60428500

CONTROL DATA
CORPORATION

CONTROL DATA®
FA722–A/B/C
DISK STORAGE CONTROLLER
DT220–C
SECOND CHANNEL FEATURE
DT716–A
DOUBLE DENSITY OPTION

GENERAL DESCRIPTION
OPERATION
THEORY OF OPERATION

Volume 1 of 3

HARDWARE MAINTENANCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
A	Manual released. Includes engineering change orders listed on correlation sheet.
(11-30-74)	
B	Manual revised; includes Engineering Change Orders 35876 (B), 35758, and 35759. Pages 1-1,
(2-5-75)	6-279, 7-7, 7-11, and 7-12 are revised by ECO 35876. Pages viii (Volume 2), 4-4, 4-5, 4-8, 4-9,
	6-263, 6-265, 6-267, 6-269, 6-271, 6-273, 6-275, A-431, A-432, and A-433 are revised by ECO
	35758. Pages 6-13, 6-15, 6-17, 6-19, 6-21, 6-23, 6-25, 6-27, A-421, A-422, A-423, A-424,
	A-425, A-426, and A-427 are revised by ECO 35759. Page 4-6 is deleted.
C	Manual revised; includes Engineering Change Order 35960. Page 6-261 is revised.
(2-5-75)	
D	Manual revised; includes Engineering Change Order 35954. Pages 6-147, A-394, A-395, A-396,
(2-5-75)	A-397, A-398, A-399, A-400, A-401, A-402, A-403, A-404, A-405, A-406, A-408, A-409, A-410,
	A-411, A-412, A-413, A-414, A-415, A-416, A-417, A-418 and A-419 are revised.
E	Manual revised; includes Engineering Change Order 35681. Pages 6-1, 6-35, 6-67, 6-69, 6-75,
(4-4-75)	6-79, A-106, A-108, A-310, A-364, A-369 and A-406 are revised. An editorial correction is
	made to page 4-9.
F	Manual revised; includes Engineering Change Order 36039. Pages vii, viii, 6-9, 6-11, 6-29, 6-31,
(4-4-75)	A-6, A-7, A-8, A-9, A-11, A-12, A-14, A-15, A-16, A-32, A-33, A-34, A-137, A-138, A-139,
	A-140, A-141, A-143, A-147, A-148, A-150, A-151, A-152, A-153, A-154, A-158, A-159, A-160,
	A-162, A-163, A-165, A-184, A-194, A-195, A-196, A-197 and A-380 are revised. Miscellaneous
	changes made to pages 6-13, 6-15, 6-17, 6-19, 6-21, 6-23, 6-25, 6-27, 6-61, 6-239, 6-241,
	6-243, 6-245, 6-247, 6-249, 6-251, A-29 and A-187. Pages 5-43 through 5-51 are added.
G	Engineering Change Order 36080. This information is included in Revision D.
(4-4-75)	
H	Manual revised; includes Engineering Change Order 36247 (publications change only).
(4-30-75)	Title Page (Vol. 1), Revision Record, Correlation Sheet, vii (Vol. 1), 1-1, 3-2 through 3-7, 3-9
	through 3-12, 3-229, 3-250, 3-287, 3-289, 3-292, 3-294, 3-300, Comment Sheet (Vol. 1), Title
	Page (Vol. 2), Revision Record, Correlation Sheet, viii (Vol. 2), 4-11, 6-5, 6-147, 6-171, 6-205,
	6-210, 6-211, 6-212, 6-213, 6-217, 6-252 through 6-257, 6-259, 6-261, 6-277, Comment Sheet
	(Vol. 2), Front Cover (Vol. 3), Title Page (Vol. 3), Revision Record, Correlation Sheet, vii (Vol. 3),
	A-3, Comment Sheet (Vol. 3), are the pages changed. New Pages added are 6-211.1, 6-213.1,
	6-257.0, 6-257.1, DT716-A Yellow Divider, A-437 through A-446. Appendix B Divider deleted.
J	Manual revised; includes Engineering Change Order 36235. Pages 4-9, 6-1, 6-7, 6-9, 6-11, 6-15,
(4-30-75)	6-23, 6-29, 6-31, 6-33, 6-37, 6-39, 6-61 of (Volume 2), A-5, A-6, A-9, A-12, A-15, A-16, A-18,
	A-37, A-136, A-137, A-147, A-153, A-156, A-162, A-164, A-165, A-179, A-187, and A-204 of
	(Volume 3) are revised. Engineering Change Order 35791 is included in this revision.
K	Manual revised; includes Field Change Order 36293. Pages 6-1, 6-9, 6-31, 6-61, and A-29 are
(7-11-75)	revised. Miscellaneous changes are made to Cover, Title Page, v in Volumes 1, 2, and 3, 1-1,
Publication No. 60428500	

Address comments concerning this
manual to:

Control Data Corporation
Publications and Graphics Division
4201 North Lexington Avenue
Arden Hills, Minnesota 55112

or use Comment Sheet in the back of
this manual.

REVISION RECORD (CONT'D)

REVISION	DESCRIPTION
K (Cont'd)	7-1, and 7-2.
L (7-23-75)	Manual revised; includes Field Change Order 36377 and Engineering Change Order 36079. Field Change Order 36377. Pages 6-213, 6-219, A-88, A-354, and A-361 are revised. Engineering Change Order 36079. Pages A-440, A-441, A-442, A-443, and A-446 are revised. Miscellaneous changes are made to vii, 6-7, 6-13, 6-15, 6-17, 6-19, 6-21, 6-23, 6-25, 6-27, 6-33, 6-37, 6-87, and 6-279. Pass On-Pass Back Wiring (Divider) and A-447 through A-454 are added.
M (9-26-75)	Manual revised; includes Field Change Order 36470. Pages 6-1 and 6-61 are revised. Editorial correction is made to page 6-63.

Publication No.
60428500

MANUAL TO EQUIPMENT LEVEL CORRELATION SHEET

SHEET 1 OF 1

		EQUIPMENT					
MANUAL REV	FCO OR ECO	FA722	FA722	DT220	DT716	FA722	
A	Released	A01	B01	C01	A01	A01	
-	ECO35758	A01	B01	C01	A01		
-	ECO35759	A01	B01	C01	A01		
B	ECO35876	A01	B01	C01	A01		
C	ECO35960	A01	B01	C01	A01		
D	ECO35954	A01	B01	C01	A01		
E	ECO35681	A01	B01	C01	A01		
F	ECO36039	A01	B01	C01	A01		
G	ECO36080	A01	B01	C01	A01		
H	ECO36247	A01	B01	C01	A01		
J	ECO36235	A02	B02	C02	A01	C02	
	ECO35791	A02	B02	C02	A01	C02	
K	FCO36293	A03	B03	C03	A01	C03	
L	ECO36079	A03	B03	C03	A01	C03	
	FCO36377	A04	B04	C04	A01	C04	
M	FCO36470	A05	B05	C04	A01	C04	

PREFACE

This manual provides level 2 hardware maintenance information for the CONTROL DATA ® FA722-A/B/C Disk Storage Controller and some of its related options. The FA722-A/B serves as the primary control element in CDC® 7054/844 Disk Storage Subsystems. Level 2 maintenance includes the following activities.

Preventive maintenance

Replacement of defective logic cards, power modules, and indicator lights

Adjustments

Cable and backpanel repair

The manual is divided into three volumes. Volume 1 contains equipment descriptions and operational theory. Volume 2 contains installation procedures, maintenance information, and diagrams. Volume 3 contains parts data, wire lists, and option information.

The following equipment manuals provide maintenance and reference information for equipment used in disk storage subsystems. The product manuals provide information pertaining to subsystem interaction with computer systems. Refer to the Literature Distribution Services catalog for the latest revisions of all manuals.

EQUIPMENT MANUALS

Control Data Publication	Manual Type	Publication No.
FA710/FA719/FA720/FA722 Disk Storage Controller	Hardware Reference	60364500
DC144-A/B, DT198-A, DC401-A Mass Storage Extender	Customer Engineering	60392600
BR30X Disk Storage Unit	Maintenance	70618400
BR30X Disk Storage Unit	Diagrams and Wire Lists	70618500
BR30X Disk Storage Unit	Parts List	70618600
84X/975X Disk Storage Unit	Maintenance Aids	70619100
TF201 Maintenance Console	Customer Engineering	58032700
BB372-A Core Storage Module/ FV444-A Power Regulator Module	Customer Engineering	60346500
TTL Key to Logic Symbols	Customer Engineering	60406000

PRODUCT MANUALS

Control Data Publication	Manual Type	Publication No.
Disk Storage Subsystem	General Information	60364400
Disk Storage Subsystem	Operation and Programming	60363900
Section 2 Stations/Subsystems	Site Preparation	60277500
SCOPE 3.4 Operating System	Installation Handbook	60307400
KRONOS 2.1 Operating System	Installation Handbook	60407500
Network Operating System (NOS) 1.0.0	Installation Handbook	60435700
System Maintenance Monitor (SMM)	Reference	60160600
Buffer Controller Diagnostics	Reference	60400300
Buffer Controller Assembly Language (BUCAL)	Reference	60372000

CONTENTS

VOLUME 1

FIGURES

TABLES

SECTION 1

GENERAL DESCRIPTION

GENERAL DESCRIPTION

INTRODUCTION

The FA722-A/B/C Disk Storage Controller (controller) interfaces up to eight CDC® BR302/BR303/BR308 Disk Storage Units with one peripheral processor unit (PPU) data channel from one of the following computer systems.

CDC 6000 Series
CDC® CYBER 70 Models 72, 73, 74
CDC® CYBER 170 Models 172, 173, 175

The FA722-A requires 60-Hertz (Hz) power, and the FA722-B may use either 50- or 60-Hz power. The FA722-C is functionally identical to the FA722-B. The difference between the B and C is in cabinet color, FA722-B is gold and the FA722-C is blue-gray.

EXPANSION OPTIONS

Options are available to increase the number of PPU data channels or disk storage units serviced by the controller. The DT220-C Second Channel Feature adds a second PPU data channel access to the controller. DC144-A/B, DT198-A, and DC401-A Mass Storage Extenders permit up to 64 standard density disk storage units to be attached logically to a controller. Double density disk storage units must connect directly to the controller. The second channel feature is described in this manual. Refer to the preface for publication numbers of mass storage extender manuals.

DOUBLE DENSITY OPTION

The [redacted] DT716A Double Density Option must be installed in the controller to permit operation with double density BR308 Disk Storage Units. The double density option requires wiring changes, the replacement of two control logic modules, and the addition of one control logic module.

CONTROLWARE MA 710 A

Many of the operating characteristics of the controller are determined by software which executes in the subsystem processor and control logic portions of the controller. This software is called controlware. The equipment configurator for controlware designed to execute in the controller is MA710. Refer to the disk controller hardware reference manual listed in the preface for controlware reference information.

FUNCTIONAL DESCRIPTION

Figure 1-1 illustrates the relationship of major controller components.

COUPLER

The coupler performs assembly/disassembly of 12-bit PPU words and provides an interface between the PPU and either the subsystem processor or the control logic.

MA 710 = CONTROLWARE A08 REVISION LEVEL.

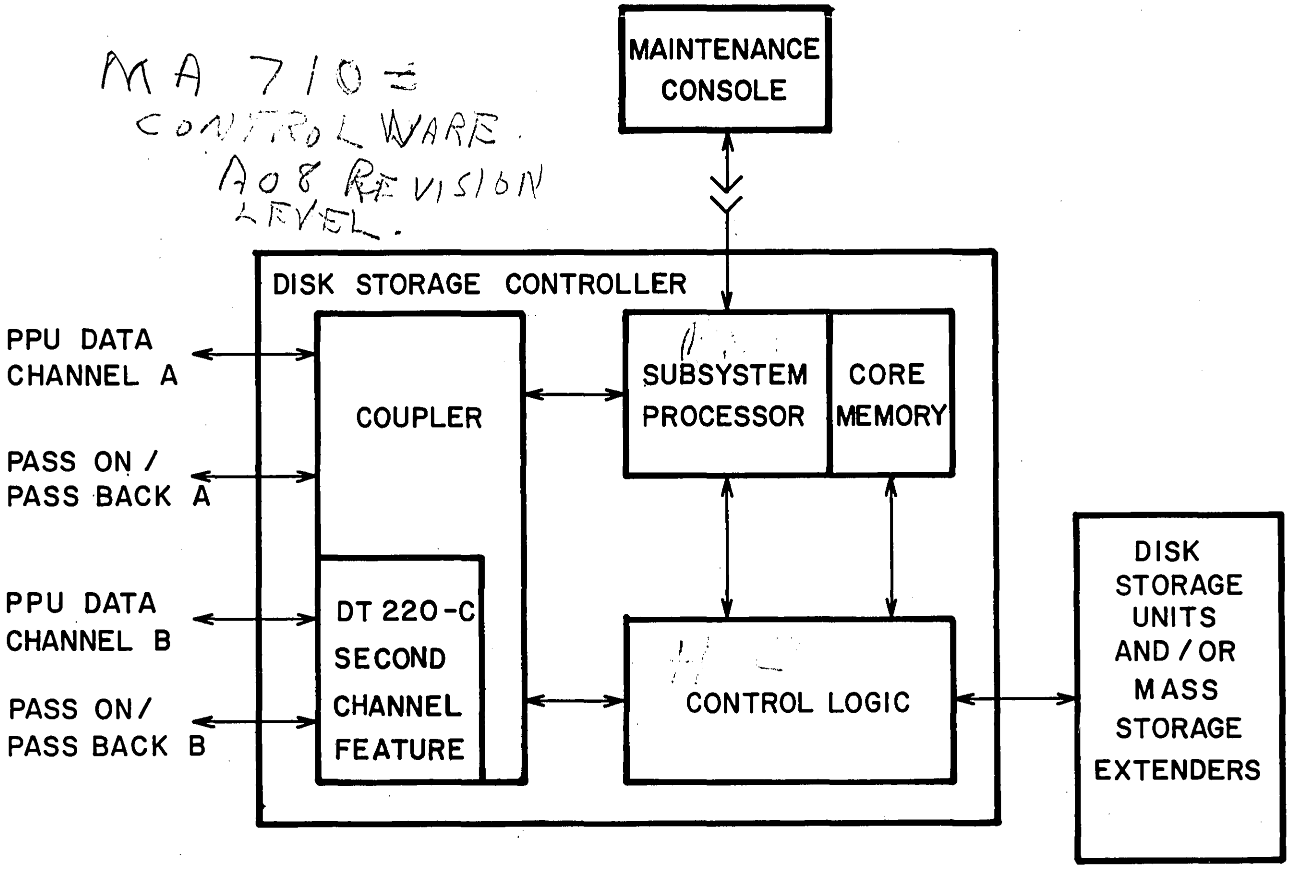


Figure 1-1. Controller Block Diagram

SUBSYSTEM PROCESSOR

The subsystem processor is a general purpose processor that uses a 16-bit instruction set and serves as the major control element of the controller. The subsystem processor communicates with a PPU via the coupler and controls the activities of the control logic. It receives function codes from the PPU and then selects a list of directors (control logic instructions) for execution by the control logic. The subsystem processor also passes status information from the control logic to the PPU. Although the subsystem processor controls data transfer operations, data passes directly between the control logic and the PPU via the coupler.

The subsystem processor has a control section, an arithmetic section, and various interfaces.

- The control section processes instructions and controls sequences.
- The arithmetic section performs two's complement arithmetic operations by means of an adder, two index registers, a shift network, and an accumulator register.
- A block transfer interface accommodates block transfers (such as autoload information) between the PPU and core memory.
- A normal channel interface provides communication with the PPU via the coupler and with the control logic. Control and status signals are exchanged on the normal channel interface.
- A storage interface links the subsystem processor to core memory through the memory scanner of the control logic.
- A maintenance interface provides for attachment of a portable maintenance console.

CONTROL LOGIC

The control logic is a special purpose processor that controls disk storage unit physical activity, transfers data to and from disk storage units, and performs error detection and correction. The control logic performs these functions by executing directors (control logic instructions) stored in core memory. The instructions are called directors to distinguish them from subsystem processor instructions. Directors vary in length from 16 to 64 bits.

When the subsystem processor has accepted a valid function code from the PPU, it selects a director sequence and informs the control logic. The control logic accesses core memory and begins loading the director sequence into its director buffer. The subsystem processor initiates execution of the director sequence which controls disk storage unit operation and data handling and processing. All data is passed directly between the control logic and the PPU (via the coupler) without being processed by the subsystem processor.

A memory scanner, which is part of the control logic, controls all accesses to the core memory and assigns priorities based on real-time constraints. The order of priority is:

1. Data handling references
2. Director buffer references
3. Subsystem processor references

Although the subsystem processor is assigned the lowest priority, it is normally not tied to real-time constraints and therefore has a larger proportion of access time available to it.

CORE MFMORY

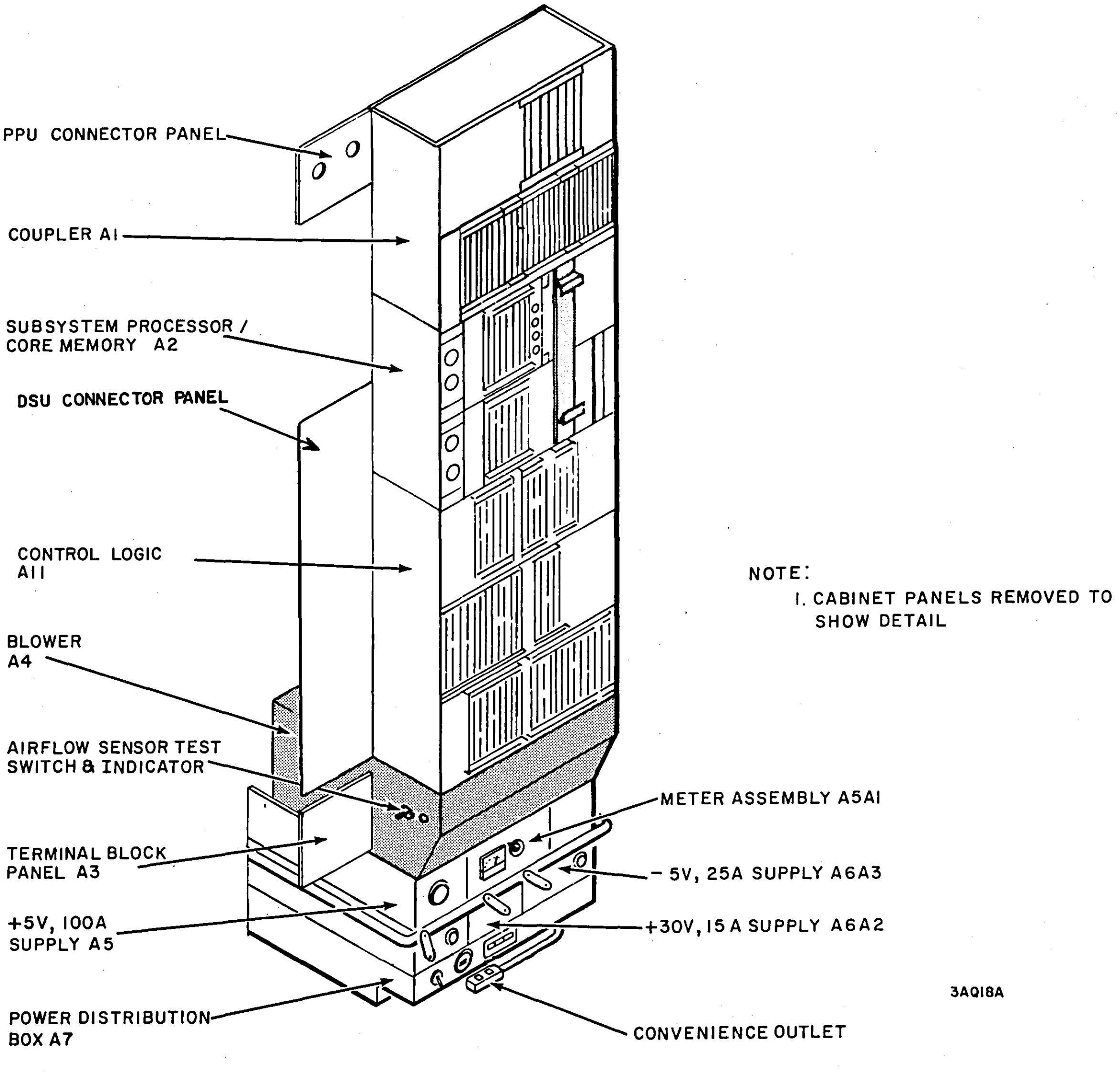
The core memory contains 4096 16-bit words and is shared by the subsystem processor and the control logic. The subsystem processor uses the memory to control overall operation of the controller. The control logic references memory to obtain director sequences which control disk storage unit and data handling operations.

To avoid memory conflicts, all memory references are routed through the memory scanner in the control logic so that the highest priority references are handled first. Although the core memory has a potential cycle time of 750 nanoseconds, the effective cycle time depends upon the operations being performed by the subsystem processor and the control logic.

PHYSICAL DESCRIPTION

Figure 1-2 illustrates and identifies major controller components.

The logic consists of TTL integrated circuits encapsulated in 14-pin or 16-pin dual in-line packages (chips) which are mounted on printed circuit modules. A module con-



NOTE:
1. CABINET PANELS REMOVED TO SHOW DETAIL

Figure 1-2. Controller Component Locations

tains up to 25 chips. These modules are called TTL 25-paks and plug into the logic chassis.

A logic chassis consists of two or three printed circuit boards mounted in a metal frame. Each board has several rows of pins which pass through the board and protrude from the front and back. Modules are inserted by sliding them over metal guides in the frame until they plug into the pins protruding from the front of the board. Backpanel pins are interconnected by a wirewrap method.

Doors and removable panels on the cabinet provide access to site maintainable components. Coupler logic occupies the top two rows of the cabinet and consists of 35 modules (including the second channel feature). The next two rows contain the subsystem processor, comprised of 23 modules, a core memory, a memory power regulator, a connector panel for the maintenance console, and an auxiliary operator panel. The lower three rows contain 71 modules of control logic. Most power related components are located in the base of the cabinet. A vertical bus bar carries +5 volts to horizontal bus bars in each logic chassis row. Individual power wires distribute other voltages to logic chassis bus bars.

Power supplies operate from 120/208-volt, 400-Hz, 3-phase power. The blower, elapsed time meter, and convenience outlet operate from 60-Hz or 50-Hz power.

NOTE

Since the convenience outlet is protected by a 7-ampere circuit breaker, it is suitable for use only with low power equipment.

The blower draws room air through a filter, circulates it around the power supplies and logic modules, and forces it out through exhaust ducts at the top of the cabinet.

DIMENSIONS

Height	76 inches	(193 centimeters)
Depth	25 inches	(63.5 centimeters)
Width	39 inches	(91.4 centimeters)
Weight	300 pounds	(136 kilograms)

ENVIRONMENTAL REQUIREMENTS

Cooling method	Forced air (blower)
Heat dissipation	2800 Btu per hour (705.6 Kg-cal/hr)
Temperature	
Operating	40°F (4.4°C) to 120°F (49°C)
Nonoperating	-30°F (-34.4°C) to 150°F (65.6°C)
Recommended operating	75°F (23.9°C)
Maximum gradient	20°F (11.0°C) per hour
Humidity (no condensation)	
Operating	10% to 90% relative humidity
Nonoperating	5% to 95% relative humidity
Maximum operating altitude	6000 feet (1830 meters) above sea level

POWER REQUIREMENTS

120/208 volts	400 Hz	3-phase
120 volts	50/60 Hz	1-phase

SECTION 2

OPERATION

OPERATION

INTRODUCTION

This section describes controls, indicators, and operating procedures for the controller and disk storage unit. For initial power-up and checkout procedures, refer to Installation and Checkout, section 4. For complete subsystem operating procedures, refer to the Disk Storage Subsystem Operation and Programming manual. For controlware and coupler reference information, refer to the disk controller hardware reference manual. These manuals are listed in the preface.

AUXILIARY OPERATOR PANEL CONTROLS AND INDICATORS

Figure 2-1 illustrates the auxiliary operator panel. Table 2-1 describes auxiliary operator panel controls and indicators.

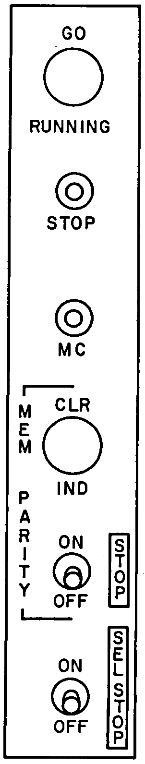


Figure 2-1. Auxiliary Operator Panel

TABLE 2-1. AUXILIARY OPERATOR PANEL CONTROLS AND INDICATORS

Nomenclature	Device	Function
GO/RUNNING	Momentary close switch/indicator light	Pressing the switch causes the subsystem processor to start executing instructions. The indicator lights when the subsystem processor is running.
STOP	Momentary close switch	Pressing the switch stops the subsystem processor.
MC	Momentary close switch	Pressing this switch clears the subsystem processor's control section, addressable registers, and output channels. The GO/RUNNING indicator must be unlighted when the MC switch is pressed.
MEM PARITY CLR/IND	Momentary close switch/indicator light	The indicator lights when a memory parity error is detected. Pressing the switch clears the error status and causes the light to go out.
MEM PARITY STOP ON/OFF	Toggle switch	When this switch is ON, the subsystem processor stops running upon detection of a memory parity error. When this switch is OFF, the subsystem processor continues to run after detecting a memory parity error.
SEL STOP ON/OFF	Toggle switch	When this switch is ON, the subsystem processor stops anytime it reads a 00XX instruction. When this switch is OFF, the subsystem processor ignores a 00XX instruction and goes to the next instruction.

POWER CONTROLS AND INDICATORS

Figure 2-2 illustrates the front panels of the power supplies and the power distribution box. Table 2-2 describes power indicators and controls.

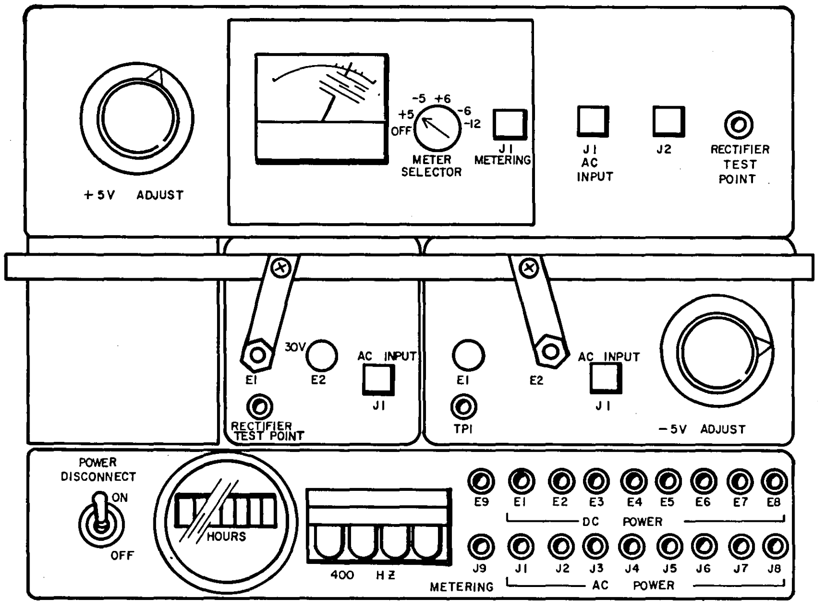


Figure 2-2. Power Controls and Indicators

POWER-UP/POWER-DOWN PROCEDURE

For this procedure, all logic voltages should be at their nominal values as described in the initial power-up procedure in section 4. The power-down sequence is listed first since it must be performed correctly for power-up to be effective.

POWER-DOWN

1. Press the STOP switch on the auxiliary operator panel. The GO/RUNNING indicator should go out.
2. Turn OFF the POWER DISCONNECT switch.

POWER-UP

1. Turn ON the 400 HZ circuit breaker and the POWER DISCONNECT switch. The blowers should operate.

2. Press the GO/RUNNING switch/indicator on the auxiliary operator panel. The indicator should light. The controller is now ready for operation.

TABLE 2-2. POWER CONTROLS AND INDICATORS

Control/Indicator	Function
+ 5V ADJUST rotary knob	Adjusts +5-volt power output. The adjustment is monitored on the percent meter.
Percent meter	Indicates the percent difference between the voltage input to the meter and the nominal voltage for that input. The METER SELECTION switch determines the voltage input to the meter.
- 5V ADJUST rotary knob	Adjusts -5-volt power output. The adjustment is monitored on the percent meter.
POWER DISCONNECT switch	Applies power to the blowers, elapsed time meter, and 400 Hz circuit breaker.
Elapsed time meter	Indicates the amount of time that 50/60-Hz power has been applied to the controller.
400 HZ circuit breaker	Protects the power supplies from overload.
AIRFLOW SENSOR BLOWER (on side of plenum)	Disables 50/60-Hz power input to blower during airflow sensor test.
AIR FLOW SENSOR RELAY INDICATOR (on side of plenum)	Indicates that the 130°F (55°C) thermostat has opened the relay supplying power to the 400 Hz circuit breaker.

DISK STORAGE UNIT CONTROLS

Figure 2-3 illustrates disk storage unit controls and indicators. Table 2-3 describes each of these items.

TABLE 2-3. DISK STORAGE UNIT CONTROLS AND INDICATORS

Nomenclature	Device	Function
START	Alternate action pushbutton/indicator	When the indicator is unlighted and all interlocks are properly set, pressing this switch applies power to the disk storage unit and initiates disk pack rotation. When the indicator is lighted, pressing this switch removes power from the disk storage unit and stops disk pack rotation.
READY	Indicator	Lights when the disk pack has reached normal speed and the heads are loaded.
FAULT	Pushbutton switch/indicator	Lights to indicate a fault condition in the disk storage unit. Pressing the switch clears the fault indication.
MAINTENANCE	Indicator	Lights when the controller is not controlling the power application to this disk storage unit.
TEMPERATURE	Indicator	Lights when an overtemperature condition exists in the disk storage unit.
Unlabeled Indicator	Indicator	Lights when the controller selects the disk storage unit.

DISK PACK EXCHANGE PROCEDURE

1. Make certain that the START indicator is not lighted and that the disk pack is not spinning. If the START indicator is lighted, press it to turn off the light and stop the disk storage unit.

2. When the disk pack has stopped, press the main cover latch (Figure 2-3) and lift the main cover. The dust cover opens with the main cover.

3. Place a disk pack cover over the loaded disk pack so that it engages the spindle. Turn counterclockwise until the spindle clicks and lift the cover and disk pack from the disk storage unit.

CAUTION

Place one hand under the disk pack to prevent the disk pack from falling free of the cover.

4. Using its cover as a handle, place the new disk pack slowly over the spindle until it engages the spindle drive unit. Turn the disk pack cover clockwise until it reaches a stop. Lift the disk pack cover from the disk storage unit.

5. Close the main cover, making sure that it latches. If the cover is not securely latched, the dust cover interlock remains open and prevents power application.

6. Press the START switch. When the disk pack is at operating speed (heads loaded), the READY indicator lights. The disk storage unit is now ready for operation.

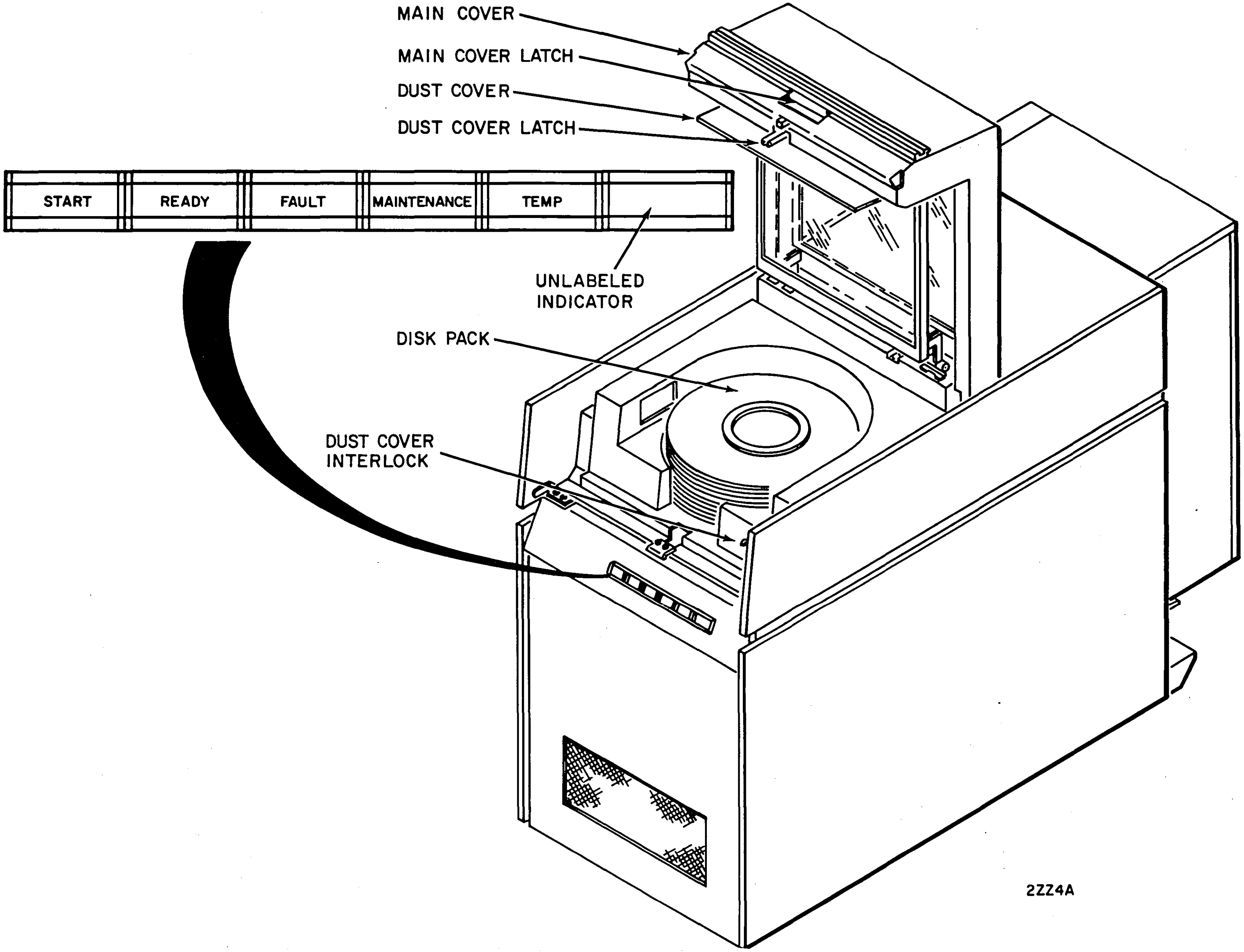


Figure 2-3. DSU Controls and Indicators

SECTION 3

THEORY OF OPERATION

THEORY OF OPERATION

INTRODUCTION

This section provides component theory and operational sequences for each of the major assemblies in the disk controller. In addition, the first part of this section describes the PPU interface, pass on/pass back interfaces, DSU interface, maintenance console interface, and interfaces linking controller major assemblies.

INTERFACES

Figure 3-1 illustrates the interfaces described in the following paragraphs.

PPU INTERFACE

This interface connects the disk controller to a PPU. Dual access controllers have two PPU interfaces, one for each PPU. All data and control signals on the PPU interface are 25-nanosecond pulses which are synchronized with 1-MHz and 10-MHz clock signals also present on the interface.

Signals From PPU Only

1 MHZ CLOCK/10 MHZ CLOCK

The controller uses these signals to synchronize all communication with a PPU.

ACTIVE

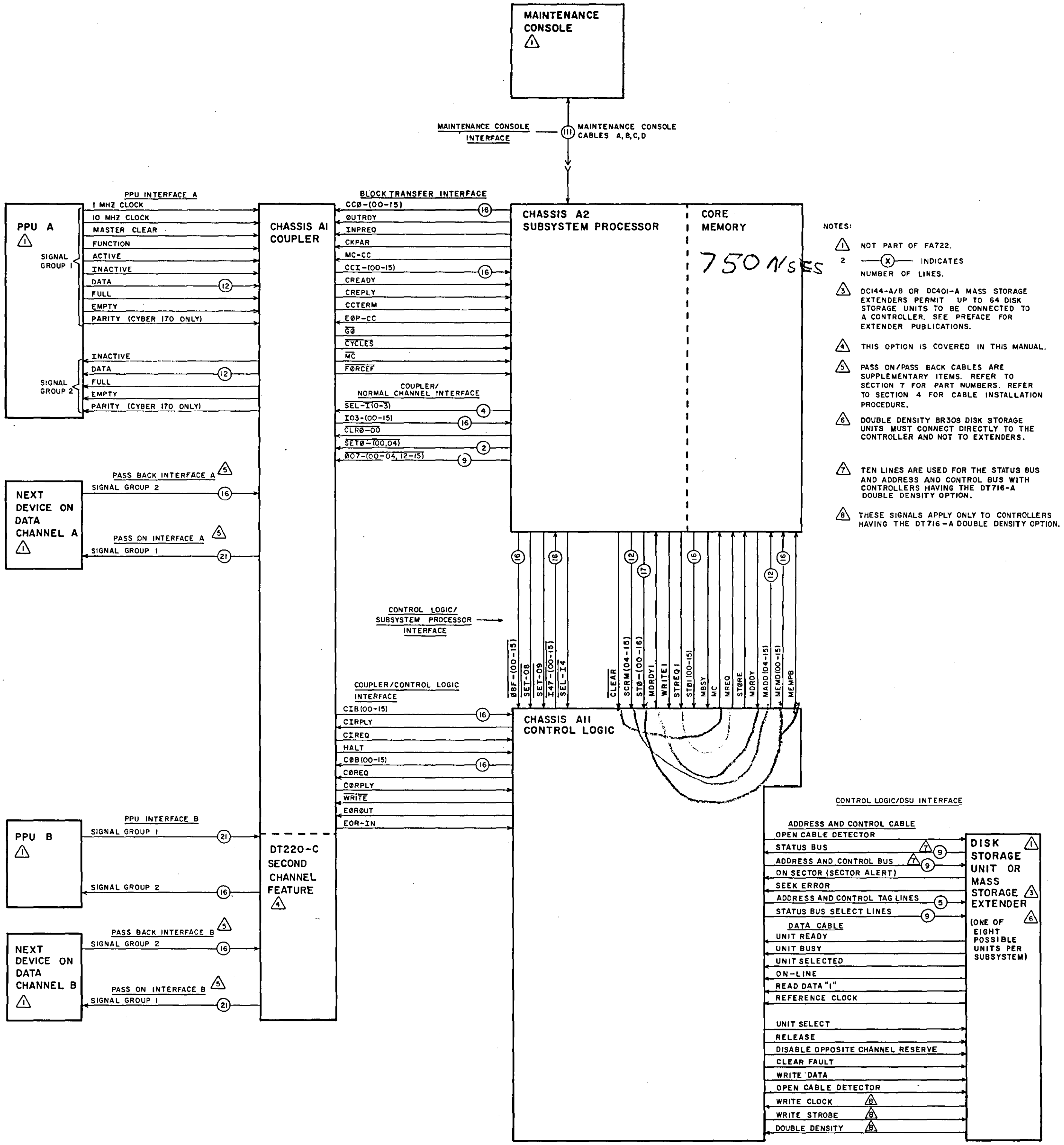
The PPU sends this signal to initiate or resume communication with the controller. An active signal accompanies each function signal.

FUNCTION

This signal identifies the accompanying 12 data signals as a function code.

MASTER CLEAR

The PPU sends this signal to clear the controller just after a computer system deadstart.



NOTES:

1 NOT PART OF FA722.

2 —(X)— INDICATES NUMBER OF LINES.

3 DC144-A/B OR DC401-A MASS STORAGE EXTENDERS PERMIT UP TO 64 DISK STORAGE UNITS TO BE CONNECTED TO A CONTROLLER. SEE PREFACE FOR EXTENDER PUBLICATIONS.

4 THIS OPTION IS COVERED IN THIS MANUAL.

5 PASS ON/PASS BACK CABLES ARE SUPPLEMENTARY ITEMS. REFER TO SECTION 7 FOR PART NUMBERS. REFER TO SECTION 4 FOR CABLE INSTALLATION PROCEDURE.

6 DOUBLE DENSITY BR308 DISK STORAGE UNITS MUST CONNECT DIRECTLY TO THE CONTROLLER AND NOT TO EXTENDERS.

7 TEN LINES ARE USED FOR THE STATUS BUS AND ADDRESS AND CONTROL BUS WITH CONTROLLERS HAVING THE DT716-A DOUBLE DENSITY OPTION.

8 THESE SIGNALS APPLY ONLY TO CONTROLLERS HAVING THE DT716-A DOUBLE DENSITY OPTION.

Figure 3-1. Interface Diagram

Bidirectional PPU Interface Signals

DATA

These 12 signals carry data, function, and status words between the PPU and the controller.

PARITY (CDC CYBER 170 Only)

This signal provides odd parity for the 12 data signals it accompanies.

FULL

This signal indicates to the receiver that the word formed by the 12 data signals can be accepted.

EMPTY

This signal indicates to the transmitter that the receiver has accepted the word accompanying the last full signal.

INACTIVE

The controller replies to a function signal (and its associated function code) with an inactive signal. An inactive signal may also be used by either the PPU or the controller to terminate communication.

PASS ON/PASS BACK INTERFACES

These interfaces transfer all the signals described under HLP Interface between the PPU and other devices connected behind the controller on the same data channel. Signals on the pass on interface lag corresponding signals from the PPU by 100 nanoseconds. Likewise, signals on the pass back interface lag corresponding signals from the other device(s) on the data channel by 100 nanoseconds. Each access of dual access controllers may have pass on/pass back cables installed.

CONTROL LOGIC/DISK STORAGE UNIT INTERFACE

Signals between the control logic and a disk storage unit or a mass storage extender are carried on cables A (the address and control cable) and B (the data cable).

Address and Control Cable

Status Bus

The status bus consists of nine lines (10 lines for double density option) which carry status information from the unit to the controller. Information on the status bus is determined by status select signals sent from the controller to the unit. Table 3-1 shows the relationship of the status select lines to the status bus.

TABLE 3-1. STATUS SELECT LINE FUNCTIONS

	Status Select Lines								
Status Bus	Read Cyl. Sel.	Read Diff. Cnt.	Read Head Reg.	Read Sec. Cntr.	Read Sec. Reg.	Read Interlock	Read Position	Read Fault	Read Control
Bit 0	1	1	1	1	1	Pack on	Forward	W+E+R· ⑤ $\overline{\text{On Cyl.}}$	Sector
Bit 1	2	2	2	2	2	Sector block ②	Reverse	(W+E)· R⑥	Pack unsafe
Bit 2	4	4	4	4	4	Heads loaded	Cylinder pulse	Current	Seek ⑦ error
Bit 3	8	8	8	8	8	Brush cycle	End of travel	+ Volt	On cylinder
Bit 4	16	16	16	16	16	Start switch	Fine servo	- Volt	Index
Bit 5	32	32				Local remote	Speed	Seek	Amplitude⑧ monitor 1
Bit 6	64	64				Spindle motor on		AC write① fault	End of cylinder
Bit 7	128	128				Power ③ supply temperature			Amplitude④ monitor 2
Bit 8	256	256		Cntr Valid		Logic ④ temperature			Amplitude④ monitor 3
Bit 9①	512①	512 ①							

① Used only with DT716-A Double Density Option.
② This is Control Interlock for double density option.
③ This is Logic Temp. for double density option.
④ Not used with double density option.
⑤ This is W+R· $\overline{\text{On Cyl}}$. for double density option.
⑥ This is W· R for double density option.
⑦ This is Seek Incomplete for double density option.
⑧ This is EOT Seek Error for double density option.

Address and Control Bus

The address and control bus consists of nine lines (10 lines for double density option) which carry information from the controller to the unit. Information on these lines is determined by address and control tag signals which are sent from the controller to the unit. Table 3-2 shows the relationship of the tag lines to the address and control bus.

On Sector (Sector Alert)

This is an unselected status line which indicates that the addressed sector is one sector away from the heads. The on sector signal is one sector in length and occurs once per revolution until a read gate, clear, or release is received. The on sector signal is disabled while a seek command is in process.

Seek Error

This is an unselected status line. It indicates that the unit was unable to complete a move within 500 milliseconds or that the carriage has moved to a position outside the recording field. A return to zero seek command cleans the seek error condition, returns the heads to cylinder zero, and enables an on cylinder signal to the controller.

TABLE 3-2. TAG LINE FUNCTIONS

Address and Control Bus	Tag Line				
	Difference Select	Cylinder Select	Sector Select	Head Select	Control Select
Bit 1	1	1	1	1	Write gate
Bit 1	2	2	2	2	Read gate
Bit 2	4	4	4	4	Seek forward
Bit 3	8	8	8	8	Head advance
Bit 4	16	16	16	16	Erase gate**
Bit 5	32	32			Seek reverse
Bit 6	64	64			RTZ
Bit 7	128	128			Data strobe early
Bit 8	256	256			Data strobe late
Bit 9 *	512	512			

*Used only with DT716-A Double Density Option.
**Not used in double density disk units.

Address and Control Tag Lines (Received by the Unit) (Refer to Table 3-2)

Difference Select: This signal indicates that the address and control bus contains positioning information (the difference between the unit's present cylinder address and the controller's new cylinder address).

Cylinder Select: This signal indicates that the address and control bus contains the controller's new cylinder address.

Sector Select: This signal indicates that the address and control bus contains the address of the sector which will generate the next on sector signal.

Head Select: This signal indicates that the address and control bus contains the address of the head which is to be selected.

Control Select: This signal indicates that the address and control bus contains control information with the following bit assignments.

	Signal	Function
Bit 0	Write gate	Enables write driver.
Bit 1	Read gate	Enables digital read data on the read data lines.
Bit 2	Seek forward	Initiates the positioning sequence in the forward direction if the difference counter contains a number greater than zero. If the difference counter equals zero, a forward carriage offset results.
Bit 3	Head advance	Increments the head counter by one. NOTE This signal occurring with the end of cylinder signal clears the head counter.
Bit 4	Erase gate	Enables the erase driver. (Not used in double density disk units.)
Bit 5	Seek reverse	Initiates the positioning sequence in the reverse direction if the difference counter contains a number greater than zero. If the difference counter equals zero, a reserve carriage offset results.

	Signal	Function
Bit 6	Return to zero	Initiates positioning to cylinder zero, and clears seek error condition if one exists.
Bit 7	Data strobe early	When this signal is true, the data strobe is moved to the early margin position.
Bit 8	Data strobe late	When this signal is true, the data strobe is moved to the late margin position.

Status Bus Select Lines (Refer to Table 3-1)

Read Cylinder Select (Controller to Unit): This signal gates the contents of the unit cylinder register to the controller on the status bus.

Interlock Select (Controller to Unit): This signal gates unit interlock status to the controller with the following bit assignments on the status bus.

	Signal	Function
Bit 0	Pack on	Indicates a pack is mounted on the spindle.
Bit 1	Sector block	Indicates the unit cover is closed and sector block is in position to sense sector disk.
	or	
	Control interlock	Indicates the unit cover is closed, all dc circuit breakers are closed, and the start switch is on. (Applies to double density option only.)
Bit 2	Heads loaded	Indicates heads have been loaded on the disk pack.
Bit 3	Brush cycle	Indicates pack brush cycle is in progress.
Bit 4	Start switch	Indicates start switch is on.
Bit 5	Local/remote	Indicates unit sequencing power is under control.
Bit 6	Spindle motor	Power is applied to the spindle motor.
Bit 7	Power supply temperature	Indicates power supply temperature normal.
	or	
	Logic temperature	Indicates logic chassis temperature normal. (Applies to double density option only.)
Bit 8	Logic temperature	Indicates logic chassis temperature normal. (Not used with double density option.)

Read Head Register Select: This signal gates the contents of the unit head register to the controller.

Read Difference Counter Select: This signal gates the one's complement of the contents of the unit difference counter to the controller on the status bus.

Read Sector Counter Select: This signal gates the contents of the unit sector counter to the controller on the status bus. Bit 8 is sector counter valid signal. When this line is true, the output of the sector counter is not in the process of changing. The counter contents are one number ahead of the actual sector count.

Read Sector Register Select: This signal gates the contents of the unit sector register to the controller on the data bus.

Read Position Status Select: This signal gates positioner control status to the controller on the data bus with bit assignment as follows:

	Signal	Function
Bit 0	Forward	Indicates the forward latch is set.
Bit 1	Reverse	Indicates the forward latch is cleared.
Bit 2	Cylinder pulse	Transmits cylinder pulses during a positioning function.
Bit 3	End of travel	Indicates the heads have been positioned beyond the useable recording field in either the forward or reverse direction.
Bit 4	Fine servo	Indicates the heads are being positioned under fine servo control and are less than 1/2 track from the final destination.
Bit 5	Speed & mtr on	Indicates the pack is rotating at a speed safe for flying the heads.

