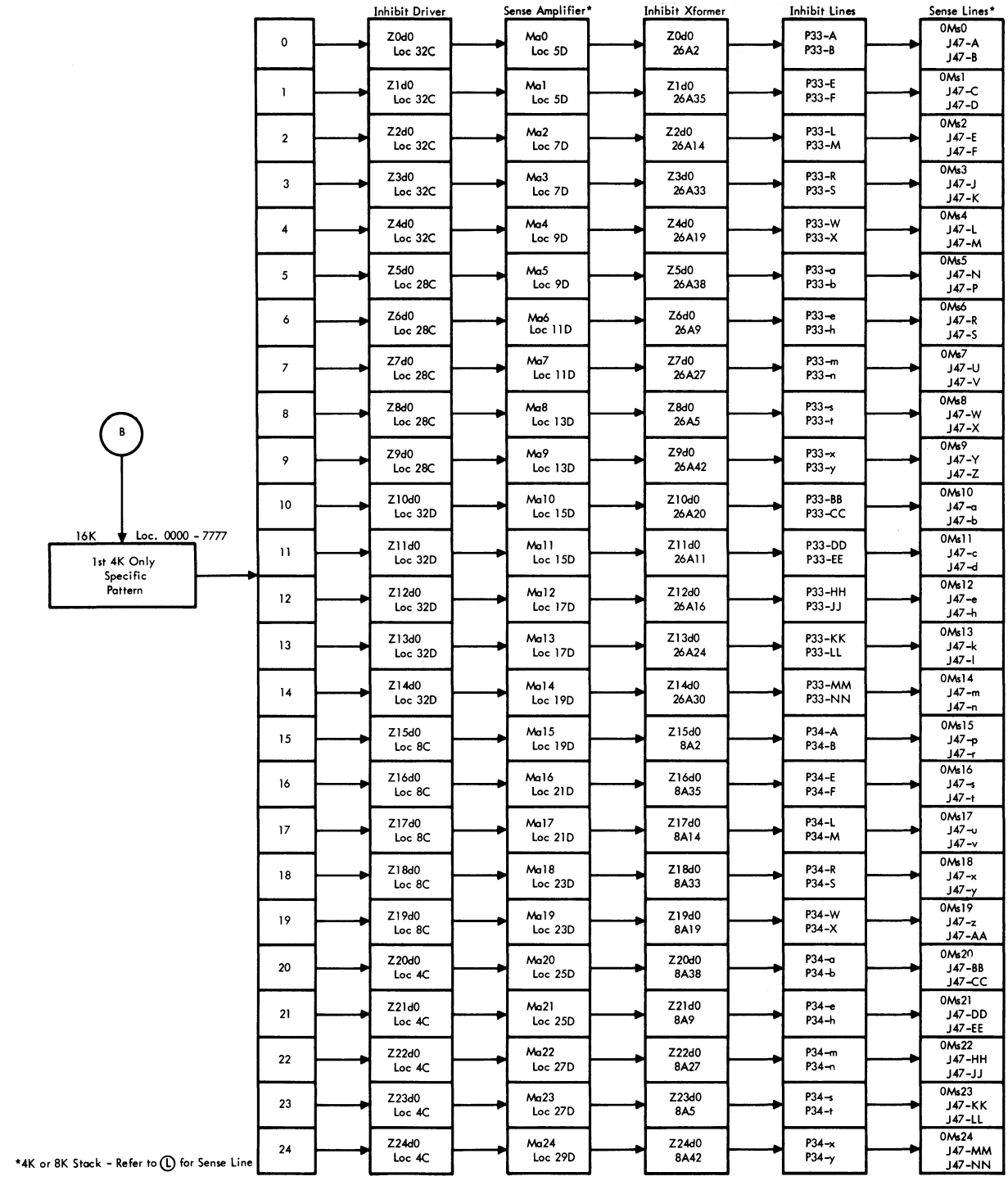
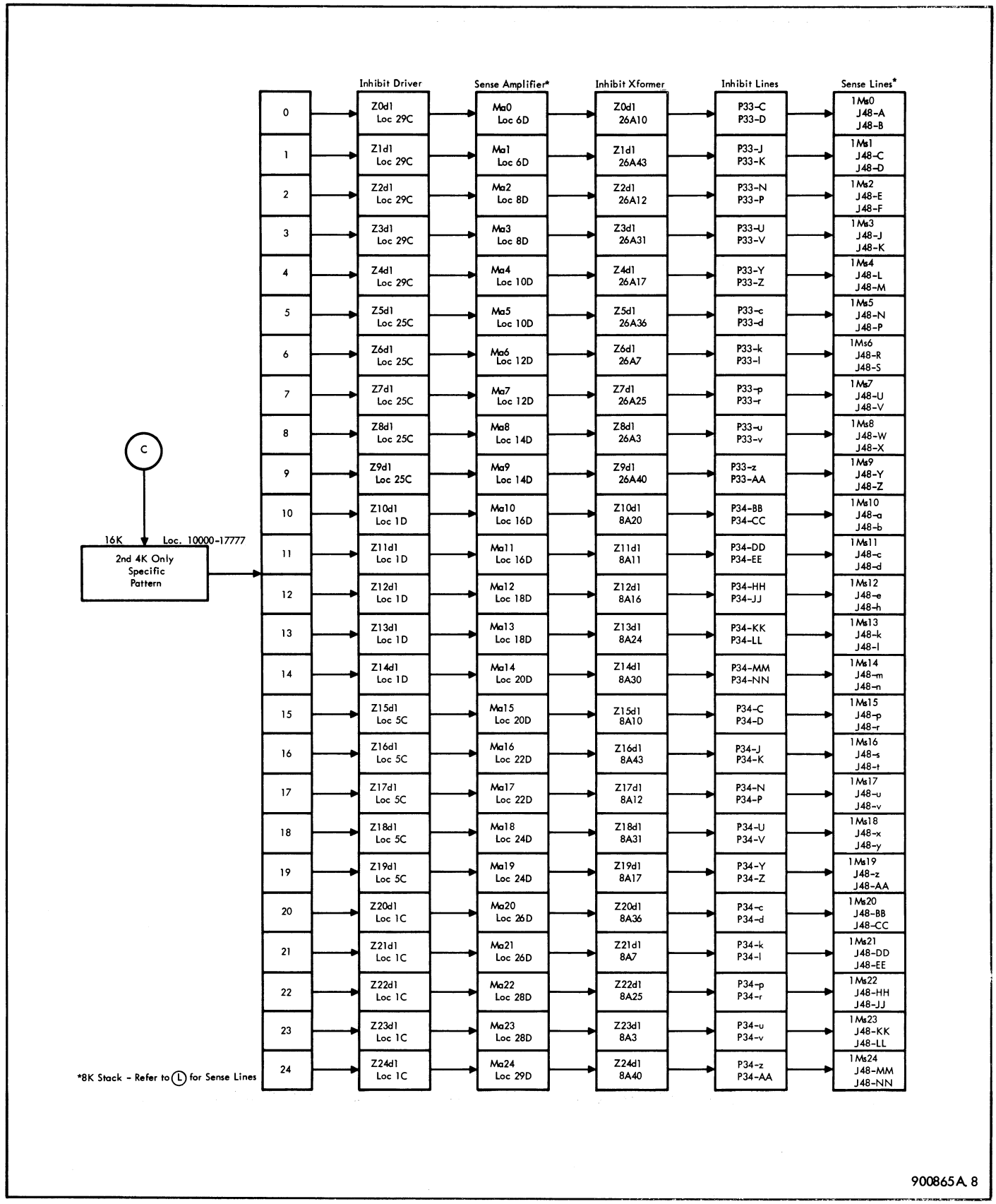