Read Fault Status Select: This signal gates the contents of the fault register to the controller on the status bus with the following bit assignments.

	Signal	Function
Bit 0	$W+E+R \cdot \overline{\text{On Cyl}}$	Indicates a write, erase, or read gate has been received while unit was off cylinder.
	or	
	$W+R \cdot \overline{\text{On Cyl}}$	Indicates a write or read gate has been received while unit was off cylinder. (Applies to double density option only.)
Bit 1	$(W+E) \cdot R$	Indicates a write or erase gate has been received while the read gate is true.
	or	
	$W \cdot R$	Indicates a write gate has been received while the read gate is true. (Applies to double density option only.)
Bit 2	Current	Indicates one or more of the following fault conditions. More than one head selected Erase and no write driver on Write drive on and no erase Both write drivers on Write gate without write data
Bit 3	+Volt	Indicates abnormal positive voltage in the unit.
Bit 4	-Volt	Indicates abnormal negative voltage in the unit.
Bit 5	Seek	Indicates a seek error has occurred.
Bit 6	AC write fault	Indicates write gate without write data. (Applies to double density option only.)
Bit 7	Not used	

Read Control Status Select: This signal gates the following control functions to the controller on the status bus.

	Signal	Function
Bit 0	Sector mark	Indicates start of record.
Bit 1	Pack unsafe	Indicates a unit fault has occurred.

Signal		Function
Bit 2	Seek error	Indicates unit was unable to complete a move within 500 milliseconds or that the carriage has moved to a position outside the recording field.
	or	
	Seek incomplete	Indicates unit was unable to complete a move within 575 ± 175 milliseconds or that the carriage has moved to a position outside the recording field. (Applies to double density option only.)
		A return-to-zero seek command clears the seek error or seek incomplete condition, returns the heads to cylinder zero, and enables an on cylinder signal to the controller.
Bit 3	On cylinder	Indicates unit has completed a move to the addressed cylinder.
Bit 4	Index	Track reference signal. This signal occurs once per revolution. The leading edge is coincident with the leading edge of the sector zero sector mark.
Bit 5	Amplitude monitor 1	Indicates the average amplitude of the data signal envelope has dropped below a selected minimum value
	or	
	EOT seek error	Indicates that the carriage has moved to a position outside the recording field. A return-to-zero seek command will clear the EOT seek error condition, return the heads to zero, and enable an on-cylinder signal to the control logic. (Applies to double density option only.)
Bit 6	End of cylinder	This line indicates the head counter has been advanced beyond head 18.
Bit 7	Amplitude monitor 2	This signal is a monitor of short-duration changes in data signal amplitude with a threshold of approximately 25 percent of the average amplitude; it indicates that marginal data may follow. (Not used with double density option.)

Signal		Function
Bit 8	Amplitude monitor 3	This signal is a monitor of short-duration changes in data signal amplitude with a threshold of approximately 50 percent of the average amplitude. This function is intended as an aid in certifying the pack during the formatting process. The format pattern should be all ones or zeros. This line should be monitored only during the data field. (Not used with double density option.)

NOTE

Amplitude monitor 2 and amplitude monitor 3 have a pluse forming circuit in the output with minimum pulse duration of 5 microseconds.

Data Cable

Unit Ready (Unit to Controller)

This signal indicates the selected unit has completed a first seek. It is cleared when the heads have unloaded with the spindle motor off. The unit does not accept seek commands until ready.

Unit Busy (Unit to Controller)

This signal indicates the selected unit is reserved by the other channel. It is applicable to dual channel use only.

Unit Selected (Unit to Controller)

This signal indicates the selected unit is available and that its unit select input is true.

On-Line (Unit to Controller)

This is a single-ended line operated from the on-line switch. An open condition indicates an off-line status, and the controller should disable the interface to this unit.

Read Data 1 (Unit to Controller)

This line transmits the recovered data in the form of pulses-on-ones information after the data has been separated from the read clock.

Reference Clock (Unit to Controller)

The reference clock defines the beginning of a data cell. It is an internally derived reference signal and is synchronous with recorded data.

Unit Select (Controller to Unit)

This line selects the unit for communication with the controller unless the unit is busy or off-line. The unit returns a unit selected signal in response to this signal.

Release (Controller to Unit)

This line clears the reserve in the selected unit. Unit select and release may occur at the same time.

Disable Opposite Channel Reserve (Controller to Unit)

This signal releases all units reserved by the opposite channel without regard for unit status. It should be used only when it is known that the opposite channel is inoperative.

Clear Fault

This line clears the fault flip-flop and the fault register in the selected unit.

Write Data

This line carries data which is to be recorded on the disk pack. Coding is modified frequency modulation (MFM) and the bit rate is 6.8 MHz. (For double density option, coding is non-return to zero (NRZ) and the bit rate is 6.451 MHz.)

Open Cable Detector

One of these lines is in each cable; it is a signal to the unit indicating that the unit is connected to the control logic.

Write Clock (Unit to Controller)

This line carries a 6.451 MHz clock used to generate NRZ write data in the controller and returned to the unit as write strobe. This is a continuous frequency synchronized with unit spindle speed variations for maximum data capacity. (Applies to double density option only.)

Write Strobe (Controller to Unit)

This line returns the write clock to the unit to compensate for I/O cable propagation differences by synchronizing NRZ write data with unit internal write clock before converting to MFM compensated write data. (Applies to double density option only.)

Double Density (Unit to Controller)

This line indicates that a double density unit is selected. Both sides of this differential signal are connected to ground in the unit. (Applies to double density option only.)

MAINTENANCE CONSOLE INTERFACE

This interface consists of the 111 signals carried by cables A, B, C, and D between a maintenance console and the subsystem processor. The interface allows the subsystem processor to be autoloaded, monitored, and controlled from the maintenance console. Refer to Table 2-1 in the TF201 Maintenance Console CE Manual for interface signal descriptions.

BLOCK TRANSFER INTERFACE

The block transfer interface is the normal data path between the coupler and the subsystem processor. This interface carries function codes, function parameters, and autoload data from the PPU to the subsystem processor and status information from the subsystem processor to the PPU. CCØ CCI

Coupler Ready (CREADY)

This signal indicates that the coupler will respond immediately to an input block transfer instruction or an output block transfer instruction in the subsystem processor.

Coupler Reply (CREPLY)

This pulse indicates that the coupler has accepted a data word from the subsystem processor or that the coupler has a data word available for the subsystem processor. The coupler responds to an input request or output ready with this signal.

Coupler Terminate (CCTERM)

Following an input request or an output ready, the coupler can terminate an input block transfer or output block transfer instruction by sending this signal. The subsystem processor does not send a parity strobe.

Input Data (CCI-00 through CCI-15)

These 16 lines carry input data from the coupler to the subsystem processor.

Output Ready (OUTRDY)

B.C.

This pulse indicates that the subsystem processor has data available for the coupler.

Input Request (INPREQ)

This pulse indicates that the subsystem processor can receive another word of data from the coupler.

Parity Strobe (CK-PAR)

This pulse indicates that the subsystem processor has accepted a data word from the coupler.

Master Clear (MC-CC)

This pulse clears registers and control in the coupler.

End of Operation (EOP-CC)

This pulse accompanies the last output ready signal during a block transfer to indicate that the transfer is complete. It is also returned to the coupler following a terminate signal from the coupler or following the reply to the last input request during a block transfer.

Output Data (CCO-00 through CCO-15)

These 16 lines carry output data from the subsystem processor to the coupler.

Not Cycle Stop ($\overline{\text{CYCLES}}$)

AUTO
LOAD
0414

The coupler sends this pulse to the subsystem processor to halt execution of the processor program at the end of the current memory cycle (regardless of its relationship to the end of the current instruction).

AUTO LOAD

Not Master Clear ($\overline{\text{MC}}$)

The coupler sends this pulse to clear the subsystem processor and its memory.

AUTO LOAD

Not Force Function ($\overline{\text{FORCEF}}$)

This pulse forces the subsystem processor to initialize a block transfer input instruction.

AUTO LOAD

Not Go ($\overline{\text{GO}}$)

This pulse starts execution of the input block transfer instruction.

COUPLER/NORMAL CHANNEL INTERFACE

This interface consists of normal input and output channels between the coupler and the subsystem processor. Normal input channels 00 through 03 are available to the coupler for sending information to the subsystem processor. Normal output channels 00, 01, 02, and 04 carry information from the subsystem processor to the coupler.

The coupler provides registers for storing normal input/output channel data. The subsystem processor has no normal channel registers and provides only input/output control signals and output channel data signals.

Normal Input Channel Data (I03-00 through I03-15)

These 16 lines carry data for normal input channels 00 through 03. The coupler provides information on these lines according to the input channel select signals sent by the subsystem processor.

Not Input Channel Select ($\overline{\text{SEL-I0}}$ through $\overline{\text{SEL-I3}}$)

These four lines define the contents of the normal input channel data lines. A zero on one of the lines selects the associated channel. Only one line may be selected at a time.

Not Normal Output Channel Data ($\overline{\text{O07-00}}$ through $\overline{\text{O07-15}}$)

These 16 lines carry complement data for normal output channels 00, 01, 02, and 04. The coupler accepts information from these lines according to set and clear output channel signals sent by the subsystem processor.

Not Set Normal Output Channel ($\overline{\text{SET-O0, O1, O2, O4}}$)

These four lines define the contents of the normal output channel data lines. A 0 on one of the lines sets bits in the associated coupler channel register for 1's on the normal output channel data lines.

Not Clear Normal Output Channel 00 ($\overline{\text{CLR-O0}}$)

This signal clears bits in the coupler channel 00 register for 1's on the normal output channel data lines.

CONTROL LOGIC/SUBSYSTEM PROCESSOR INTERFACE

This interface carries control signals, memory data, and address signals between the control logic and the subsystem processor.

Normal Input Channel Data (I47-00 through I47-15)

These 16 lines carry data for normal input channel 04. The control logic provides information on these lines when the subsystem processor sends a $\overline{\text{SEL-I4}}$ signal.

Not Select Input Channel 04 ($\overline{\text{SEL-I4}}$)

The subsystem processor sends this signal to enable status from the control logic onto the normal input channel data lines.

Not Normal Output Channel Data ($\overline{\text{O8F-00 through O8F-15}}$)

These 16 lines carry complement data for normal output channels 08 and 09. The control logic accepts information from these lines according to the $\overline{\text{SET-O8}}$ and $\overline{\text{SET-O9}}$ signals sent by the subsystem processor.

Not Set Normal Output Channel ($\overline{\text{SET-O8}}$, $\overline{\text{SET-O9}}$)

A 0 on one of these lines sets bits in the associated control logic channel register for 0's on the normal output channel data lines.

Not Clear ($\overline{\text{CLEAR}}$)

This signal is a 750-microsecond pulse which is generated at power-on time and whenever the MC switch on the auxiliary operator panel is pressed. The signal master clears the control logic and is relayed to the core memory as signal MC.

Storage Request One (STREQ1)

This is a 90- to 110-nanosecond pulse from the subsystem processor which initiates a memory request through the control logic memory scanner.

Storage Address (SCRM04 through SCRM15)

These 12 lines carry the memory address from the subsystem processor to the control logic. The lines are stable at the leading edge of the storage request and remain stable until MDRDY1 is received from the control logic indicating the memory cycle is complete.

Write Data (STO-00 through STO-15)

These 16 lines carry data from the subsystem processor to the control logic. The control logic passes the data to the memory when the processor request is honored.

Write Command (WRITE1)

A 1 on this line indicates that the data on the write data lines is to be stored into memory. This signal is passed to the memory as a store command by the control logic when the processor memory request is honored. When this line is a 0, the storage request is defined as a read request.

Read Data (STO-100 through STO-115)

These 16 lines carry data from memory to the control logic. Memory data is routed to the subsystem processor by backpanel wires, since both of these assemblies are in the same chassis.

Memory Data Ready One (MDRDY1)

This is a 60-nanosecond pulse from the control logic to the subsystem processor which indicates that the read data lines contain data requested by the subsystem processor.

Memory Request (MREQ)

This is a 100-nanosecond pulse from the control logic which starts the memory timing chain.

Master Clear (MC)

This is a signal from the control logic which master clears the memory. The signal is 8 microseconds in length when generated by the maintenance console and 750 microseconds in length when generated by the auxiliary operator panel or by a power-on master clear. The 750-microsecond signal is required to clear the memory power regulator.

Parity (STO-16 and STO-116)

These bits provide odd parity for the write data lines and read data lines, respectively.

Memory Address (MADD04 through MADD15)

These 12 lines carry the memory address from the control logic to the memory. These lines are stable at the leading edge of the memory request and remain stable until memory data ready is received from memory.

Memory Data (MEMD00 through MEMD15)

These 16 lines carry write data from the control logic to the memory. The lines are stable at the leading edge of the memory request and remain stable until memory data ready is received from memory.

Memory Data Ready (MDRDY)

This signal from the memory to the control logic goes from a 0 to a 1 350 nanoseconds after a memory request is received by memory and indicates that the memory request has been honored. It will remain at the 1 level until the next memory request is received.

Write Parity (MEMPB)

This signal from the control logic to the memory provides odd parity for the word on the memory data lines.

Memory Busy One (MBSY1)

This signal from the memory to the control logic indicates that the memory is performing a storage cycle.

COUPLER/CONTROL LOGIC INTERFACE

This interface carries information between the coupler and the control logic.

Input Data Request (CIREQ)

This signal generated in the coupler signifies that data is available on the input data lines to the control logic. It drops to a 0 on the leading edge of the input data reply (CIRPLY).

Input Data Reply (CIRPLY)

This is a 50-nanosecond (minimum) pulse from the control logic whose leading edge indicates that the input data has been accepted.

Output Data Request (COREQ)

This signal from the control logic indicates on its leading edge that the output data is available to the coupler. It remains a 1 until the leading edge of an output data reply (CORPLY).

Output Data Reply (CORPLY)

This signal from the coupler indicates on its leading edge that output data has been accepted. It drops to a 0 on the trailing edge of COREQ.

End of Record Output (EOROUT)

This is a 50-nanosecond (minimum) pulse from the control logic which indicates that the last data word has been transmitted to the coupler.

End of Record Input (EOR-IN)

This is a 150- to 200-nanosecond pulse from the coupler which indicates that the last input data word has been transmitted to the control logic.

Not Write ($\overline{\text{WRITE}}$)

This signal from the control logic indicates on the transition from a 1 to a 0 that a write operation has been initiated. In its 1 state, this signal indicates that a read operation is enabled.

Output Data (COB-00 to COB-15)

These 16 lines from the control logic to the coupler carry output data. These lines remain stable from the leading edge of the output data request (COREQ) to the leading edge of the output data reply (CORPLY).

Input Data (CIB-00 to CIB-15)

These 16 lines from the coupler to the control logic carry input data. These lines remain stable from the leading edge of the input data request (CIREQ) to the leading edge of the input data reply (CIRPLY).

Halt (HALT)

This signal from the coupler causes the control logic to stop director execution, stop director loading, stop any data transfers from memory after the data buffer releases the scanner, and send a halt status bit to the subsystem processor. The halt status and accompanying conditions can be cleared by any master clear.

COUPLER THEORY

The coupler performs the following functions.

Signal conversion — Converts 25-nanosecond pulses from the PPU data channel into TTL levels, and TTL levels into channel pulses.

Data format modification — Converts PPU data formats to formats required by the controller.

Data path switching — Establishes a data path between any two of the following.

PPU
Subsystem processor
Control logic

A DT220-C Second Channel Feature can be installed in the standard coupler to permit connection of a second PPU data channel to the controller.

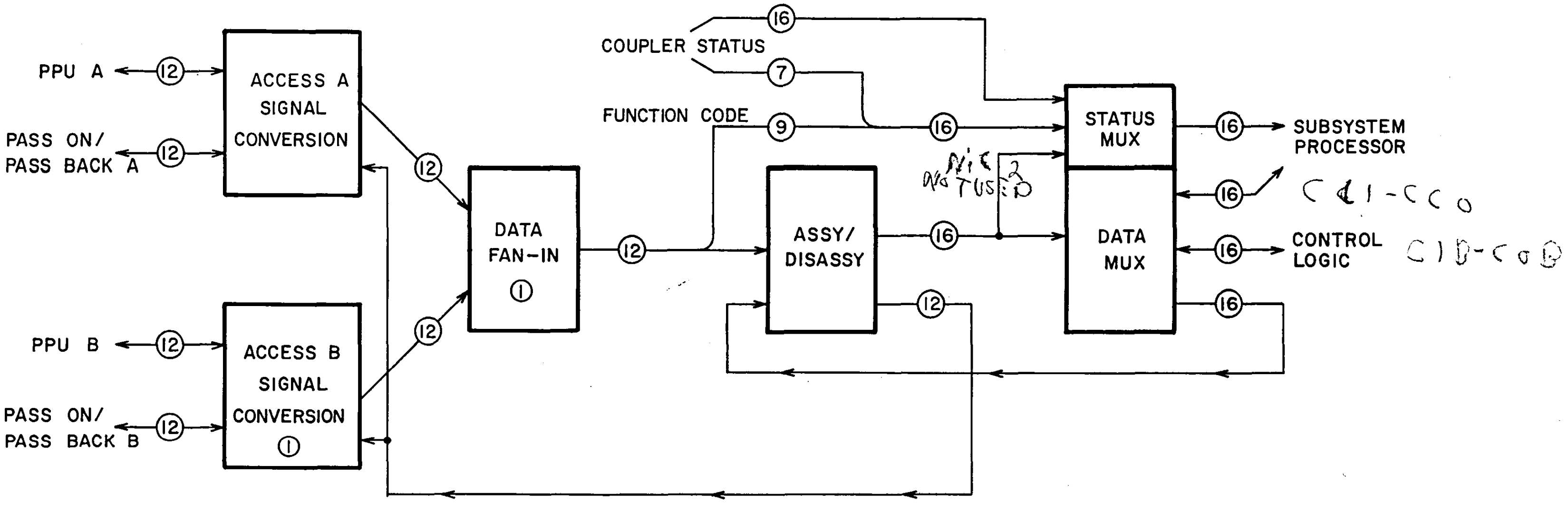
All cards called out in coupler theory are located in chassis A1 unless otherwise indicated.

COMPONENT THEORY

The coupler can be divided into four sections: signal conversion, control, assembly/disassembly, and data multiplexer. Figure 3-2 illustrates data and status flow through the coupler.

Signal Conversion

Signal conversion logic is concentrated in row A of the coupler chassis and is constructed of two basic circuits, pulsed to TTL converters (receivers) and TTL to pulsed converters (transmitters). The coupler uses receivers to pick up pulsed signals from the PPU and from another device on the channel (via pass back cable). Transmitters handle signals from the coupler to the PPU and signals from the coupler to another device on the channel (via pass on cable).



NOTE:

① DT 220-C SECOND CHANNEL FEATURE

Figure 3-2. Coupler Block Diagram

Channel control and clock fanout cards contain circuitry which synchronizes signal conversion logic with the PPU. The clock fanout card uses tuned lengths of wire to generate four clock signals from the PPU 10 MHz clock. These clock signals are then distributed to receiver/transmitter cards to time channel and pass on/pass back activity.

The coupler provides parity check/generate capability on the PPU data channel, although this capability is used only when the controller is attached to a CDC CYBER 170 series computer system. A switch on the parity card disables the parity capability when the controller is attached to a CDC CYBER 70 or 6000 series computer system.

The DT220-C Second Channel Feature adds a second set of converter, channel control, parity, and clock fanout cards to the standard coupler. DT220-C also adds cards containing connect logic, PPU data fanin logic, and status fanout logic.

Circuit descriptions for the basic receiver and transmitter circuits follow.

Pulsed to TTL Converter (Receiver)

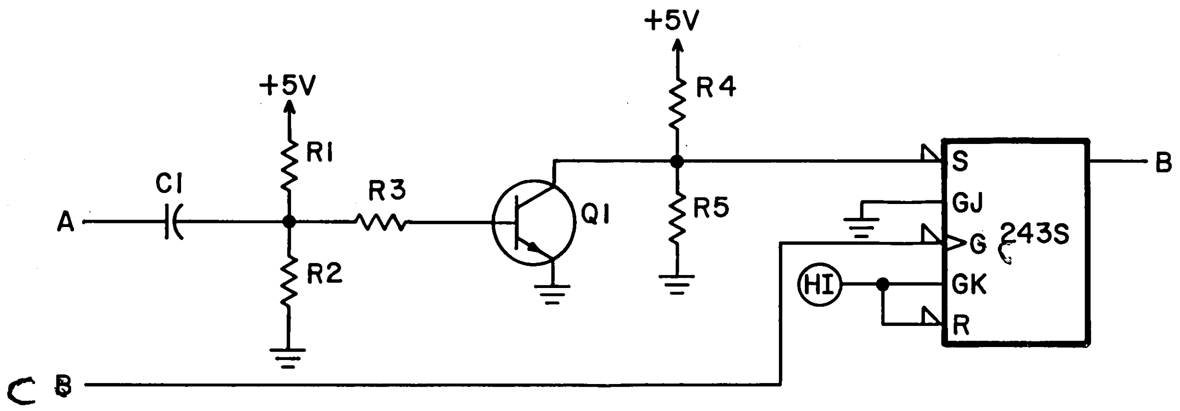


Figure 3-3. Pulsed to TTL Converter

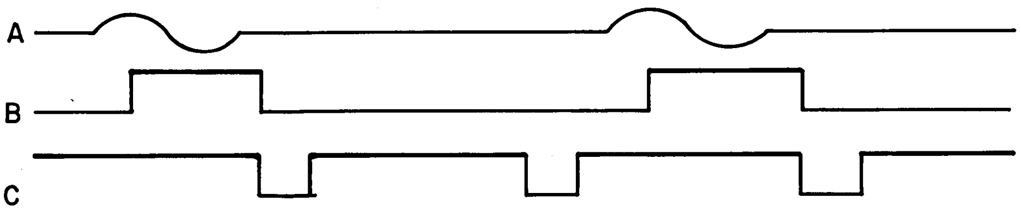


Figure 3-4. Waveforms for Pulsed to TTL Converter

Initially, R1, R2, and R3 hold the base of Q1 just below threshold, preventing Q1 from conducting. With Q1 not conducting, R4 and R5 hold the S input of the 243S chip high, thereby disabling it.

C1 couples the positive portion of waveform A through the R1/R2/R3 network to the base of Q1. This causes Q1 to conduct, dropping the S input of the 243S chip and setting the flip-flop.

After the received bit has been accepted by the coupler, channel control logic issues pulse C which clears the flip-flop in preparation for the next pulse A.

TTL to Pulsed Converter (Transmitter)

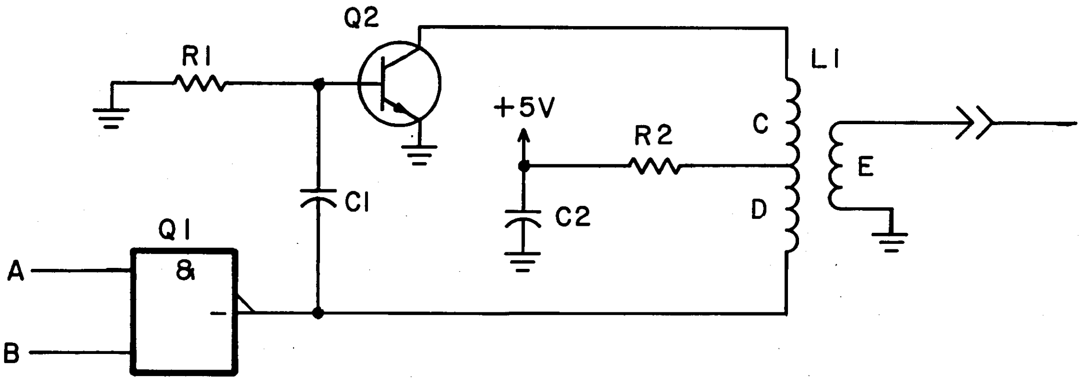


Figure 3-5. TTL to Pulsed Converter

Initially, Q1 and Q2 are off so that their collectors are at the source voltage, +5 volts. No current flows in either primary winding of L1. When inputs A and B are both high, the collector of Q1 goes low, and current flows in winding D of L1. (In practice, one of the inputs to Q1 is a level, and the other input is a 25-nanosecond pulse. The width of the input pulse is critical for proper circuit operation.) The voltage drop across R2 and the step-down action of L1 produce a voltage pulse in winding E of L1. The pulse duration on the line follows the 25-nanosecond pulse applied to Q1.

C1 couples the change in Q1 collector voltage to the base of Q2. The values of C1 and R1 have been selected to cause Q2 to conduct about 25 nanoseconds after Q1. When Q2 conducts, current flows in winding C of L1. The resulting induced voltage in winding E of L1 is equal to but of opposite polarity from the previous voltage. The net effect of this is to provide an equal but opposite magnetizing force on L1 to prevent L1 from building up a bias on the line.

When the level input to Q1 is high, the circuit outputs the previously described pulse. When the level input to Q1 is low, the circuit remains in its initial state. Figure 3-6 illustrates waveforms associated with inputs and outputs of this circuit.

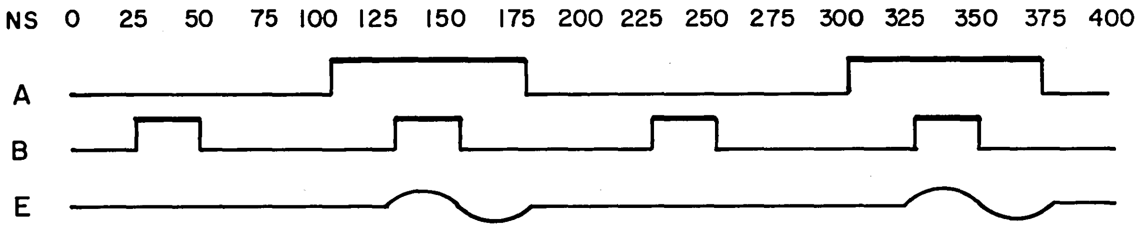


Figure 3-6. Waveforms for TTL to Pulsed Converter

Control

Coupler control logic performs four functions.

Checks all function codes for valid function and equipment number of 0.
Reserves the coupler and connects it (dual access only) to one of the accesses.
Controls PPU autoloads of the controller.
Coordinates data transfer through the coupler.

Function/Equipment Number Check

Since the equipment number of the controller is hardwired to 0, the control section ignores all other equipment numbers. The equipment number is the octal digit formed by channel bits 2^9, 2^{10}, and 2^{11}.

The coupler becomes reserved only when the received function code ranges from 0000 to 0477_8. A dual access coupler examines each function for an autoload (0414_8) or status (0012_8) function code. The standard coupler checks each function only for an autoload function code. Autoload and status (dual access only) functions are processed by the coupler. All other functions are passed to the subsystem processor for decoding.

Reserve/Connect

The coupler becomes reserved upon receipt of a legal function code containing equipment number 0. A deadman timer starts each time the reserved latch sets. This timer (which ensures that a subsystem malfunction cannot cause the PPU to wait more than 4 seconds for a response) resets whenever data is transferred with the subsystem processor or the control logic. The reserved latch clears when the PPU sends an inactive, when the subsystem sends an inactive or disconnect, when the coupler is master cleared, or when the processor deadman timer expires.

Connect circuitry is part of the DT220-C Second Channel Feature. A scanning connect switch connects the coupler to the access which first receives a legal function code containing equipment number 0. When the first access is connected, activity on the other access is limited to responses to status functions. These status responses set bit 2^{10} to 1 to indicate that the coupler is connected to the opposite access.

Although it does not provide an immediate response to a nonstatus function on an unconnected access, the coupler does retain the function code for processing after the connect on the other access drops. The PPU can clear this function by sending an inactive or master clear to the coupler.

Autoload Control

This logic checks all functions for the autoload function code (0414_8). When an autoload function code is detected, the coupler initiates the activity described under the autoload from PPU operational sequence.

Data Transfer Coordination

This logic generates ready/resume type signals which sequence data transfers with the PPU, the subsystem processor, and the control logic. Data transfer coordination logic also adjusts the assembly/disassembly mode and the path through the data multiplexer according to codes provided by the subsystem processor over normal output channels.

Assembly/Disassembly

Figure 3-7 illustrates the components which form the assembly/disassembly (A/D) section.

Data Register

This 32-bit register holds PPU or controller words during the A/D process. During most operations, the upper data register holds words 0, 2, 4, and so on, and the lower

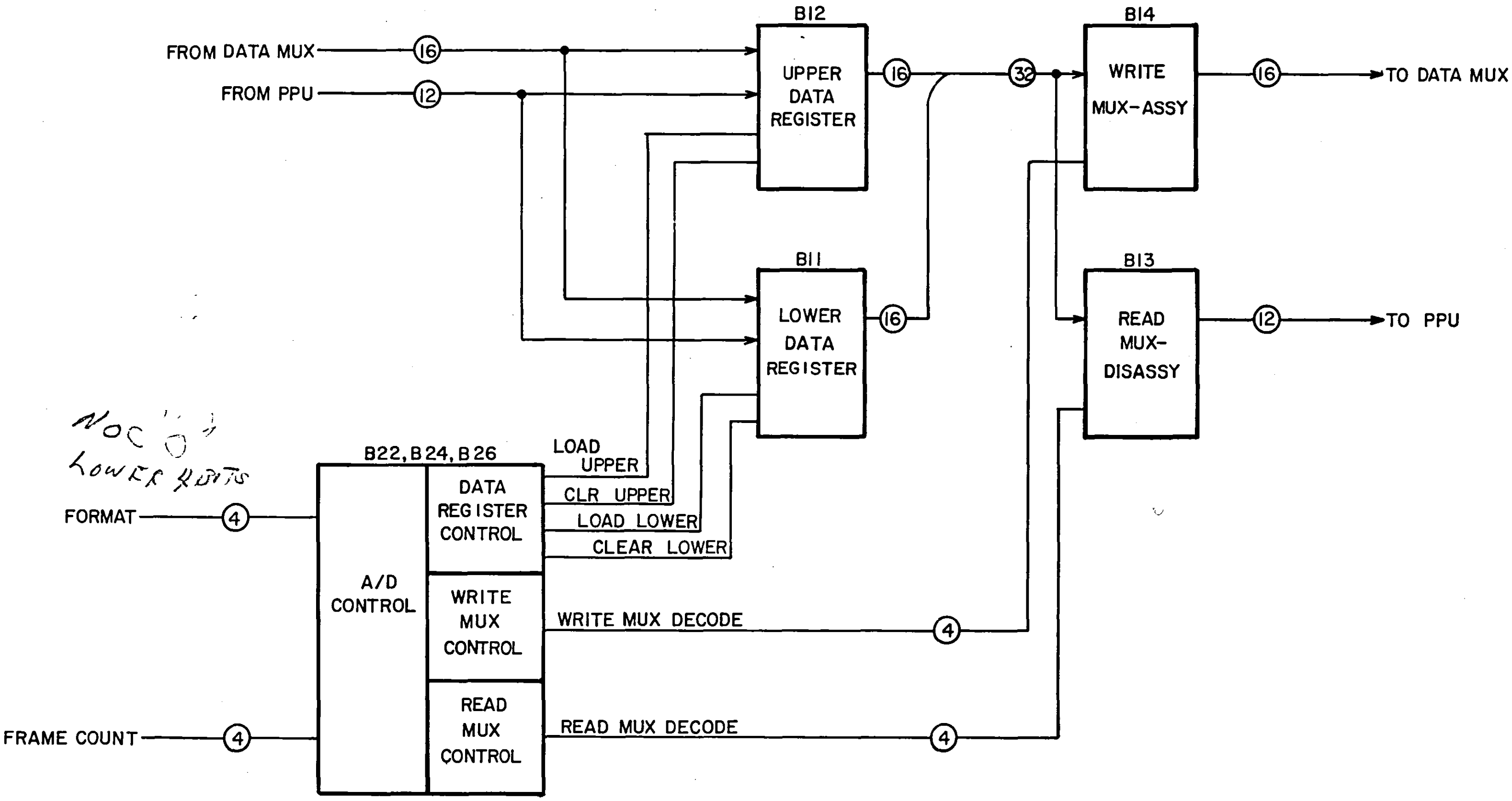


Figure 3-7. Assembly/Disassembly Components

data register holds words 1, 3, 5, and so on. During format 6/7 operations, the upper data register holds all words. Both the upper and lower data register hold 12-bit PPU words in the highest order bit positions of the register.

A/D Multiplexers

These two multiplexers are designated the write mux and the read mux. The write mux, which assembles PPU words into controller words, consists of 16 16 by 1 multiplexer chips. The read mux, which disassembles controller words into PPU words, consists of 12 16 by 1 multiplexer chips.

Both the write mux and the read mux work on the same principle. The 16 (write mux) or 12 (read mux) input pins associated with each format/frame count combination are connected to appropriate bit positions of the data register. When a 4-bit mux decode is then applied to the mux, it outputs a word in the format associated with the mux decode.

A/D Control

This logic processes format codes and current frame counts to produce data register clear/load signals, the write mux decode, and the read mux decode.

A maximum frame count is associated with each format as follows:

Format	Maximum Frame Count
0	1
1	4
2	3
3	4
4	2
5	2
6	1
7	1
8	7
9	10

The maximum frame count for a format is the number of words that must be removed from the write mux or from the read mux before the format pattern repeats. Format patterns are illustrated in the disk controller hardware reference manual listed in the preface.

The format is selected by the subsystem processor on normal output channel 00, bits 2^0 through 2^3. Even formats are used for assembly, and odd formats are used for disassembly.

Whenever the connected PPU activates the data channel, the coupler sets the frame counter to the maximum frame count associated with the current format. The frame counter is decremented whenever a word is taken from the write mux or from the read mux. When the frame counter decrements to 0, the coupler resets it to the maximum frame count.

Clearing and loading of the data register is also synchronized with format and frame count.

Table 3-3 indicates the signals generated by A/D control for each combination of format and current frame count.

Data Multiplexer

The data multiplexer consists of three 16-bit data multiplexers and a 16-bit status multiplexer. Figure 3-8 illustrates the paths through the multiplexers.

Data Paths

The subsystem processor outputs a 2-bit code on normal output channel 4 to select one of the following data paths.

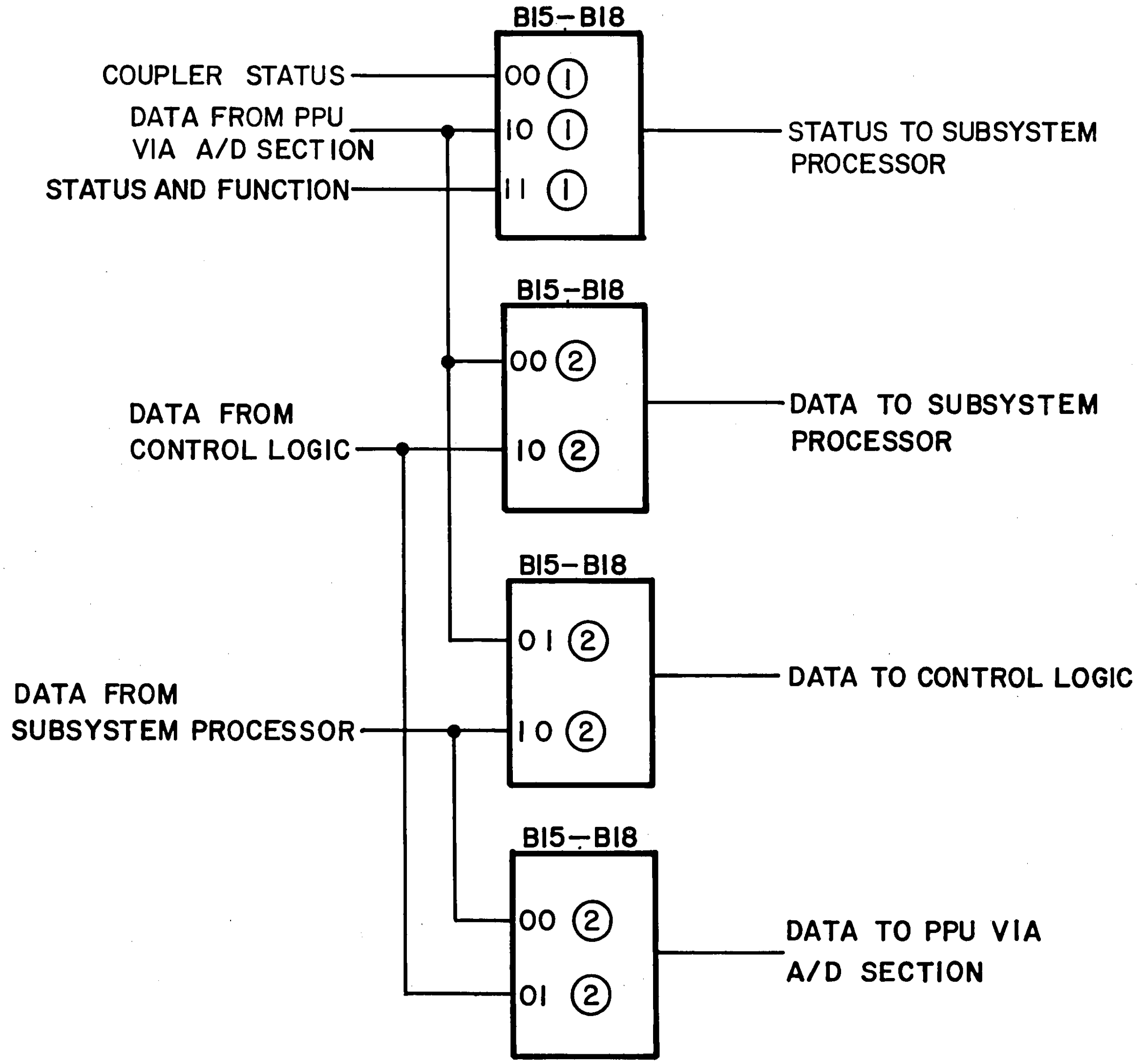
NOC 4
BIT14,15

Path Select Code	Path
00	Subsystem processor/PPU
01	Control logic/PPU
10	Subsystem processor/control logic

EVEN FORMAT TO MUX
ODD FORMAT FROM MUX

TABLE 3-3. A/D CONTROL SIGNALS

Format	Current Frame Count	Clear/Load Data Register	Write Mux Decode	Read Mux Decode
0	1	Both	0000	-
1	4	-	-	0000
1	3	Upper	-	0001
1	2	-	-	0010
1	1	Lower	-	0011
2	3	Upper	0001	-
2	2	Lower	0010	-
2	1	Both	0011	-
3	4	-	-	0100
3	3	Upper	-	1101
3	2	Lower	-	0110
3	1	Both	-	0111
4	2	Upper	0100	-
4	1	Lower	1110	-
5	2	Upper	-	0100
5	1	Lower	-	1000
6	1	Upper	0101	-
7	1	Both	-	1000
8	7	Upper	0110	-
8	6	Both	0111	-
8	5	Lower	1000	-
8	4	Both	1001	-
8	3	Upper	1010	-
8	2	Lower	1011	-
8	1	Both	1110	-
9	10	-	-	1010
9	9	Upper	-	1011
9	8	Lower	-	1001
9	7	-	-	1100
9	6	Upper	-	1101
9	5	Lower	-	1110
9	4	-	-	0100
9	3	Upper	-	1101
9	2	Lower	-	1111
9	1	Upper	-	0111


B15–B18
COUPLER STATUS
00 ①
DATA FROM PPU VIA A/D SECTION
10 ①
STATUS AND FUNCTION
11 ①
STATUS TO SUBSYSTEM PROCESSOR
B15–B18
00 ②
DATA FROM CONTROL LOGIC
10 ②
DATA TO SUBSYSTEM PROCESSOR
B15–B18
01 ②
DATA FROM SUBSYSTEM PROCESSOR
10 ②
DATA TO CONTROL LOGIC
B15–B18
00 ②
01 ②
DATA TO PPU VIA A/D SECTION

NOTES:

① STATUS PATH SELECT CODE

② DATA PATH SELECT CODE

Figure 3-8. Data Multiplexer Components

Although the coupler has provision for a data path from a local autoload device to the subsystem processor, this path is not used.

Status Path

The coupler provides status to the subsystem processor over the I03-xx interface lines according to SEL-IX signals sent by the subsystem processor. These signals are encoded into a 2-bit number which determines the output of the status multiplexer. Since the subsystem processor receives control logic status directly over normal input channel 4, the control logic status path provided by the coupler is not used.

OPERATIONAL SEQUENCES

The following sequences highlight coupler operation during typical controller operations.

Coupler Connect (Second Channel Feature Only)

Figure 3-9 illustrates connect logic for dual access couplers. The standard (single access) coupler requires no connect logic, and therefore, jumpers the function signal from card A07 to function logic at card B21.

The coupler must be connected to an access before the access can reserve the coupler or issue nonstatus function codes to the controller. When one access is connected, status requests from the other access receive coupler reserved status responses until the connect drops on the first access.

When neither access is connected, the access A data fanin path at card B10 is enabled. The access B path is enabled only when access B is connected.

The dual access connect sequence proceeds as follows: assume that the coupler is not reserved and that the PPU attached to access A functions the controller first.

1. The function pulse from PPU A is caught on card A12 and then stored in the function hold A flip-flop on card A07.
2. When the function hold A flip-flop sets, it fires a 60-nanosecond one-shot on card B08 which sets the A function latch.

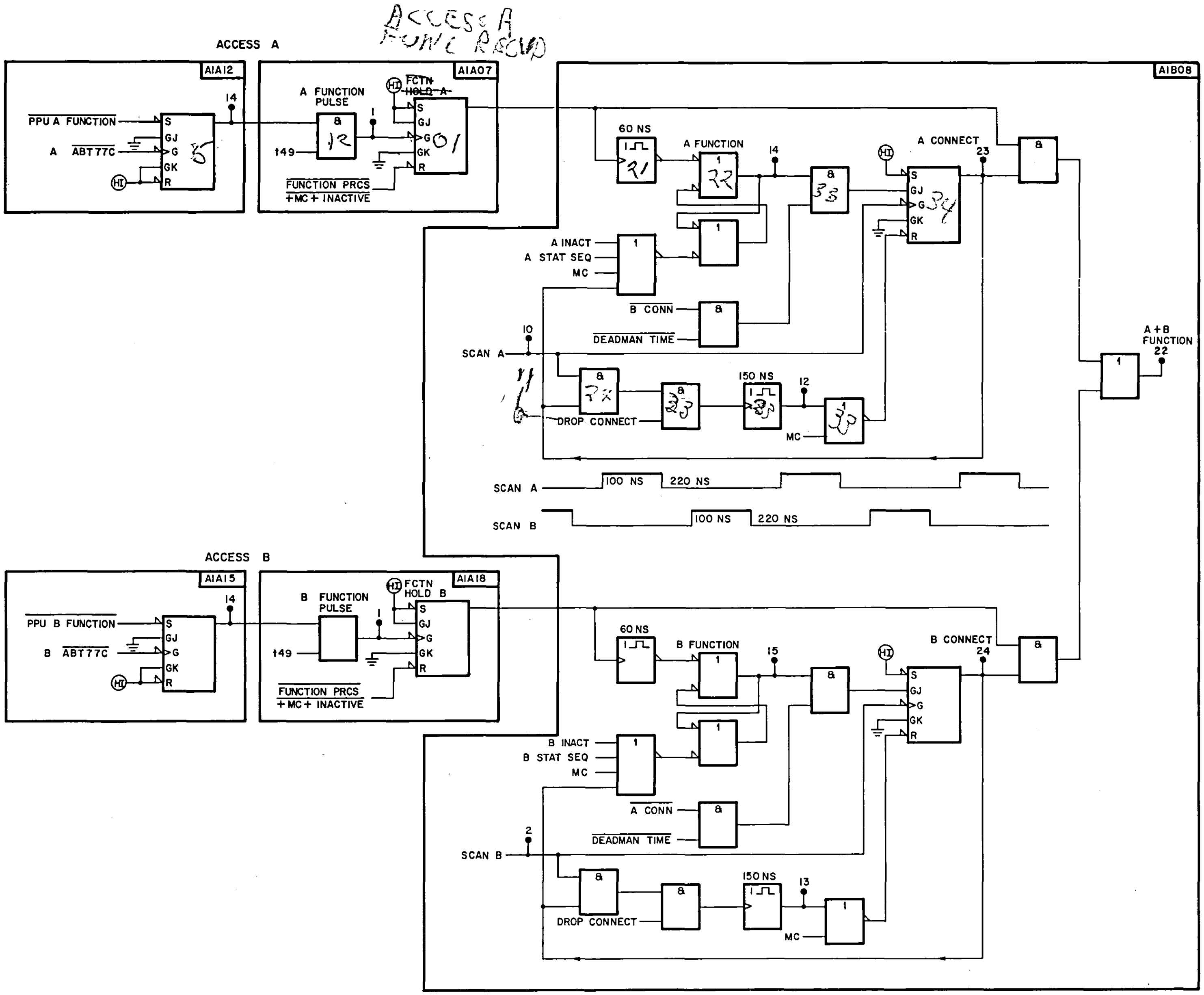


Figure 3-9. Connect Logic

3. Since access B is not connected and the deadman timer has not expired, the output of the A function latch is gated to the A connect flip-flop.

4. The next low transition of the scan A signal sets the A connect flip-flop, which in turn clears the A function latch and enables drop connect logic.

5. The outputs of the A connect and function hold A flip-flops combine to generate the A+B function signal. This signal feeds logic on card B21 which checks for equipment number 0 and identifies the function code.

6. The A connect flip-flop is cleared by the first high transition of the scan A signal which occurs after one of the following events.

   a. An equipment number other than 0 is detected on card B21.

   b. The subsystem processor issues an inactive or disconnect to the coupler.

   c. The coupler is master cleared.

   d. The deadman timer expires.

7. Access B connect logic operates just like the access A logic. Since the scan A and scan B signals are out of phase with each other, the possibility of a tie between the two accesses is eliminated.

Autoload

Types of Autoloads

The controller can be autoloaded from a maintenance console or from the PPU. A maintenance console autoload does not involve the coupler, so its sequence is not described here. Refer to the maintenance console customer engineering manual for information concerning this type of autoload.

Although the coupler has provision for attachment of a local autoload device, such a device is not part of, nor used with, the controller.

Autoload From PPU

This operation is best understood when broken down into three types of activity: autoload initiation (Figure 3-10), PPU/coupler activity (Figure 3-11), and subsystem processor/coupler activity (Figure 3-12). The PPU autoload sequence proceeds as follows:

1. The PPU sends a 0414_8 function. When the controller is a dual access model, the coupler performs a connect sequence before passing the function signal to card B21. Single access couplers jumper the function signal directly to card B21.

2. Function logic on card B21 receives the function signal, preclears the autoload function latch, and sets the function in process latch.

3. The 414_8 portion of the function code sets the autoload function latch and generates a pulse which does the following.

   a. Sets the reserved latch on card B21.

   b. Master clears the coupler on card B24.

   c. Replies to the function by returning an inactive to the PPU on card A07 (PPU A) or A18 (PPU B).

   d. Starts transmission of autoload initiation signals to the subsystem processor from card B19. The signals and their functions are as follows:

      Cycle stop — Halts the subsystem processor at the completion of its current memory reference.

      Master clear — Clears the control section and addressable registers of the subsystem processor, thereby forcing the impending autoload to begin at memory address 0000. Generates (in the subsystem processor) an MC-CC signal which is returned to the coupler to force the 00 (PPU/subsystem processor) data path. Also forces all bits of all normal output channels to 0, forcing a 0000 A/D format in the coupler. The 0000 A/D format is used for autoload data transfers.

      Force function — Forces a large block count in the subsystem processor A register and forces a block transfer instruction in the F register. The EN-OUT signal is grounded in the subsystem processor at card A2A26, so all forced functions are decoded as input block transfers.

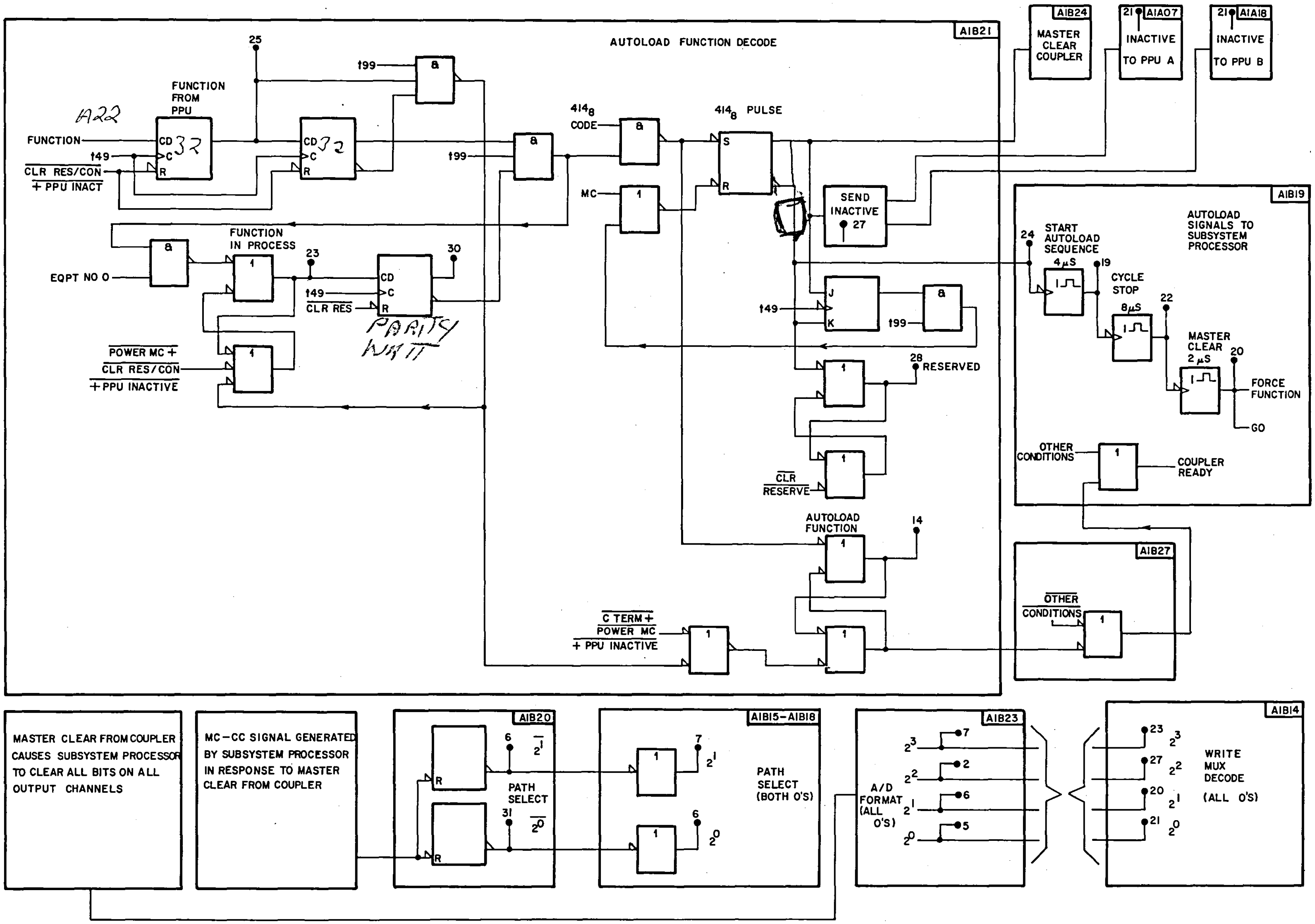


Figure 3-10. Autoload Initiation Logic

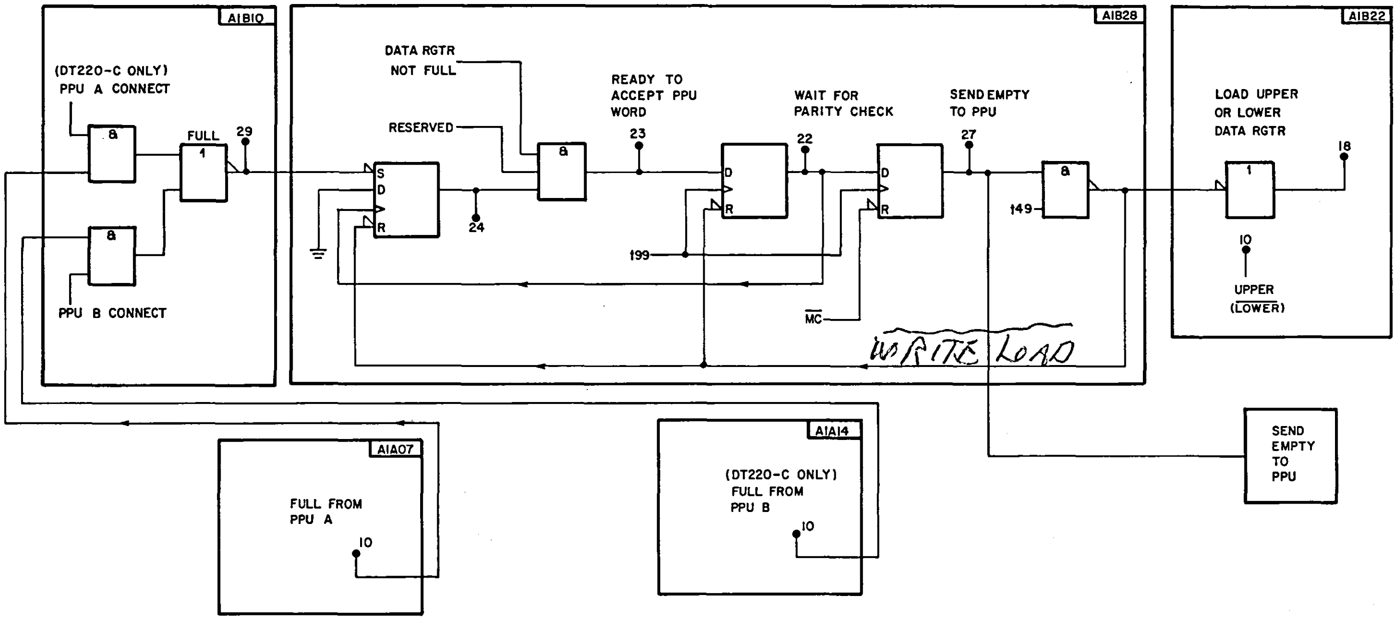


Figure 3-11. PPU/Coupler Activity During Autoload or PPU Write Operations

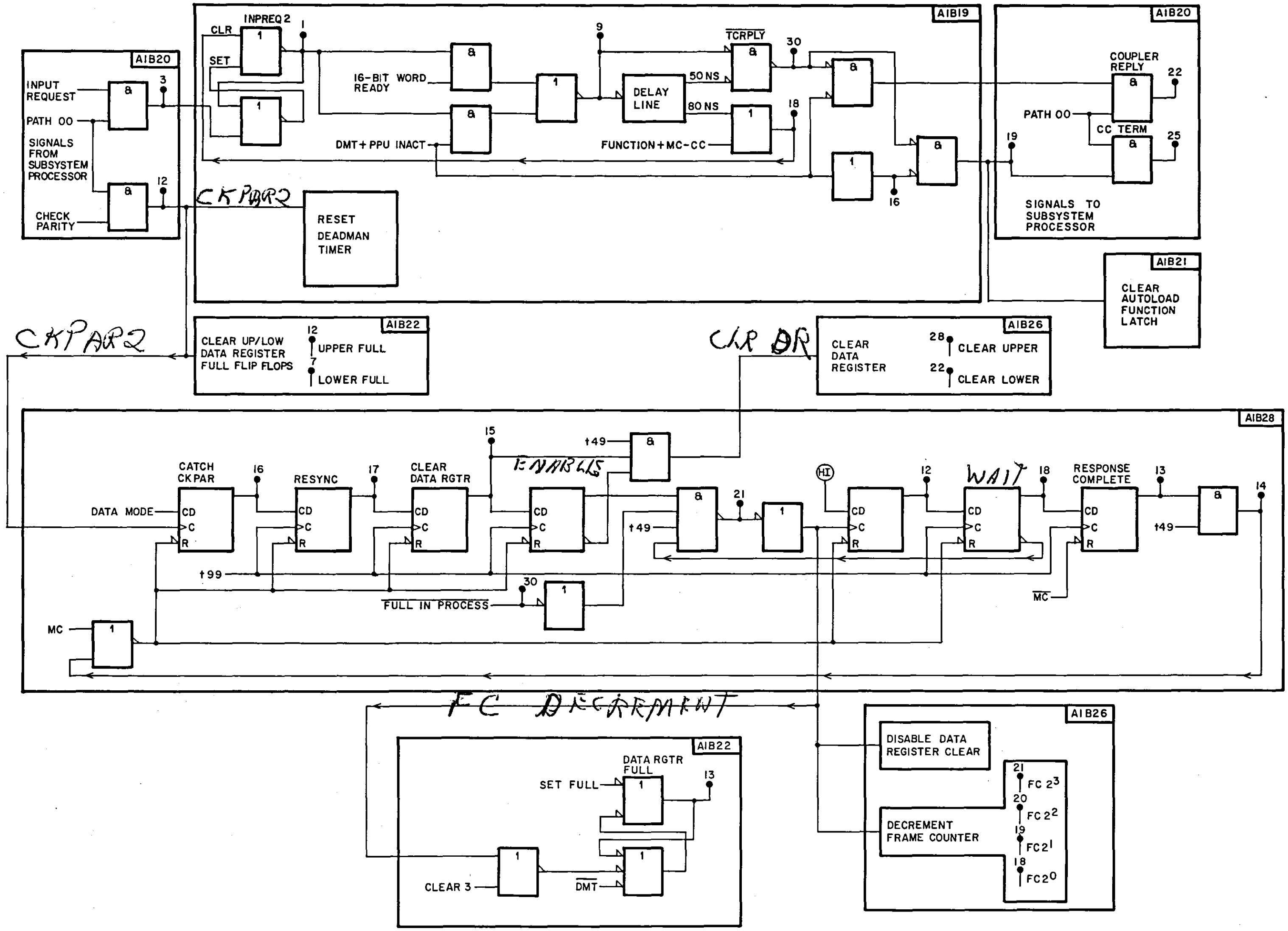


Figure 3-12. Subsystem Processor/Coupler Activity During Autoload

Go — Initiates subsystem processor execution of the forced input block transfer instruction. The subsystem processor immediately returns an input request signal to the coupler.

Coupler ready — This signal is active whenever the autoload function latch on card B21 is set. Coupler ready prevents a subsystem processor exit from the input block transfer until the coupler terminates the transfer with a CCTERM signal.

4. When the PPU detects the function reply from the coupler (inactive channel), it activates the channel and starts sending autoload data. Each 12-bit word is accompanied by a full signal. The data mode latch on card B22 sets when the PPU activates the channel while the coupler is reserved.

5. The coupler handles the PPU/subsystem processor transfer as follows:

   a. The PPU sends two 12-bit words, each word accompanied by a full signal.

   b. The coupler loads the first word in the upper 12 bits of the upper data register (card B12), returns an empty to the PPU, loads the second word in the upper 12 bits of the lower data register (card B11), and returns an empty to the PPU. Further empties to the PPU are inhibited until the data register becomes not full.

   c. The 00 data path select and the 0000 A/D format cause the lower eight bits of both 12-bit words in the data register to be combined into one 16-bit word. This 16-bit word emerges from the data multiplexer as CCI-00 through CCI-15 on cards B15 through B18.

   d. An input request from the subsystem processor causes the coupler to transmit the 16-bit word from the data multiplexer to the subsystem processor. This word is accompanied by a coupler reply signal from card B20.

   e. When the subsystem processor receives the 16-bit word, it returns a check parity signal to the coupler and then stores the word in memory.

   f. The check parity signal causes the coupler to reset the deadman timer, set the data register empty, clear the data register, and decrement the frame counter. Since format 0000 has a maximum frame count of

1, the frame counter returns to a count of 1 immediately after being decremented.

g. When the data register becomes not full, the coupler is free to accept the next two words of autoload data from the PPU. These two words are merged and transferred to the subsystem processor exactly the same as were the preceding two words.

6. After the last 12-bit word of autoload data has been transferred by the PPU to the coupler, the PPU inactivates the channel. This causes the coupler to send a CCTERM signal to the subsystem processor along with the last 16-bit word of autoload data.

7. The CCTERM signal from the coupler causes the subsystem processor to exit from the input block transfer instruction and continue instruction execution at memory address 0001.

Coupler Function Activity

Whenever it receives a function signal from the PPU, the coupler checks for equipment number 0 and legal function code. If the function is an autoload function (0414_8) or a status function (0012_8) to an unconnected access (dual access controllers only), the coupler responds to the function. These coupler responses are described under Autoload and Coupler Status Replies. The coupler passes all other legal function codes to the subsystem processor for decoding. The following sequence describes coupler activity upon receipt of one of these function codes from the PPU. Figure 3-13 illustrates function logic.

1. The PPU sends the function. When the controller is a dual access model, the coupler performs a connect sequence before passing the function signal to card B21. Single access couplers jumper the function signal directly to card B21.

2. Logic on card B21 receives the function signal and preclears the function in process latch. If the equipment number accompanying the function is 0, the function in process latch sets at the next t99.

3. The parity wait flip-flop delays the output of the function in process latch for an additional 50 nanoseconds to allow time for a parity check on the function code (CDC CYBER 170 only).

4. The output of the parity wait flip-flop sets the reserved latch and also feeds the status multiplexer on card B15. The subsystem processor can then send a status path code of 000 to examine function and reserved status on normal input channel 00.

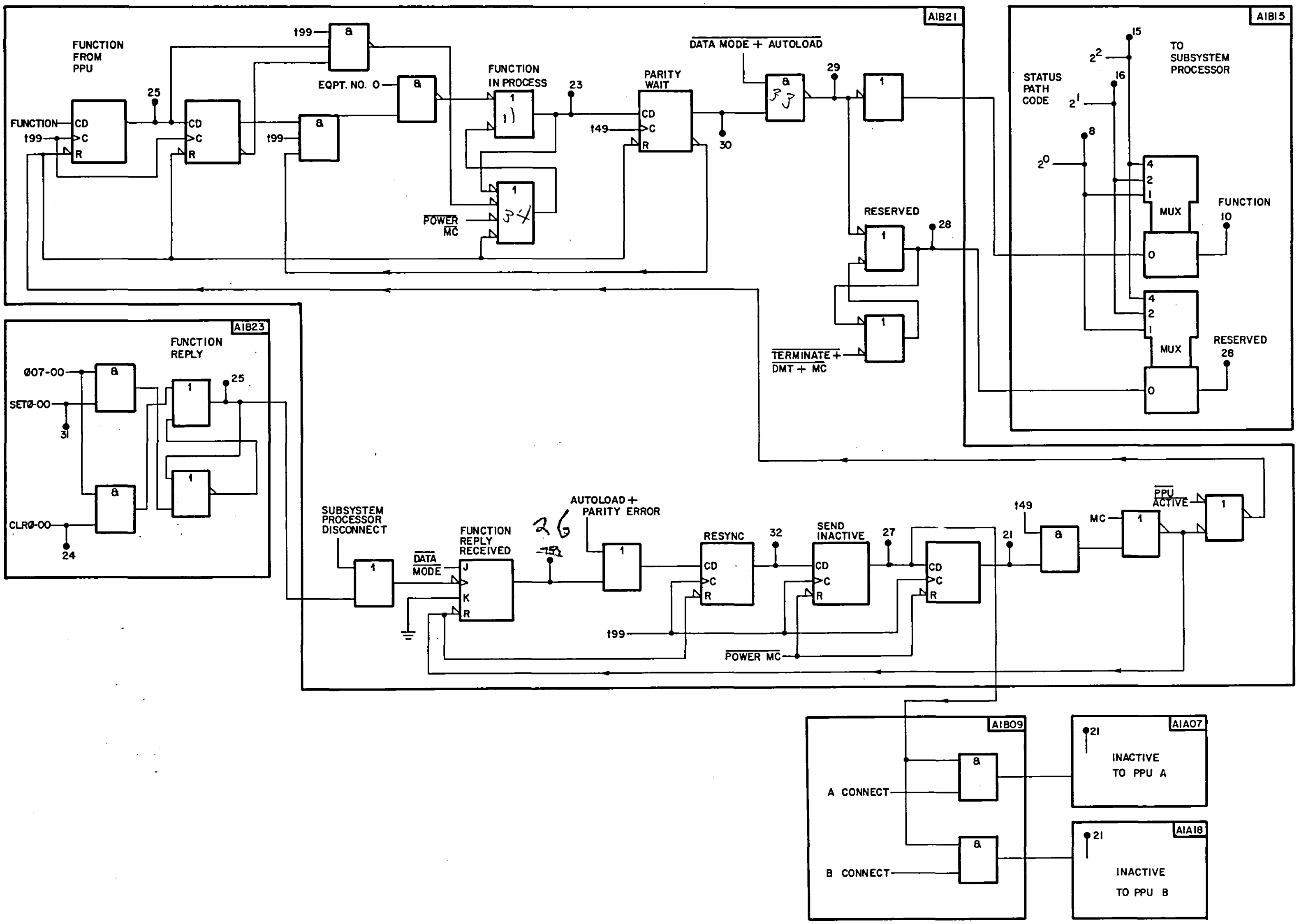


Figure 3-13. Function Logic

5. After it detects function status, the subsystem processor inputs the function code on normal input channel 03 and sets and then clears bit 0 of normal output channel 00. This toggles the function reply latch on card B23, which in turn sets the function reply received flip-flop on card B21. After being resynchronized, the output of the function reply received flip-flop sets the send inactive flip-flop. This enables an inactive signal to be sent to the connected PPU as a function reply.

6. At the next t49, all function logic except the reserved latch is cleared. The reserved latch remains set during the current operation to enable function parameter transfers and data transfers with the PPU. The reserved latch clears when the current operation terminates, a master clear occurs, or the deadman timer expires.

Coupler Status Replies

Dual access couplers process two kinds of status replies, subsystem processor generated status and coupler generated status. Single access couplers process only subsystem processor generated status. Sequences for each type of status are as follows:

Subsystem Processor Generated Status Sequence

Figure 3-14 illustrates coupler logic associated with this sequence.

1. The PPU issues a status request function. The coupler performs connect (dual access only) and function activity as previously described.

2. The subsystem processor inputs the status request function code, replies to the function by toggling bit 0 of normal output channel 00, selects the 00 (PPU to subsystem processor) data path, selects the appropriate A/D format, and initiates an output block transfer of appropriate word length.

3. After the PPU receives the function reply (inactive channel), it activates the channel and initiates a block input instruction. The coupler transfers status words from the subsystem processor to the PPU as follows:

    a. The active channel from the PPU raises data mode which combines with read format and data register not full signals to generate CREADY. CREADY enables the subsystem processor to initiate the output block transfer instruction.

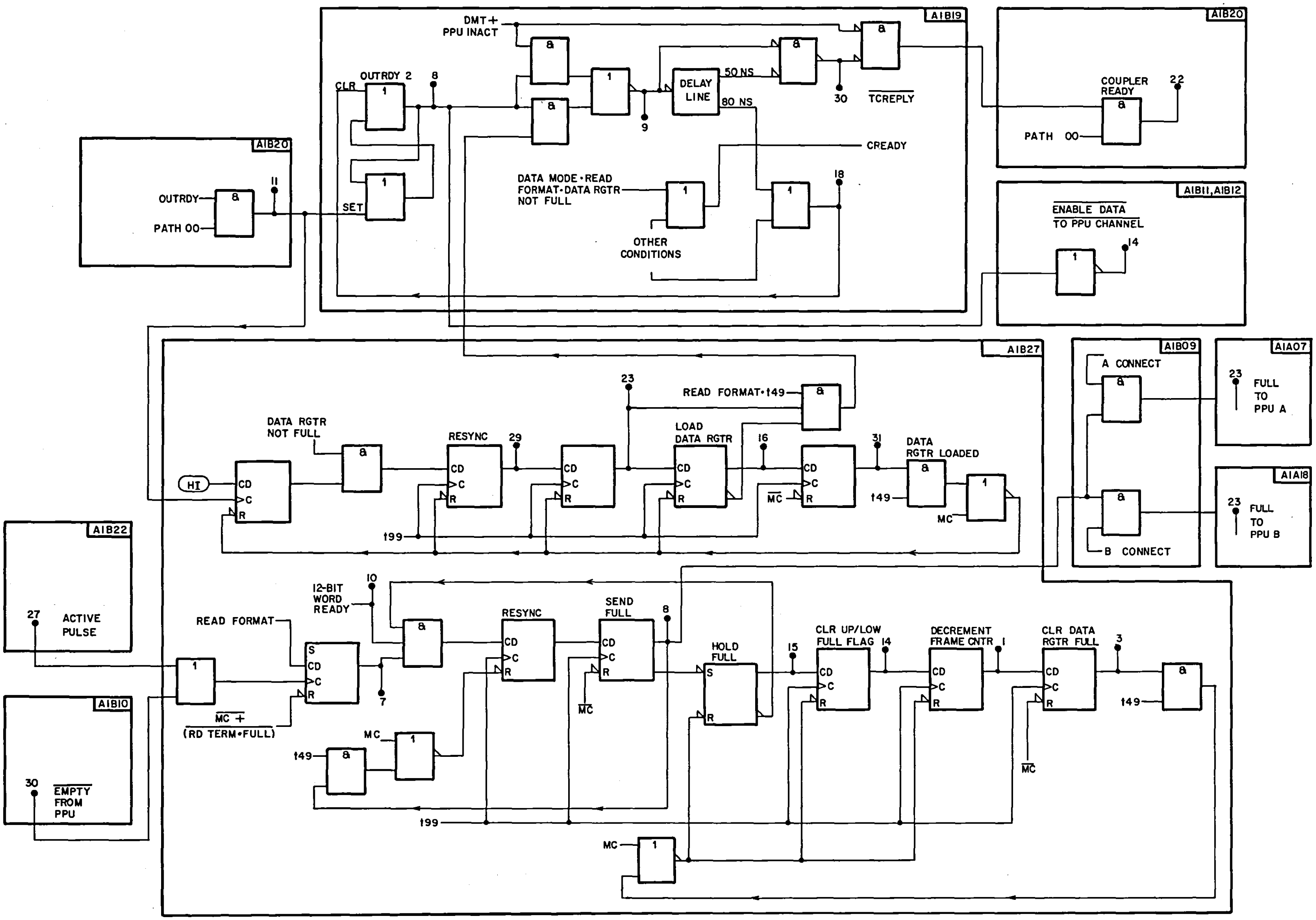


Figure 3-14. Subsystem Processor Generated Status Reply

b. The subsystem processor reads a 16-bit status word from core memory, places it on the CCO-xx lines, and sends an OUTRDY pulse to the coupler.

c. When the coupler receives the OUTRDY pulse, it sets the OUTRDY2 latch and responds with a CREPLY pulse to the subsystem processor, providing the data register is able to accept the status word.

d. The status word passes through the data multiplexer and A/D section of the coupler according to the A/D format and path select established by the subsystem processor before the transfer started. Logic on card B27 determines when a 12-bit word is ready for transfer to the PPU.

e. Either an active pulse (first word) or an empty signal (remaining words, if any) from the PPU initiates the remaining transfer activity for each word. The coupler places the 12-bit word on the data channel and sends a full signal to the PPU. Then the coupler clears the appropriate upper/lower full flags, decrements the frame count, and clears the data register full flip-flop.

4. When the subsystem processor completes the output block transfer and the PPU has accepted the last 12-bit word and returned an empty signal, the coupler inactivates the channel to the PPU. This terminates the status operation.

Coupler Generated Status Sequence

Dual access couplers provide a limited status capability to inform a requesting PPU whether or not the opposite access is connected. The coupler generates status for a PPU only when the opposite access is connected. Otherwise the coupler passes the status request to the subsystem processor for processing.

Since the coupler provides only the reserved status bit (bit 2^{10}) to the PPU, all other status bits are meaningless for status replies generated by the coupler.

Figure 3-15 illustrates logic associated with coupler status generation for access B. The sequence proceeds as follows: assume that the coupler is connected to PPU A when PPU B sends a status request (0012_8) function.

1. The status request function from PPU B sets the B function latch on card B08. The output of this latch combines with the 0012_8 decode and the A connect signal to enable the J input of the B status sequence flip-flop on card B09.

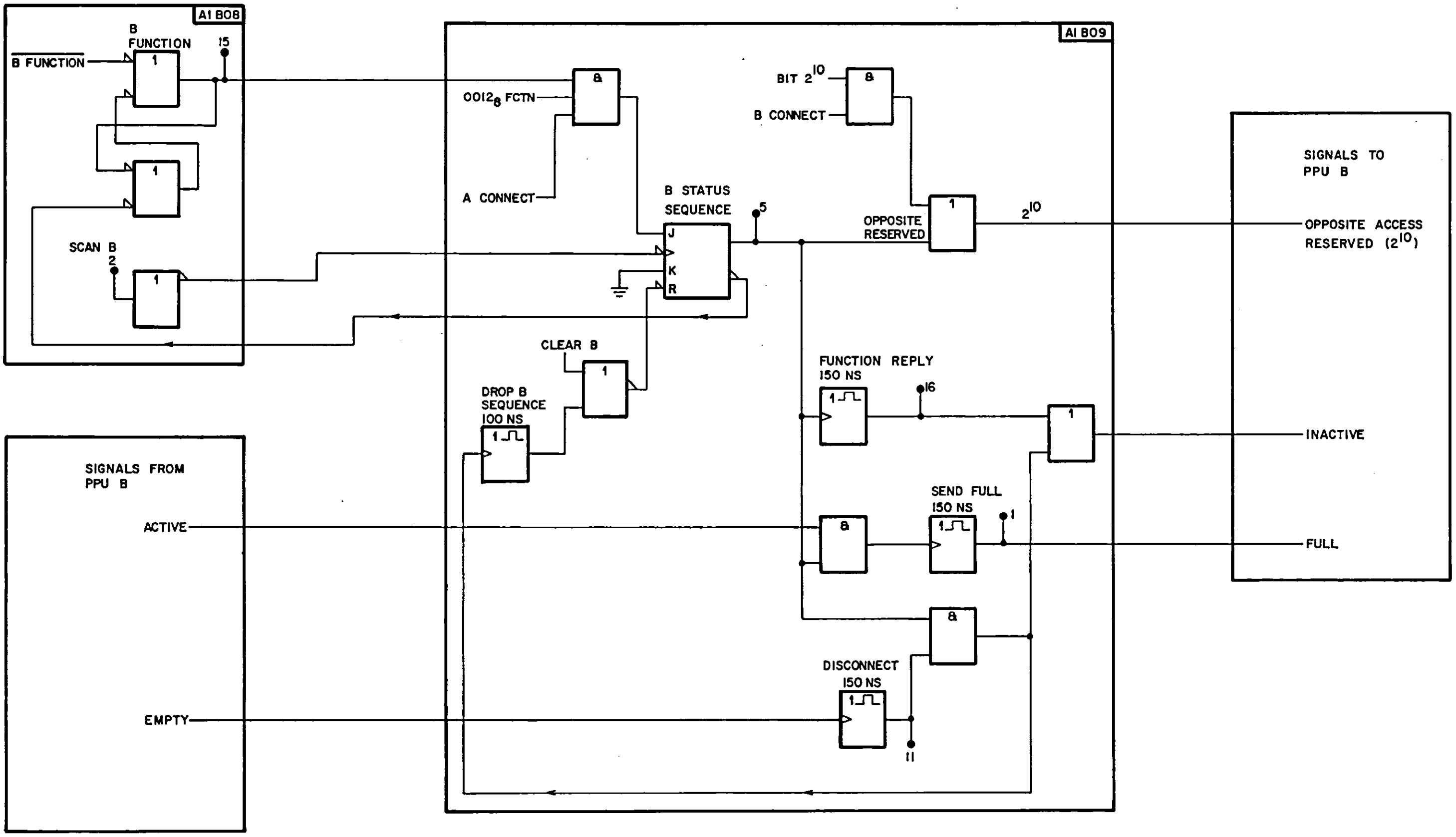


Figure 3-15. Coupler Generated Status Reply

2. The next low transition of the scan B signal sets the B status sequence flip-flop. The following activity occurs when the B status sequence flip-flop sets.

   a. The opposite access reserved bit (bit 2^{10}) becomes active. This bit is taken directly from the B status sequence flip-flop and routed to the PPU B interface as bit 2^{10}.

   b. The function reply one-shot fires to reply to the status request function by inactivating the channel.

   c. When PPU B activates the channel, the send full one-shot fires to provide a full signal which accompanies the 12-bit PPU word containing the opposite access reserved bit.

   d. The empty response from PPU B triggers the disconnect one-shot. This causes the coupler to terminate the function with an inactive signal. The output of the disconnect one-shot also triggers the drop B sequence one-shot which clears the B status sequence flip-flop.

3. When PPU B is connected and PPU A sends a status request function, A status sequence logic performs just like the already described B sequence logic.

PPU Write Operation

During a PPU write operation, the coupler transfers data from a PPU to the control logic. The connect and function decoding portions of a PPU write operation proceed as described in previous sequences. Before it sends a function reply to the PPU, the subsystem processor conditions the control logic for a write operation, selects an assembly format, and specifies data path 01 (control logic/PPU) in the coupler. Figure 3-11 illustrates coupler logic associated with PPU to coupler transfers. Figure 3-16 illustrates coupler logic associated with coupler to control logic transfers. The PPU write operation proceeds as follows:

1. After it has received a function reply from the subsystem processor and transferred function parameters (if any), the PPU activates the channel and starts sending write data. Each 12-bit word is accompanied by a full signal.

2. The coupler loads the write data into the data register on cards B11 and B12 according to the specified assembly format and the current frame count. After each 12-bit word is loaded, the coupler returns an empty signal to the PPU. When the data register becomes full, further empties are inhibited until the control logic removes one or more 16-bit words from the data register via the write multiplexer.

FIRST BYTE OF DATA WRITE
REST OF DATA [illegible] CIRPLY

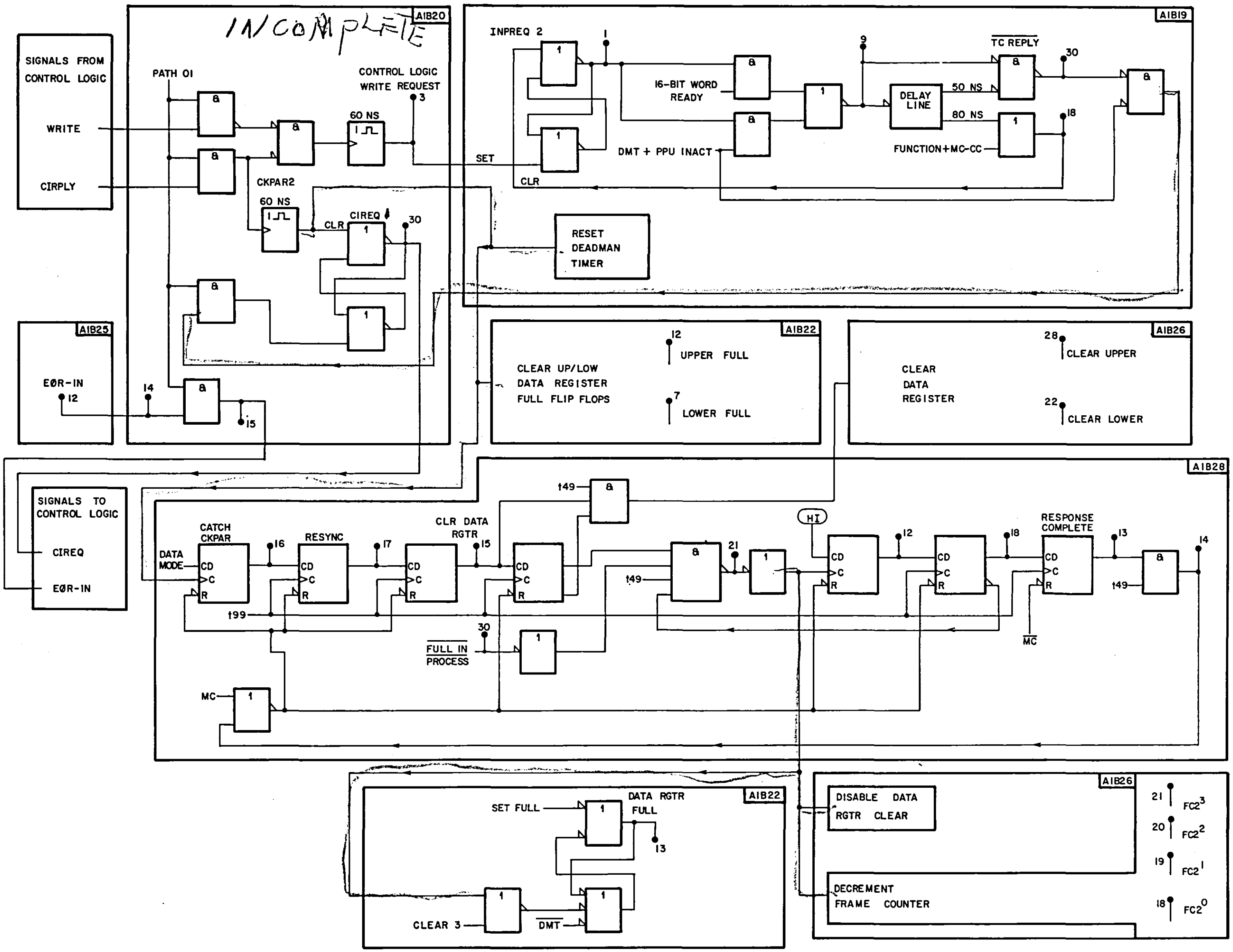


Figure 3-16. Coupler to Control Logic Transfers

3. When the subsystem processor conditioned the control logic for a write operation, the control logic returned a static write signal to the coupler. The leading edge of this signal then triggered the control logic write request one-shot which set the INPREQ2 latch on card B19.

4. When a 16-bit word becomes available from the write multiplexer on card B14, the delay line circuit on card B19 sets the CIREQ1 latch on card B20. The output of this latch informs the control logic that the data on the CIB-xx lines (cards B15 through B18) may be sampled.

5. After the control logic has accepted data from the CIB-xx lines, it returns a CIRPLY signal to card B20 in the coupler. The CIRPLY signal triggers the following activity in the coupler.

   a. Fires the CKPAR2 one-shot to clear the CIREQ1 latch on card B20 and generate CKPAR2. CKPAR2 causes the coupler to reset the deadman timer, set the data register not full, enable clearing of the data register, and decrement the frame count.

   b. Fires the control logic write request one-shot which sets the INPREQ2 latch on card B19.

6. When the data register becomes not full, the coupler is free to return an empty signal to the PPU and accept the resulting write data. The coupler transfers the remaining portion of the write data as described previously.

7. The write operation terminates when the PPU inactivates the channel after the last 12-bit word has been sent. The inactive signal generates an EOR-IN pulse on card B25 after the last 16-bit word has been accepted by the control logic. The EOR-IN pulse is gated to the control logic on card B20.

PPU Read Operation

During a PPU read operation, the coupler transfers data from the control logic to a PPU. The connect and function decoding portions of a PPU read operation proceed as described in previous sequences. Before it sends a function reply to the PPU, the subsystem processor conditions the control logic for a read operation, selects a disassembly format, and specifies data path 01 (control logic/PPU) in the coupler. Figure 3-17 illustrates coupler logic associated with control logic to coupler transfers. Figure 3-18 illustrates coupler logic associated with coupler to PPU transfers. The PPU read operation proceeds as follows:

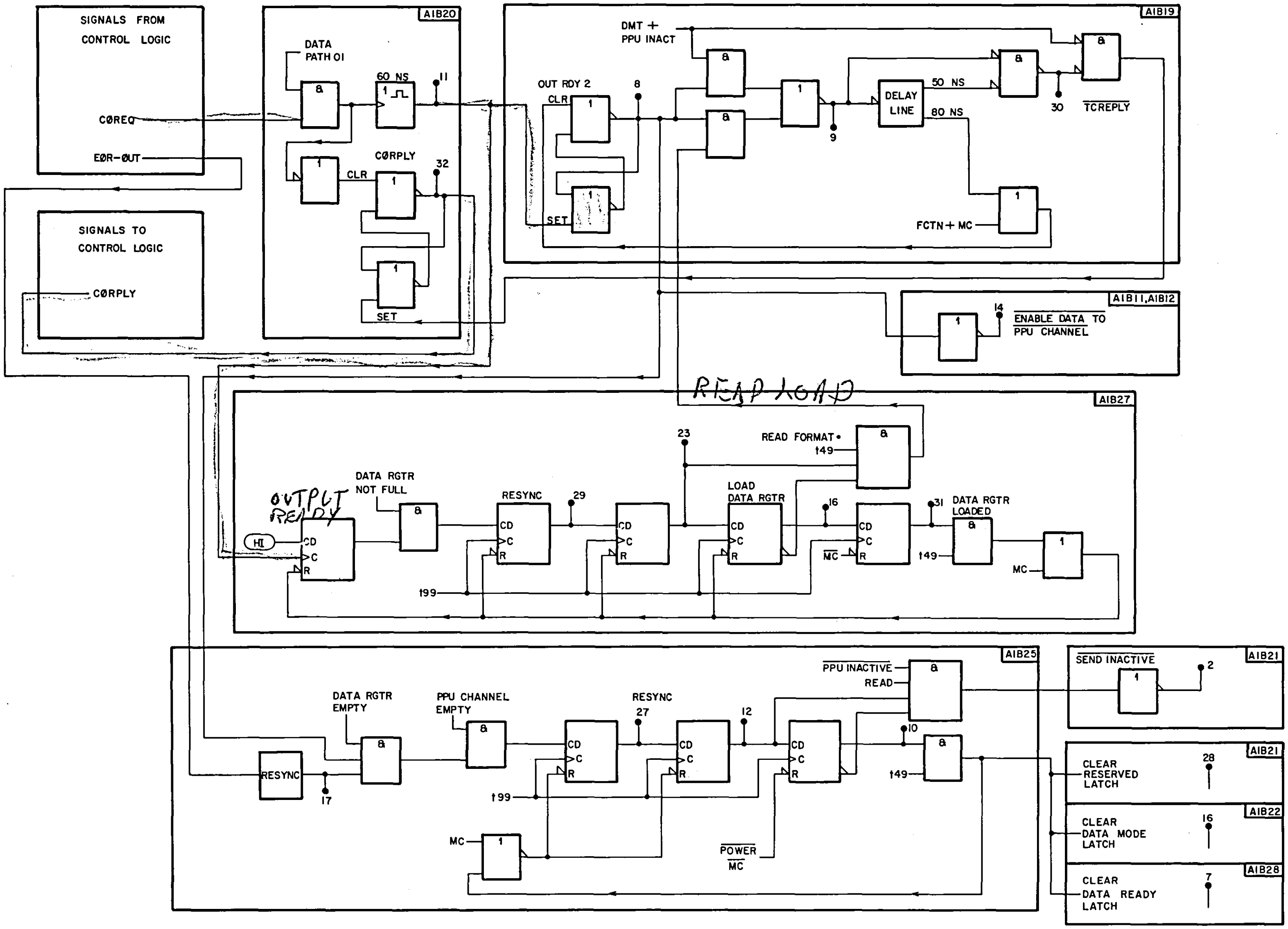


Figure 3-17. Control Logic to Coupler Transfers

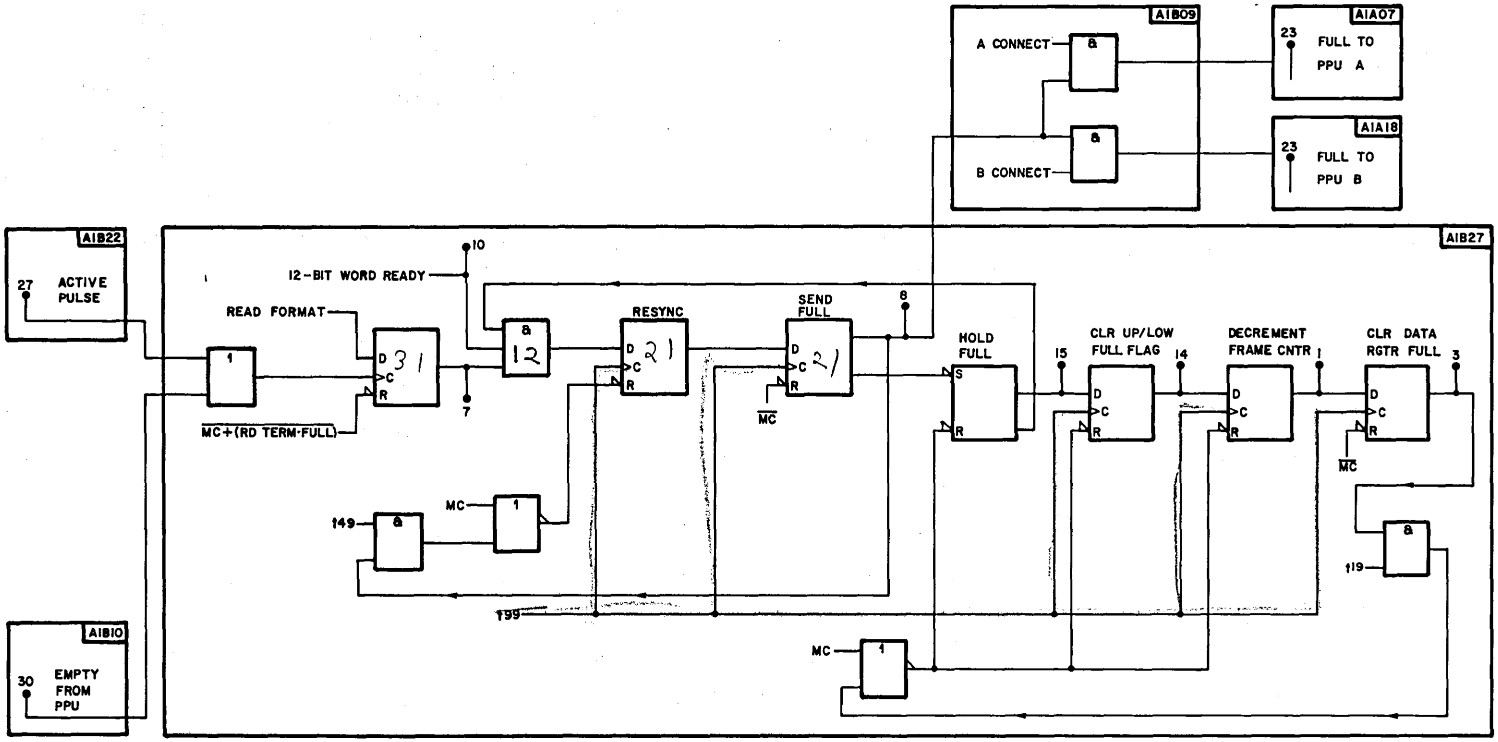


Figure 3-18. Coupler to PPU Transfers

1. After it has received a function reply from the subsystem processor and transferred function parameters (if any), the PPU activates the channel and waits for the channel to go full.

2. When a 16-bit read data word is ready for transfer, the control logic places the word on the COB-xx lines and sends a COREQ signal. The COREQ signal clears the CORPLY latch on card B20 and then triggers a one-shot which sets the OUTRDY2 latch on card B19. Logic on card B27 determines when a 12-bit word is ready for transfer to the PPU.

3. If the data register is able to accept the word from the control logic, the coupler responds with a CORPLY signal. The data word from the control logic passes through the data multiplexer and A/D section of the coupler according to the A/D format and path select established by the subsystem processor before the transfer started.

4. Either an active pulse (first word) or an empty signal (remaining words) from the PPU initiates the remaining transfer activity for each word. The coupler places the 12-bit word on the channel and sends a full signal to the PPU. Then the coupler clears the appropriate upper/lower full flags, decrements the frame count, and clears the data register full flip-flop.

5. The coupler sends an inactive to the PPU after the control logic has sent an EOR-OUT signal and the PPU has accepted the last 12-bit word and returned an empty signal. This terminates the read operation.

Controller Internal Transfers

The coupler supports a data path which connects the block transfer interface of the subsystem processor to the control logic. This data path is used by some sections of the controller diagnostic to transfer data from the core memory to a disk storage unit. The subsystem processor/control logic data path is not used during normal controller operations.

Before a subsystem processor/control logic transfer, the subsystem processor selects the 10_2 data path in the coupler. Since the data to be transferred does not pass through the A/D section of the coupler, an A/D format is not selected.

Subsystem Processor to Control Logic Transfer

1. The subsystem processor conditions the control logic for a write operation, places a 16-bit word on the CCO-xx lines, and sends an OUTRDY signal to the coupler.
2. The coupler routes the word through the data multiplexer (on cards B15 through B18), places the word on the CIB-xx lines, and sends a CIREQ signal to the control logic.
3. When the control logic has accepted the data, it returns a CIRPLY signal to the coupler, which in turn sends a CREPLY signal to the subsystem processor.
4. The previous sequence repeats until the subsystem processor sends an EOP-CC signal to the coupler. This causes the coupler to send an EOR-IN signal to the control logic, thereby terminating the transfer.

Control Logic to Subsystem Processor Transfer

1. The subsystem processor conditions the control logic for a read operation and sends an INPREQ signal to the coupler.
2. The control logic places a 16-bit word on the COB-xx lines and sends a COREQ signal to the coupler.
3. The coupler routes the word through the data multiplexer, places the word on the CCI-xx lines, and sends a CREPLY signal to the subsystem processor.
4. When the subsystem processor has accepted the data, it returns a CKPAR signal to the coupler, which in turn sends a CORPLY to the control logic.
5. The previous sequence repeats until the control logic sends an EOR-OUT signal to the coupler. This causes the coupler to send a CCTERM signal to the subsystem processor, thereby terminating the transfer.

SUBSYSTEM PROCESSOR/CORE MEMORY THEORY

The subsystem processor is the major control element in the disk controller. It receives function codes from the coupler, initiates control logic activity, and provides status to the coupler. Controlware stored in the core memory contains the 16-bit instructions, operands, and addresses used by the subsystem processor.

COMPONENT THEORY

Like most processors, the subsystem processor/core memory can be divided into four sections: control, arithmetic, input/output, and storage. The control section processes instructions, and the arithmetic section performs two's complement arithmetic and logical operations. The input/output section provides communication over several input/output channels and a block transfer channel. The subsystem processor's storage section is the core memory.

Figure 3-19 illustrates data flow between the functional elements of the subsystem processor/core memory.

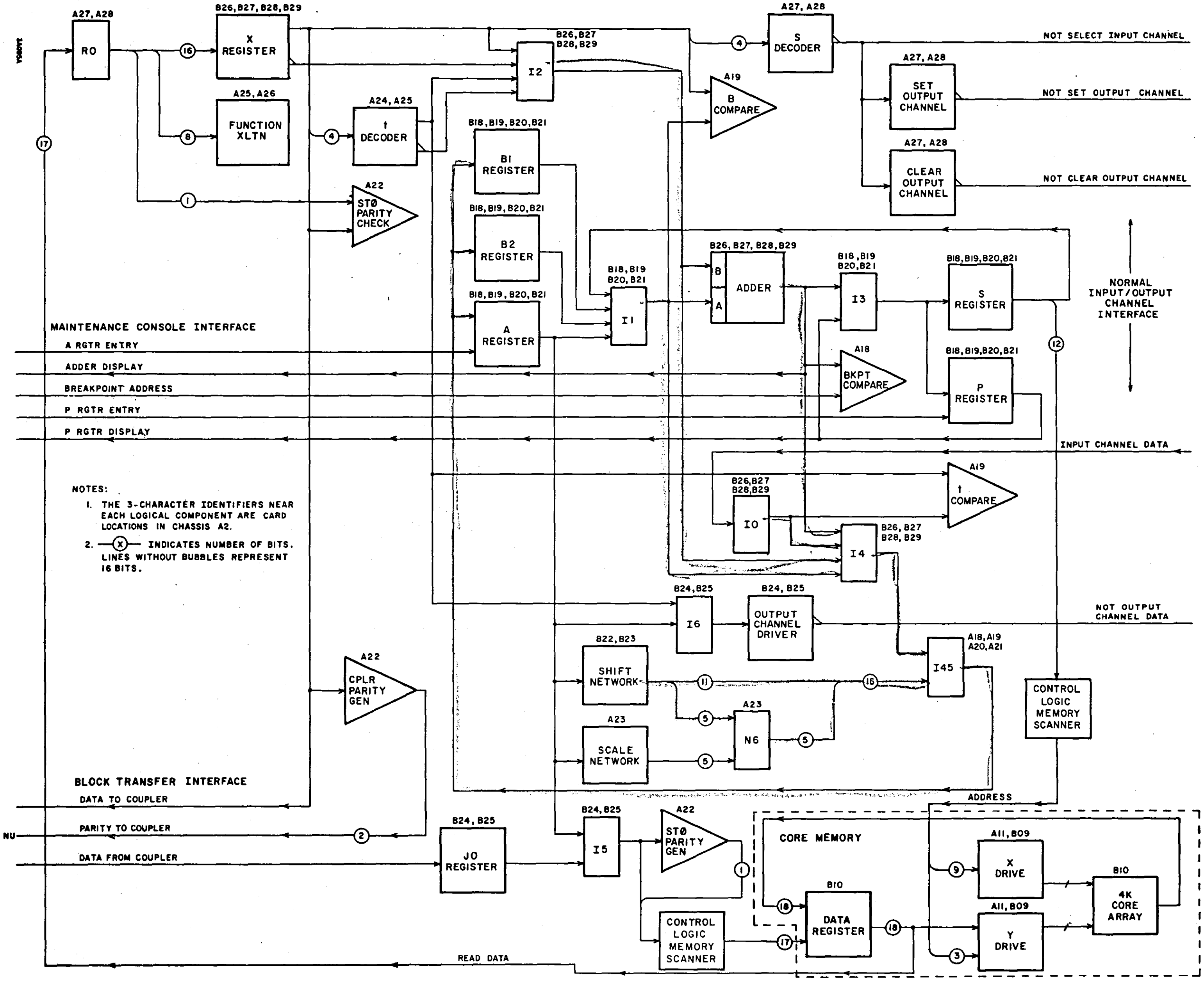
Control Section

Timing Chain

The timing chain consists of an 11-stage rank of flip-flops tied to a crystal oscillator. A go signal applied to stage 0 initiates generation of clock pulses which synchronize subsystem processor activity.