Figure 5-1. Memory Troubleshooting, Flow Diagram



	Inhibit Driver	Sense Amplifier*	Inhibit Xformer	Inhibit Lines	Sense Lines*
0	Z0d0 Loc 32C	Ma0 Loc 5D	Z0d0 26A2	P33-A P33-B	0Ms0 J47-A J47-B
1	Z1d0 Loc 32C	Ma1 Loc 5D	Z1d0 26A35	P33-E P33-F	0Ms1 J47-C J47-D
2	Z2d0 Loc 32C	Ma2 Loc 7D	Z2d0 26A14	P33-L P33-M	0Ms2 J47-E J47-F
3	Z3d0 Loc 32C	Ma3 Loc 7D	Z3d0 26A33	P33-R P33-S	0Ms3 J47-J J47-K
4	Z4d0 Loc 32C	Ma4 Loc 9D	Z4d0 26A19	P33-W P33-X	0Ms4 J47-L J47-M
5	Z5d0 Loc 28C	Ma5 Loc 9D	Z5d0 26A38	P33-a P33-b	0Ms5 J47-N J47-P
6	Z6d0 Loc 28C	Ma6 Loc 11D	Z6d0 26A9	P33-e P33-h	0Ms6 J47-R J47-S
7	Z7d0 Loc 28C	Ma7 Loc 11D	Z7d0 26A27	P33-m P33-n	0Ms7 J47-U J47-V
8	Z8d0 Loc 28C	Ma8 Loc 13D	Z8d0 26A5	P33-s P33-t	0Ms8 J47-W J47-X
9	Z9d0 Loc 28C	Ma9 Loc 13D	Z9d0 26A42	P33-x P33-y	0Ms9 J47-Y J47-Z
10	Z10d0 Loc 32D	Ma10 Loc 15D	Z10d0 26A20	P33-BB P33-CC	0Ms10 J47-a J47-b
11	Z11d0 Loc 32D	Ma11 Loc 15D	Z11d0 26A11	P33-DD P33-EE	0Ms11 J47-c J47-d
12	Z12d0 Loc 32D	Ma12 Loc 17D	Z12d0 26A16	P33-HH P33-JJ	0Ms12 J47-e J47-h
13	Z13d0 Loc 32D	Ma13 Loc 17D	Z13d0 26A24	P33-KK P33-LL	0Ms13 J47-k J47-l
14	Z14d0 Loc 32D	Ma14 Loc 19D	Z14d0 26A30	P33-MM P33-NN	0Ms14 J47-m J47-n
15	Z15d0 Loc 8C	Ma15 Loc 19D	Z15d0 8A2	P34-A P34-B	0Ms15 J47-p J47-r
16	Z16d0 Loc 8C	Ma16 Loc 21D	Z16d0 8A35	P34-E P34-F	0Ms16 J47-s J47-t
17	Z17d0 Loc 8C	Ma17 Loc 21D	Z17d0 8A14	P34-L P34-M	0Ms17 J47-u J47-v
18	Z18d0 Loc 8C	Ma18 Loc 23D	Z18d0 8A33	P34-R P34-S	0Ms18 J47-x J47-y
19	Z19d0 Loc 8C	Ma19 Loc 23D	Z19d0 8A19	P34-W P34-X	0Ms19 J47-z J47-AA
20	Z20d0 Loc 4C	Ma20 Loc 25D	Z20d0 8A38	P34-a P34-b	0Ms20 J47-BB J47-CC
21	Z21d0 Loc 4C	Ma21 Loc 25D	Z21d0 8A9	P34-e P34-h	0Ms21 J47-DD J47-EE
22	Z22d0 Loc 4C	Ma22 Loc 27D	Z22d0 8A27	P34-m P34-n	0Ms22 J47-HH J47-JJ
23	Z23d0 Loc 4C	Ma23 Loc 27D	Z23d0 8A5	P34-s P34-t	0Ms23 J47-KK J47-LL
24	Z24d0 Loc 4C	Ma24 Loc 29D	Z24d0 8A42	P34-x P34-y	0Ms24 J47-MM J47-NN