An interrupt circuit between stages 5 and 6 permits the control logic to disable the timing chain during control logic data or director memory accesses. When a subsystem processor memory request is tied up in the control logic memory scanner, the interrupt circuit stops the timing chain. As soon as the memory scanner passes the subsystem processor request to memory and memory data becomes available, the timing chain restarts at stage 6. The subsystem processor thus runs asynchronously, with processing speed dependent upon control logic memory activity.

SUBSYSTEM PROCESSOR/CORE MEMORY DATA FLOW



NOTES:

1. THE 3-CHARACTER IDENTIFIERS NEAR EACH LOGICAL COMPONENT ARE CARD LOCATIONS IN CHASSIS A2.
2. —(X)— INDICATES NUMBER OF BITS. LINES WITHOUT BUBBLES REPRESENT 16 BITS.

Figure 3-19. Subsystem Processor/Core Memory Data Flow

Function Translator

The function translator consists of an 8-bit function register and a decoding network. The function register normally receives function codes from core memory, although the coupler can force block transfer function codes into the function register. After a function code is received by the function register, the decoding network translates the function code into appropriate enabling signals and determines the addressing mode.

Internal Cycle Control

A set of status flip-flops controls the routing of all words read from core memory.

Status Flip-flop	Function
Read next instruction (RNI) status	Causes the subsystem processor to treat the next word from memory as an instruction.
Read address (RADR) status	Causes the subsystem processor to treat the next word from memory as an address.
Read operand (ROP) status	Causes the subsystem processor to treat the next word from memory as an operand.
Store (STO) status	Causes the subsystem processor to write a word at the execution address.

Instruction Flags

The control section has two hardware flags which are acted upon by certain instructions. The condition flip-flop can indicate the result of an internal test or reflect the state of a bit on an input channel. The endoff flip-flop records adder overflow during execution of add or subtract instructions.

Arithmetic Section

A (Accumulator) Register

This is the main arithmetic register. The initial contents of A is often an operand for an arithmetic or logical operation, and the final contents is the result of the operation. The contents of A can be examined and modified from a maintenance console.

P (Program Address) Register

This register contains the core memory address of the instruction to be executed. During program execution, the subsystem processor transfers the contents of P to the S register and then increments P. When the subsystem processor is stopped, the address in P is always one greater than the address of the current instruction. The contents of both P and S can be examined and modified from a maintenance console.

S (Memory Address) Register

This register is loaded during each subsystem processor cycle. The S register provides the core memory with the address of the word to be read or written.

B1, B2 (Index) Registers

These registers store numbers used for compare tests and for address formation. The contents of either register can be examined from a maintenance console.

X (Exchange) Register

This register is an integral part of both the arithmetic and input/output sections. All words from memory pass through the X register, which holds these words in one's complement form. The contents of X can be examined from a maintenance console in either true or one's complement form.

Adder

The 16-bit adder is made up of four 4-bit binary adder chips. An incrementer circuit generates a carry input to the least significant bit of the adder. This carry input converts one's complement numbers to two's complement and also serves to increment or decrement two's complement numbers. A carry output from the most significant bit of the adder feeds the endoff flip-flop.

Shift Network

The shift network performs endoff and circular right shifts of from 0 to 15 bit positions on A register numbers.

Scale Network

The scale network forms a 5-bit count of the leading 0's in the A register.

Logical Operation Gates

The arithmetic section contains a logical product gate and an exclusive OR gate for performing logical operations on two 8-bit or two 16-bit operands.

Input/Output Section

NOTE

1. The disk controller hardware reference manual listed in the preface describes the use of the input/output channels and the block transfer channel.
2. The subsystem processor has no registers for storing input/output channel data. These registers are provided by the coupler and the control logic. The subsystem processor provides only input/output control signals and output channel data signals.

Input Channels

The input/output section has provision for up to 16 16-bit input channels but uses only five channels. There are two input channel related instructions. The 09XX instruction causes the condition flip-flop to assume the state of a selected bit on a selected input channel. The 0CX0 instruction transfers 16 bits from a selected input channel to the A register.

Output Channels

The input/output section has provision for up to 16 16-bit output channels but uses only six channels. Output channel 0 has more flexibility than the other output channels, that is, it is able to handle selective bit changes as well as six types of data transfers from the A register. The other output channels can accommodate single-bit selective set operations and two types of data transfers from the A register.

Block Transfer Channel

This 16-bit bidirectional channel is used for data transfers between consecutive core memory locations and the coupler. Transfers can be initiated by instruction or forced from the coupler.

During output block transfers, data is read from core memory, placed in the X register, and then sent to the coupler. During input block transfers, data from the coupler is buffered in the J0 register before being written in core memory.

Storage Section

NOTE

> This manual does not contain operational theory for the core memory or the power regulator module, since these assemblies are not field repairable. If more detailed information is required, refer to the BB372-A customer engineering manual listed in the preface.

Core Memory

This 2-1/2 dimension memory provides read and write capability for 4096 18-bit words, although the subsystem processor and control logic use only 17 bits of each word (16 data bits and one parity bit). The core memory has a read access time of about 350 nanoseconds and a total cycle time of about 750 nanoseconds. As illustrated in Figure 3-19, core memory functional elements include a data register, X and Y drive circuits, and a core array.

Figure 3-20 illustrates the relationship of signals between the core memory and control logic/subsystem processor for a single memory cycle. Refer to Control Logic/Subsystem Processor Interface in this section for descriptions of these signals.

Memory address, data, parity, and store signals from control logic to memory vary according to whether the memory request is a subsystem processor, data, or director request. The signals shown in Figure 3-20 are worst case with respect to the memory.

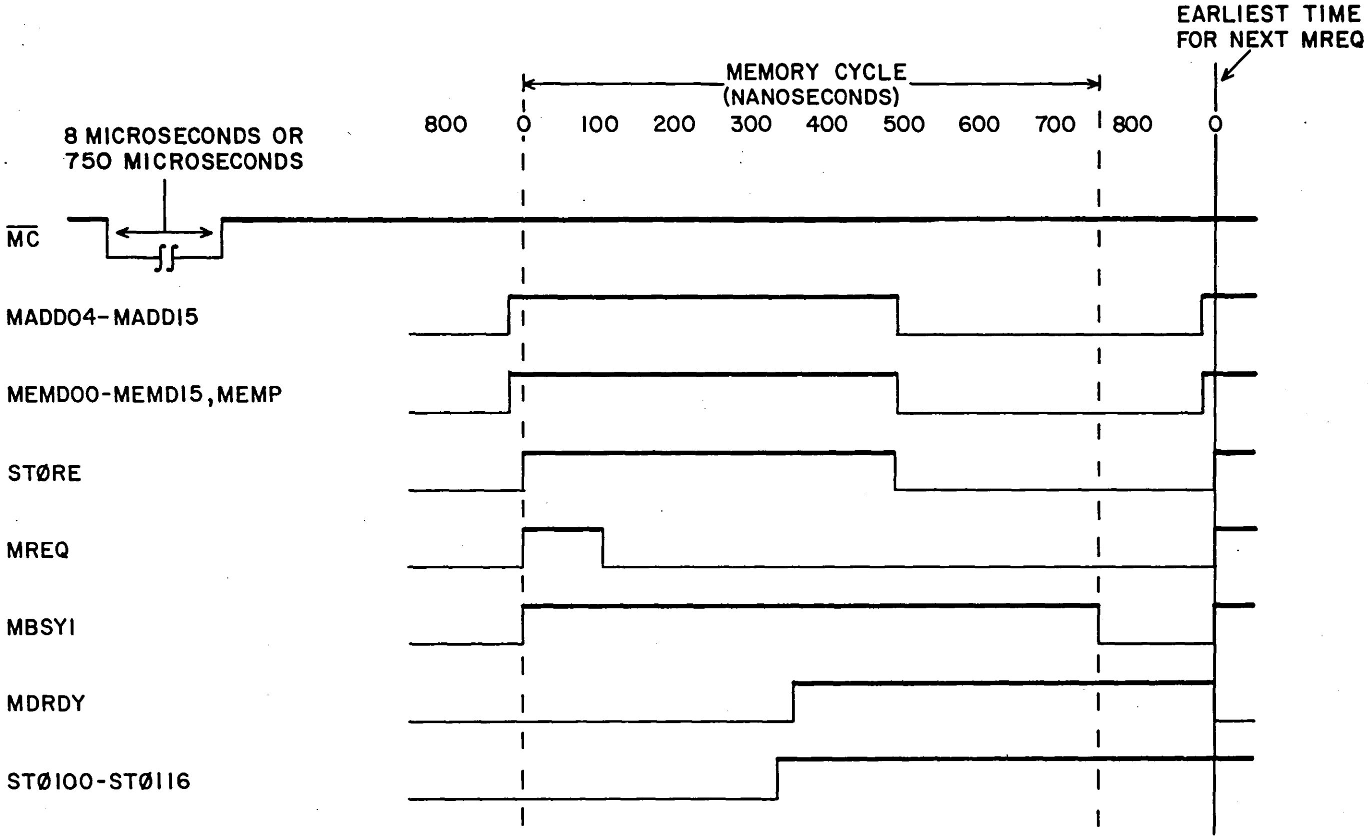


Figure 3-20. Memory Interface Timing

Power Regulator Module

The power regulator module regulates the current and voltage supplied to the core memory. It also turns off all power to the memory when a current overload occurs. After the current overload condition has been eliminated, the power regulator module is restored to normal operation by pressing the MC switch on the subsystem processor auxiliary operator panel.

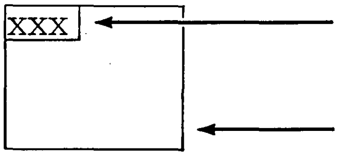
SUBSYSTEM PROCESSOR INSTRUCTION SEQUENCES

This part describes the four types of subsystem processor internal cycles and then provides control sequences for subsystem processor instructions. All card locations referenced in this part are in chassis A2.

Figure 3-19 illustrates the data flow mentioned in the instruction sequences.

Instruction Figures

The figures accompanying most of the sequences provide card location and test point information for major signals associated with an instruction. The logic depicted by these figures has been simplified to highlight the instruction under study. If in doubt about the function of a section of logic, refer to the appropriate logic diagrams in section 6.



Card location(s); multiple numbers indicate multiple locations (B18-B21 means B18, B19, B20, B21).

Box encloses logic elements on the card(s) indicated.

1	OR function
&	AND function
=, =1	Exclusive OR function
D	Data input for CD flip-flop
C	Clock input for CD flip-flop
◺	Low polarity active indicator (0 is active state of signal).
● x	Test point; x=test point number. A not bar over the test point number indicates the signal observed is the complement of the signal called out.

Figure 3-21. Key to Instruction Sequence Symbols

Internal Cycles

Figure 3-22 illustrates the major signals associated with the four subsystem processor internal cycles. The numerals I through IV indicate the starting point for each cycle. The solid vertical lines indicate the points at which the subsystem processor may be interrupted while waiting for a word from memory.

The four cycles are shown as they would occur for a subsystem processor instruction requiring four memory references, for example, a replace add instruction using indirect addressing (84XX). In the replace add case, the RNI cycle (I) reads the 84XX instruction from memory and generates the address of the execution address, the RAD (II) cycle reads the address from memory, the ROP (III) cycle reads the operand from the execution address and performs the add, and the STO (IV) cycle writes the result of the add at the execution address.

RNI Cycle

After a master clear, RNI status is active to enable the next word from memory into the F register (A26) for function translation and into the lower half of the X register (B26, B27) for immediate use or for subsequent address formation. RNI status drops if the current instruction requires an RAD, ROP, or STO cycle, and remains down until the required cycles are complete.

RAD Cycle

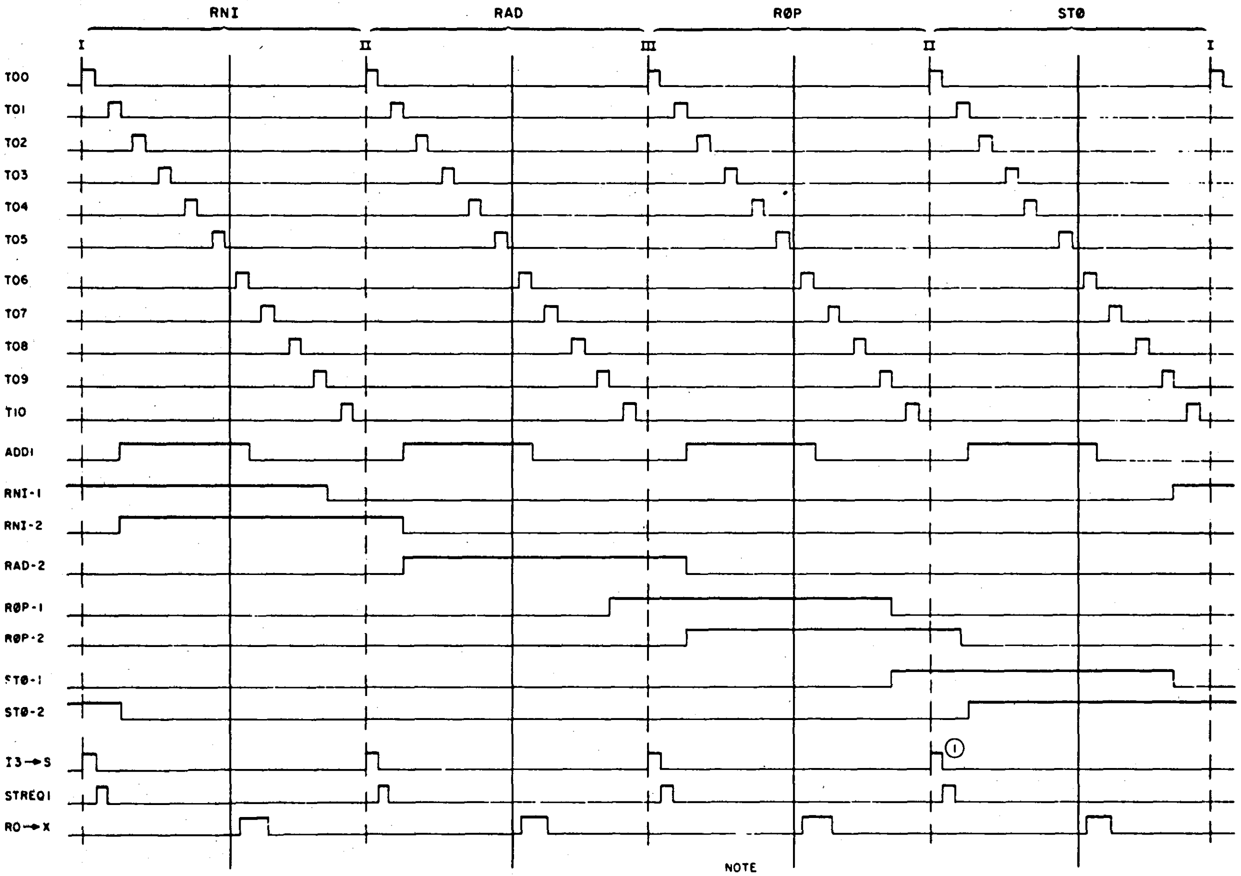
This cycle occurs for all format 2 instructions requiring indirect addressing. During the RAD cycle, the execution address (M) is obtained from the memory and loaded into a subsystem processor register for use either in the current instruction or in later instructions.

ROP Cycle

This cycle occurs for the load from (A) instruction, for instructions requiring arithmetic or logical operands from memory, and for output block transfers from memory. ROP status remains active during the subsystem processor cycle performing the load from (A), arithmetic, or logical operation. During output block transfers, ROP status remains active until the transfer terminates.

STO Cycle

This cycle occurs for instructions requiring words to be written into memory. For single-word writes, STO status remains active for one subsystem processor cycle. During input block transfer instructions, STO status remains active until the transfer terminates.



NOTE
① THIS PULSE IS DISABLED FOR REPLACE (80XX-9FXX) INSTRUCTIONS

Figure 3-22. Subsystem Processor Internal Cycles

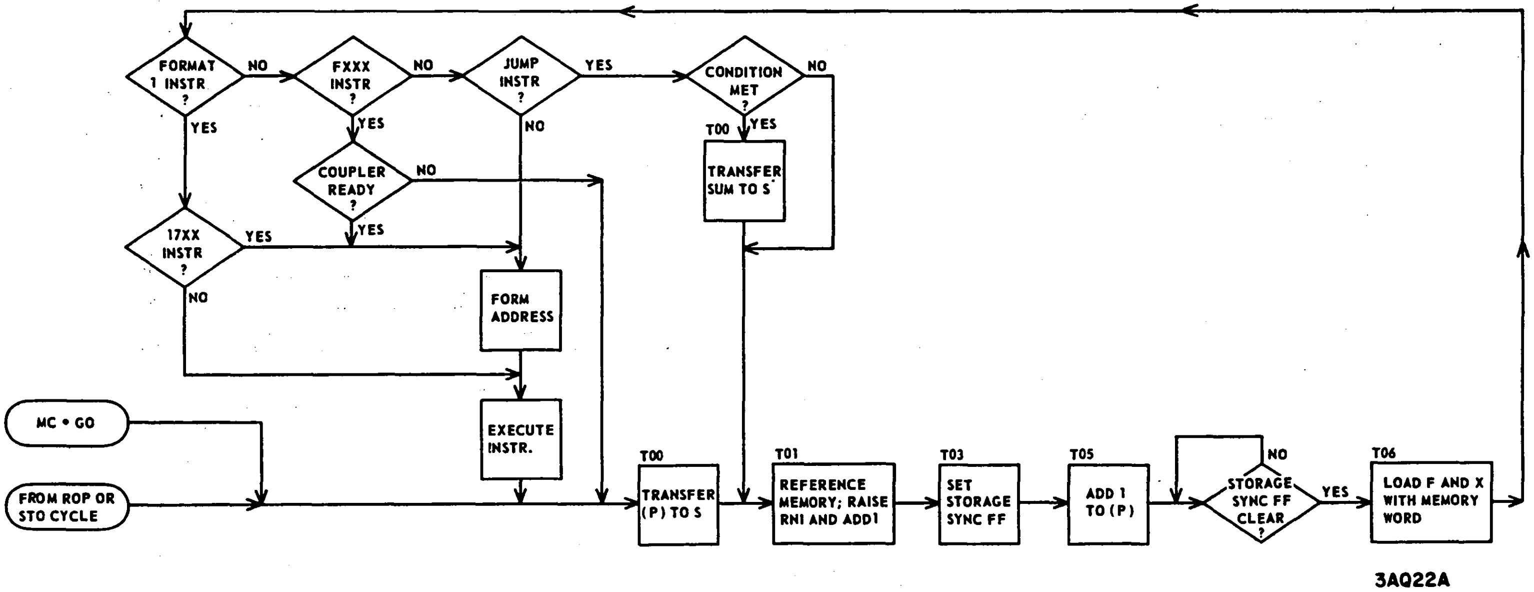


Figure 3-23. Common RNI Activity

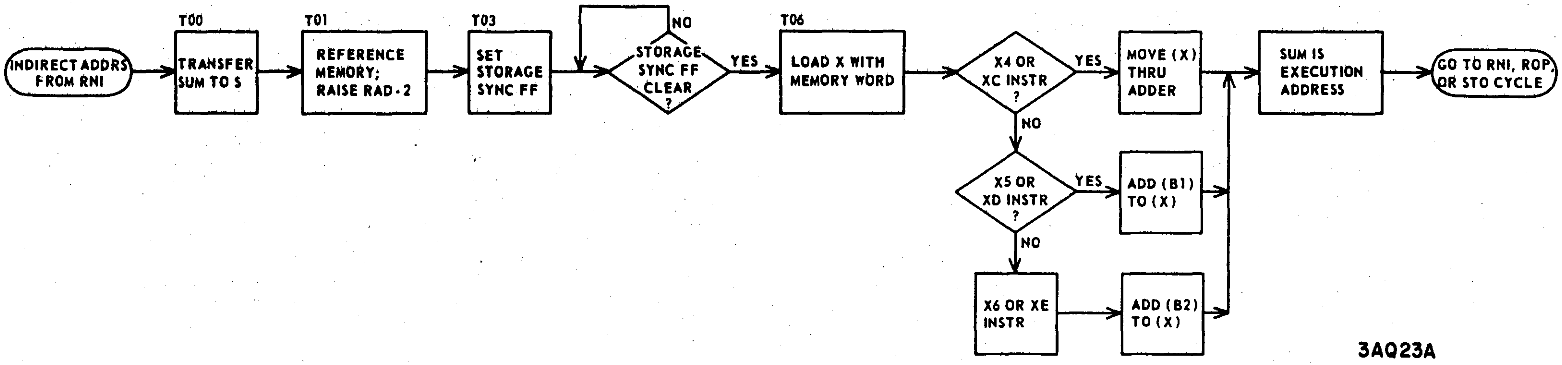


Figure 3-24. Common RAD Activity

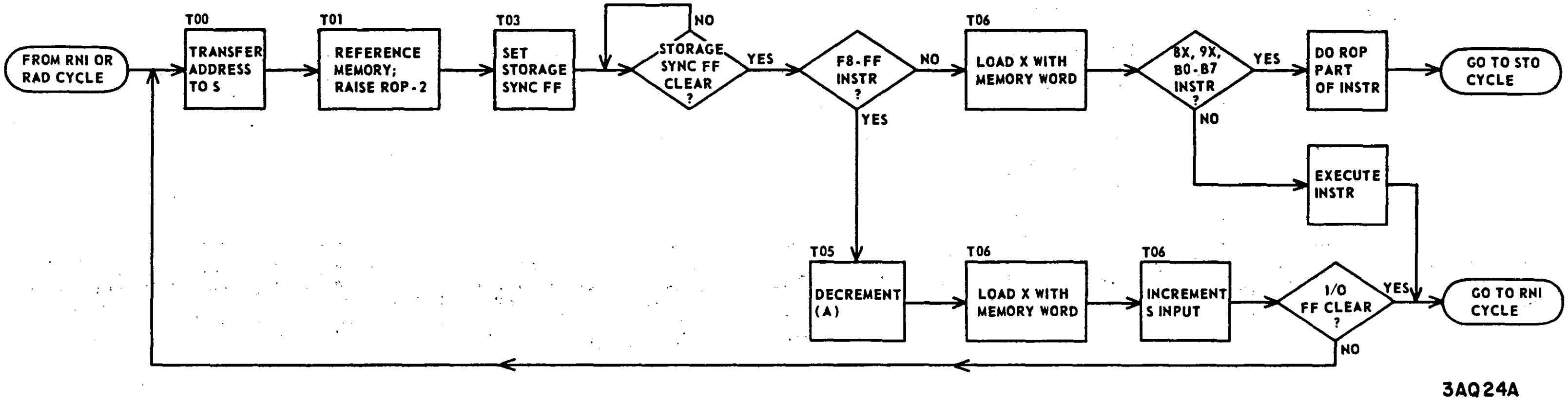


Figure 3-25. Common ROP Activity

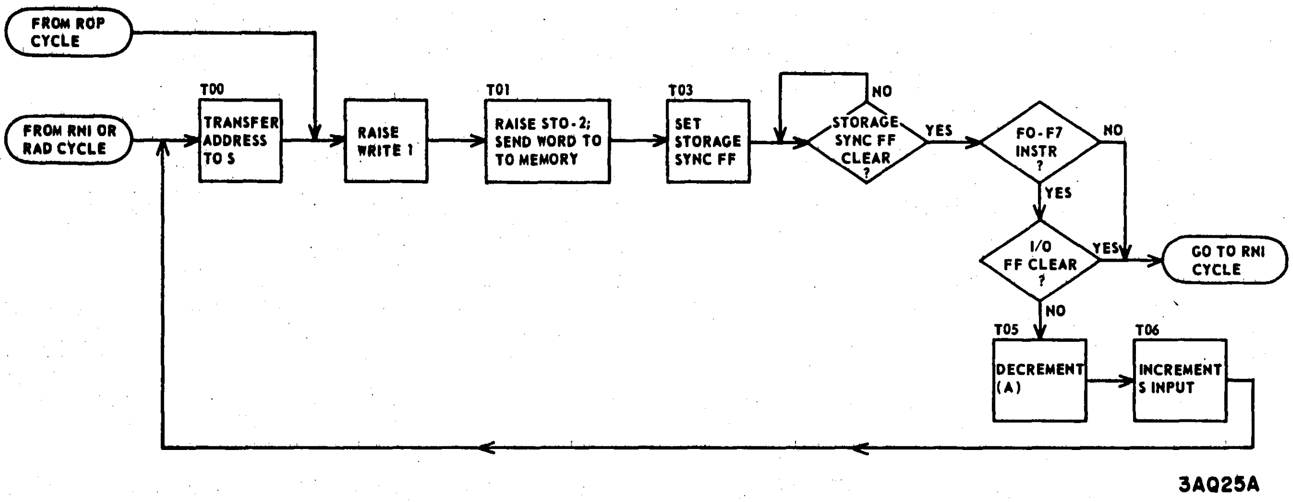


Figure 3-26. Common STO Activity

Format 1 Instructions

Format 1 instructions have the following bit format.

00 f 04	05 a 07	08 s 11	12 t 15

fa	Function code
s	Hexadecimal digit
t	Hexadecimal digit

Except for the load from (A) (17XX) instruction, all format 1 instructions require one RNI cycle. The load from (A) instruction requires one RNI cycle and one ROP cycle.

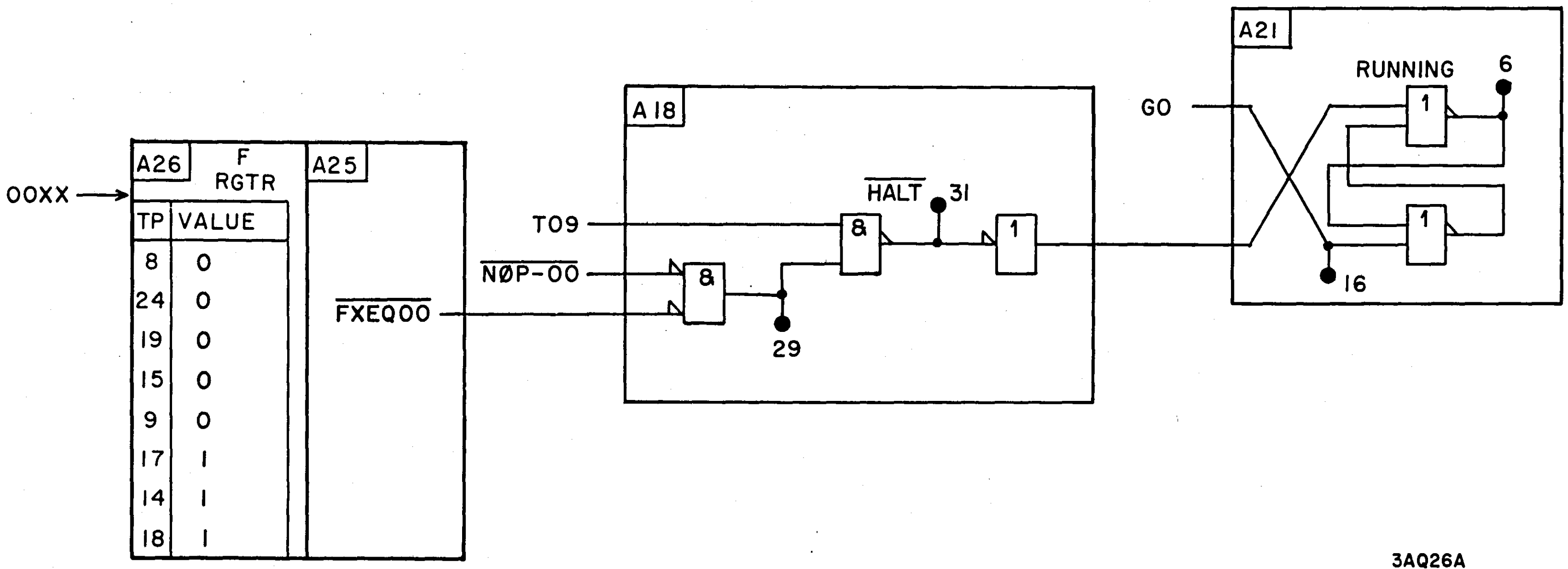
Figure 3-23 illustrates subsystem processor activity common to the RNI cycle for any format 1 or format 2 instruction.

Selective Stop (00XX)

A 00 function decode generates FXEQ00 which combines with the selective stop switch input NOP-00 to stop the subsystem processor.

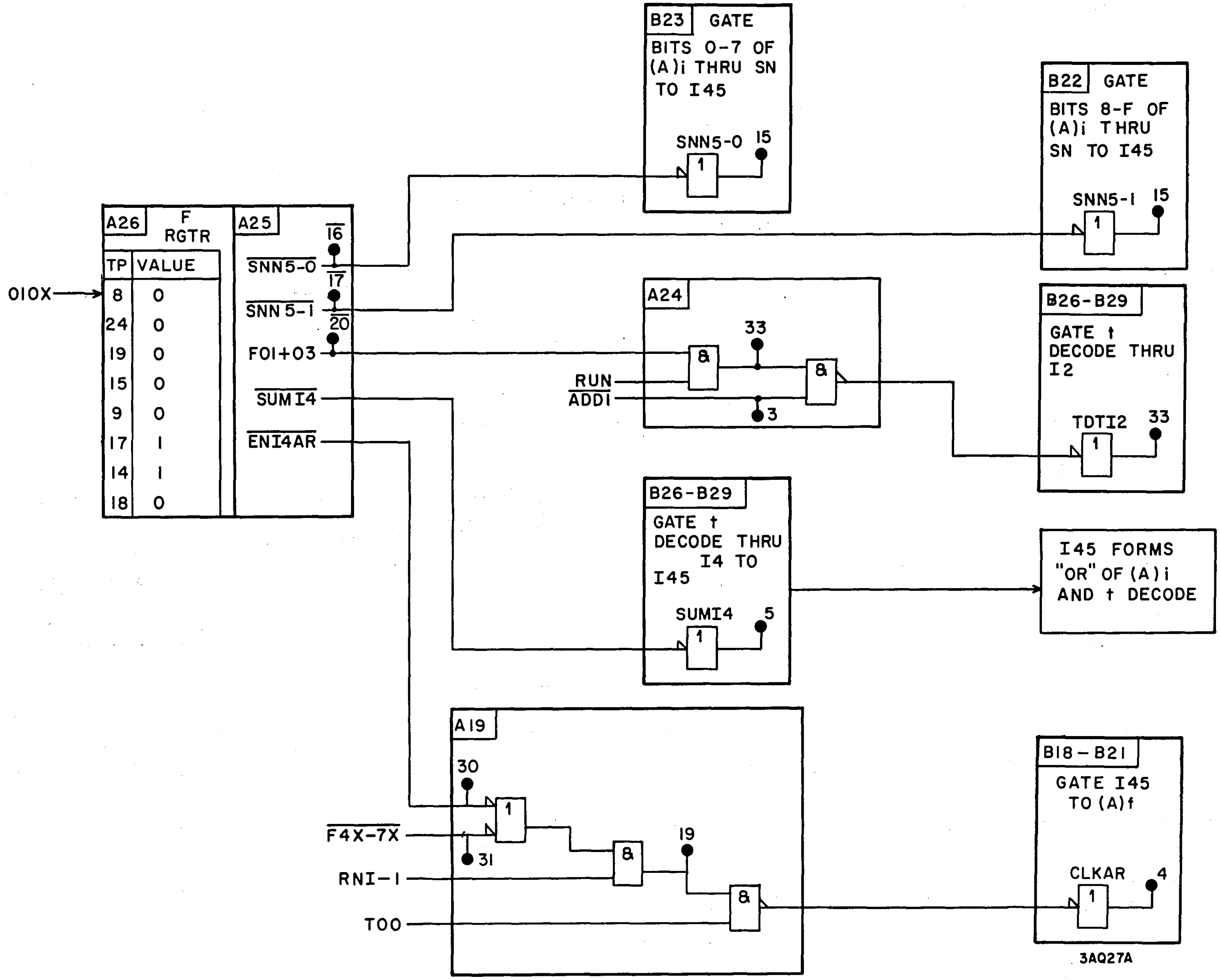
Selective Set Bit t of A (010X)

1. A 01 function decode generates SNN5-0, SNN5-1, F01+03, SUMI4, and ENI4AR.
2. SNN5-0 and SNN5-1 gate the initial contents of A through the shift network to I45.
3. F01+F03 generates TDT12 which gates the t decode through I2.
4. SUMI4 gates the t decode through the adder and I4.
5. I45 forms the OR of the initial contents of A and the t decode.
6. ENI4AR enables CLKAR which gates I45 to A to form the final contents of A.



TP	VALUE
8	0
24	0
19	0
15	0
9	0
17	1
14	1
18	1

Figure 3-27. Selective Stop (00XX)



TP	VALUE
8	0
24	0
19	0
15	0
9	0
17	1
14	1
18	0

Figure 3-28. Selective Set Bit t of A (010X)

Selective Clear Bit t of A (020X)

1. A 02 function decode generates ENARI1, FXEQ02, ANDI4, and ENI4AR.
2. ENARI1 generates ARTOI1 which gates the initial contents of A to I1.
3. FXEQ02 generates TDFI2 which gates the complement of the t decode to I2.
4. ANDI4 gates the AND of I1 and I2 through I4 to I45.
5. ENI4AR enables CLKAR which gates I45 to A to form the final contents of A.

Selective Complement Bit t of A (030X)

1. A 03 function decode generates ENARI1, F01+03, ENI4AR, and EOTOI4.
2. ENARI1 generates ARTOI1 which gates the initial contents of A to I1.
3. F01+03 generates TDTI2 which gates the t decode to I2.
4. EOTOI4 causes I4 to form the exclusive OR of I1 and I2 and feed I45.
5. ENI4AR enables CLKAR which gates I45 to A to form the final contents of A.

Count of Leading Zeros in A_i to A_f (04XX)

1. A 04 function decode generates SCTON6 and ENI4AR.
2. The scale network forms a 5-bit count of the leading zeros in the initial contents of the A register.
3. SCTON6 gates the 5-bit count from N6 to I45.
4. ENI4AR enables CLKAR which gates I45 to A to form the final contents of A.

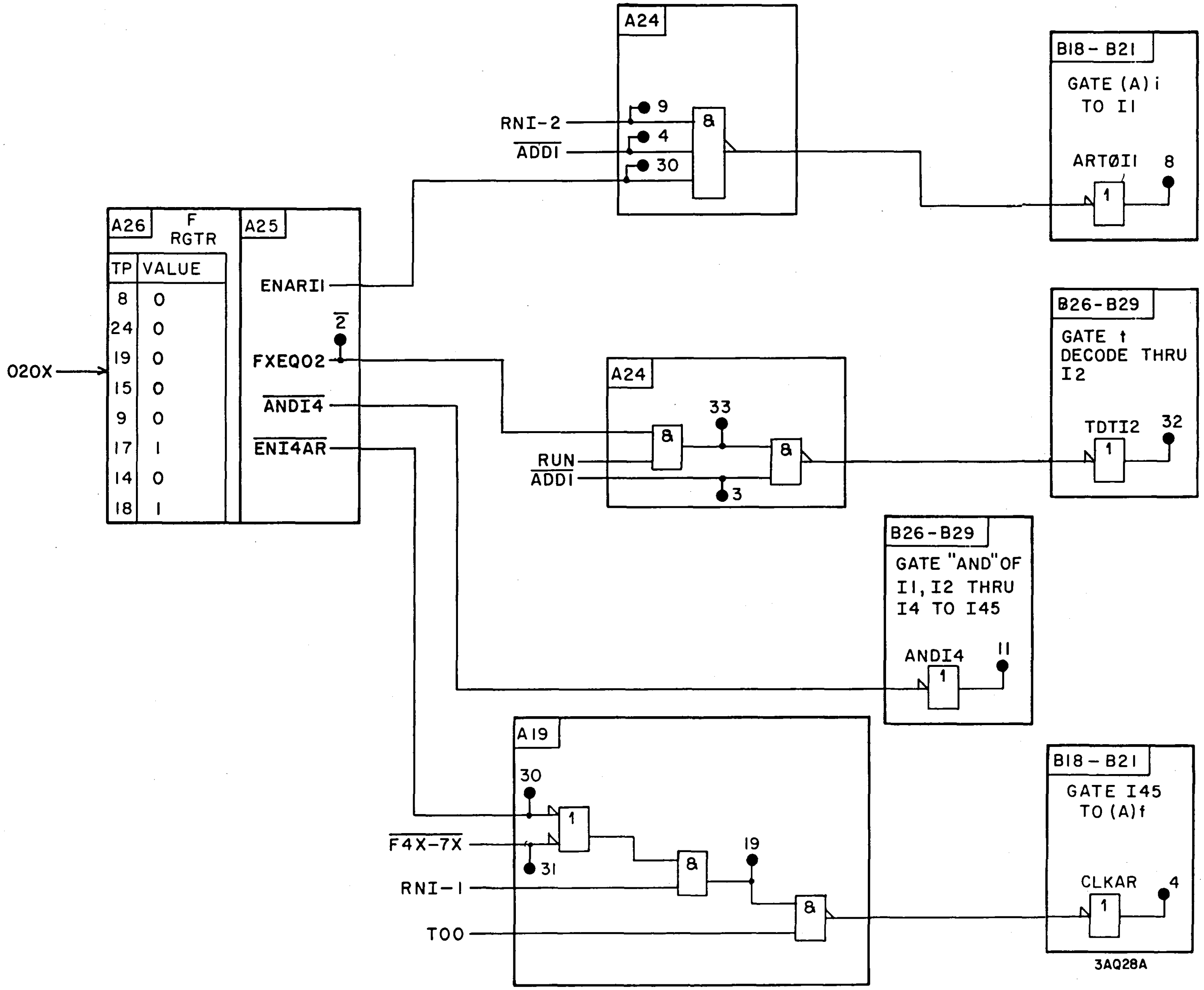


Figure 3-29. Selective Clear Bit t of A (020X)

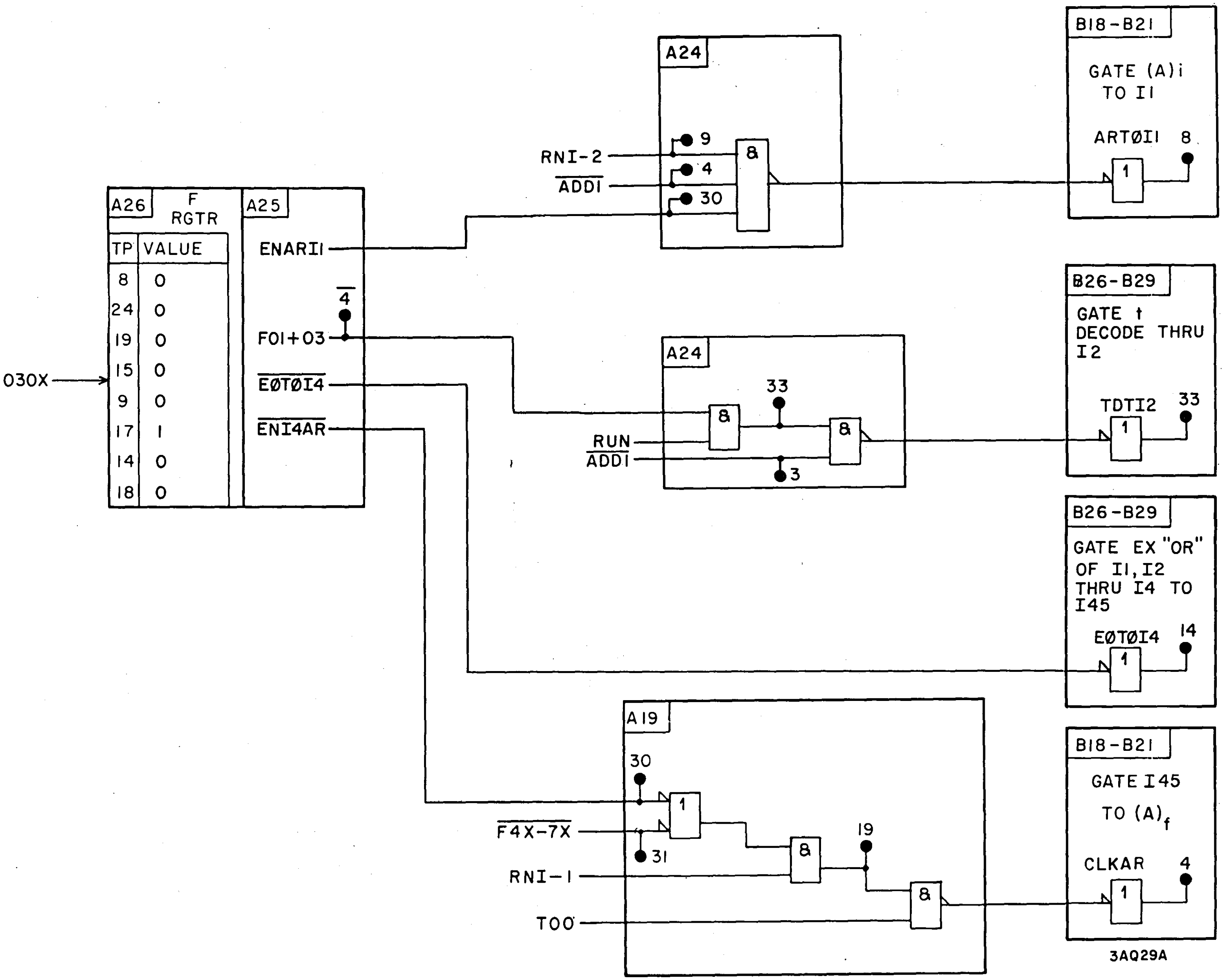


Figure 3-30. Selective Complement Bit t of A (030X)

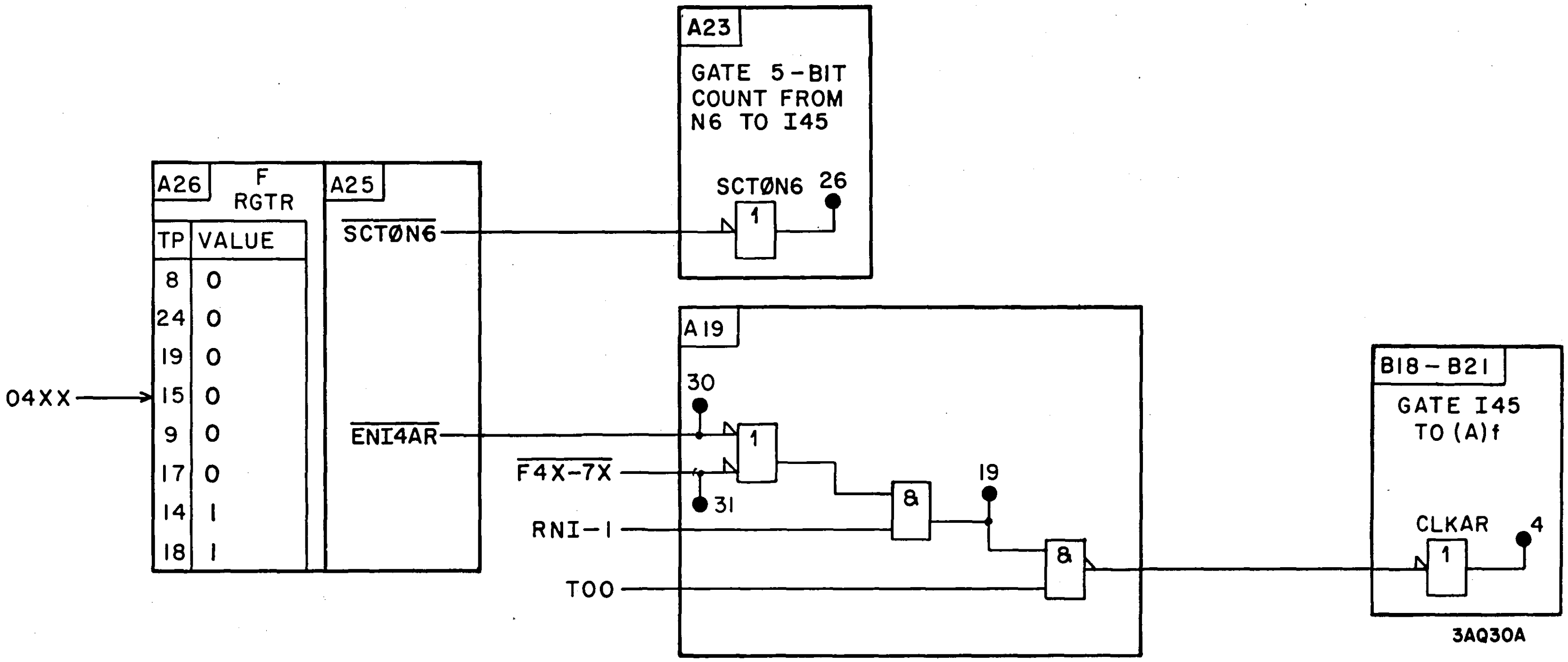


Figure 3-31. Count of Leading Zeros in A_i to A_f (04XX)

Shift (A) Right, t Places (05XX)

1. A 05 function decode generates F05+08, SNN5-0, SNN5-1, and ENI4AR.
2. F05+08 enables the lower 4 bits from X, XRS-12 through XRS-15, to generate the appropriate right shift signal(s). The right shift signals gate the initial contents of A through the shift network.
3. XRS-08 combines with F05-08 to generate STRIP which determines whether the shift is endoff (050X) or endaround (058X).
4. SNN5-0 and SNN5-1 gate the shift network output to I45. The lower 5 bits pass through N6 before feeding I45.
5. ENI4AR enables CLKAR which gates I45 to A to form the final contents of A.

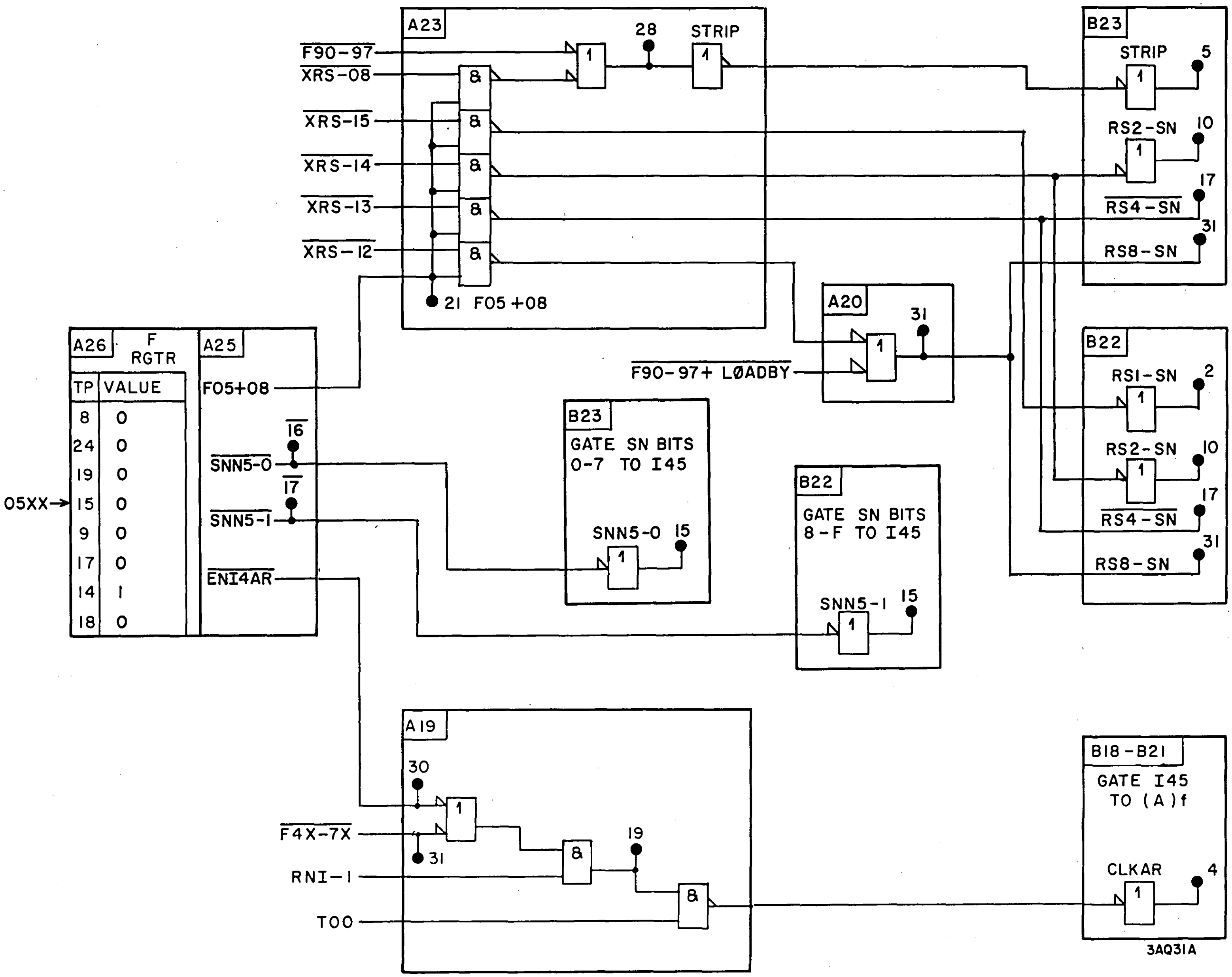
Transfer (A) + s, t to B1 (06XX)

Transfer (A) + s, t to B2 (07XX)

1. A 06 or 07 function decode generates ENARI1, ENXRI2, SUMI4, and ENI4B1 or ENI4B2.
2. ENARI1 enables ARTOI1 which gates the initial contents of A to I1.
3. ENXRI2 enables XRFI22 which gates the s and t bits from the X register to I2.
4. The adder forms the sum of I1 and I2. SUMI4 gates the result of the addition through I4.
5. ENI4B1 or ENI4B2 enables CLKB1 or CLKB2, one of which gates the result of the addition either to B1 or to B2.

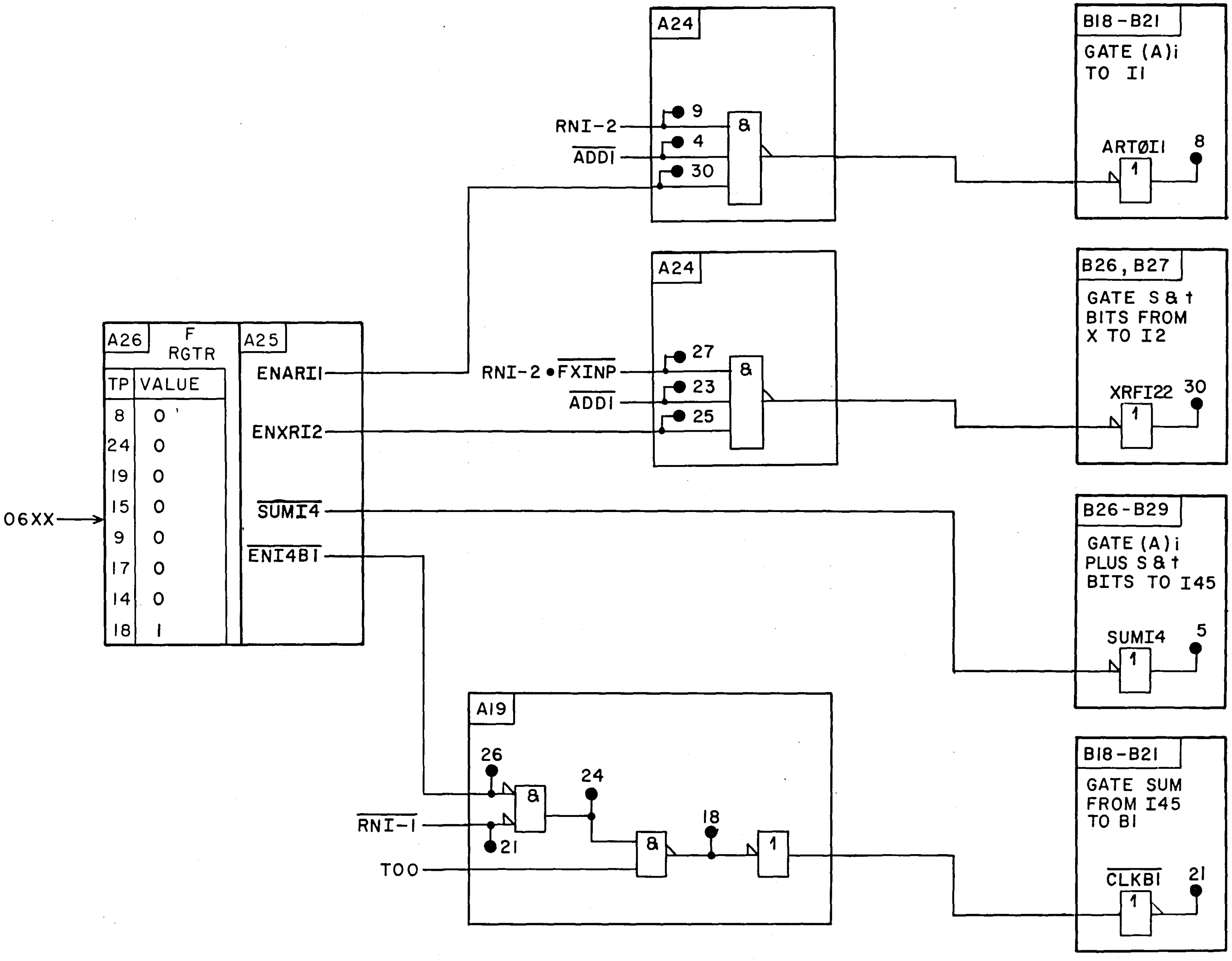
Set Condition Equal: Internal Tests (08XX)

1. A 08 function decode generates F05+08 and FXEQ08.
2. XRS-09 enables AISODD to the COND-FF.
3. XRS-10 enables the adder generate bit from the ENDF flip-flop to the COND flip-flop.
4. F05+08 enables the t field of the instruction to shift count logic. This determines the A register bit to be enabled through the shift network, to emerge as SNB15 and, subsequently, to be enabled to the COND flip-flop by XRS11.



TP	VALUE
8	0
24	0
19	0
15	0
9	0
17	0
14	1
18	0

Figure 3-32. Shift (A) Right, t Places (05XX)



3AQ32A

Figure 3-33. Transfer (A) + s, t to B1 (06XX)

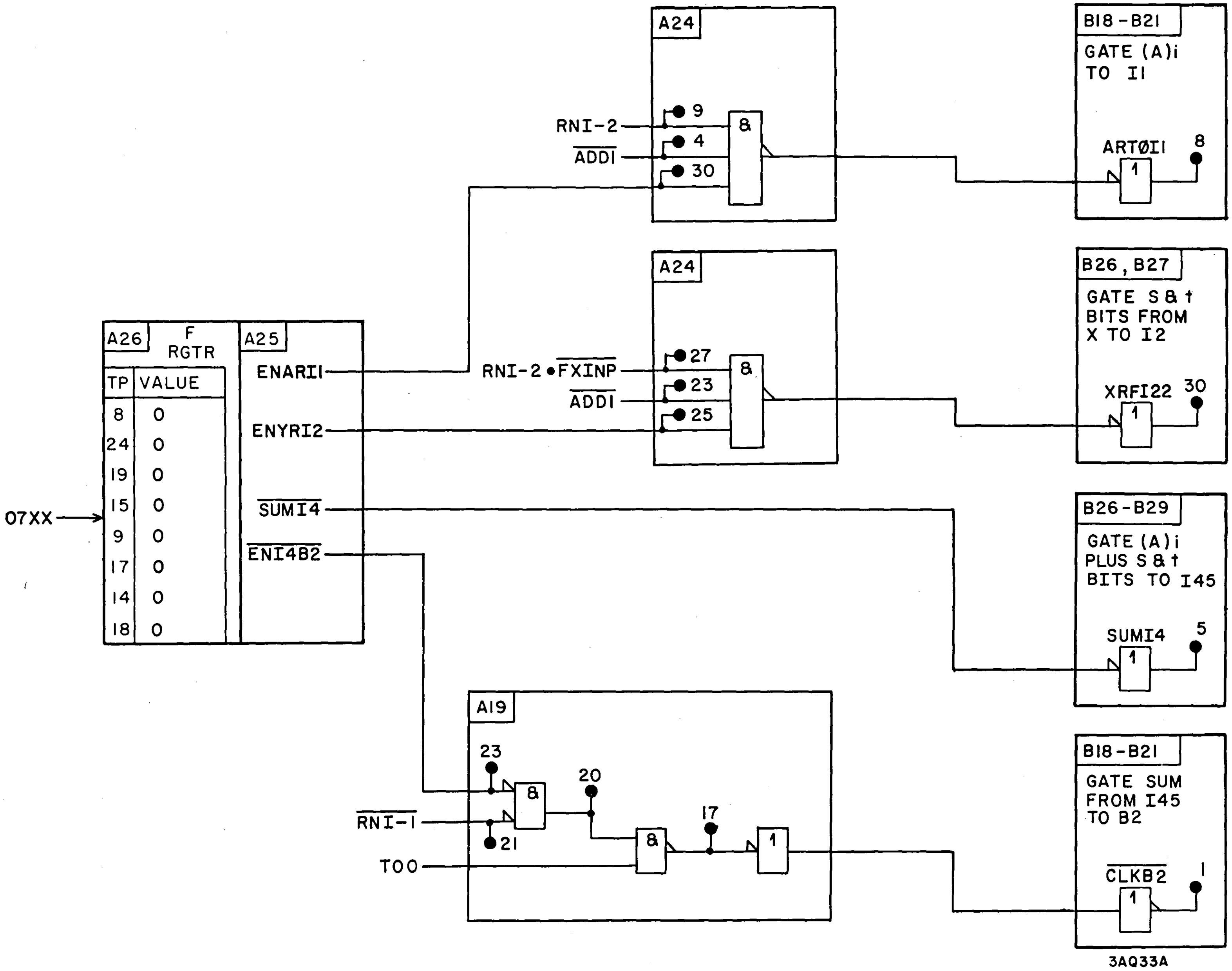
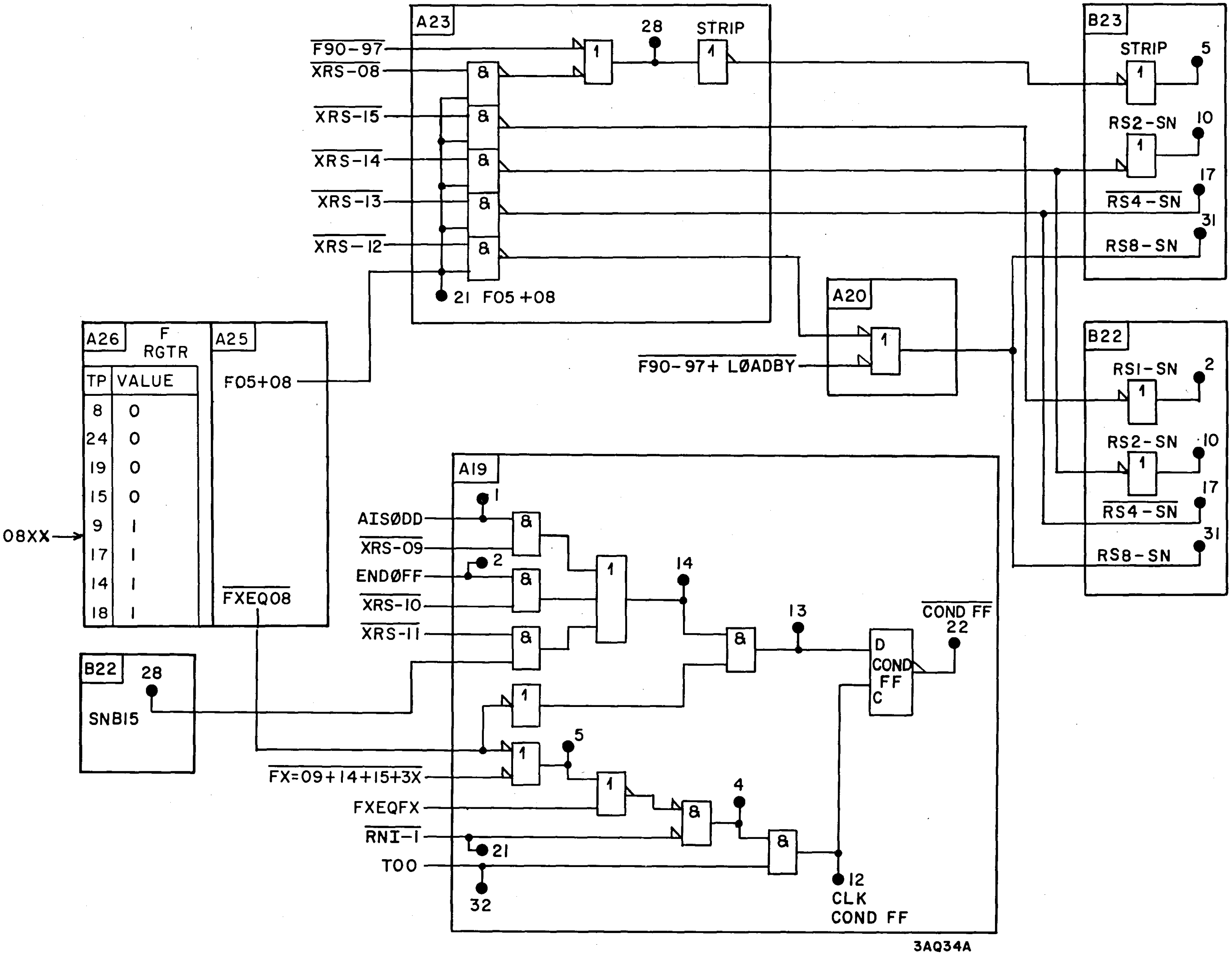


Figure 3-34. Transfer (A) + s, t to B2 (07XX)



TP	VALUE
8	0
24	0
19	0
15	0
9	1
17	1
14	1
18	1

Figure 3-35. Set Condition Equal: Internal Tests (08XX)

5. FXEQ08 enables clocking of the COND flip-flop at T00.

NOTE

The logic OR's the results of any tests specified by the s bits. If no tests are selected, the COND flip-flop is cleared.

Set Condition Equal: Bit t of Channel s (09XX)

1. The s decoder generates signals (SEL-IX) which select one of the channels for input to I0.
2. The t compare logic enables only the bit specified by t from the selected channel to the COND flip-flop.
3. FXEQ09 enables bit t of channel s to be clocked into the COND flip-flop at T00.

Selective Set Bit t of Channel 0 (0A0X) or Set Bit t and Clear All Other Bits of Channel s (s≠0) (0AsX)

1. FXEQ0A generates TDTOI6 and enables SETOUT.
2. TDTOI6 gates the t decode to I6 and the output channel drivers.
3. The s decoder provides the s decode to set output channel logic.
4. TC02 times SETOUT which feeds set output channel logic to generate a SET-OX signal. This signal causes bit t of channel s to be set (or remain set). If s ≠ 0, all other bits on channel s are cleared.

Selective Clear Bit t of Channel 0 (0B0X)

1. FXEQ0B generates TDTOI6, COMPCH, and enables CLROUT.
2. TDTOI6 gates the t decode to I6. COMPCH complements the t decode before the decode feeds the output channel drivers.
3. The s decoder provides the s decode to clear output channel logic.
4. TC02 times CLROUT which feeds clear output channel logic to generate a CLR-OX signal. This signal causes bit t of channel 0 to be cleared (or remain clear).

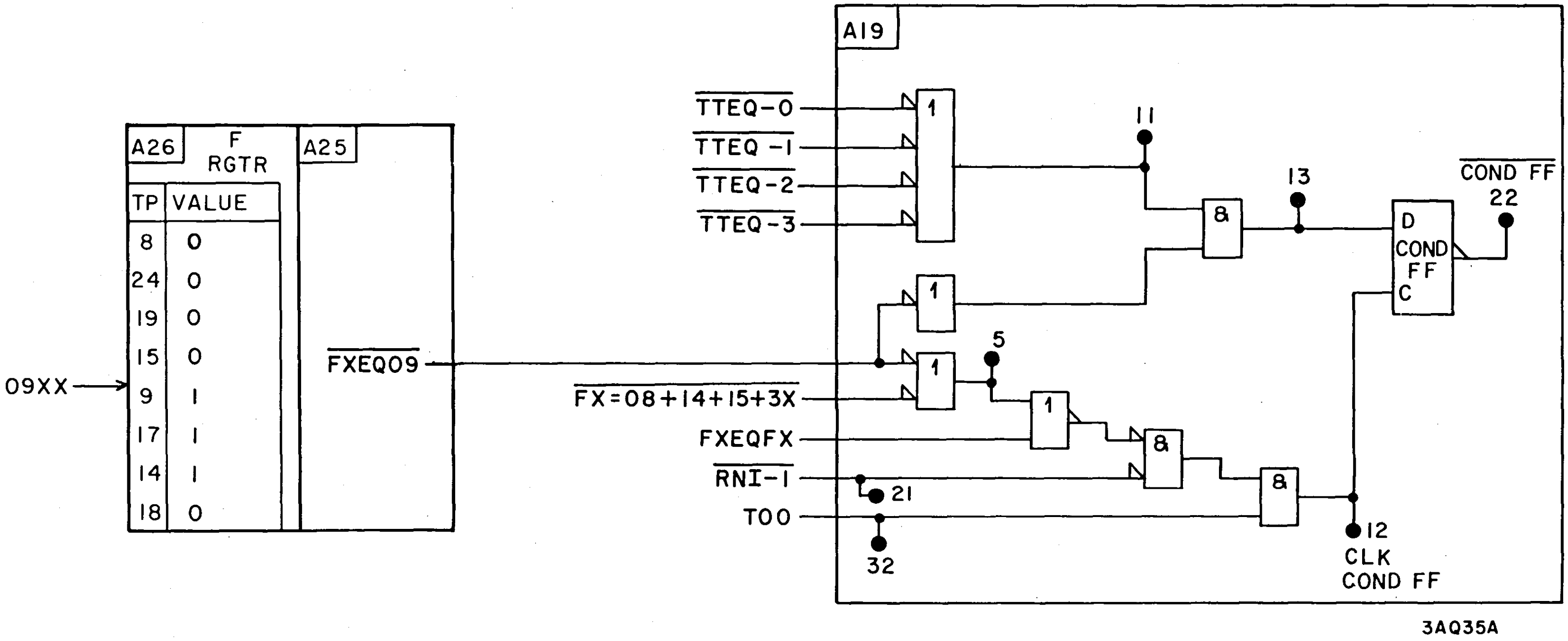
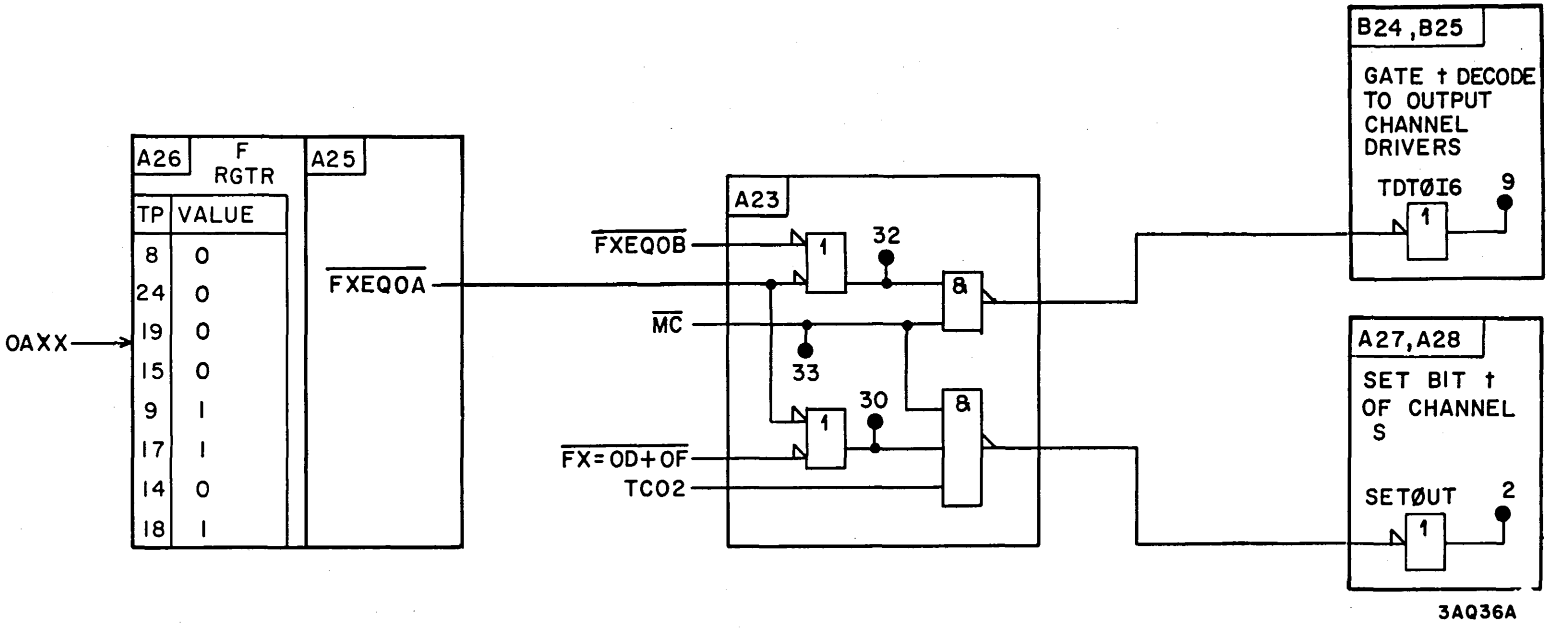


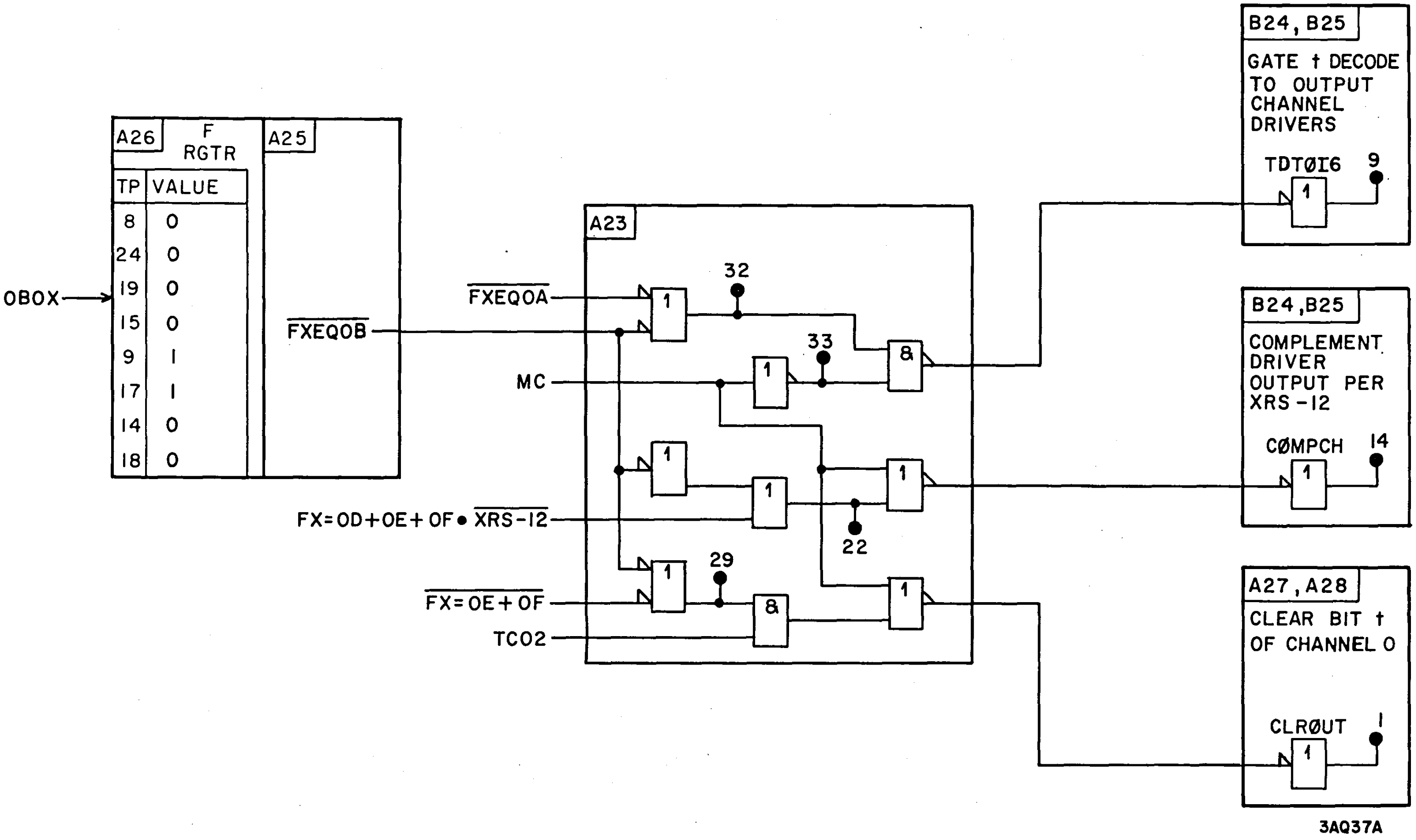
Figure 3-36. Set Condition Equal to Bit t of Channel s (09XX)



TP	VALUE
8	0
24	0
19	0
15	0
9	1
17	1
14	0
18	1

OAXX
A26
F RGTR
A25
$\overline{\text{FXEQ0A}}$
A23
$\overline{\text{FXEQ0B}}$
$\overline{\text{MC}}$
$\overline{\text{FX=0D+0F}}$
TC02
1
1
&
&
32
33
30
B24,B25
GATE t DECODE TO OUTPUT CHANNEL DRIVERS
TDTØI6
9
A27,A28
SET BIT t OF CHANNEL s
SETØUT
2
3AQ36A

Figure 3-37. Selective Set Bit t of Channel 0 (0A0X) or Set Bit t and Clear All Other Bits of Channel s (s≠0) (0AsX)



TP	VALUE
8	0
24	0
19	0
15	0
9	1
17	1
14	0
18	0

Figure 3-38. Selective Clear Bit t of Channel 0 (0B0X)

Input to A from Channel s (0CX0)

1. A 0C function decode generates FXEQ0C and I0TOI4.
2. The s decode logic generates a SEL-IX signal.
3. I0TOI4 gates the word from the selected input channel to I4 and I45.
4. FXEQ0C enables CLKAR which gates I45 to A to form the final contents of A.

NOTE

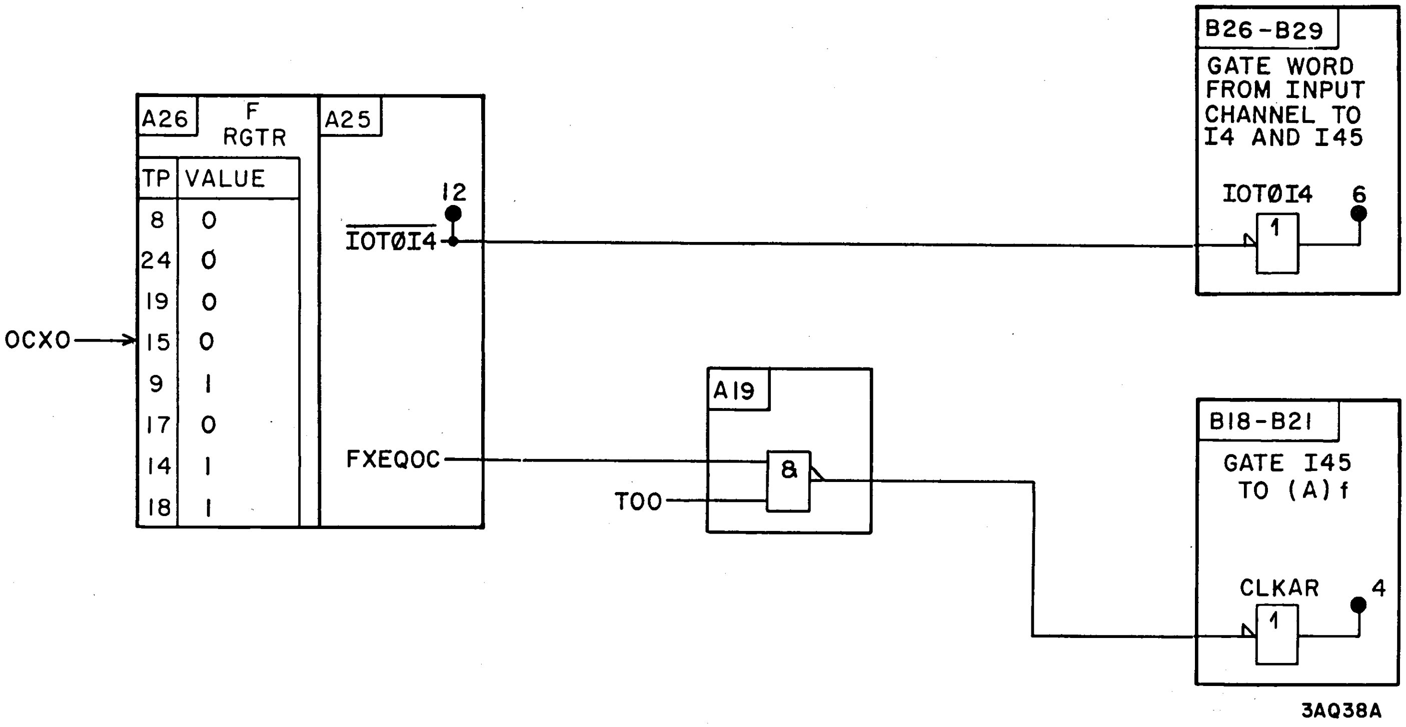
For the following three instructions (0D0X, 0E0X, 0FXX), bit 12 of the instruction word determines the state of COMPCH. COMPCH determines whether true or complement data is provided to the normal output channel drivers.

Set Channel 0 from A (0D0X)

1. FXEQ0D generates ARTOI6 and enables COMPCH and SETOUT.
2. ARTOI6 enables A register bits 00 through 15 to I6 and the normal output channel drivers.
3. The s decoder provides the s decode to set output channel logic.
4. TC02 times SETOUT which feeds set output channel logic to generate a SET-O0 signal. When COMPCH is 1, SET-O0 causes bits on output channel 0 corresponding to 1 bits in the A register to be set (or remain set). Bits corresponding to 0 bits in the A register are unchanged. When COMPCH is 0, SET-O0 causes bits on output channel 0 corresponding to 0 bits in the A register to be set (or remain set). Bits corresponding to 1 bits in the A register are unchanged.

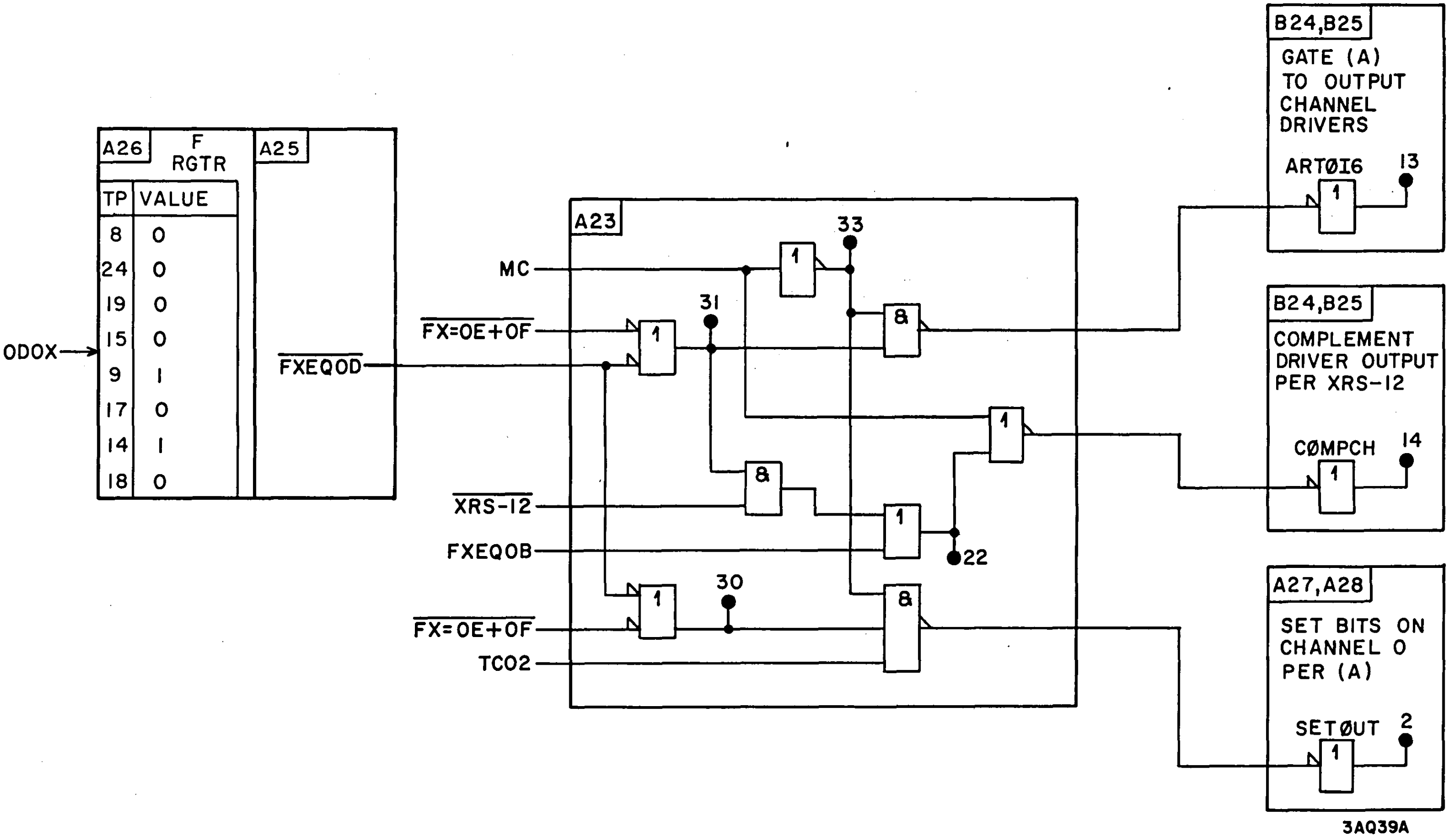
Clear Channel 0 from A (0E0X)

1. FXEQ0E generates ARTOI6 and enables COMPCH and CLROUT.
2. ARTOI6 enables A register bits 00 through 15 to I6 and the output channel drivers.
3. The s decoder provides the s decode to clear output channel logic.
4. TC02 times CLROUT which feeds clear output channel logic to generate a CLR-O0 signal. When COMPCH is 1, CLR-O0 causes bits on output channel 0 corresponding to 0 bits in the A register to be cleared (or remain clear). Bits corresponding to 1 bits in the A register are unchanged. When COMPCH is 0, CLR-O0 causes bits on output channel 0 corresponding to 1 bits in the A register to be cleared (or remain clear). Bits corresponding to 0 bits in the A register are unchanged.



TP	VALUE
8	0
24	0
19	0
15	0
9	1
17	0
14	1
18	1

Figure 3-39. Input to A from Channel s (0CX0)



TP	VALUE
8	0
24	0
19	0
15	0
9	1
17	0
14	1
18	0

Figure 3-40. Set Channel 0 from A (0D0X)

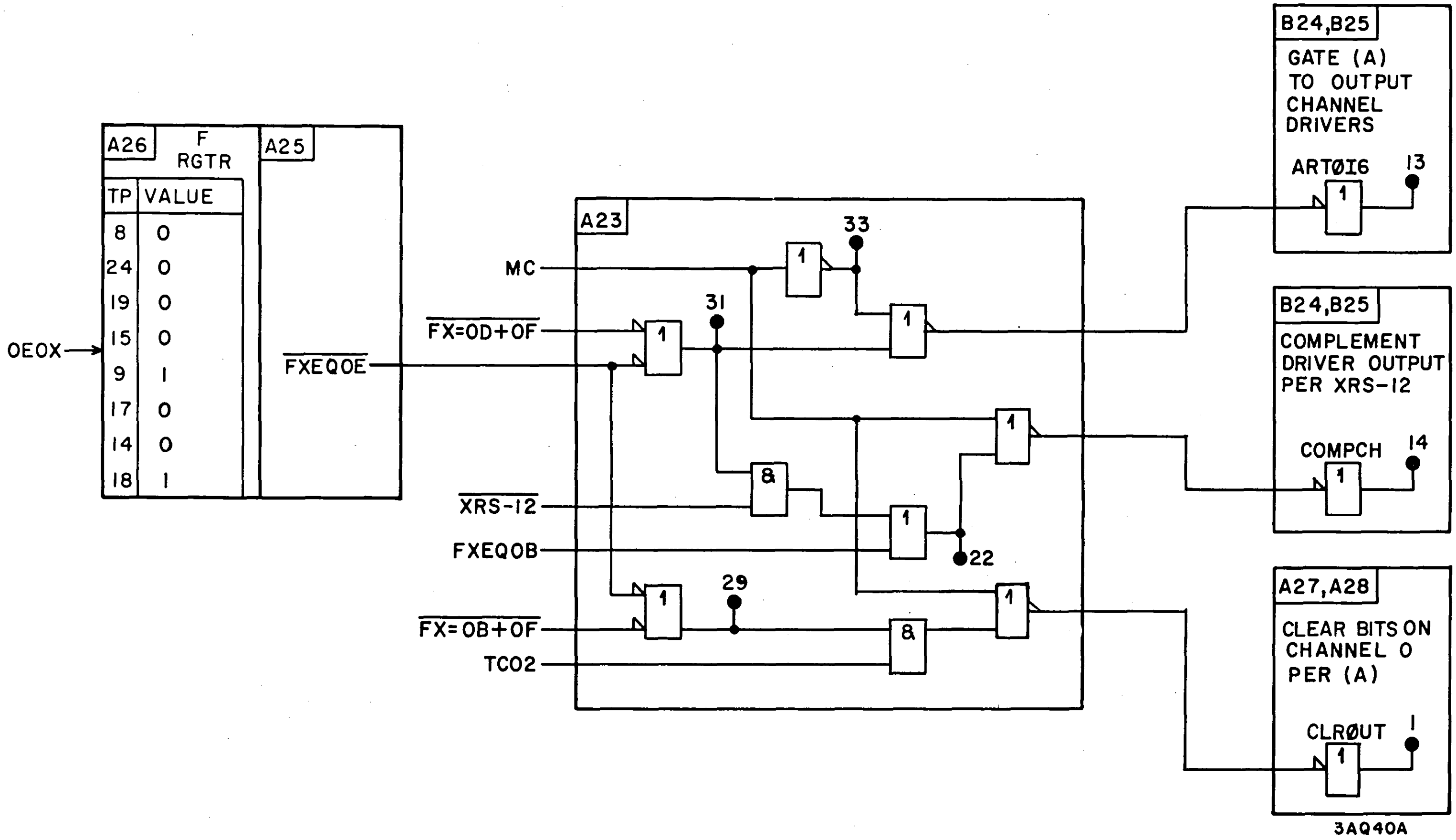


Figure 3-41. Clear Channel 0 from A (0E0X)

Transfer (A) to Channel s (0FXX)

1. FXEQ0F generates ARTOI6 and enables COMPCH, SETOUT, and CLROUT.
2. ARTOI6 enables A register bits 00 through 15 to I6 and the output channel drivers.
3. The s decoder provides the s decode to set and clear output channel logic.
4. TC02 times SETOUT and CLROUT which feed output channel logic simultaneously to generate a SET-OX and a CLR-OX signal. When COMPCH is 1, SET-OX and CLR-OX cause a true transfer of data from the A register to the selected output channel. When COMPCH is 0, SET-OX and CLR-OX cause a one's complement transfer of data from the A register to the selected output channel.

Add - No Address (10XX)

1. A 10 function decode generates ENXRI2, ENARI1, SUMI4, and F10+11.
2. ENXRI2 enables XRFI22.
3. XRFI22 gates the one's complement of X bits 08 through 15 (which is the true value of the s and t bits from memory) to I2. X bits 00 through 07 are not gated, so I2 provides 0-fill to the adder for these positions.
4. ENARI1 generates ARTOI1 which gates the initial contents of the A register to I1.
5. The adder forms the sum of the initial contents of A plus the s and t bits. SUMI4 gates the sum through I4 to I45.
6. F10+11 enables the clocking of the ENDF flip-flop at T00.
7. FXEQ1X generates ENI4AR which enables CLKAR to gate the result of the add from I45 to A and form the final contents of A.

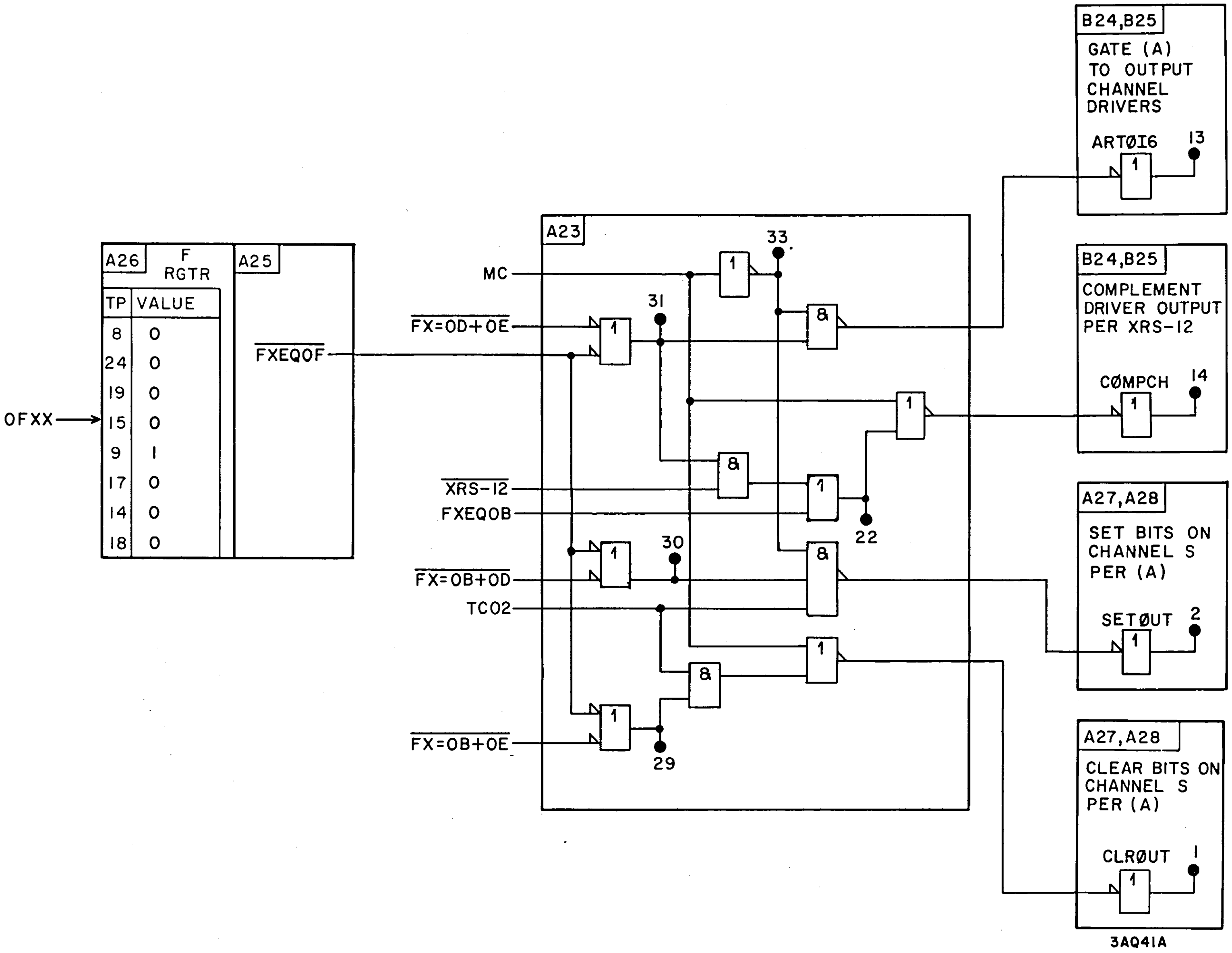


Figure 3-42. Transfer (A) to Channel s (0FXX)

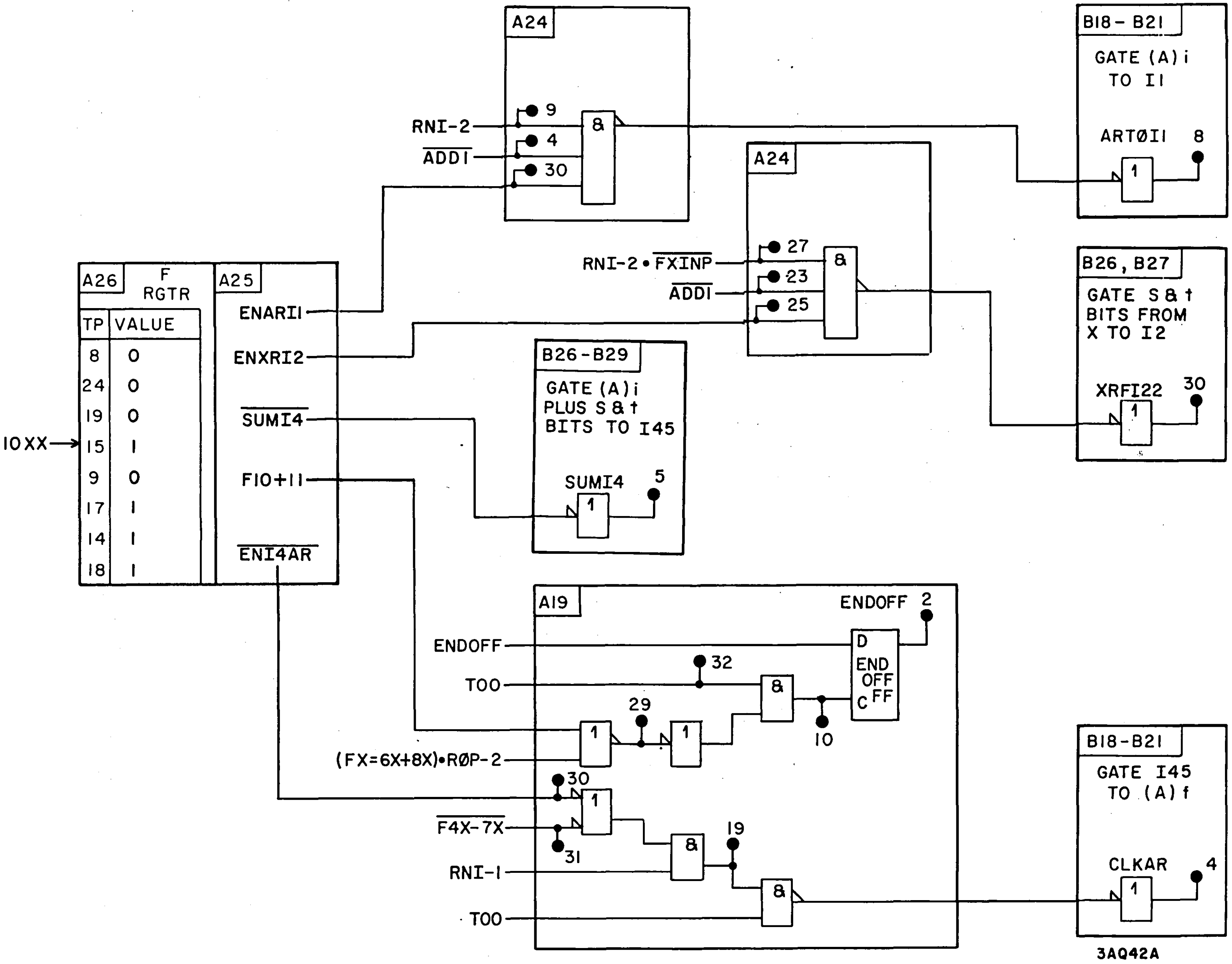


Figure 3-43. Add - No Address (10XX)

Subtract - No Address (11XX)

1. An 11 function decode generates ENNXI2, ENARI1, SUMI4 and F10+11.
2. ENNXI2 enables XRTI2, XRFI21, and CARRY4.
3. XRTI2 gates the true value of X bits 00 through 15 (which is the one's complement of bits 00 through 15 from memory) to I2. XRFI21 gates the one's complement of X bits 00 through 07 to I2. Bits 00 through 07 of I2 are 1-filled as a result of the OR of true and complement values of X bits 00 through 07 at I2.
4. ENARI1 generates ARTOI1 which gates the initial contents of A to I1.
5. CARRY4 feeds the adder to form the two's complement result as the s and t bits are subtracted from the initial contents of A. SUMI4 gates the result of the subtract operation through I4 to I45.
6. F10+11 enables the clocking of the ENDF flip-flop at T00.
7. FXEQ1X generates ENI4AR which enables CLKAR to gate the result of the subtract operation from I45 to A and form the final contents of A.

Exclusive OR - No Address (12XX)

1. A 12 function decode generates ENXRI2, ENARI1, and EOTOI4.
2. ENXRI2 generates XRFI22 which gates the one's complement of X bits 08 through 15 (which is the true value of the s and t bits from memory) to I2. X bits 00 through 07 are not gated, so I2 provides 0-fill for these positions.
3. ENARI1 generates ARTOI1 which gates the initial contents of A to I1.
4. EOTOI4 gates the exclusive OR of gates I1 and I2 through I4 to I45. Bits 00 through 07 of A are exclusive ORed with 0-fill and are thus unchanged.
5. FXEQ1X generates ENI4AR which enables CLKAR to gate the result of the exclusive OR operation from I45 to A and form the final contents of A.