Figure 5-2. Memory Troubleshooting, First 4K Stack



	Inhibit Driver	Sense Amplifier*	Inhibit Xformer	Inhibit Lines	Sense Lines*
0	Z0d1 Loc 29C	Ma0 Loc 6D	Z0d1 26A10	P33-C P33-D	1Ms0 J48-A J48-B
1	Z1d1 Loc 29C	Ma1 Loc 6D	Z1d1 26A43	P33-J P33-K	1Ms1 J48-C J48-D
2	Z2d1 Loc 29C	Ma2 Loc 8D	Z2d1 26A12	P33-N P33-P	1Ms2 J48-E J48-F
3	Z3d1 Loc 29C	Ma3 Loc 8D	Z3d1 26A31	P33-U P33-V	1Ms3 J48-J J48-K
4	Z4d1 Loc 29C	Ma4 Loc 10D	Z4d1 26A17	P33-Y P33-Z	1Ms4 J48-L J48-M
5	Z5d1 Loc 25C	Ma5 Loc 10D	Z5d1 26A36	P33-c P33-d	1Ms5 J48-N J48-P
6	Z6d1 Loc 25C	Ma6 Loc 12D	Z6d1 26A7	P33-k P33-l	1Ms6 J48-R J48-S
7	Z7d1 Loc 25C	Ma7 Loc 12D	Z7d1 26A25	P33-p P33-r	1Ms7 J48-U J48-V
8	Z8d1 Loc 25C	Ma8 Loc 14D	Z8d1 26A3	P33-u P33-v	1Ms8 J48-W J48-X
9	Z9d1 Loc 25C	Ma9 Loc 14D	Z9d1 26A40	P33-z P33-AA	1Ms9 J48-Y J48-Z
10	Z10d1 Loc 1D	Ma10 Loc 16D	Z10d1 8A20	P34-BB P34-CC	1Ms10 J48-a J48-b
11	Z11d1 Loc 1D	Ma11 Loc 16D	Z11d1 8A11	P34-DD P34-EE	1Ms11 J48-c J48-d
12	Z12d1 Loc 1D	Ma12 Loc 18D	Z12d1 8A16	P34-HH P34-JJ	1Ms12 J48-e J48-h
13	Z13d1 Loc 1D	Ma13 Loc 18D	Z13d1 8A24	P34-KK P34-LL	1Ms13 J48-k J48-l
14	Z14d1 Loc 1D	Ma14 Loc 20D	Z14d1 8A30	P34-MM P34-NN	1Ms14 J48-m J48-n
15	Z15d1 Loc 5C	Ma15 Loc 20D	Z15d1 8A10	P34-C P34-D	1Ms15 J48-p J48-r
16	Z16d1 Loc 5C	Ma16 Loc 22D	Z16d1 8A43	P34-J P34-K	1Ms16 J48-s J48-t
17	Z17d1 Loc 5C	Ma17 Loc 22D	Z17d1 8A12	P34-N P34-P	1Ms17 J48-u J48-v
18	Z18d1 Loc 5C	Ma18 Loc 24D	Z18d1 8A31	P34-U P34-V	1Ms18 J48-x J48-y
19	Z19d1 Loc 5C	Ma19 Loc 24D	Z19d1 8A17	P34-Y P34-Z	1Ms19 J48-z J48-AA
20	Z20d1 Loc 1C	Ma20 Loc 26D	Z20d1 8A36	P34-c P34-d	1Ms20 J48-BB J48-CC
21	Z21d1 Loc 1C	Ma21 Loc 26D	Z21d1 8A7	P34-k P34-l	1Ms21 J48-DD J48-EE
22	Z22d1 Loc 1C	Ma22 Loc 28D	Z22d1 8A25	P34-p P34-r	1Ms22 J48-HH J48-JJ
23	Z23d1 Loc 1C	Ma23 Loc 28D	Z23d1 8A3	P34-u P34-v	1Ms23 J48-KK J48-LL
24	Z24d1 Loc 1C	Ma24 Loc 29D	Z24d1 8A40	P34-z P34-AA	1Ms24 J48-MM J48-NN