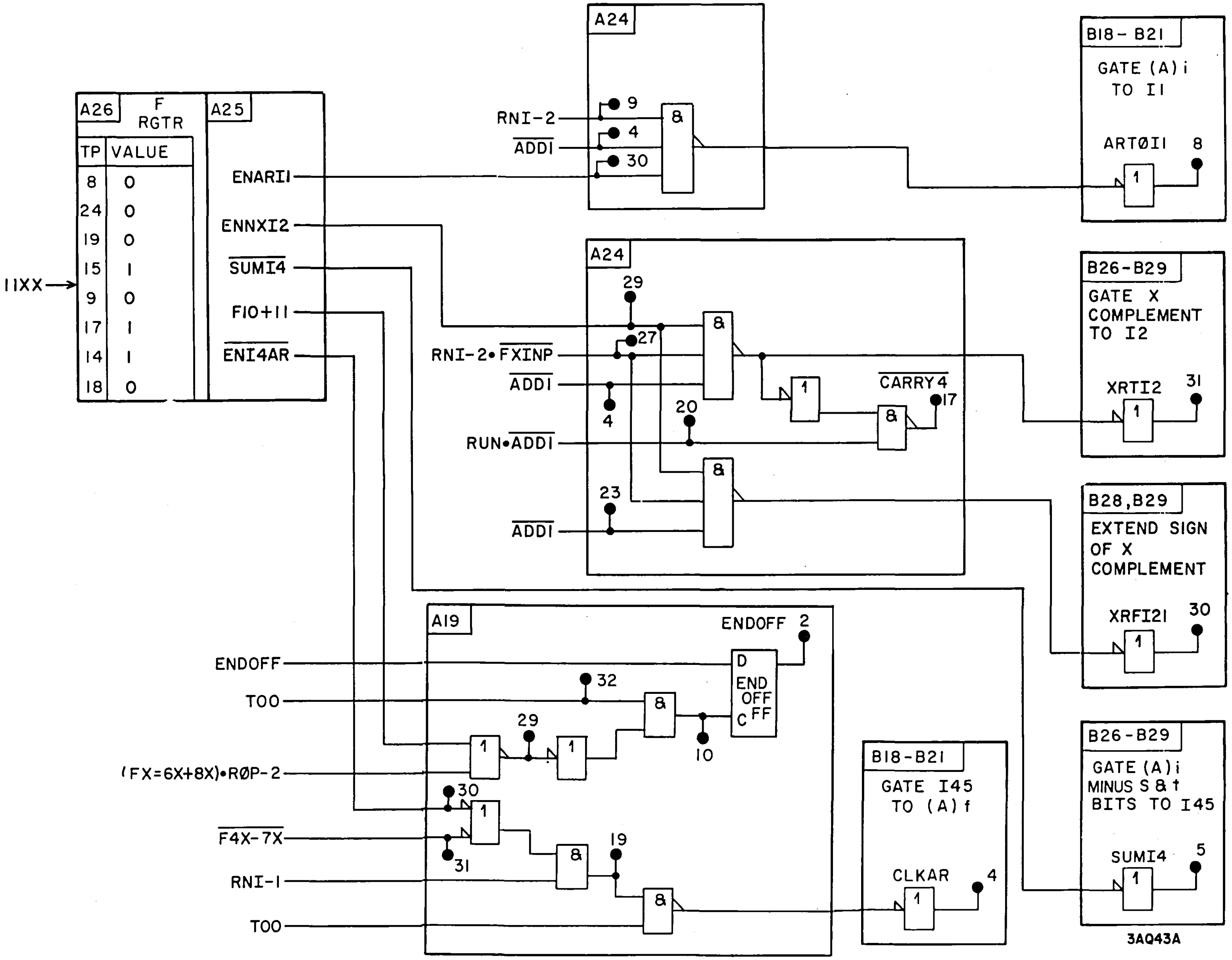
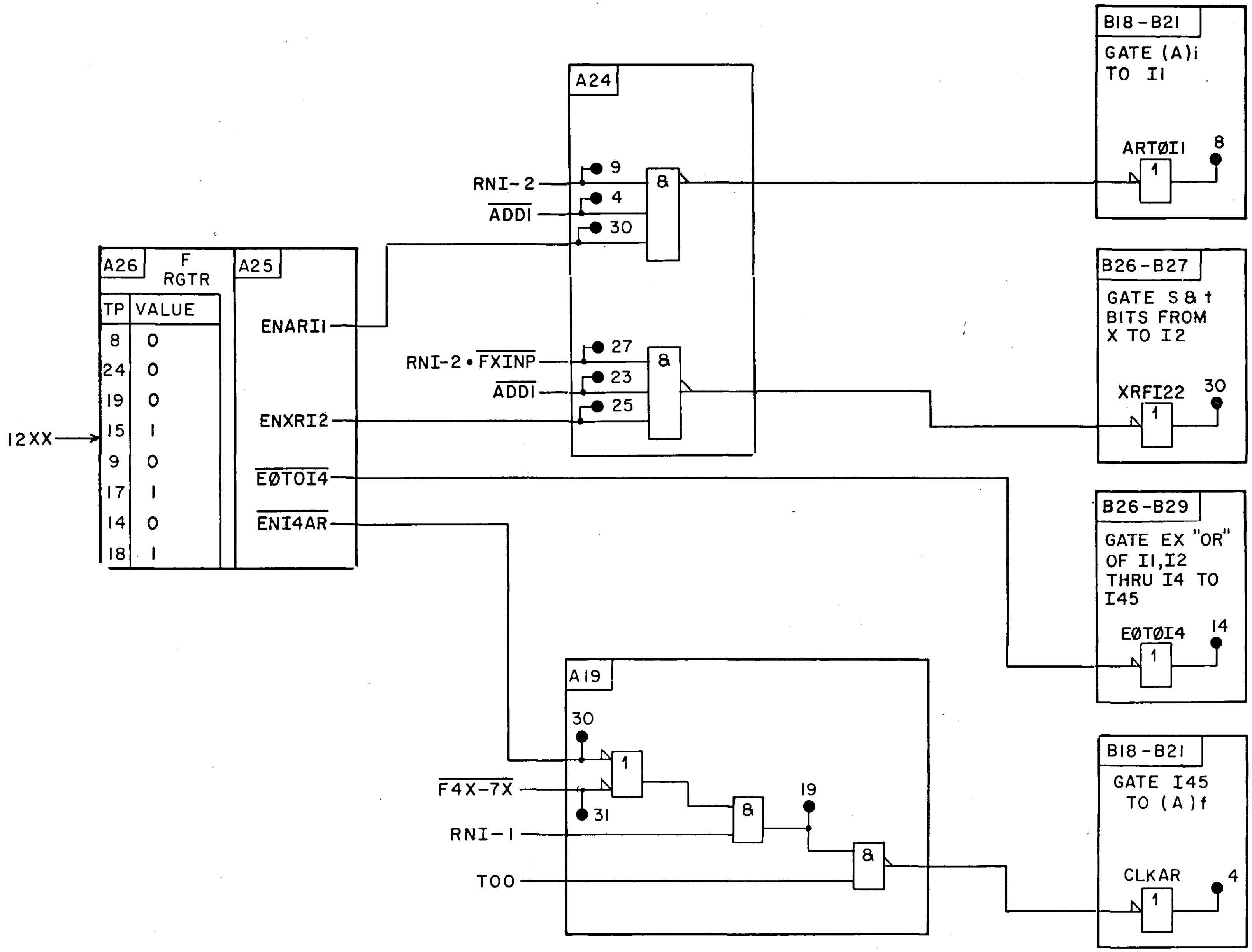


Figure 3-44. Subtract - No Address (11XX)



TP	VALUE
8	0
24	0
19	0
15	1
9	0
17	1
14	0
18	1

3AQ44A

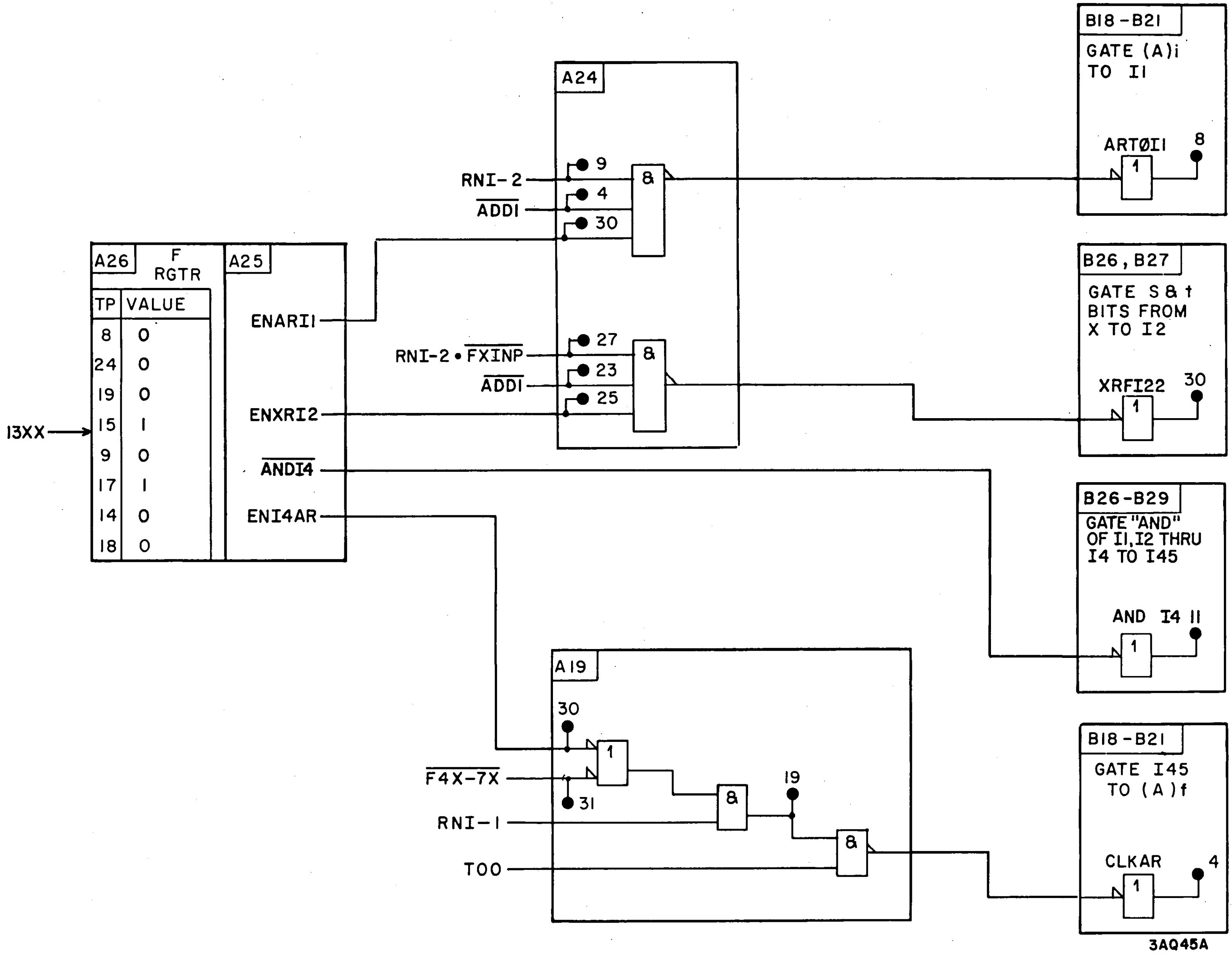
Figure 3-45. Exclusive OR - No Address (12XX)

Logical Product - No Address (13XX)

1. A 13 function decode generates ENXRI2, ENARI1, and ANDI4.
2. ENXRI2 generates XRFI22 which gates the one's complement of X bits 08 through 15 (which is the true value of the s and t bits from memory) to I2. X bits 00 through 07 are not gated, so I2 provides 0-fill for these positions.
3. ENARI1 generates ARTOI1 which gates the initial contents of A to I1.
4. ANDI4 gates the AND of I1 and I2 through I4 to I45. Bits 00 through 07 of A are ANDed with 0-fill and thus emerge from I4 as 0's.
5. FXEQ1X generates ENI4AR which enables CLKAR to gate the result of the logical product operation from I45 to A and form the final contents of A.

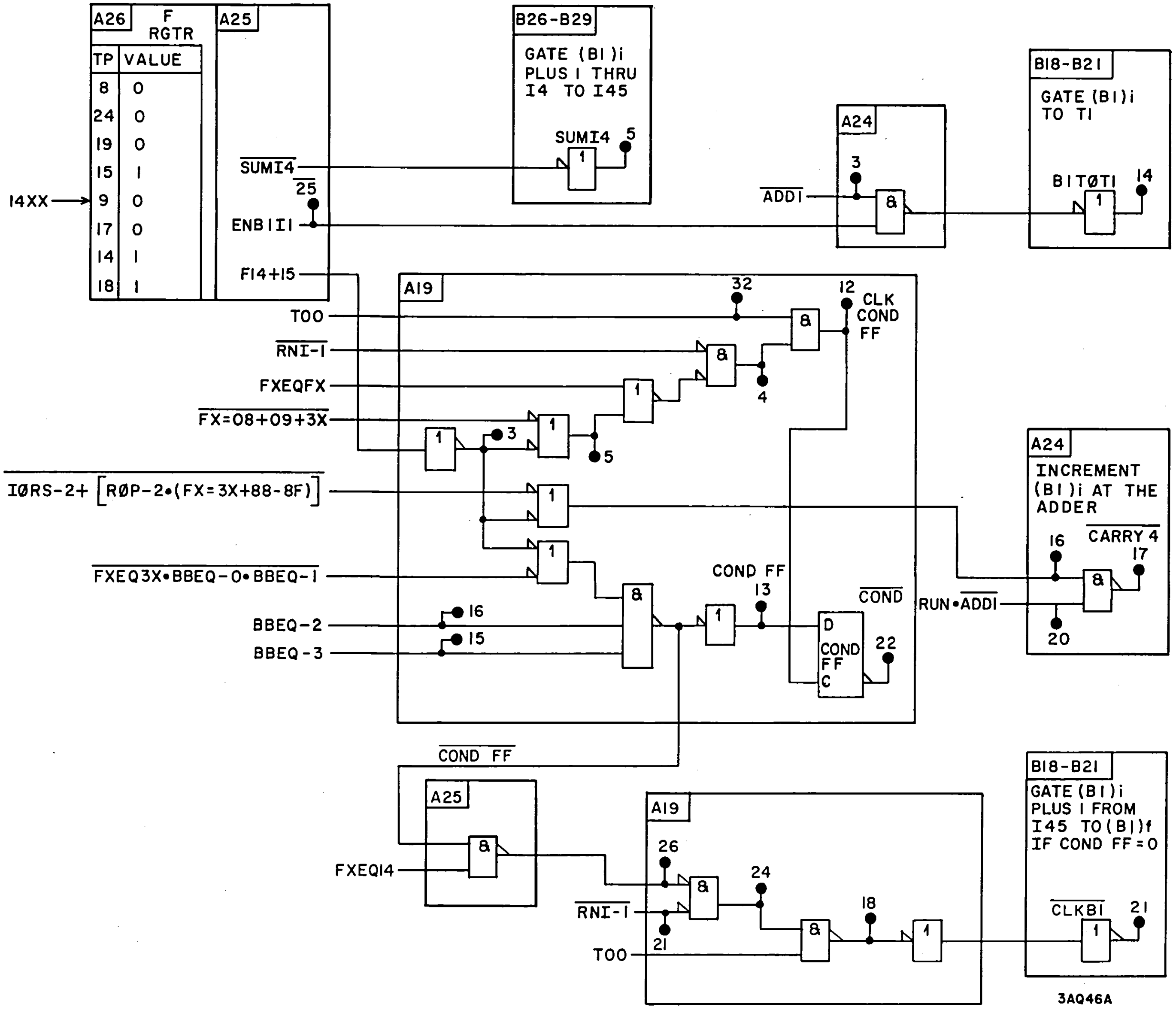
Test Index B1 - No Address (14XX)

1. A 14 function decode generates ENB1I1, F14+15, and SUMI4.
2. ENB1I1 generates B1TOI1 which gates the contents of B1 through I1 to B compare logic and the adder.
3. X bits 08 through 15, which are the one's complement of the s and t bits from memory, feed the lower 8 bits of B compare logic.
4. F14+15 enables the lower B compare result to the COND flip-flop. When a compare occurs between the s and t bits and the lower 8 bits of B1, COND-FF is active to disable ENI4B1. When no compare occurs, COND-FF is inactive to enable ENI4B1.
5. F14+15 enables clocking of the COND flip-flop at T00. F14+15 also generates FORCE which generates CARRY4 to increment the contents of B1 which are currently in the adder.
6. SUMI4 gates the incremented quantity from the adder through I4.
7. When no compare occurs (step 4), ENI4B1 enables CLKB1 which gates I4 to B1 to form the final contents of B1. When a compare occurs, B1 is not clocked and remains unchanged.



TP	VALUE
8	0
24	0
19	0
15	1
9	0
17	1
14	0
18	0

Figure 3-46. Logical Product - No Address (13XX)



TP	VALUE
8	0
24	0
19	0
15	1
9	0
17	0
14	1
18	1

Figure 3-47. Test Index B1 - No Address (14XX)

Test Index B2 - No Address (15XX)

1. A 15 function decode generates ENB2I1, F14+15, and SUMI4.
2. ENB2I1 generates B2TOI1 which gates the contents of B2 through I1 to B compare logic and the adder.
3. X bits 08 through 15 (which are the one's complement of the s and t bits from memory) feed the lower 8 bits of B compare logic.
4. F14+15 enables the lower B compare result to the COND flip-flop. When a compare occurs between the s and t bits and the lower 8 bits of B2, COND-FF is active to disable ENI4B1. When no compare occurs, COND-FF is inactive to enable ENI4B1.
5. F14+15 enables clocking of the COND flip-flop at T00. F14+15 also generates FORCE which generates CARRY4 to increment the contents of B2 which are currently in the adder.
6. SUMI4 gates the incremented quantity from the adder through I4.
7. When no compare occurs, (step 4) ENI4B2 enables CLKB2 which gates I4 to B2 to form the final contents of B2. When a compare occurs, B2 is not clocked and remains unchanged.

Load A Complement - No Address (16XX)

1. A 16 function decode generates ENNXI2 and SUMI4.
2. ENNXI2 enables XRTI2, XRFI21, and CARRY4.
3. XRT12 gates the true value of X bits 00 through 15 which is the one's complement of bits 00 through 15 from memory, to I2. XRFI21 gates the one's complement of X bits 00 through 07 to I2. Bits 00 through 07 of I2 are 1-filled as a result of the OR of true and complement values of X bits 00 through 07 at I2.
4. CARRY4 feeds the adder to convert the one's complement of the s and t bits from I2 into a two's complement number.
5. SUMI4 gates the two's complement of the s and t bits from the adder through I4 to I45.
6. FXEQ1X generates ENI4AR which enables CLKAR to gate the two's complement of the s and t bits from I45 to A to form the final contents of A.

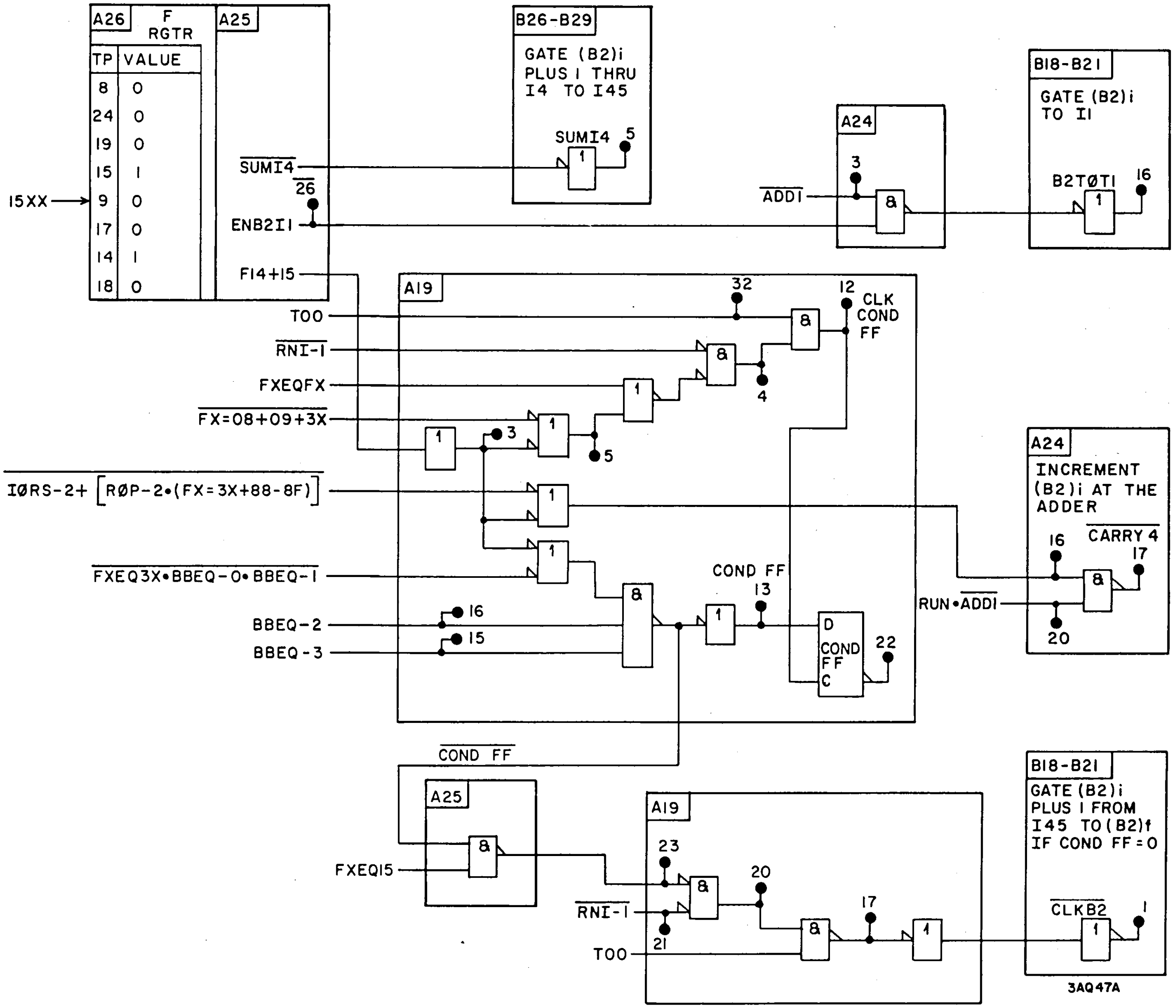


Figure 3-48. Test Index B2 - No Address (15XX)

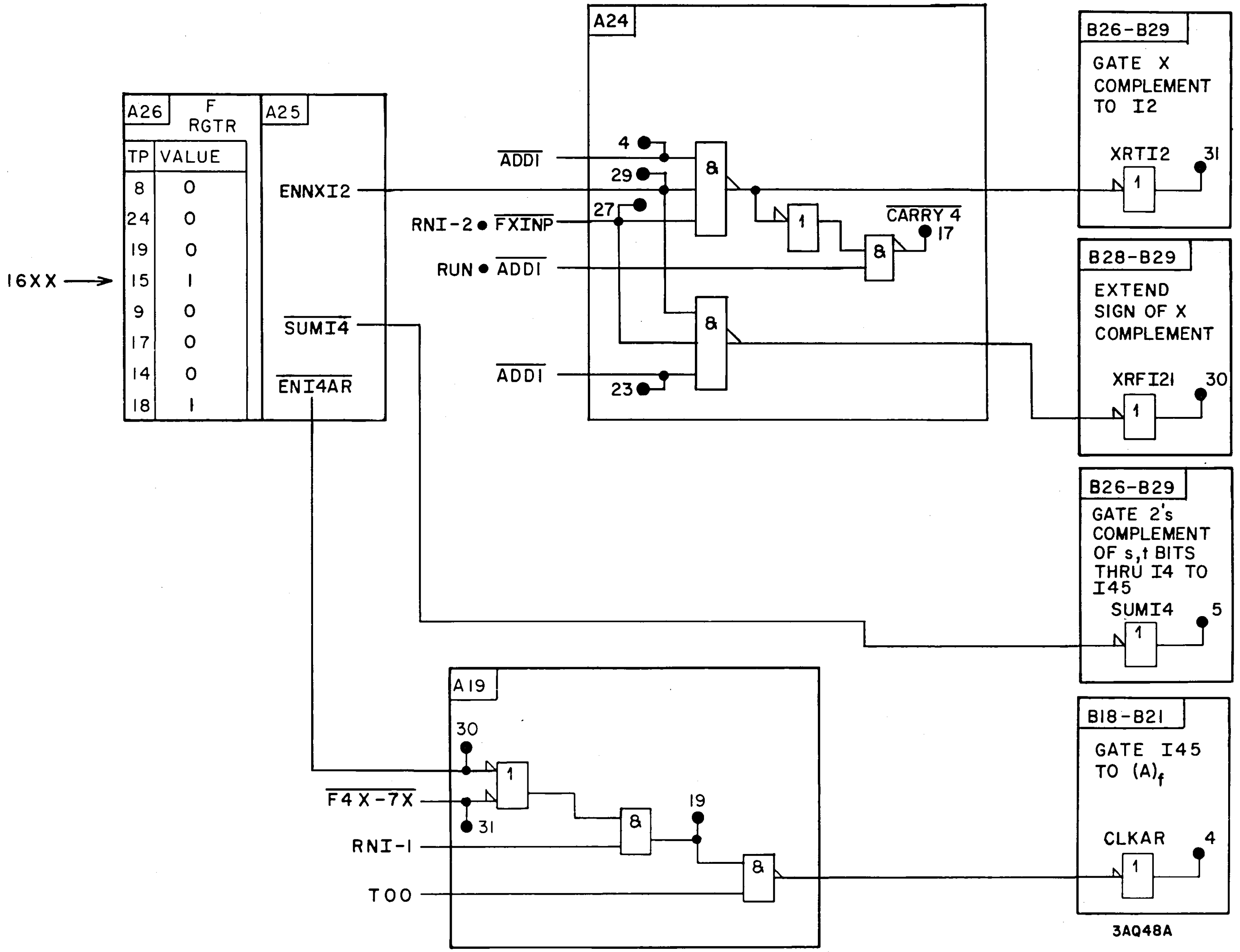


Figure 3-49. Load A Complement - No Address (16XX)

Load From (A) (17XX)

1. A 17 function decode generates ENXRI2, ENARI1, FXEQ17, and SUMI4.
2. ENXRI2 enables XRFI22 which gates the one's complement of X bits 08 through 15 (which is the true value of the s and t bits from memory) to I2. X bits 00 through 07 are not gated, so I2 provides 0-fill for these positions.
3. ENARI1 generates ARTOI1 which gates the initial contents of the A register to I1.
4. The adder forms the sum of the initial contents of A plus the s and t bits and feeds I3.
5. FXEQ17 and RNI status force RNI-NX inactive to disable PRTOI3. This enables the sum from the adder to the data inputs of the S register.
6. FXEQ17 enables setting of the ROP1 and ROP2 flip-flops.
7. ROP-2 and V100-6B generate CLKSR which gates the sum from I3 into S. The number in S is the memory address of the operand to be loaded into A.
8. The subsystem processor makes a memory reference and loads the one's complement of the operand into X.
9. ROP-2 and FXEQ17 enable XRFI21 and XRFI22 which gate the true value of the operand from X to I2.
10. The operand passes through the adder unchanged. SUMI4 gates the operand through I4 to I45.
11. FXEQ1X generates ENI4AR which enables CLKAR to gate the operand from I45 to A to form the final contents of A.

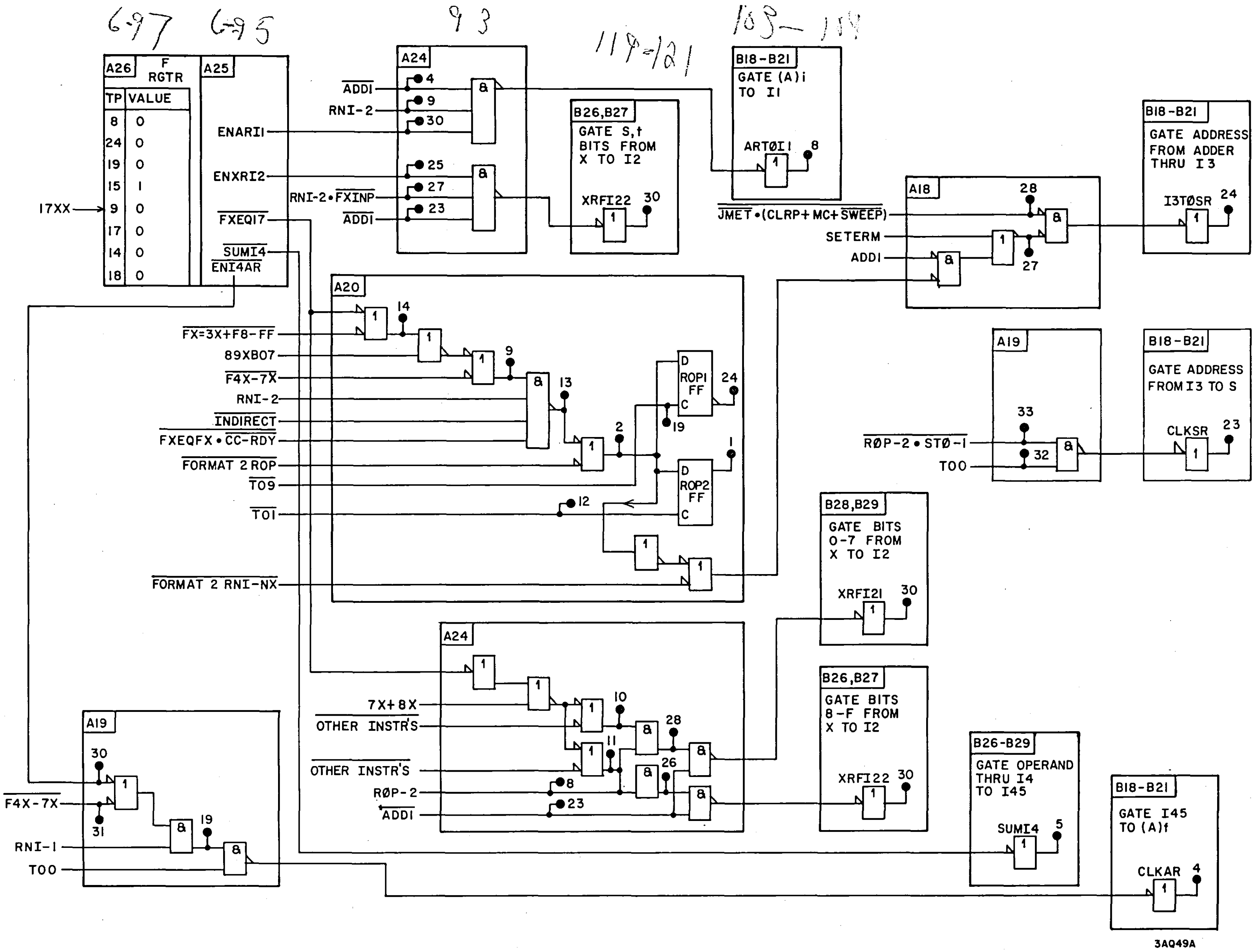


Figure 3-50. Load From (A) (17XX)

Format 2 Instructions

Format 2 instructions have the following bit format.

00 04	05	06 07	08 15
f	i	r	m

f	Function code
ir	Addressing mode designator
m	Two-hexadecimal-digit base address

The number of subsystem processor internal cycles required for a format 2 instruction depends upon the instruction and upon the selected addressing mode. The maximum number of internal cycles required per instruction is four, except for block transfer (FXXX) instructions. During block transfer instruction execution, the subsystem processor remains in STO cycle or ROP cycle mode until the transfer terminates.

Figures 3-25 and 3-26 illustrate ROP and STO cycle common to format 2 instructions.

Format 2 Instruction Address Formation

There are eight modes by which execution addresses for format 2 instructions are formed. These modes and their corresponding instruction codes are as follows:

Instruction Code	Addressing Mode
X0 or X8	Direct
X1 or X9	Direct/index B1
X2 or XA	Direct/index B2
X3 or XB	Relative forward
X4 or XC	Indirect
X5 or XD	Indirect/index B1
X6 or XE	Indirect/index B2
X7 or XF	Relative backward

The three direct modes, the relative forward mode, and the relative backward mode form the execution address during the RNI cycle of an instruction. The three indirect modes cause a RAD cycle during which the execution address is formed.

NOTE

When an indirect addressing mode is specified for a jump instruction and the jump condition is not present, the subsystem processor exits from the instruction without initiating a RAD cycle.

The entries below describe execution address formation sequences which are common to most format 2 instructions. Figure 3-23 illustrates the RNI cycle activity which results in transfer of the instruction word from memory to F and X. Figures 3-51 through 3-58 illustrate each form of address formation.

X0 or X8
Direct

1. The execution address is m and consists of the one's complement of X bits 08 through 15. ENXRI2 enables XRFI22 which gates m from X through I2 to the adder.

X1 or X9
Direct/index B1

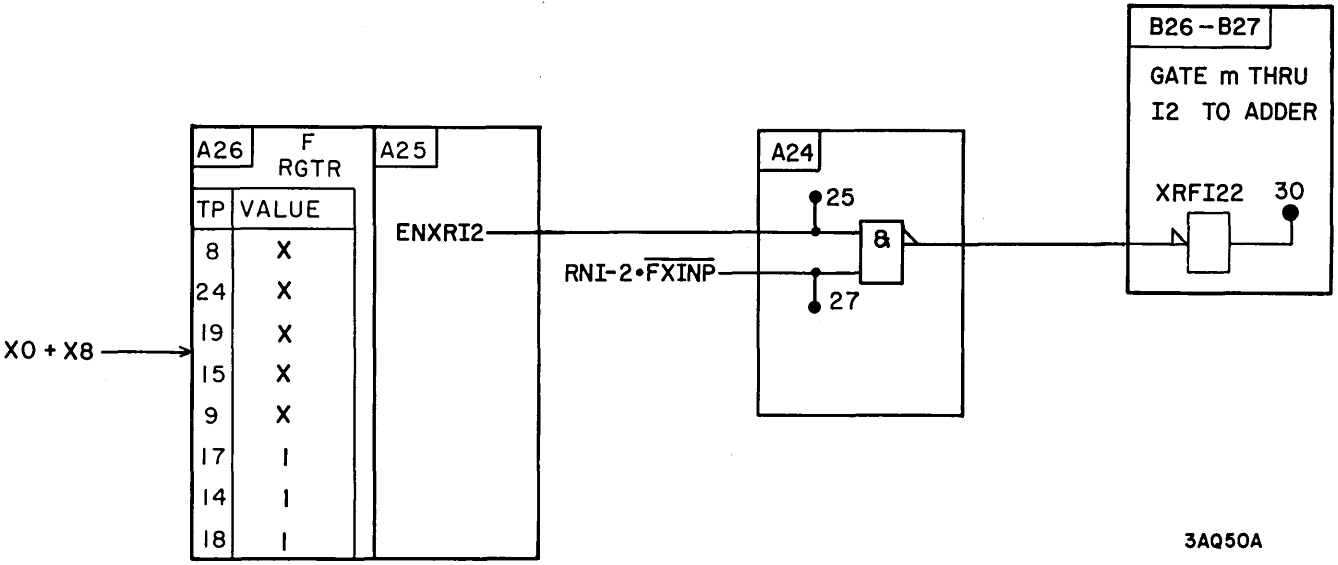
1. An X1 or X9 decode generates ENB1I1.
2. ENB1I1 enables B1TOI1 which gates (B1) through I1 to the adder.
3. ENXRI2 enables XRFI22 which gates m from X through I2 to the adder.
4. The adder forms the execution address by adding m to the contents of B1.

X2 or XA
Direct/index B2

1. An X2 or XA decode generates ENB2I1.
2. ENB2I1 enables B2TOI2 which gates (B2) through I1 to the adder.
3. ENXRI2 enables XRFI22 which gates m from X through I2 to the adder.
4. The adder forms the execution address by adding m to the contents of B2.

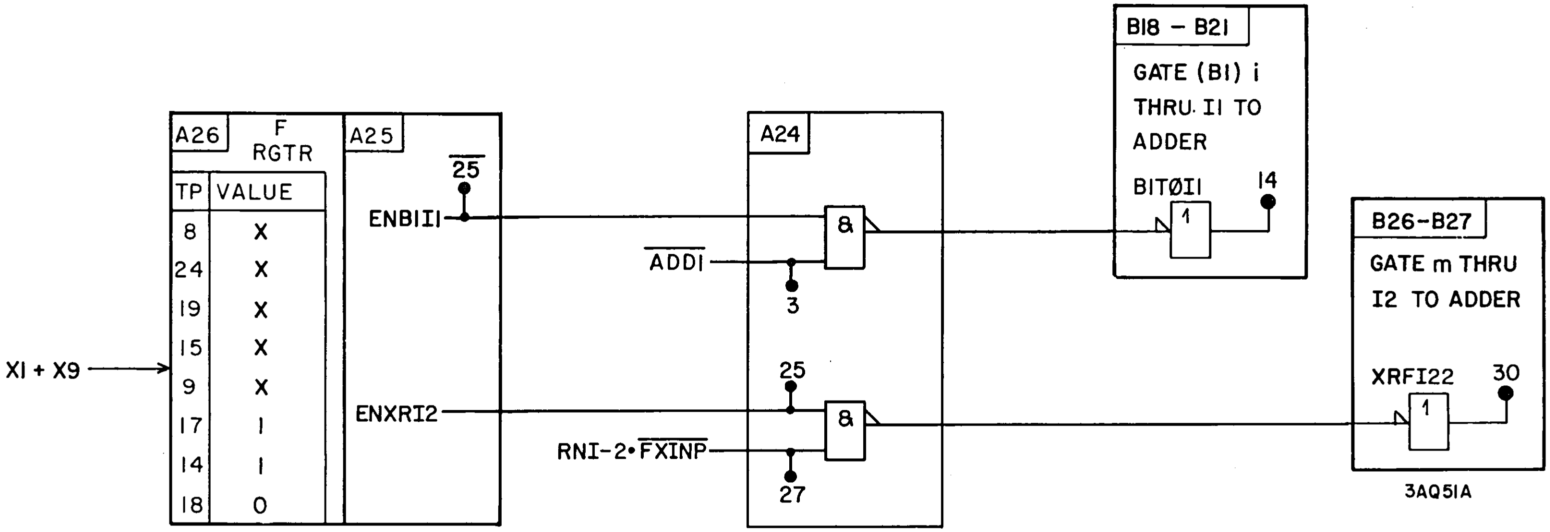
X3 or XB
Relative forward

1. FX3+XB enables SRTOI1 which gates (S) through I1 to the adder.
2. ENXRI2 enables XRFI22 which gates m from X through I2 to the adder.
3. The adder forms the execution address by adding m to the contents of S.



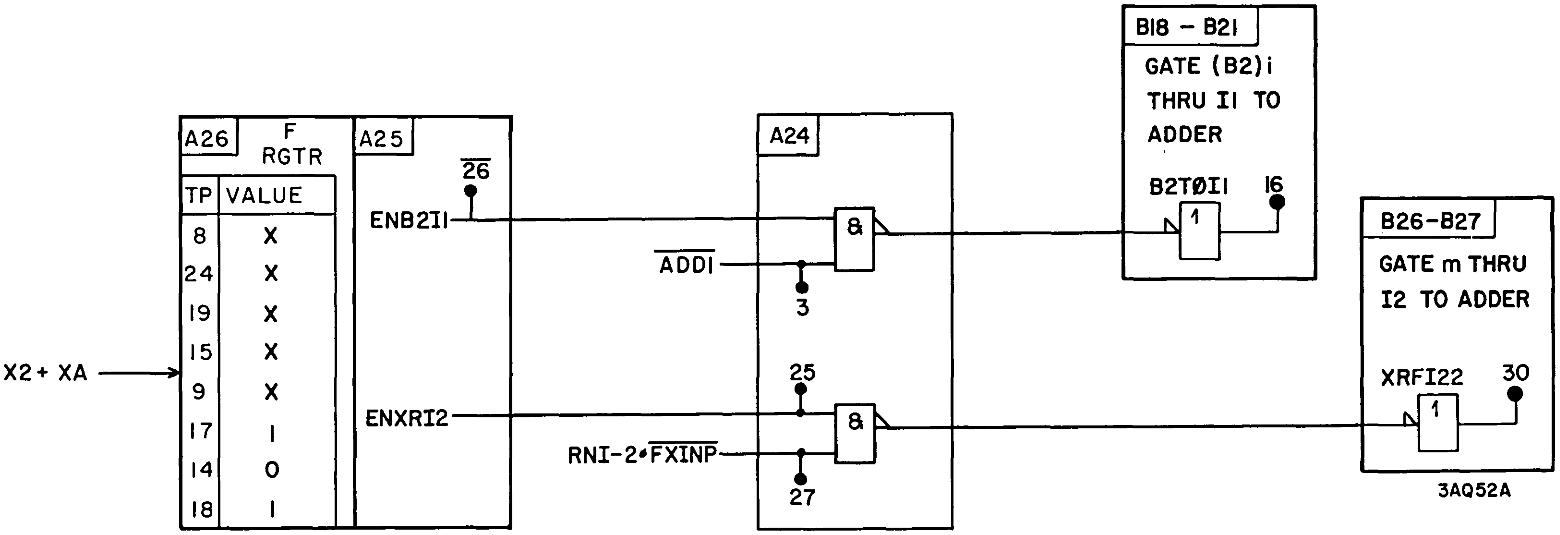
TP	VALUE
8	X
24	X
19	X
15	X
9	X
17	I
14	I
18	I

Figure 3-51. Direct (X0 or X8)



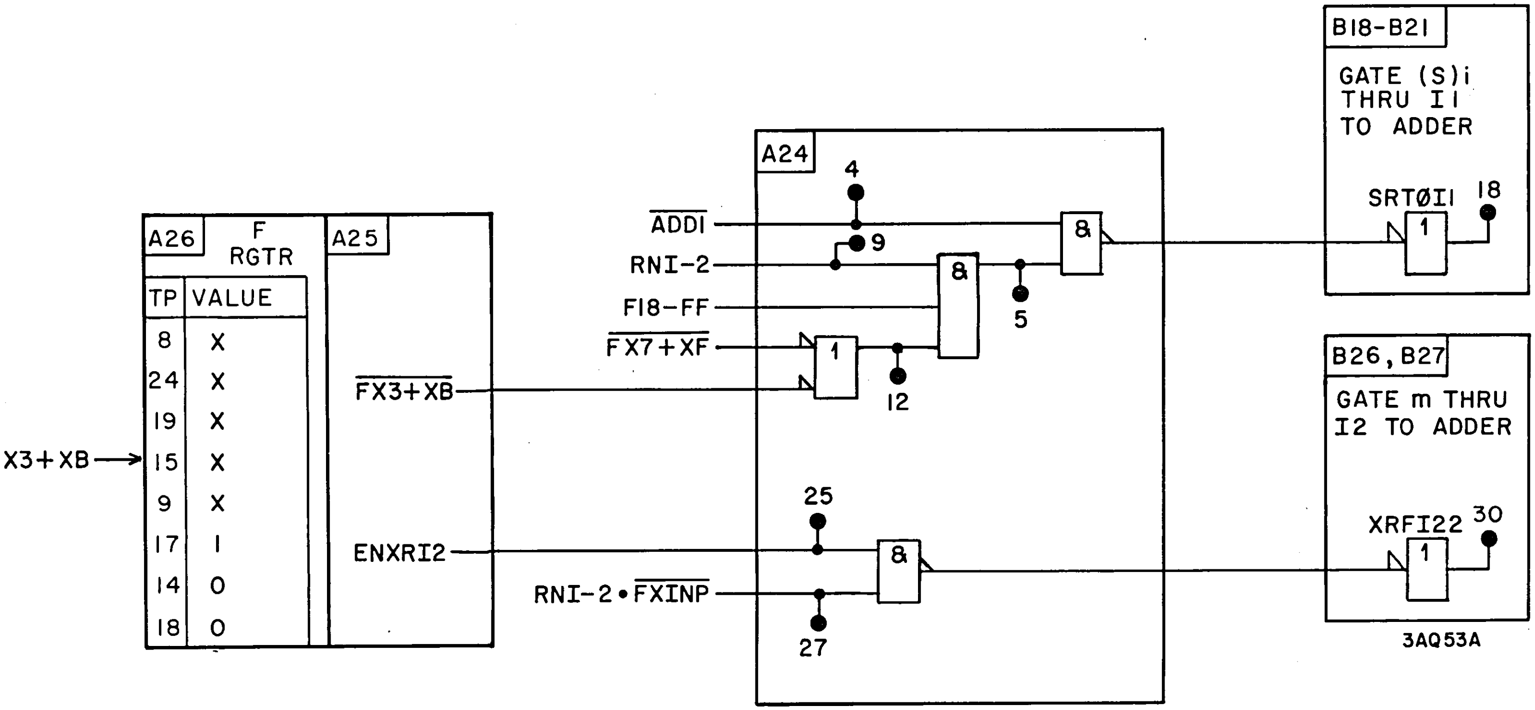
TP	VALUE
8	X
24	X
19	X
15	X
9	X
17	1
14	1
18	0

Figure 3-52. Direct/Index B1 (X1 or X9)



TP	VALUE
8	X
24	X
19	X
15	X
9	X
17	1
14	0
18	1

Figure 3-53. Direct/Index B2 (X2 + XA)



TP	VALUE
8	X
24	X
19	X
15	X
9	X
17	1
14	0
18	0

Figure 3-54. Relative Forward (X3 or XB)

X4 or XC
Indirect

1. FX4+XC forces RNI-NX inactive.
2. ENXRI2 enables XRFI22 which gates m from X through I2 to the adder. m emerges from the adder unchanged.
3. An inactive RNI-NX drives PRTOI3 to 0 so that m is gated from the adder to S by CLKSR (A07).
4. The subsystem processor makes a memory reference and loads the contents of the address specified by (S) into X.
5. FX4+XC enables the setting of the RAD2 flip-flop at T01.
6. RAD-2 enables XRFI21 and XRFI22 which gate the word from memory through I2 to the adder.
7. The output of the adder (which is the word from memory) is the execution address.

X5 or XD
Indirect/index B1

1. FX5+XD forces RNI-NX inactive.
2. ENXRI2 enables XRFI22 which gates m from X through I2 to the adder. m emerges from the adder unchanged.
3. An inactive RNI-NX drives PRTOI3 to 0 so that m is gated from the adder to S by CLKSR.
4. The subsystem processor makes a memory reference and loads the contents of the address specified by (S) into X.
5. FX5+XD enables the setting of the RAD2 flip-flop at T01.
6. RAD-2 enables XRFI21 and XRFI22 which gate the word from memory through I2 to the adder.
7. RAD-2 and X5+XD enable ENB1I1 which generates BITOI1 to gate (B1) through I1 to the adder.
8. The adder forms the execution address by adding the word from memory to the contents of B1.