Figure 5-3. Memory Troubleshooting, Second 4K Stack

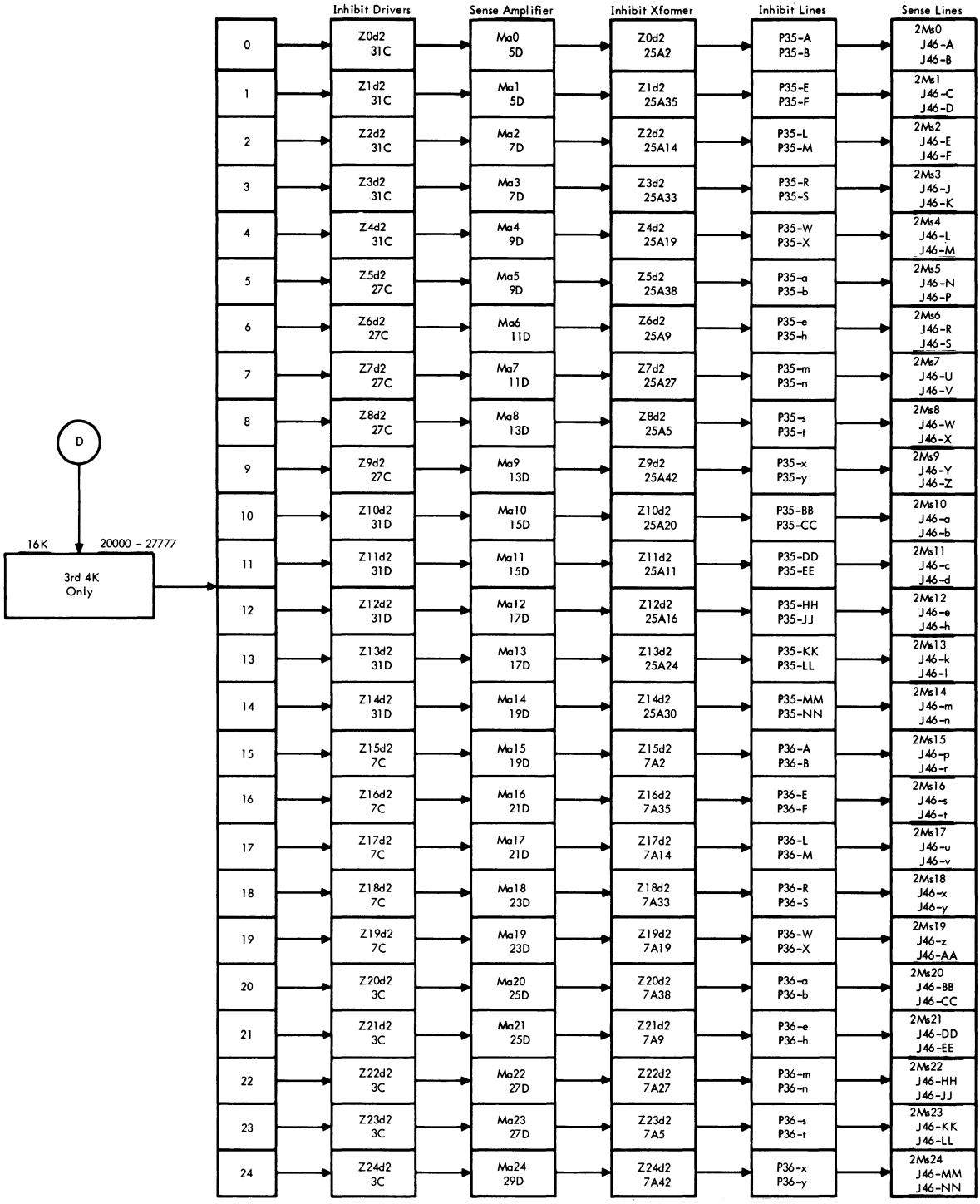


Figure 5-4. Memory Troubleshooting, Third 4K Stack

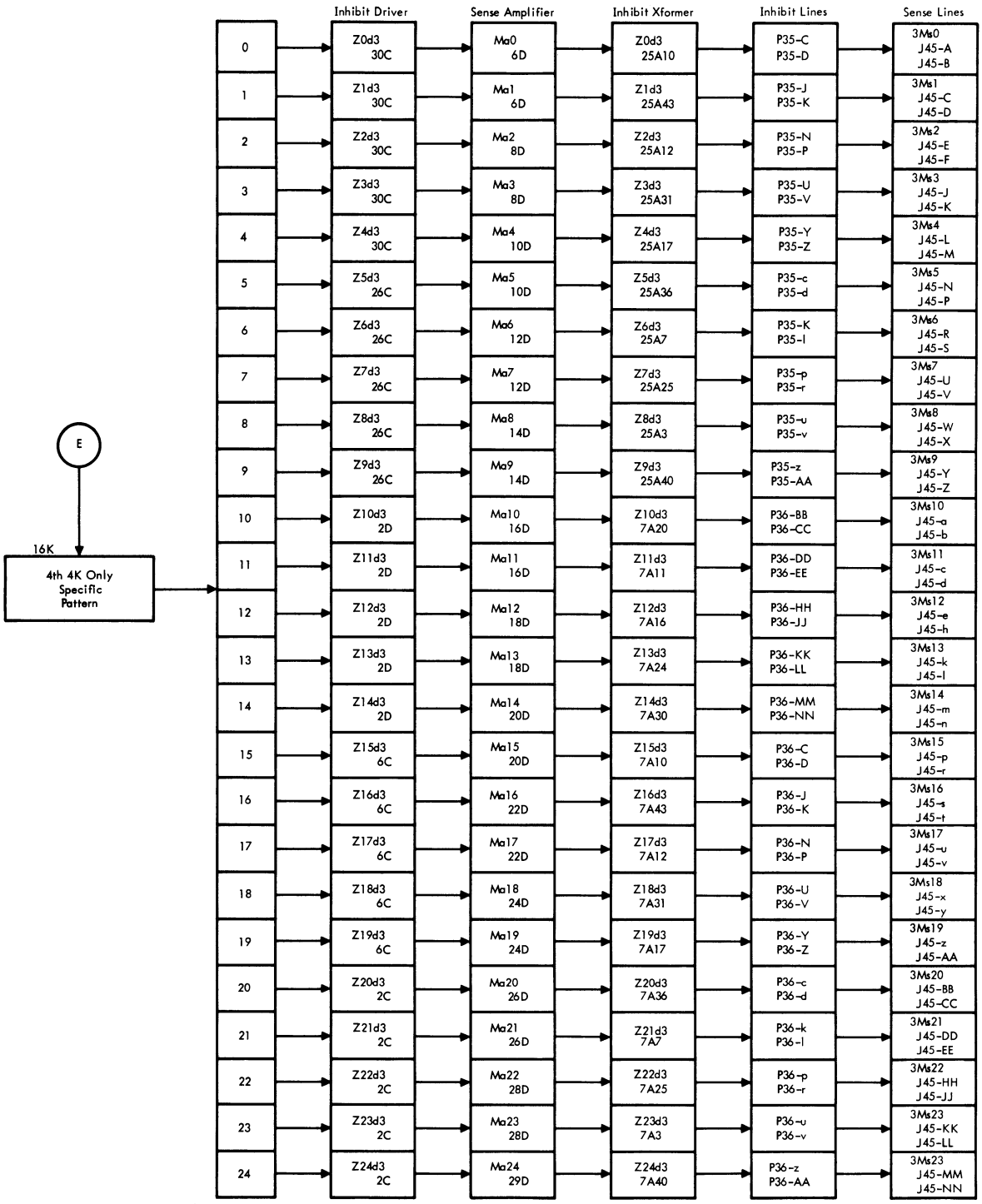


Figure 5-5. Memory Troubleshooting, Fourth 4K Stack

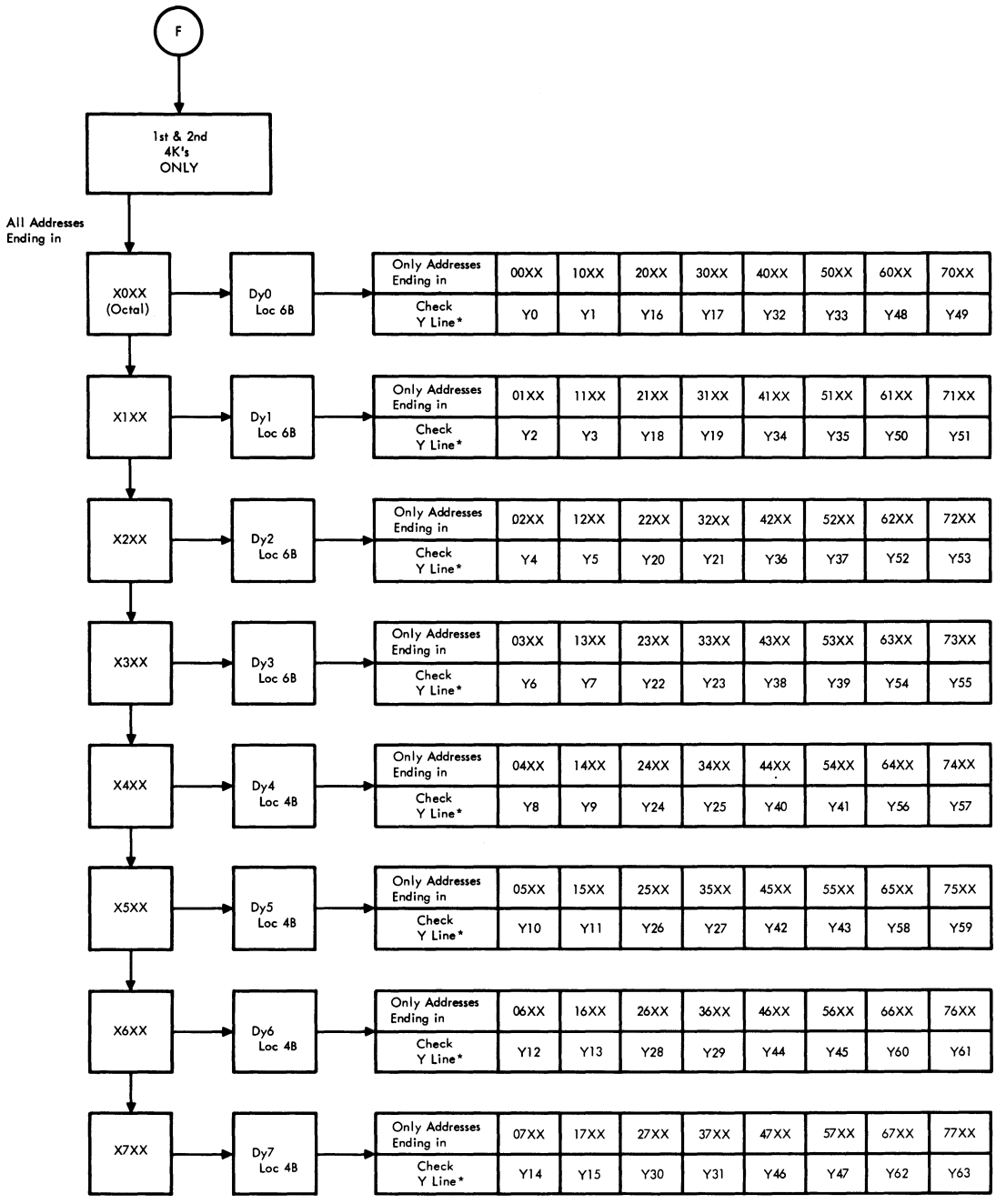


Figure 5-6. Memory Troubleshooting, First and Second 4K Stacks

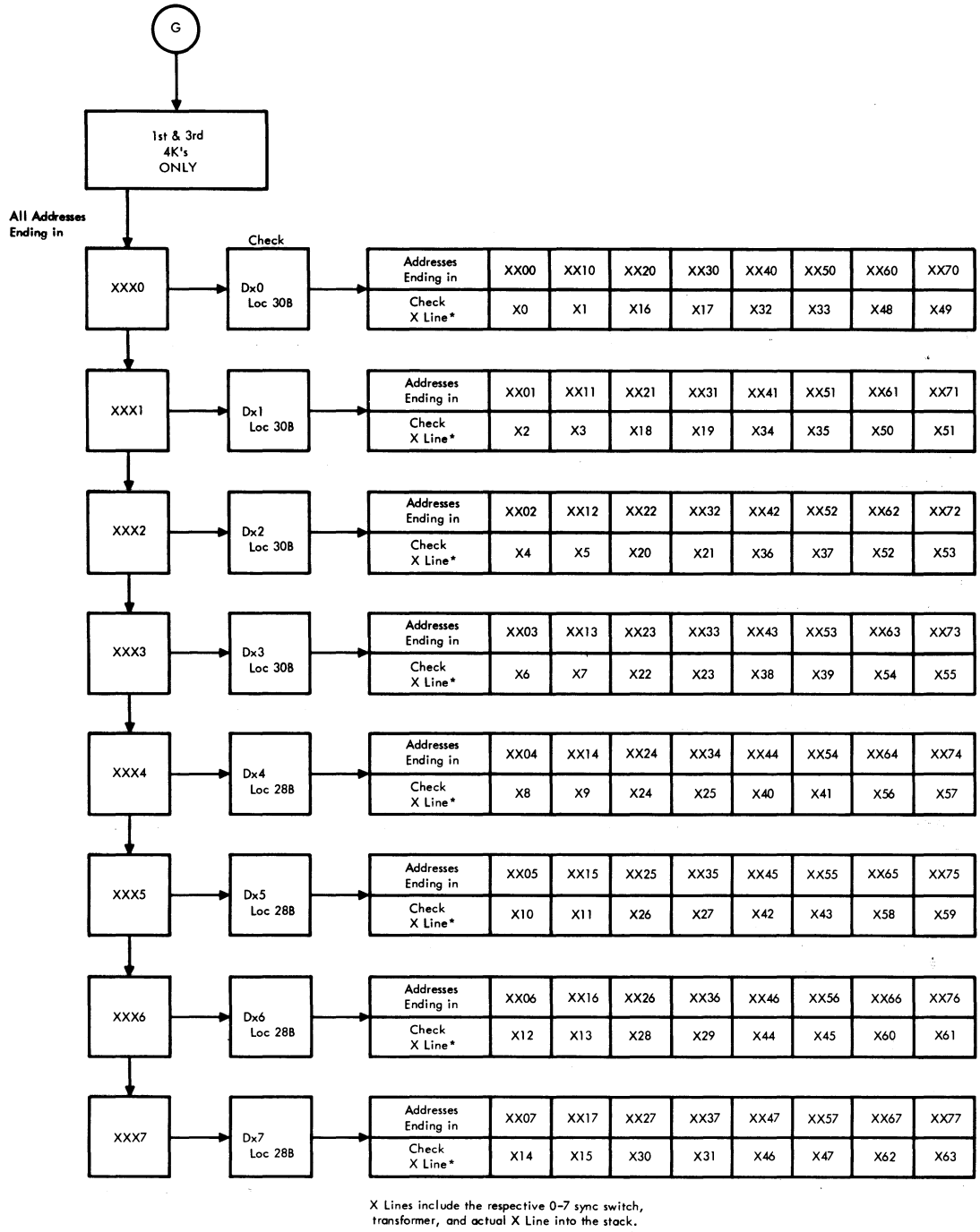


Figure 5-7. Memory Troubleshooting, First and Third 4K Stacks

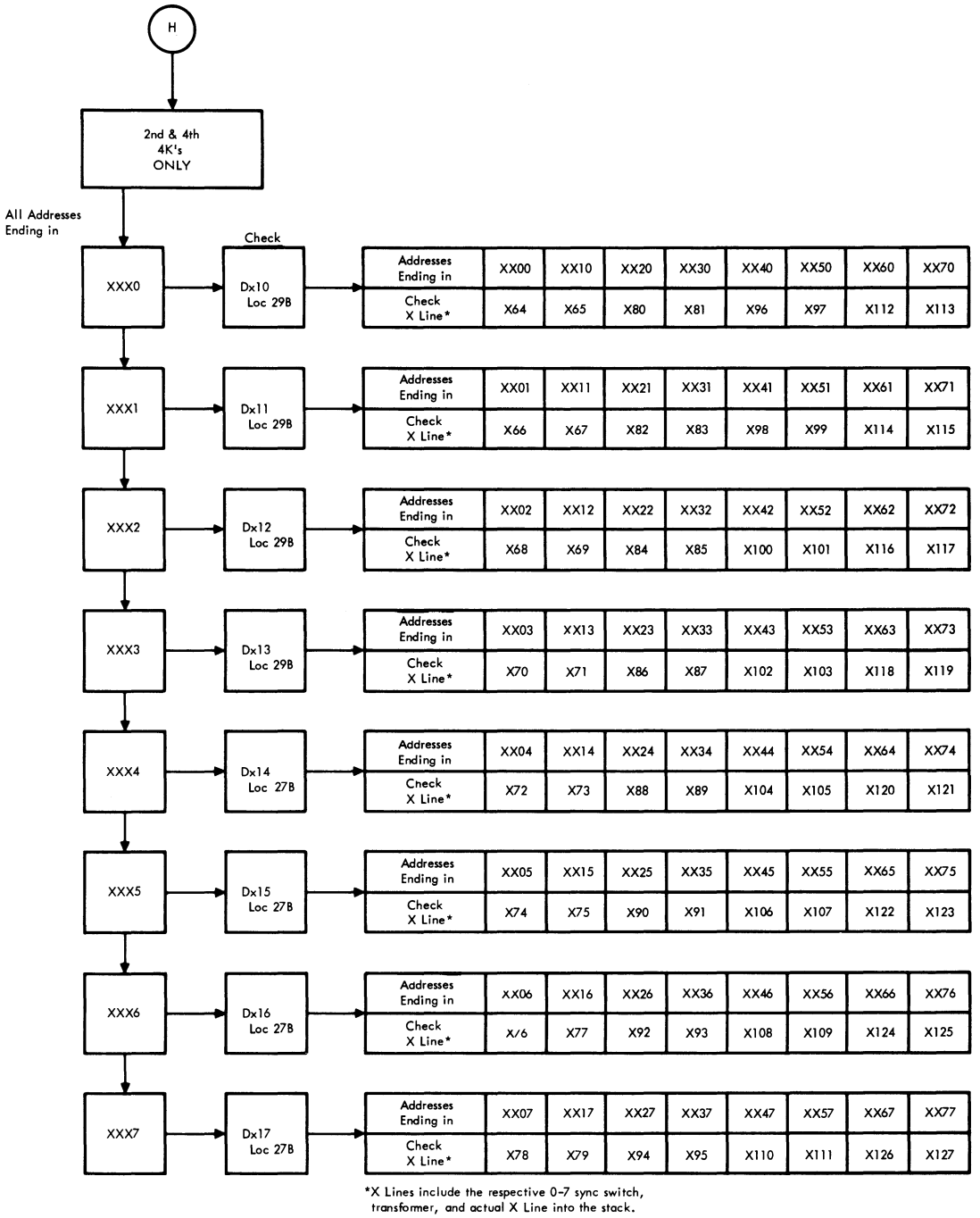


Figure 5-8. Memory Troubleshooting, Second and Fourth 4K Stacks

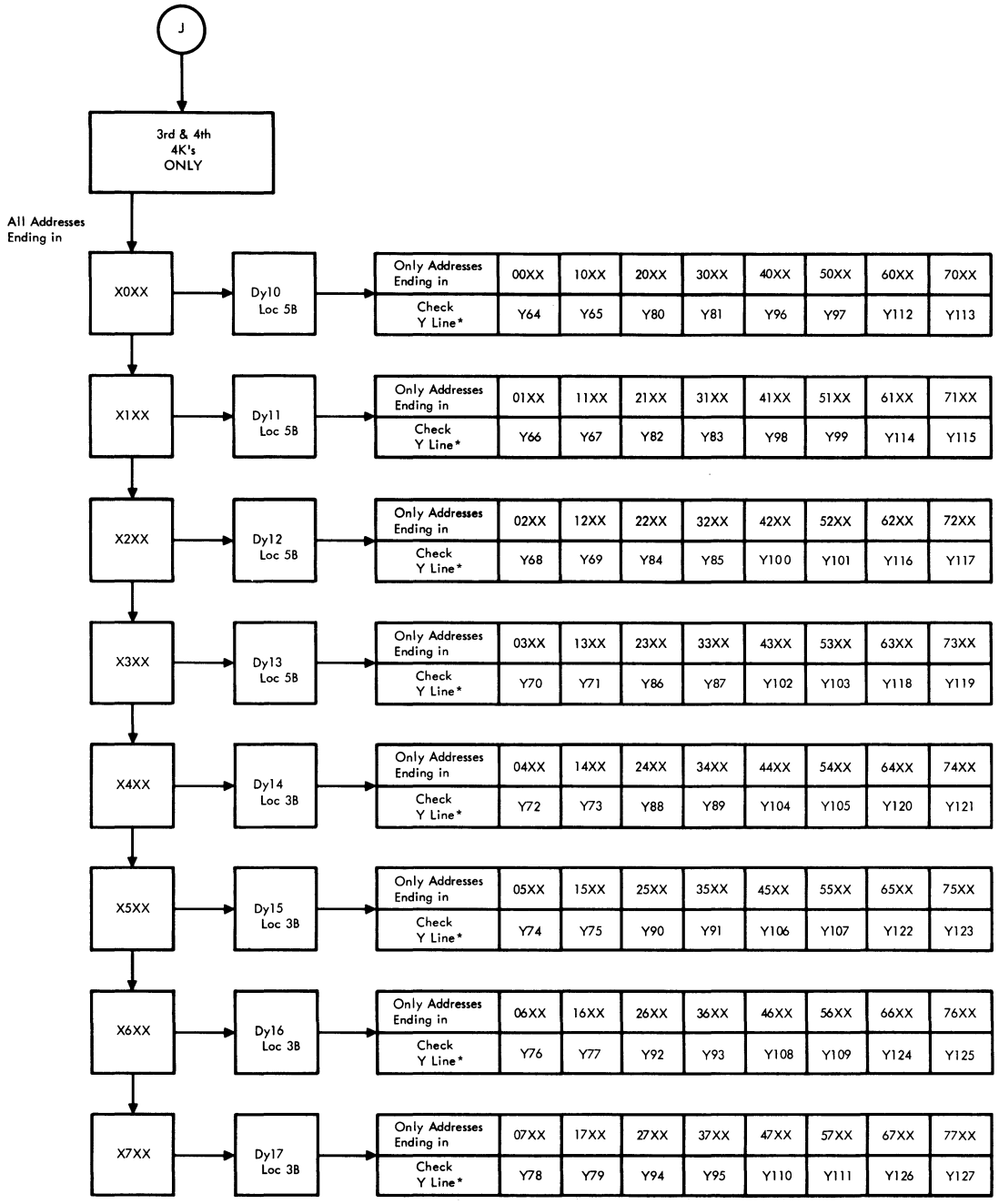


Figure 5-9. Memory Troubleshooting, Third and Fourth 4K Stacks

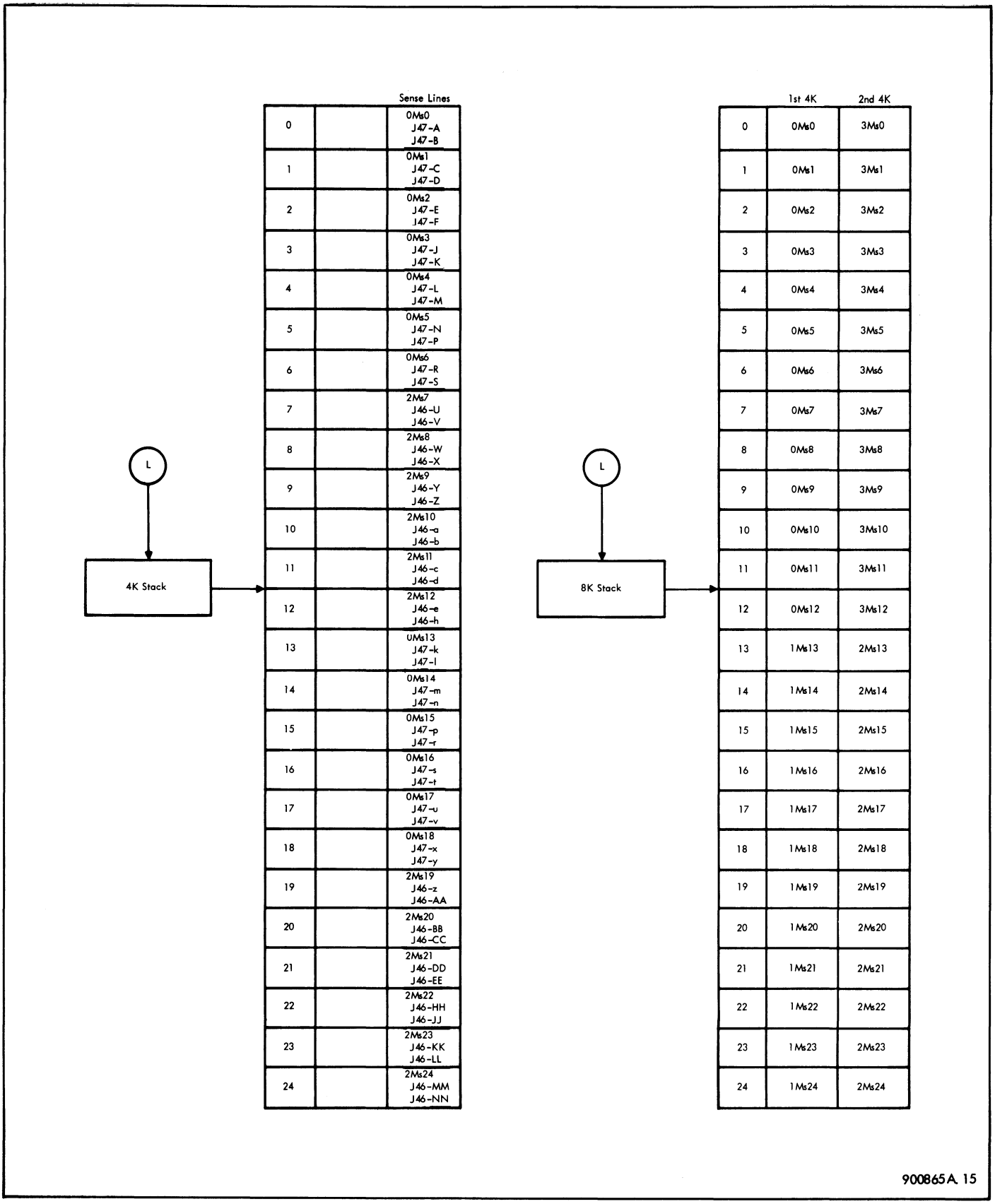


Figure 5-10. Memory Troubleshooting, 4K and 8K Stacks

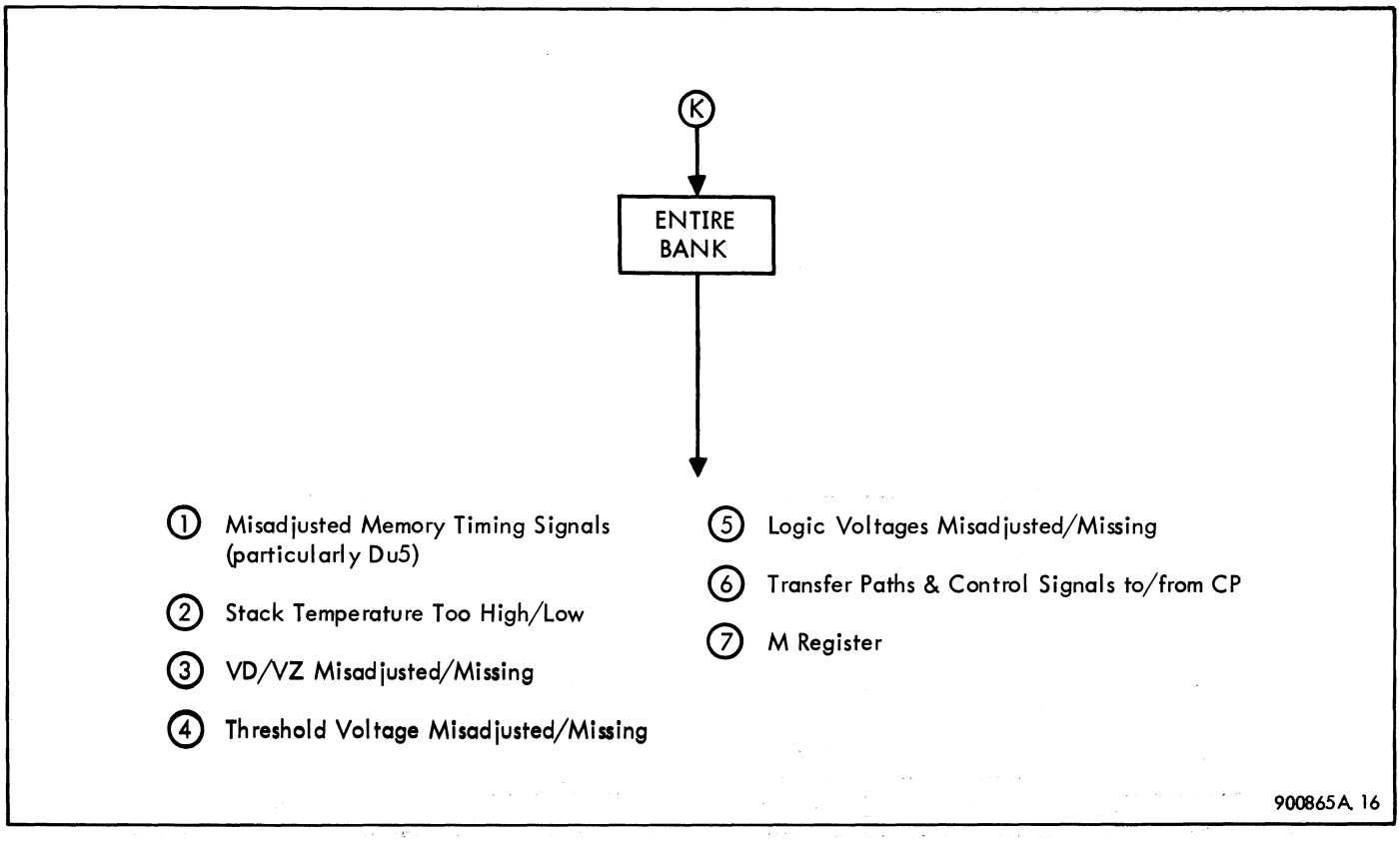


Figure 5-11. Memory Troubleshooting, Entire Bank

SECTION VI
PARTS LIST

A parts list for the SDS printed circuit modules in the 930 Computer is contained in the 925/930/9300 Computers Module Reference Data, SDS 900623.

SECTION VII
DRAWINGS

7-1 INTRODUCTION