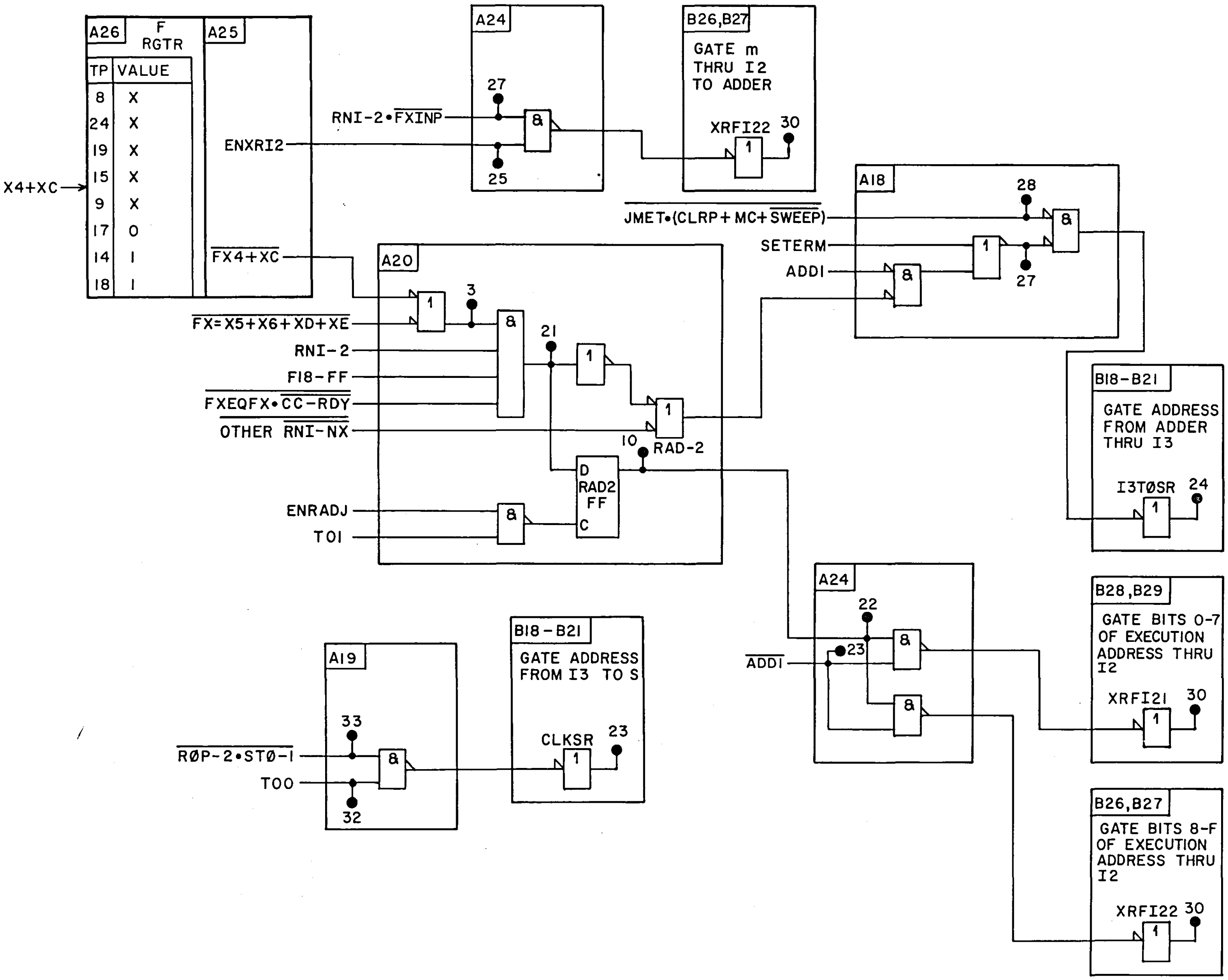


Figure 3-55. Indirect (X4 or XC)

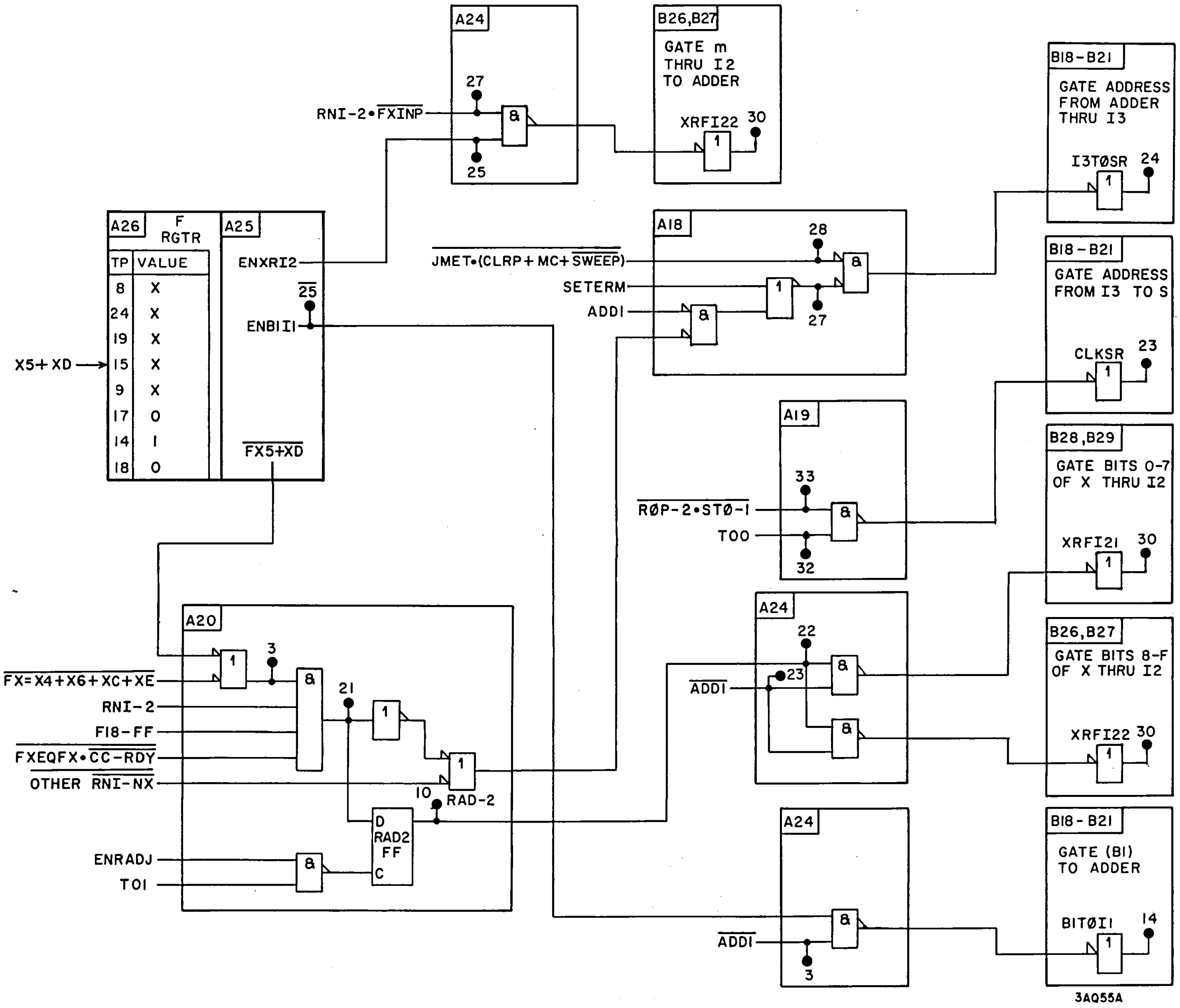


Figure 3-56. Indirect/Index B1 (X5 or XD)

X6 or XE
Indirect/index B2

1. FX6+XE forces RNI-NX inactive.
2. ENXRI2 enables XRFI22 which gates m from X through I2 to the adder. m emerges from the adder unchanged.
3. An inactive RNI-NX drives PRTOI3 to 0 so that m is gated from the adder to S by CLKSR.
4. The subsystem processor makes a memory reference and loads the contents of the address specified by (S) into X.
5. FX6+XE enables the setting of the RAD2 flip-flop at T01.
6. RAD-2 enables XRFI21 and XRFI22 which gate the word from memory through I2 to the adder.
7. RAD-2 and X6+XE enable ENB2I1 which generates B2TOI1 to gate (B2) through I1 to the adder.
8. The adder forms the execution address by adding the word from memory to the contents of B2.

X7 or XF
Relative backward

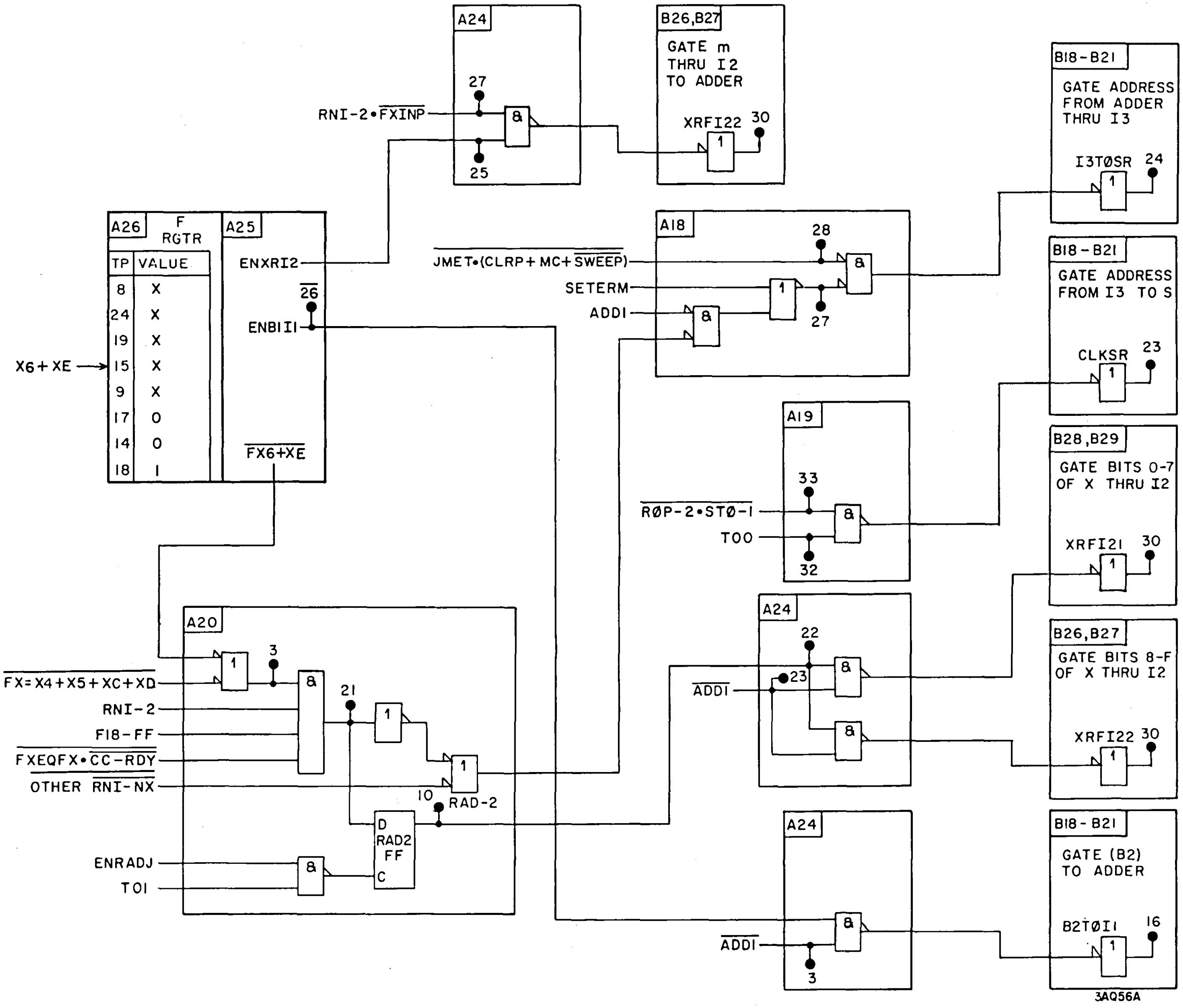
1. FX7+XF enables SRTOI1 which gates (S) through I1 to the adder.
2. ENNXI2 enables XRTI2 and XRFI22 which gate the one's complement of m (with sign extended) from X through I2 to the adder.
3. CARRY4 feeds the adder to form the two's complement result as m is subtracted from (S). The result is the execution address.

Enter A with Address (18XX through 1FXX)

1. SUMI4 gates the execution address from the adder through I4 to I45.
2. FXEQ1X generates ENI4AR which enables CLKAR to gate the execution address from I45 to A.

Enter B1 with Address (20XX through 27XX)

1. SUMI4 gates the execution address from the adder through I4 to I45.
2. F20-27 generates ENI4B1 which enables CLKB1 to gate the execution address from I45 to B1.



TP	VALUE
8	X
24	X
19	X
15	X
9	X
17	0
14	0
18	1

Figure 3-57. Indirect/Index B2 (X6 or XE)

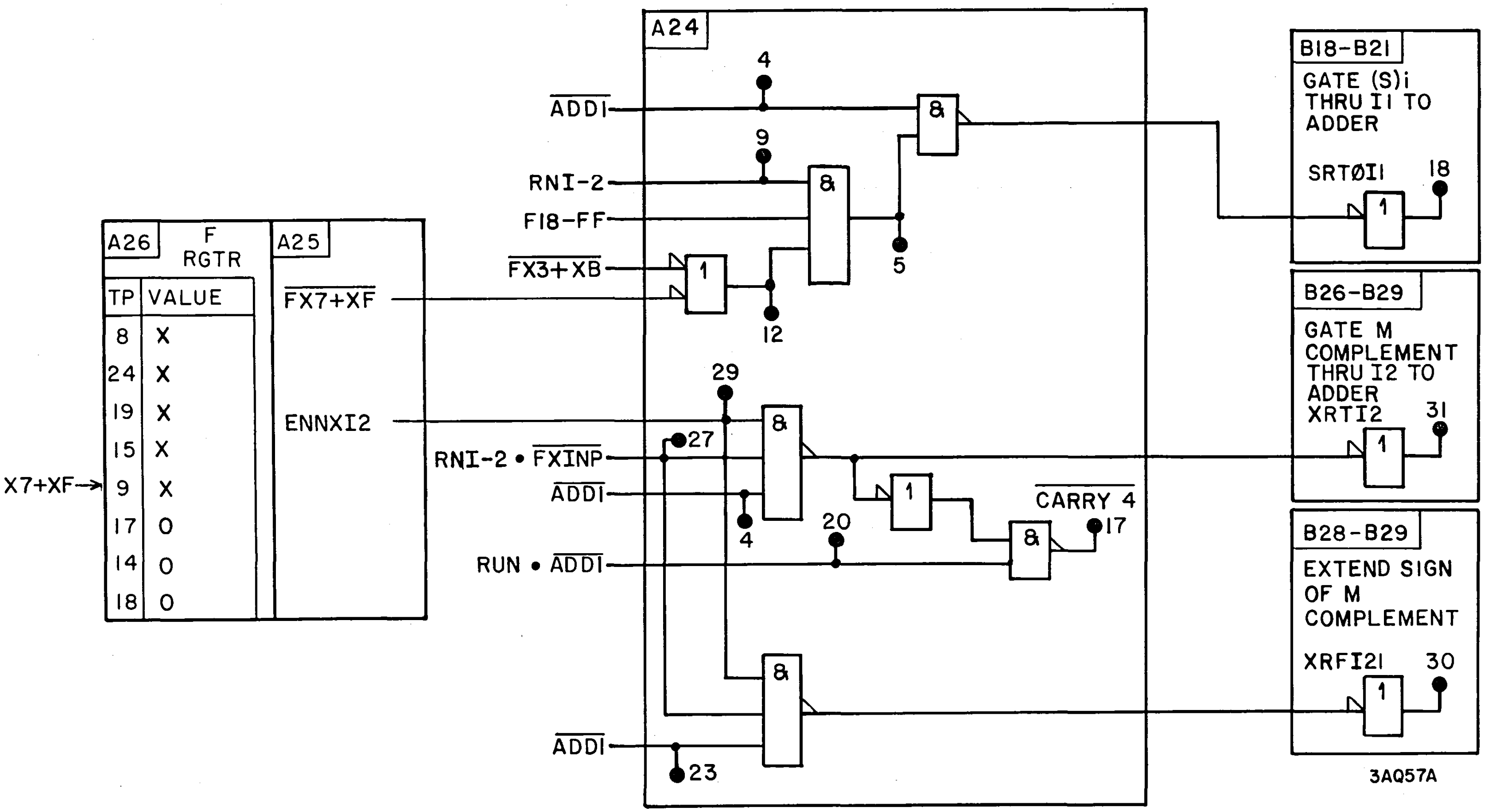
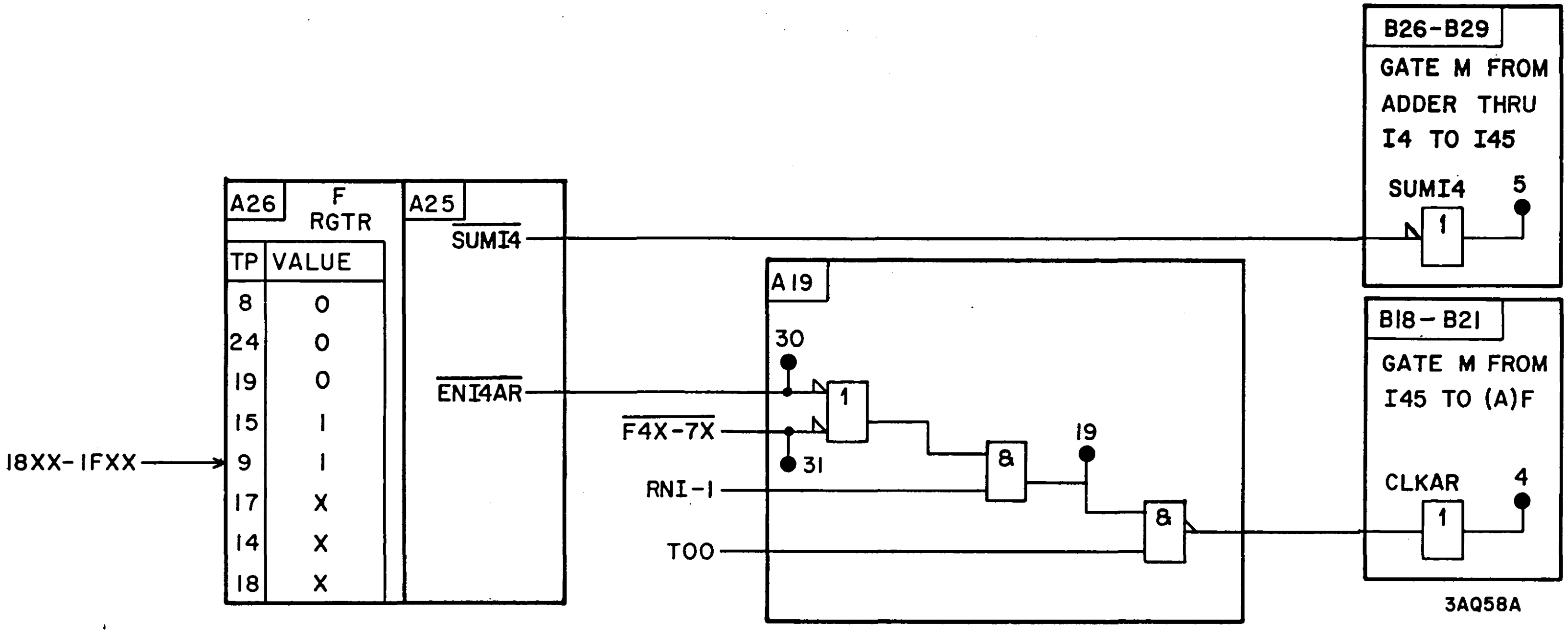


Figure 3-58. Relative Backward (X7 or XF)



TP	VALUE
8	0
24	0
19	0
15	1
9	1
17	X
14	X
18	X

A26
F RGTR
A25
$\overline{\text{SUMI4}}$
$\overline{\text{ENI4AR}}$
18XX-1FXX
$\overline{\text{F4X-7X}}$
RNI-1
T00
A19
30
31
1
&
19
&
B26-B29
GATE M FROM ADDER THRU I4 TO I45
SUMI4
5
1
B18-B21
GATE M FROM I45 TO (A)F
CLKAR
4
1
3AQ58A

Figure 3-59. Enter A with Address (18XX-1FXX)

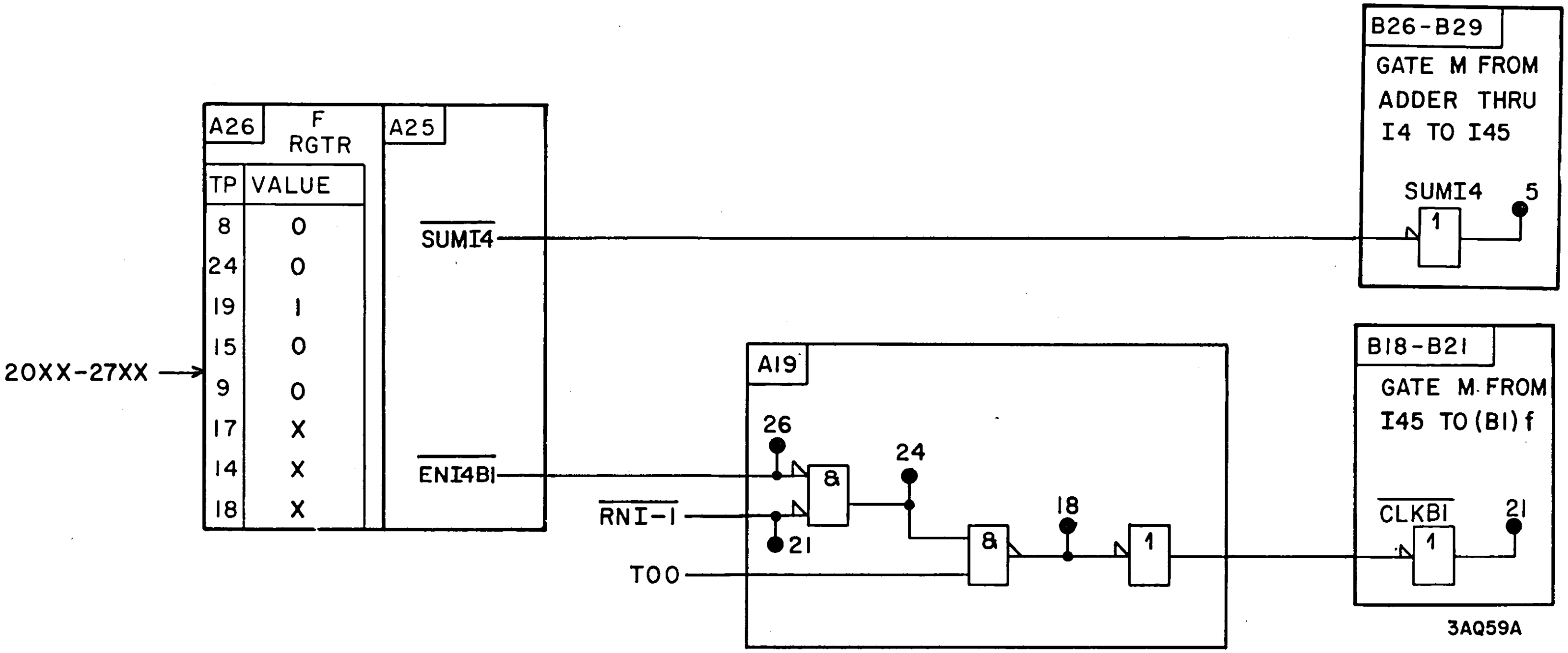


Figure 3-60. Enter B1 with Address (20XX-27XX)

Enter B2 with Address (28XX through 2FXX)

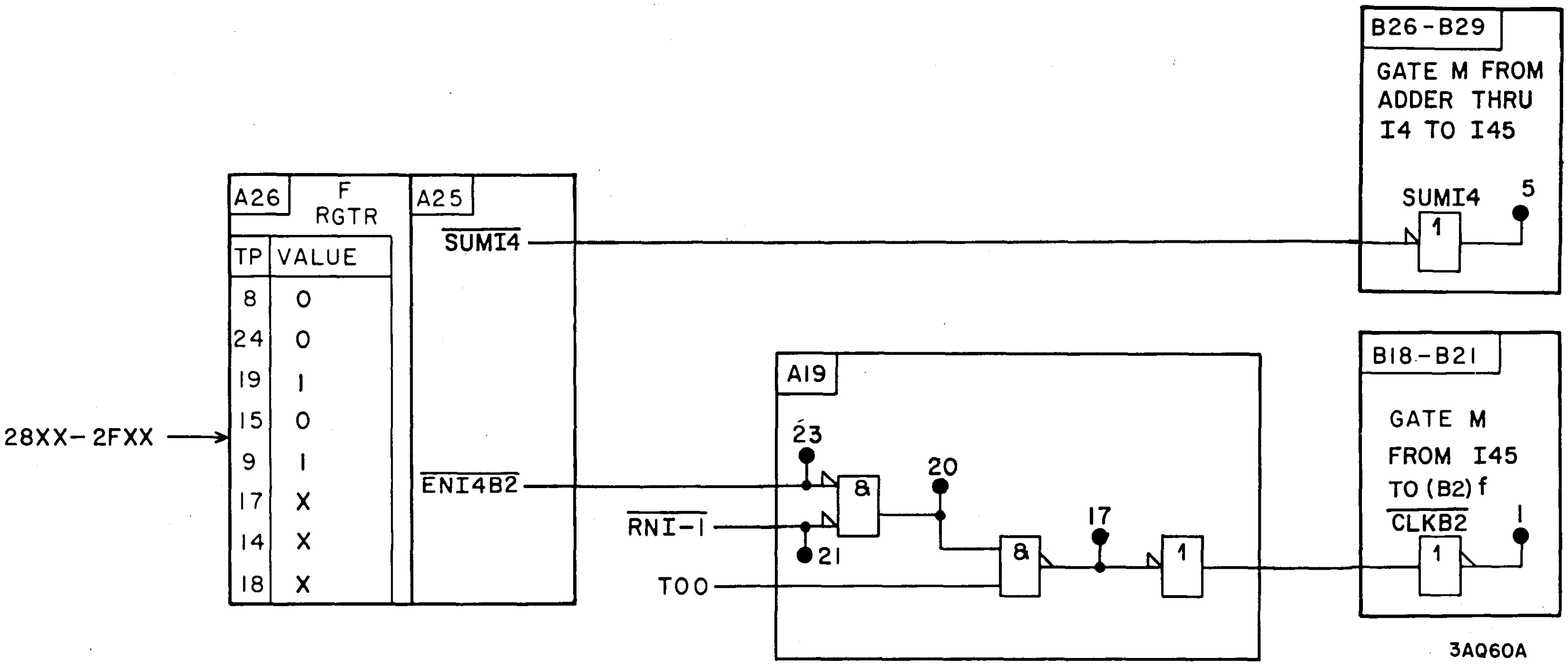
1. SUMI4 gates the execution address from the adder through I4 to I45.
2. F28-2F generates ENI4B2 which enables CLKB2 to gate the execution address from I45 to B2.

Test Index B1 (30XX through 37XX)

1. During the ROP cycle of the instruction, F30-37 enables ENB1I1 which generates B1TOI1 to gate (B1) through I1 to B compare logic and the adder.
2. The operand read from the execution address during the ROP cycle feeds B compare logic from X.
3. FXEQ3X enables FORCE to generate CARRY4 which increments (B1) in the adder.
4. SUMI4 gates (B1) + 1 from the adder through I4 to I45.
5. When a compare occurs between the operand and (B1), COND-FF is active to disable ENI4B1 and prevent CLKB1. The subsystem processor exits to an RNI cycle and the COND flip-flop is set at T00.
6. When a compare does not occur between the operand and (B1), COND-FF is inactive to enable ENI4B1 which generates CLKB1 to gate (B1) + 1 from I45 to B1. The subsystem processor exits to an RNI cycle and the COND flip-flop is cleared at T00.

Test Index B2 (38XX through 3FXX)

1. During the ROP cycle of the instruction, F38-3F enables ENB2I1 which generates B2TOI1 to gate (B2) through I1 to B compare logic and the adder.
2. The operand read from the execution address during the ROP cycle feeds B compare logic from X.
3. FXEQ3X enables FORCE to generate CARRY4 which increments (B2) in the adder.
4. SUMI4 gates (B2) + 1 from the adder through I4 to I45.



A26 F RGTR

TP	VALUE
8	0
24	0
19	1
15	0
9	1
17	X
14	X
18	X

Figure 3-61. Enter B2 with Address (28XX-2FXX)

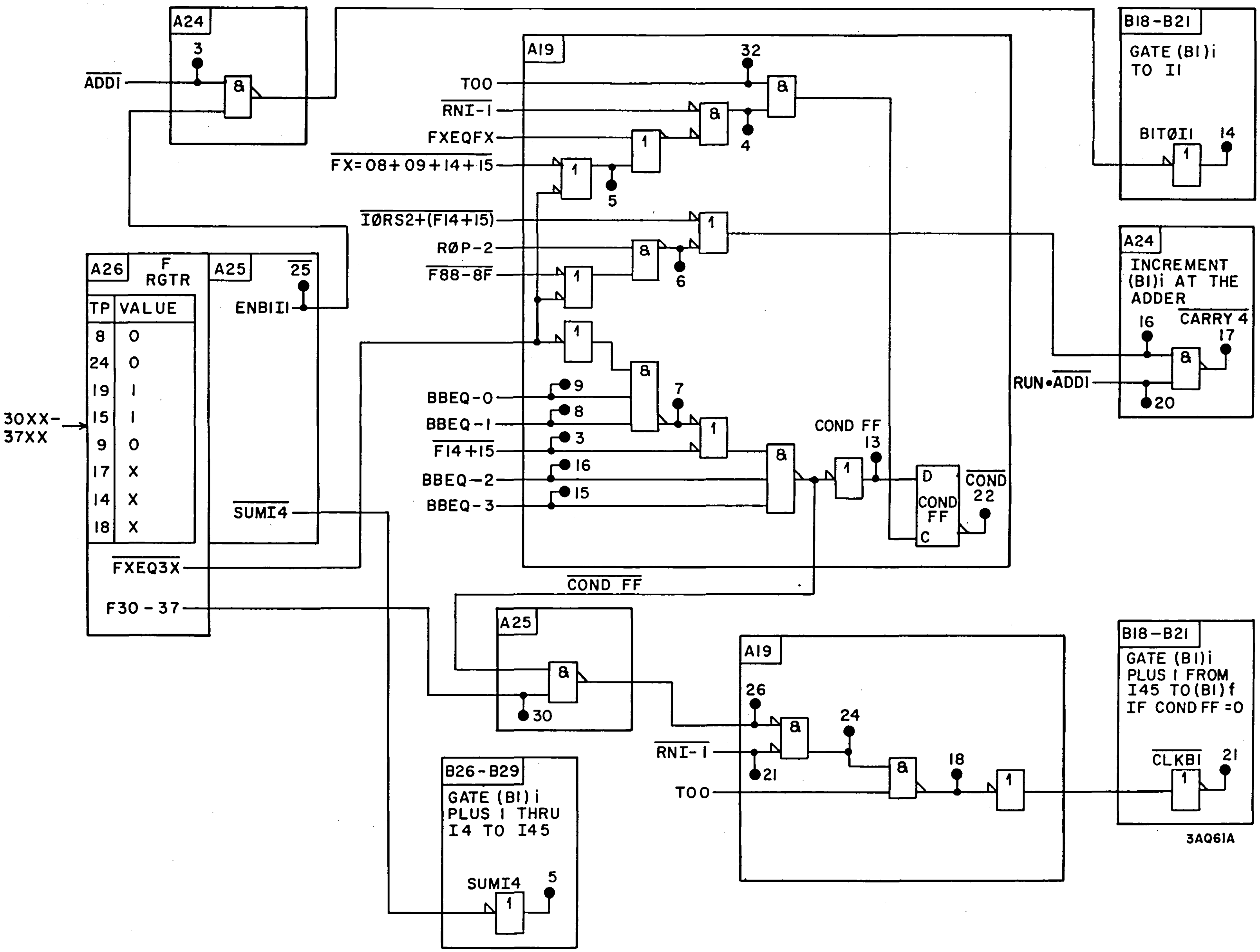


Figure 3-62. Test Index B1 (30XX-37XX)

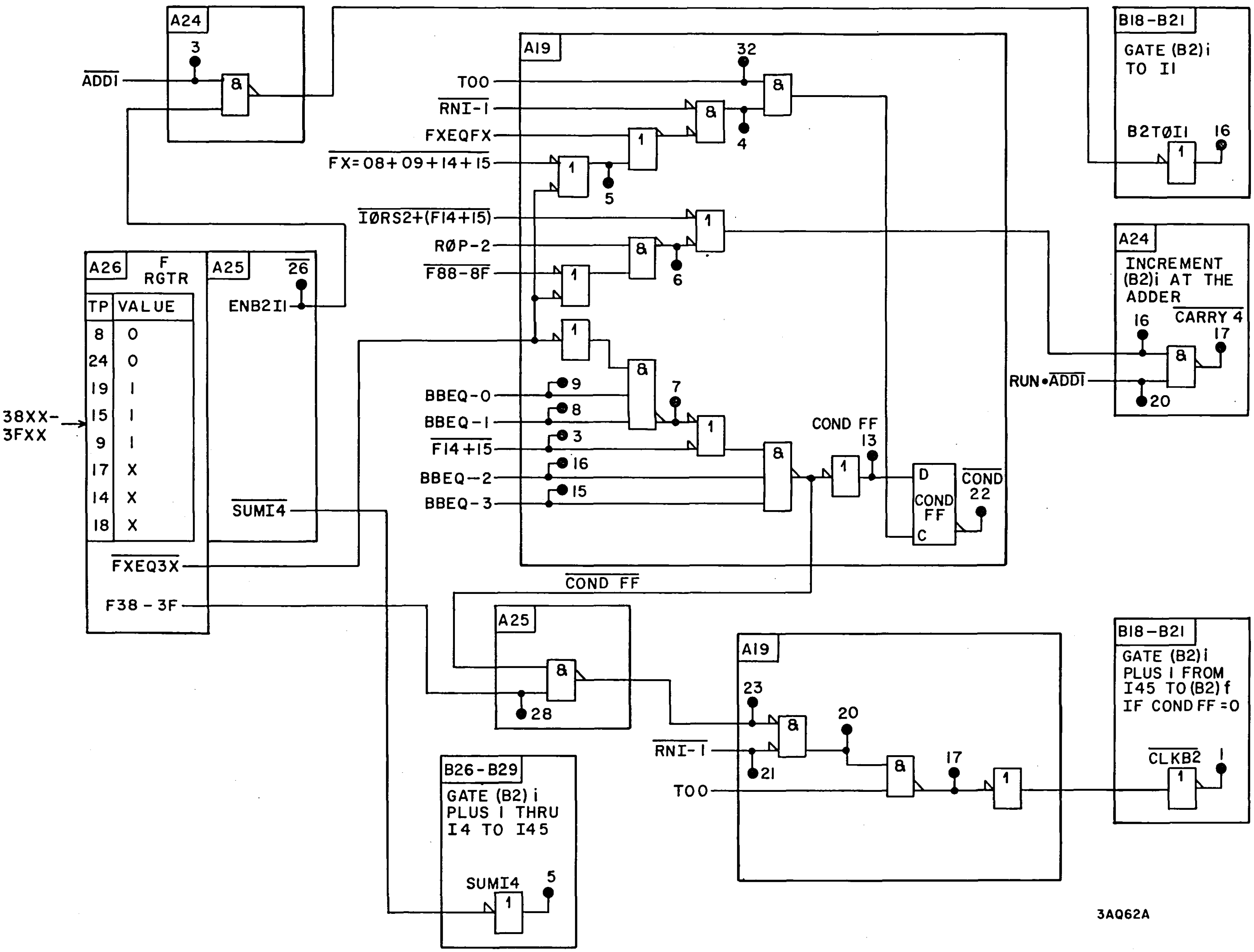


Figure 3-63. Test Index B2 (38XX-3FXX)

5. When a compare occurs between the operand and (B2), COND-FF is active to disable ENI4B2 and prevent CLKB2. The subsystem processor exits to an RNI cycle and the COND flip-flop is set at T00.
6. When a compare does not occur between the operand and (B2), COND-FF is inactive to enable ENI4B2 which generates CLKB2 to gate (B2) + 1 from I45 to B2. The subsystem processor exits to an RNI cycle and the COND flip-flop is cleared at T00.

Load A (40XX through 47XX)

1. During the ROP cycle of the instruction, an inactive FX8-XF signal combines with FXEQ4X to enable XRFI21 and XRFI22. These signals gate the operand (obtained from the execution address) from X through I2 to the adder. The operand passes through the adder unchanged.
2. SUMI4 gates the operand from the adder through I4 to I45.
3. The subsystem processor exits to an RNI cycle. F4X-7X enables CLKAR at T00 to gate the operand from I45 to A.

Load A Complement (48XX through 4FXX)

1. During the ROP cycle of the instruction, FXEQ4X combines with FX8-XF to enable XRTI2 which gates the one's complement of the operand (obtained from the execution address) from X through I2 to the adder.
2. XRTI2 enables CARRY4 which feeds the adder to convert the one's complement of the operand to two's complement form.
3. SUMI4 gates the two's complement of the operand from the adder I4 to I45.
4. The subsystem processor exits to an RNI cycle. F4X-7X enables CLKAR at T00 to gate the two's complement of the operand from I45 to A.

Load Leftmost Byte (50XX through 57XX)

1. During the ROP cycle of the instruction, F50-57 enables XRFI21 which gates the leftmost byte of the operand (obtained from the execution address) through I2 to the adder. The rightmost byte from I2 contains 0-fill.
2. SUMI4 gates the leftmost byte of the operand from the adder through I4 to I45.

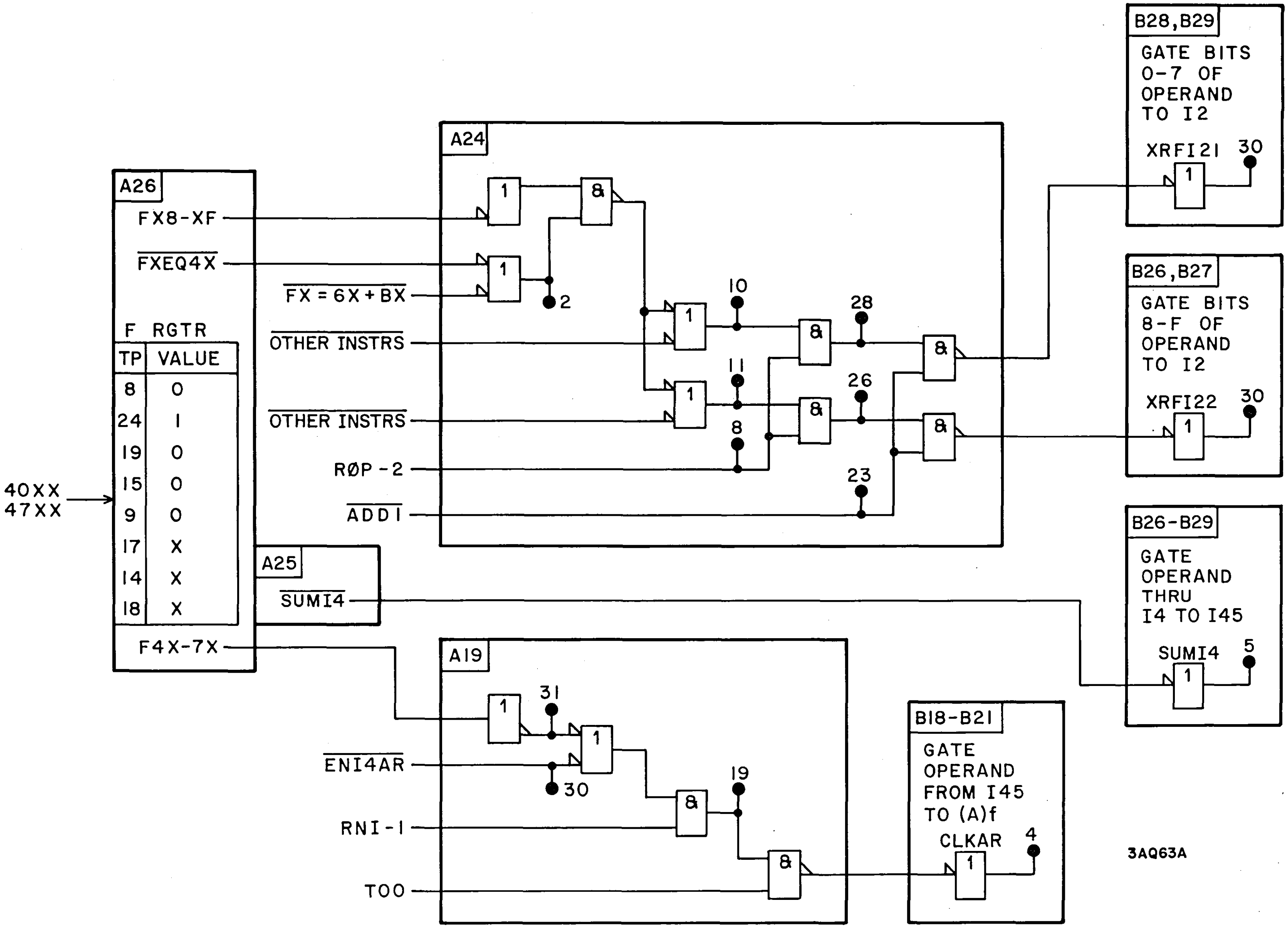


Figure 3-64. Load A (40XX-47XX)

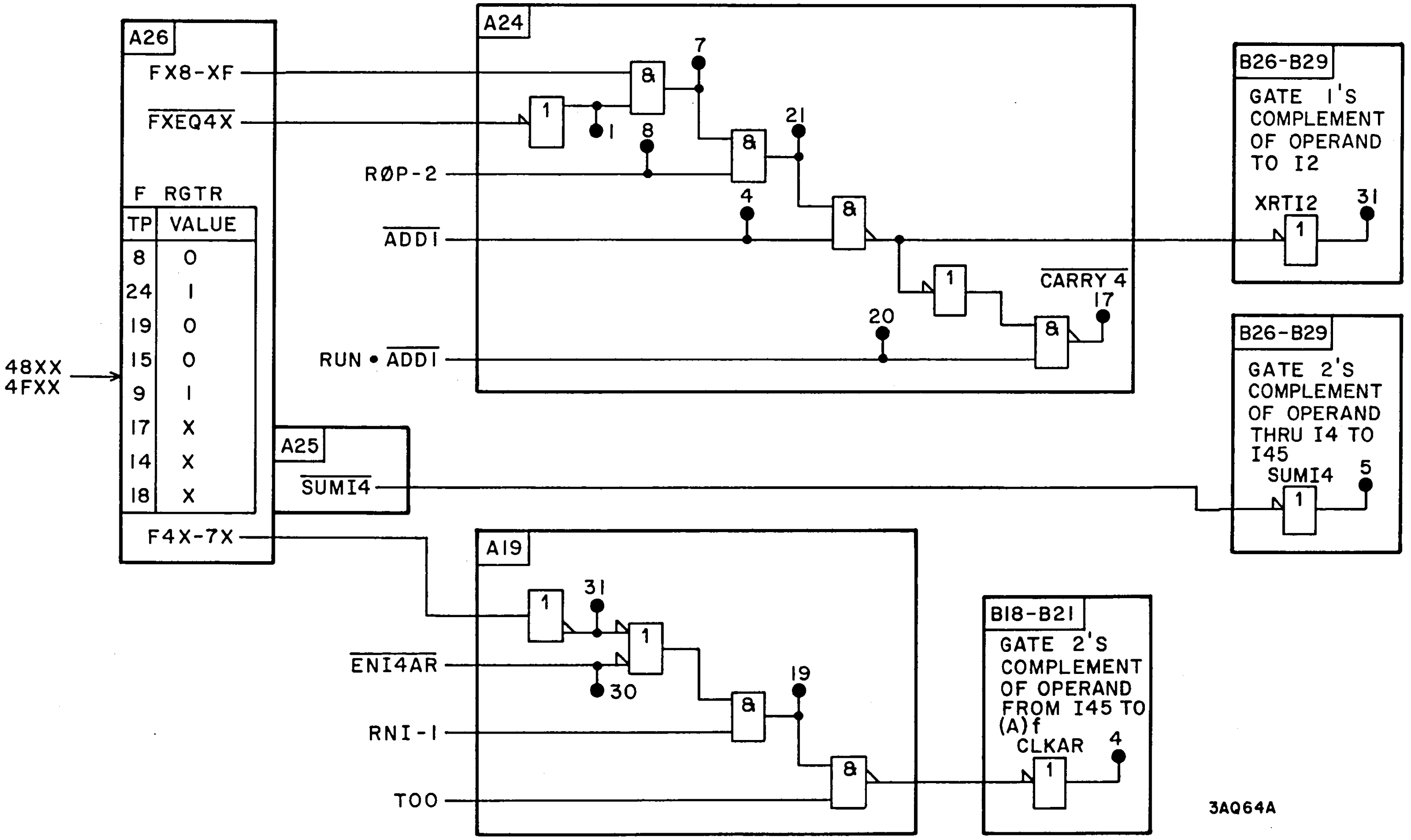


Figure 3-65. Load A Complement (48XX-4FXX)

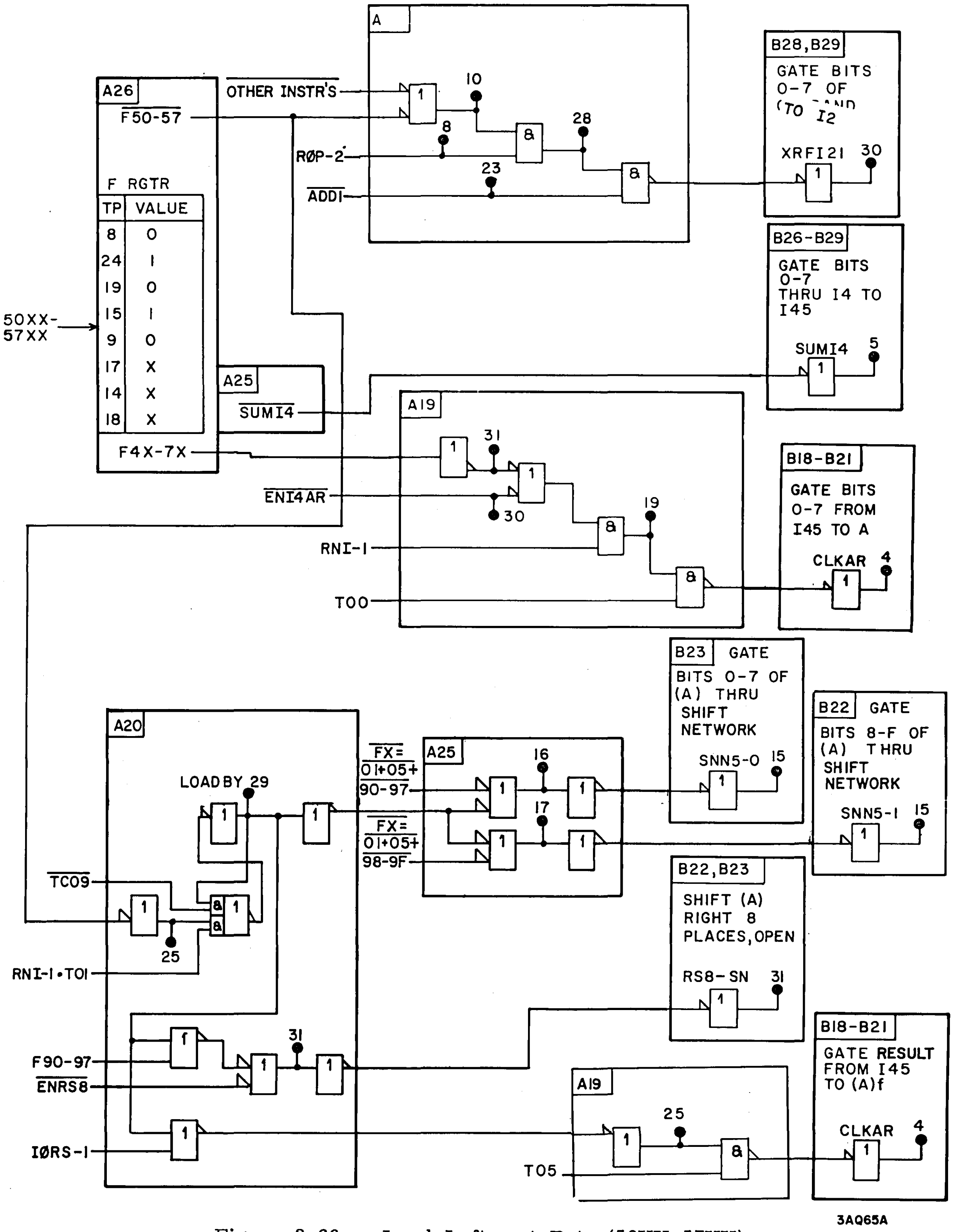


Figure 3-66. Load Leftmost Byte (50XX-57XX)

3. The subsystem processor exits to an RNI cycle. F4X-7X enables CLKAR at T00 to gate the leftmost byte of the operand from I45 to the leftmost byte position of A.

4. F50-57 generates RS8-SN, LOADBY, and IO+LBY. These signals remain active from T01 until T09 of the current RNI cycle, although the next instruction word from memory is loaded into F and X at T06.

5. RS8-SN causes the shift network to shift (A) right eight places. Since STRIP is active, the shift is endoff. After the shift, the leftmost byte of the operand emerges from the rightmost byte position of the shift network and the leftmost byte position of the shift network contains 0-fill.

6. LOADBY generates SNN5-0 and SNN5-1 which gate the shift network output to I45. The lower 5 bits pass through N6 before feeding I45.