7-2 The logic diagrams, circuit schematics, and reference drawings for the 930 Computer are given in the following publications:

Publication	Publication Number
SDS 930 Computer Central Processor, Logic Layouts	900592
930/9300 Memory, Logic Layouts	900620
925/930 Computer Basic Interrupt, Logic Layouts	900608
925/930/9300 Computers, General Reference Drawings	900619
925/930/9300 Computers, Module Reference Data	900623

7-3 The power distribution and control console schematics are not the same for all 930 Computers. The schematic drawing numbers are shown by computer serial number in table 7-1.

Table 7-1 Power Distribution and Control Console Schematics

930 COMPUTER SERIAL NO.	POWER DISTRIBUTION SCHEMATIC			CONTROL CONSOLE SCHEMATIC	NOTES
	Power Supply Cabinet	Central Processor	Input/Output Cabinet		
3100-3112	107770	107933	107826		Harrison Lab Power Supply, PX13 in I/O. I/O has separate AC
3113-3136	110863	110775	109497		SDS Power Supply PX13 in I/O
3137-3155	113136	113717A	113590		SDS Power Supply PX22, PX23 in I/O
3156-3161	113841	113717B	113590		Control console hookup change
3162-	113841	113717C	113590		Serial No. 3170 uses 3156-61 schematic
3100-3118				107000A	
3119-3155				107000B	
3156-				107000D	

SECTION VIII
LOGIC EQUATIONS

The logic equations for the 930 Computer central processor and memory are given in SDS 930 Computer Logic Equations, SDS 900636.

ALPHABETICAL INDEX

In the following index, numbers following the topic refer to the paragraph containing information on the subject. The letter T before a number indicates the subject is contained in a table; the letter F indicates the subject is contained in a figure.

ALPHABETICAL INDEX (Cont.)

ALPHABETICAL INDEX (Cont.)

ALPHABETICAL INDEX (Cont.)