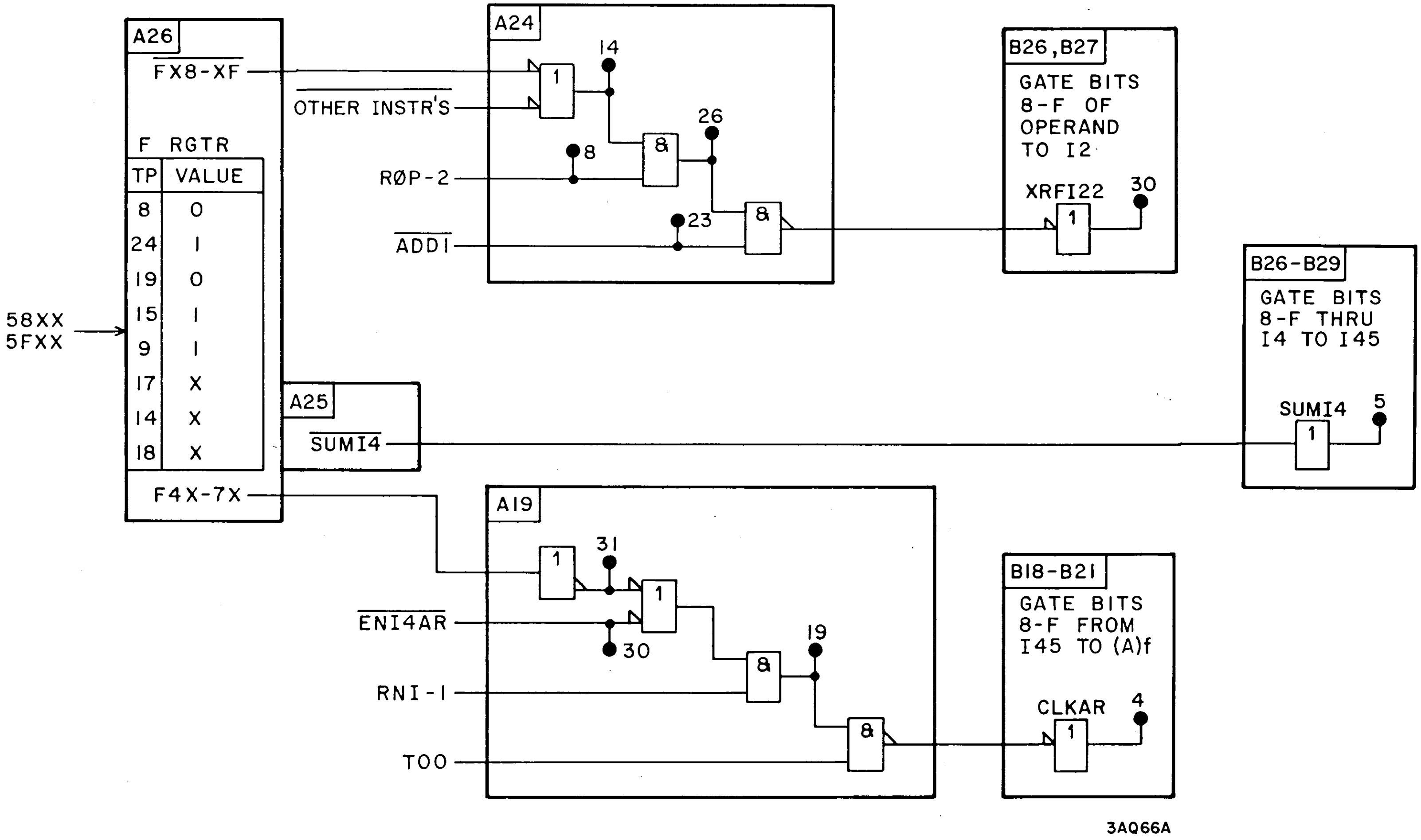
7. At T05, IO+LBY enables CLKAR which gates the shift network output from I45 to A. The rightmost byte position of A now contains the leftmost byte of the operand, and the leftmost byte position of A contains 0-fill.

Load Rightmost Byte (58XX through 5FXX)

1. During the ROP cycle of the instruction, F58-5F enables XRFI22 which gates the rightmost byte of the operand (obtained from the execution address) through I2 to the adder. The leftmost byte from I2 contains 0-fill.

2. SUMI4 gates the rightmost byte of the operand from the adder through I4 to I45.

3. The subsystem processor exits to an RNI cycle. F4X-7X enables CLKAR at T00 to gate the rightmost byte of the operand from I45 to the rightmost byte position of A. The leftmost byte position of A contains 0-fill.



TP	VALUE
8	0
24	1
19	0
15	1
9	1
17	X
14	X
18	X

Figure 3-67. Load Rightmost Byte (58XX-5FXX)

Add (60XX through 67XX)

1. During the ROP cycle of the instruction, an inactive FX8-XF signal combines with FXEQ6X to enable XRFI21 and XRFI22. These signals gate the operand (obtained from the execution address) from X through I2 to the adder.
2. FXEQ6X enables ARTOI1 which gates the initial contents of A through I1 to the adder.
3. The adder forms the sum of the initial contents of A and the operand. SUMI4 gates the sum through I4 to I45.
4. FXEQ6X enables clocking of the ENDF flip-flop at T00.
5. The subsystem processor exits to an RNI cycle. F4X-7X enables CLKAR at T00 to gate the result of the add from I45 to A.

Subtract (68XX - 6FXX)

1. During the ROP cycle of the instruction, FXEQ6X combines with FX8 through XF to enable XRTI2 which gates the one's complement of the operand (obtained from the execution address) from X through I2 to the adder.
2. FXEQ6X enables ARTOI1 which gates the initial contents of A through to the adder.
3. XRTI2 enables CARRY4 which feeds the adder to form the two's complement result as the operand is subtracted from the initial contents of A. SUMI4 gates the result of the subtract operation through I4 to I45.
4. FXEQ6X enables clocking of the ENDF flip-flop at T00.
5. The subsystem processor exits to an RNI cycle. F4X-7X enables CLKAR at T00 to gate the result of the subtract operation from I45 to A.

Exclusive OR (70XX through 77XX)

1. During the ROP cycle of the instruction, FXEQ7X enables XRFI21 and XRFI22 which gate the operand (obtained from the execution address) from X through I2.
2. FXEQ7X enables ARTOI1 which gates the initial contents of A through I1.

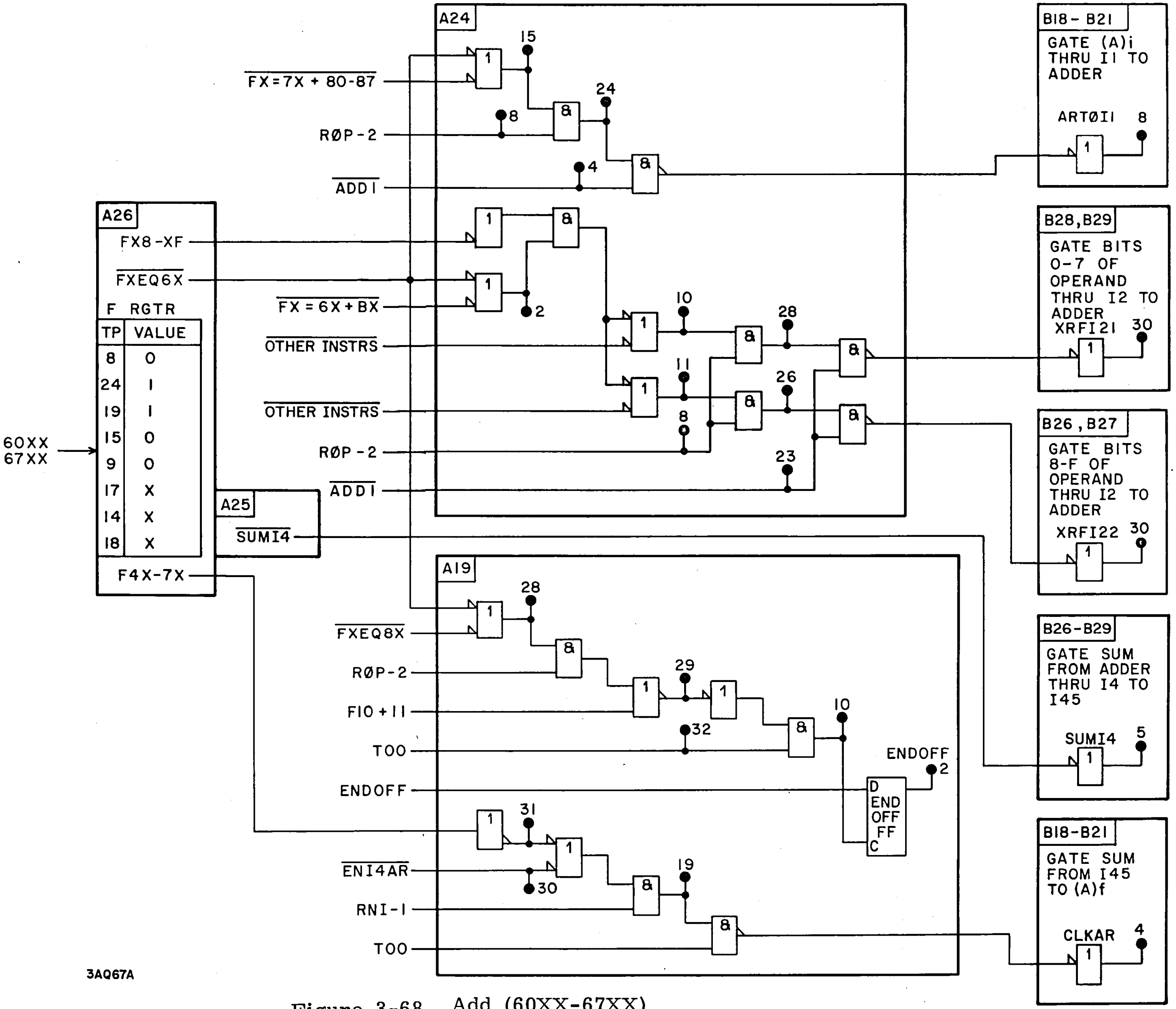


Figure 3-68. Add (60XX-67XX)

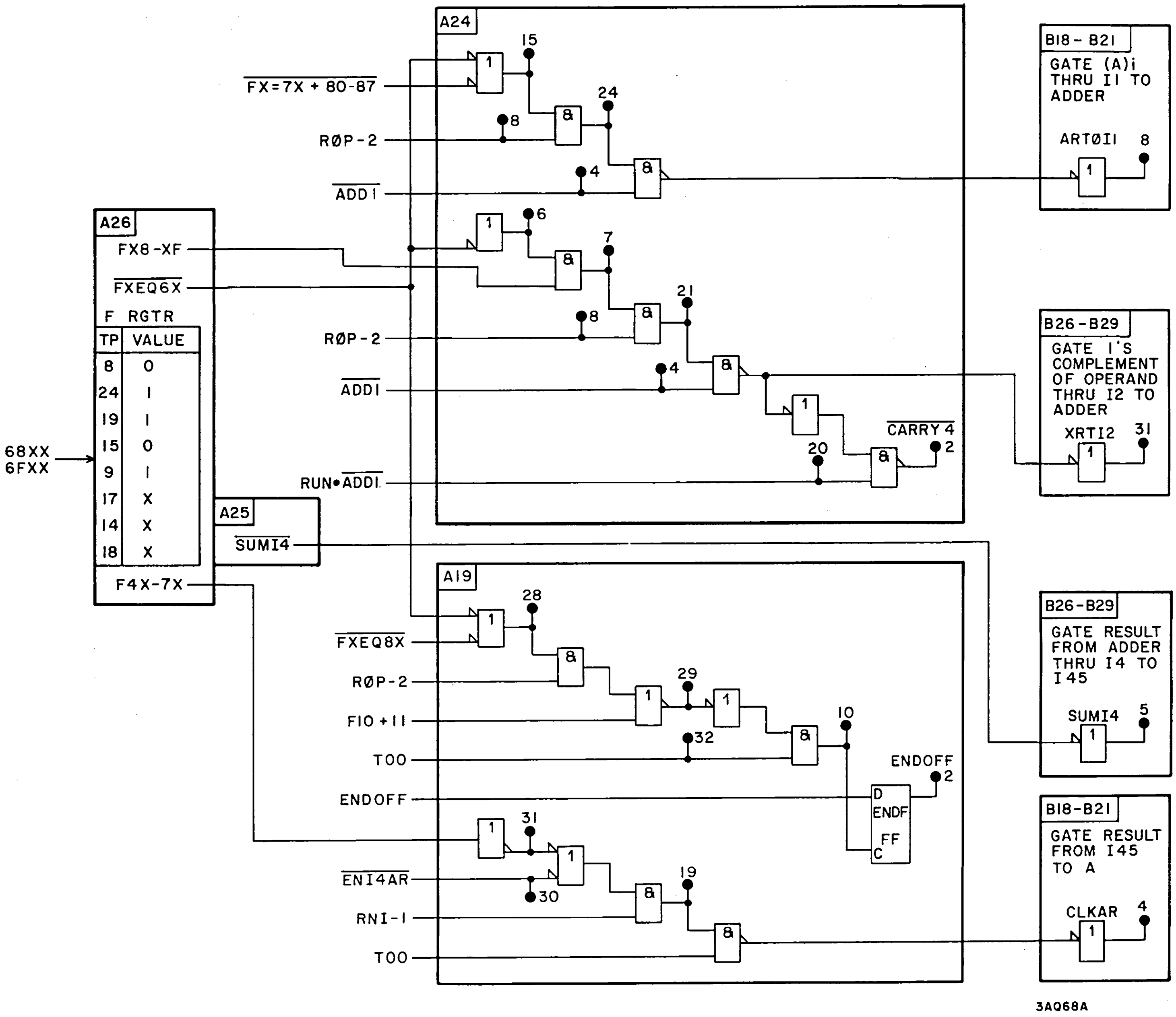


Figure 3-69. Subtract (68XX-6FXX)

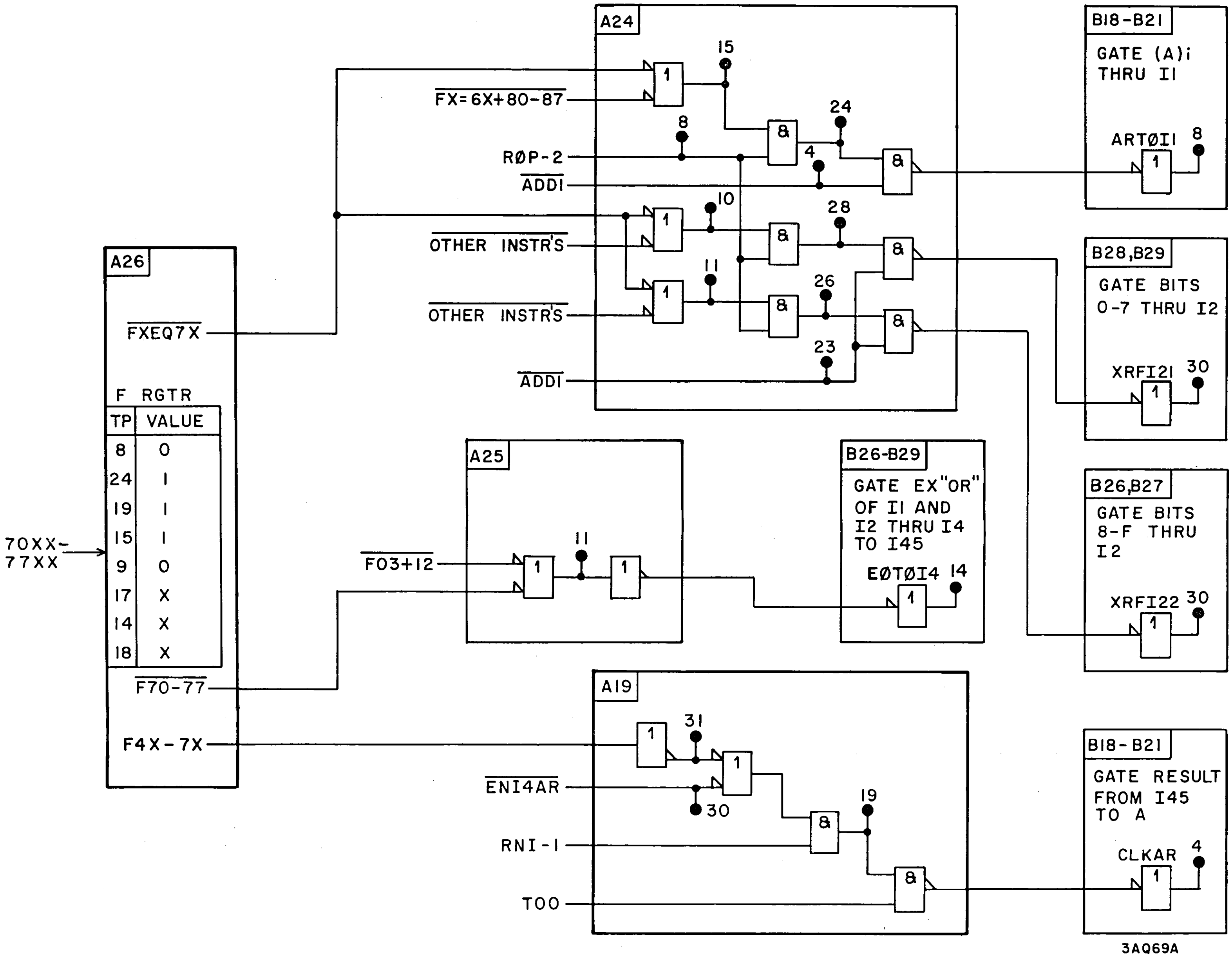


Figure 3-70. Exclusive OR (70XX-77XX)

3. F70 through 77 generates EOTOI4 which gates the exclusive OR of I1 and I2 through I4 to I45.

4. The subsystem processor exits to an RNI cycle. F4X-7X enables CLKAR at T00 to gate the result of the exclusive OR from I45 to A.

Logical Product (78XX through 7FXX)

1. During the ROP cycle of the instruction, FXEQ7X enables XRFI21 and XRFI22 which gate the operand (obtained from the execution address) from X through I2.

2. FXEQ7X enables ARTOI1 which gates the initial contents of A through I1.

3. F78-7F generates ANDI4 which gates the AND of I1 and I2 through I4 to I45.

4. The subsystem processor exits to an RNI cycle. F4X-7X enables CLKAR at T00 to gate the result of the AND from I45 to A.

Replace Add (80XX through 87XX)

1. During the ROP cycle of the instruction, FXEQ8X enables XRFI21 and XRFI22 which gate the operand (obtained from the execution address) from X through I2 to the adder.

2. FXEQ8X enables ARTOI1 which gates the initial contents of A through I1 to the adder.

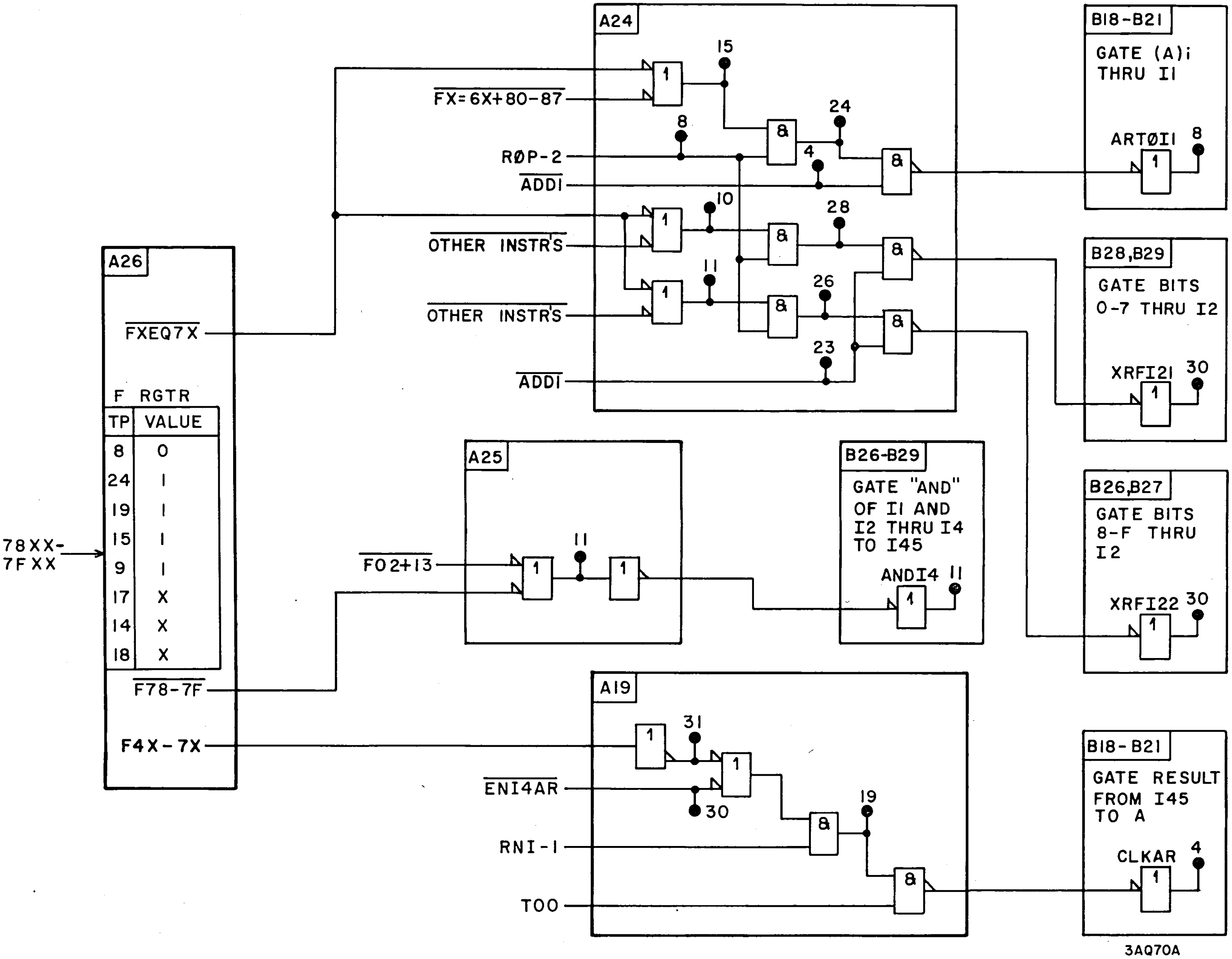
3. The adder forms the sum of the initial contents of A and the operand. SUMI4 gates the sum through I4 to I45.

4. FXEQ8X enables clocking of the ENDF flip-flop at T00.

5. 89XB07 enables CLKAR at T00 to gate the result of the add from I45 to A.

6. 89XB07 causes the subsystem processor to exit to a STO cycle. CLKSR is disabled for STO cycles immediately following ROP cycles, so the execution address for the store is the same as the address the operand was read from.

7. J0TOI5 is inactive, so the contents of A is enabled to I5 for transfer to memory. Figure 3-26 illustrates the remaining STO cycle activity for the instruction.



F RGTR

TP	VALUE
8	0
24	1
19	1
15	1
9	1
17	X
14	X
18	X

Figure 3-71. Logical Product (78XX-7FXX)

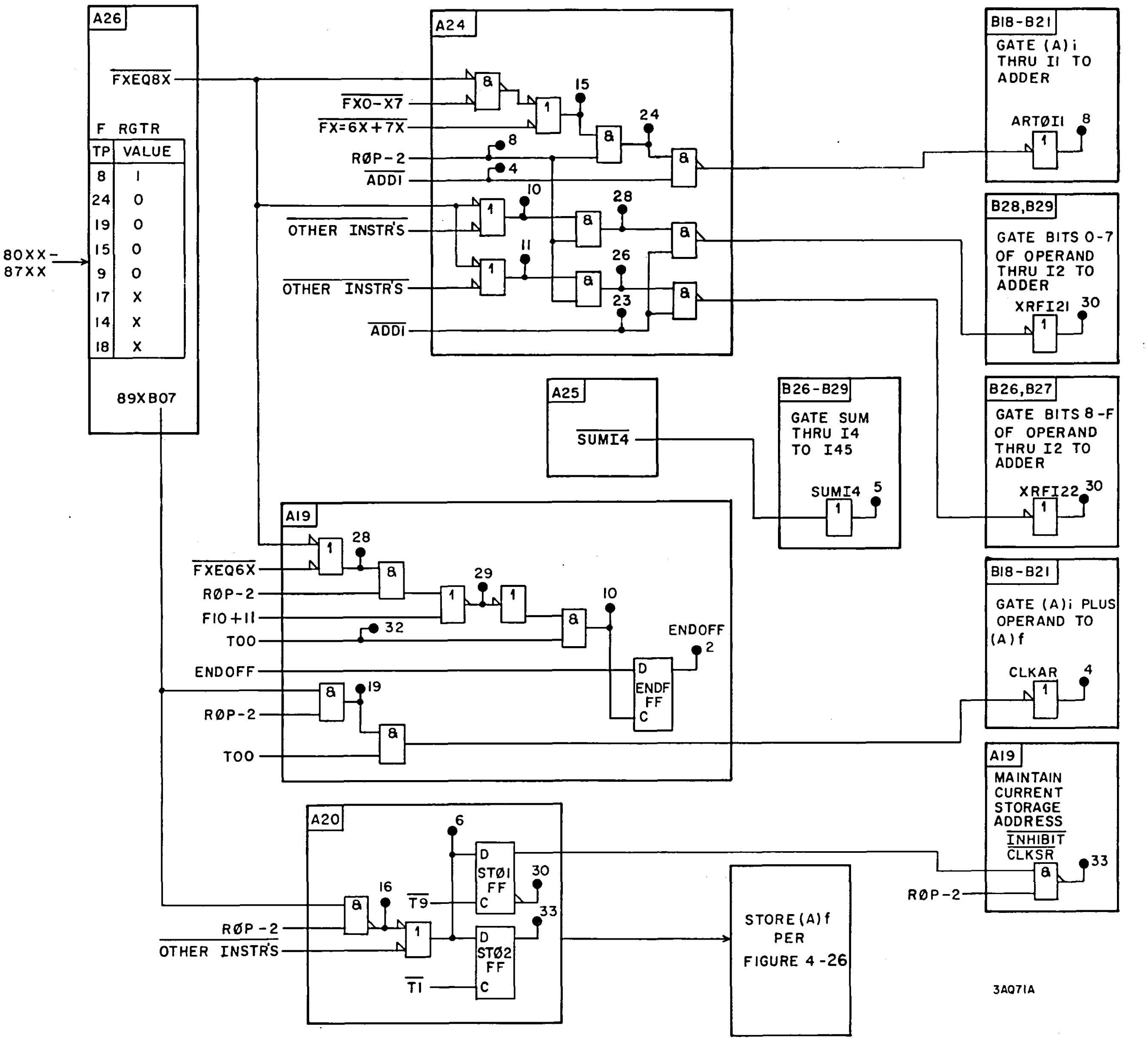


Figure 3-72. Replace Add (80XX-87XX)

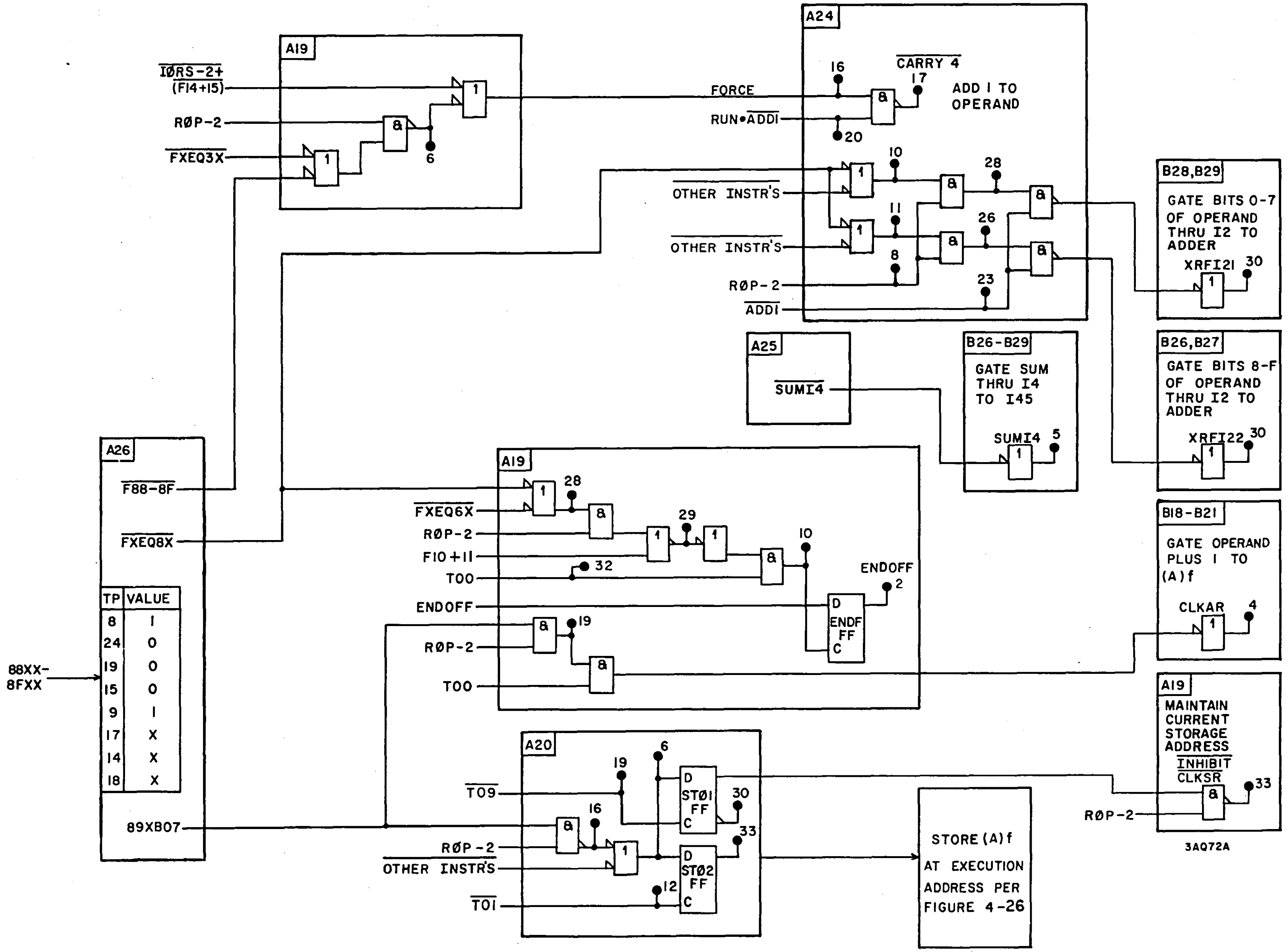
Replace Add One (88XX through 8FXX)

1. During the ROP cycle of the instruction, FXEQ8X enables XRFI21 and XRFI22 which gate the operand (obtained from the execution address) from X through I2 to the adder.
2. F88-8F generates FORCE which enables CARRY4. CARRY4 feeds the adder to increment the sum. The adder now contains the operand plus one.
3. SUMI4 gates the adder output through I4 to I45.
4. FXEQ8X enables clocking of the ENDF flip-flop at T00.
5. 89XB07 enables CLKAR at T00 to gate the result of the add from I45 to A.
6. 89XB07 causes the subsystem processor to exit to a STO cycle. CLKSR is disabled for STO cycles immediately following ROP cycles, so the execution address for. the store is the same as the address the operand was read from.
7. J0TOI5 is inactive, so the contents of A are enabled to I5 for transfer to memory. Figure 3-26 illustrates the remaining STO cycle activity for the instruction.

Replace Leftmost Byte (90XX through 97XX)

Figure 3-75 illustrates byte flow for this instruction.

1. During the ROP cycle of the instruction, F90-97 enables XRFI22 which gates the rightmost byte of the operand (obtained from the execution address) through I2 to the adder. The leftmost byte from I2 contains 0-fill.
2. SUMI4 gates the rightmost byte of the operand from the adder through I4 to I45.
3. F90-97 generates RS8-SN and STRIP which causes the shift network to perform an endaround right shift of eight places on the initial contents of A. After the shift, the rightmost byte of the initial contents of A emerges from the leftmost byte position of the shift network.
4. F90-97 generates SNN5-0 which gates the leftmost byte from the shift network (which is the rightmost byte of the initial contents of A) to I45.
5. 89XB07 enables CLKAR at T00 to gate the rightmost byte of the initial contents of A and the rightmost byte of the operand from I45 to A.



TP	VALUE
8	1
24	0
19	0
15	0
9	1
17	X
14	X
18	X

Figure 3-73. Replace Add One (88XX-8FXX)

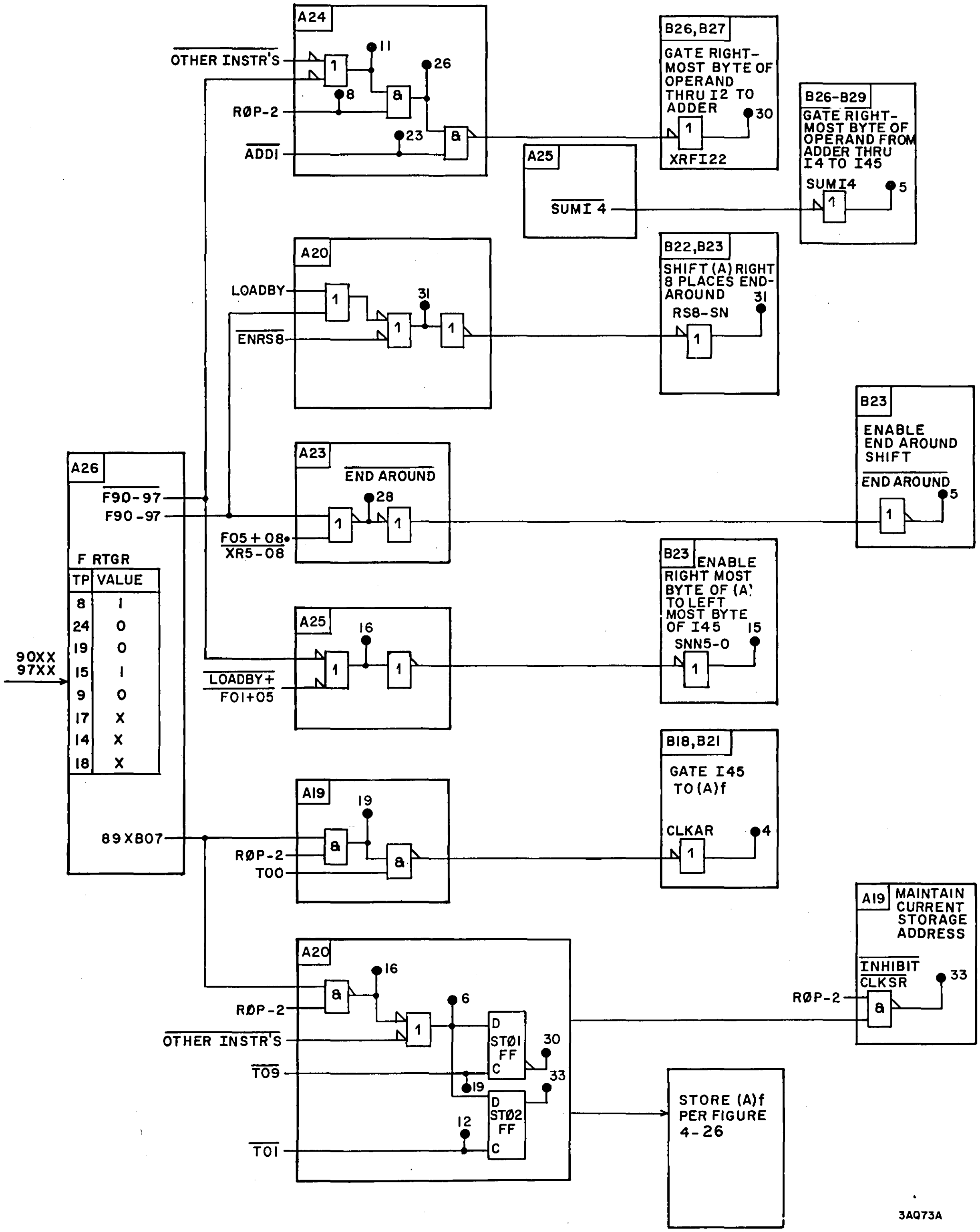


Figure 3-74. Replace Leftmost Byte (90XX-97XX)

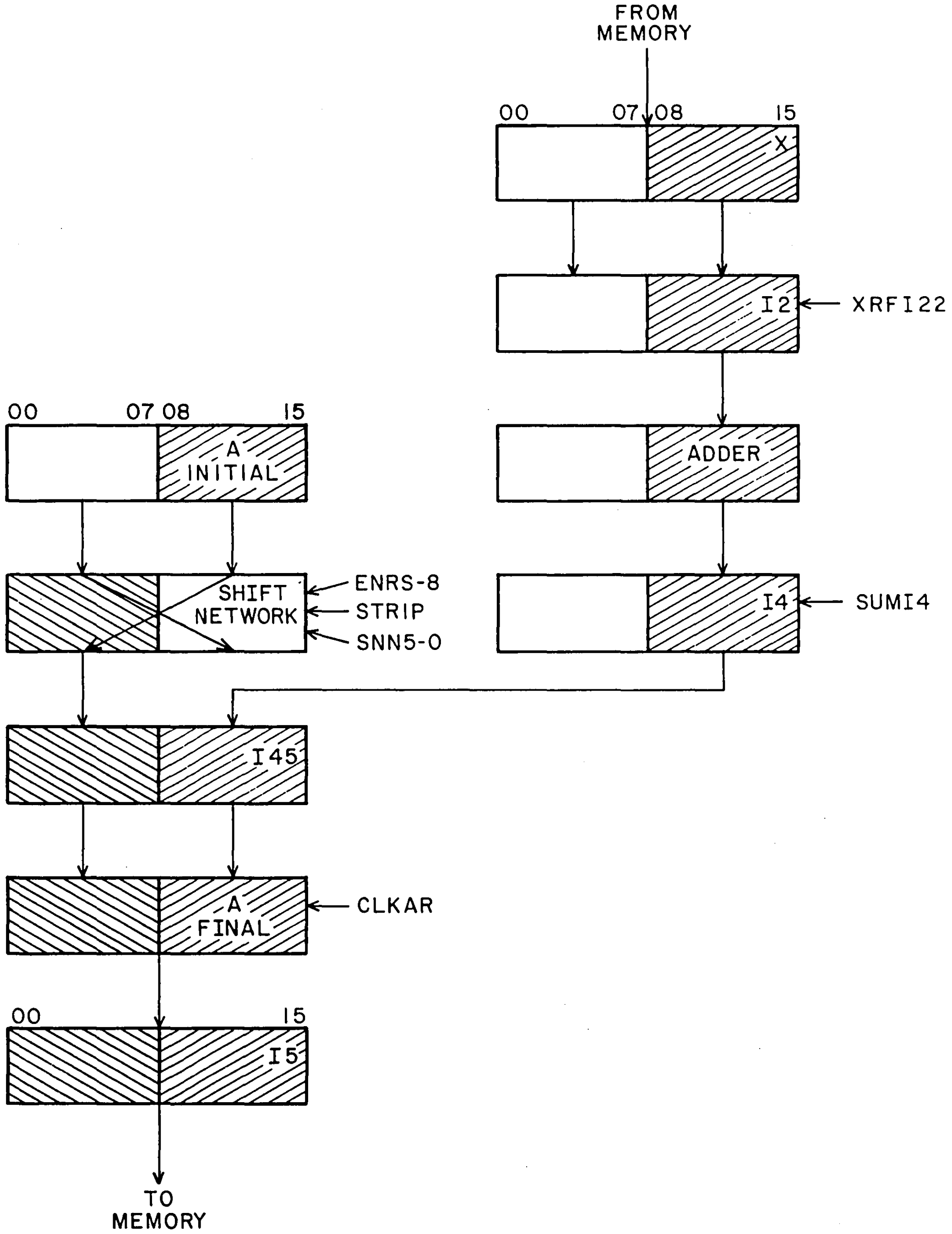


Figure 3-75. Byte Flow for Replace Leftmost Byte

6. 89XB07 causes the subsystem processor to exit to a STO cycle. CLKSR is disabled for STO cycles immediately following ROP cycles, so the execution address for the store is the same as the address the operand was read from.

7. J0TOI5 is inactive, so the contents of A is enabled to I5 for transfer to memory. Figure 3-26 illustrates the remaining STO cycle activity for the instruction.

Replace Rightmost Byte (98XX through 9FXX)

Figure 3-77 illustrates byte flow for this instruction.

1. During the ROP cycle of the instruction, F98-9F enables XRFI21 which gates the leftmost byte of the operand (obtained from the execution address) through I2 to the adder. The rightmost byte from I2 contains 0-fill.

2. SUMI4 gates the leftmost byte of the operand from the adder through I4 to I45.

3. The initial contents of A passes through the shift network unchanged. F98-9F generates SNN5-1 which gates the rightmost byte of the initial contents of A from the shift network to I45. The lower 5 bits (bits 11 through 15) from the shift network pass through N6 before feeding I45.

4. 89XB07 enables CLKAR at T00 to gate the rightmost byte of the initial contents of A and the leftmost byte of the operand from I45 to A.

5. 89XB07 causes the subsystem processor to exit to a STO cycle. CLKSR is disabled for STO cycles immediately following ROP cycles, so the execution address for the store is the same as the address the operand was read from.

6. J0TOI5 is inactive, so the contents of A is enabled to I5 for transfer to memory. Figure 3-26 illustrates the remaining STO cycle activity for the instruction.

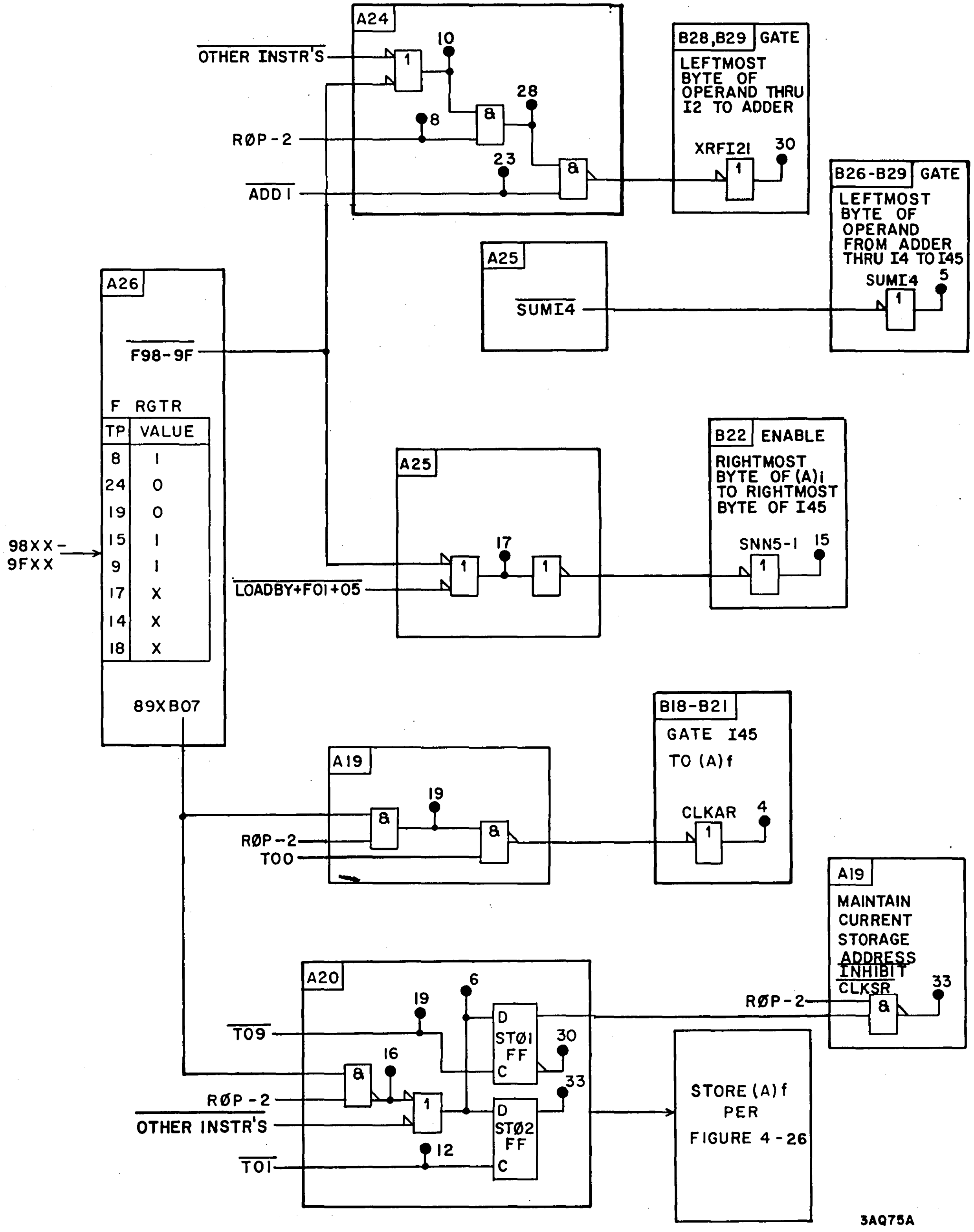


Figure 3-76. Replace Rightmost Byte (98XX-9FXX)

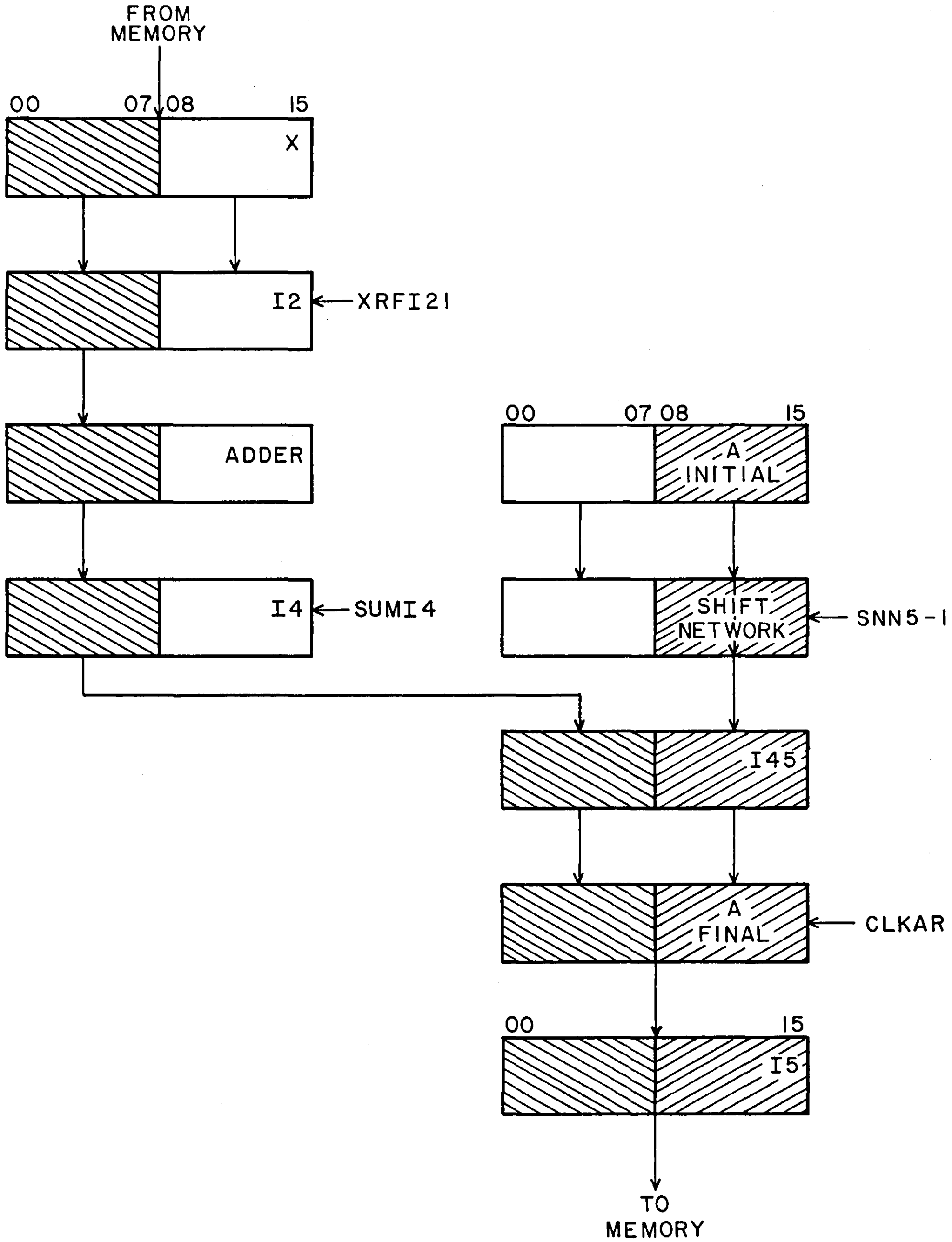


Figure 3-77. Byte Flow for Replace Rightmost Byte

Store (A0XX through A7XX)

1. When the execution address has been formed, FXEQAX raises STO cycle status.
2. J0TOI5 is inactive, so the contents of A is enabled to I5 for transfer to memory. Figure 3-26 illustrates the remaining STO cycle activity for the instruction.

Store Zeros (A8XX through AFXX)

1. When the execution address has been formed, FXEQAX raises STO cycle status.
2. STORE0 disables the A register input to I5. J0TOI5 is inactive, so 0-fill is enabled to I5 for transfer to memory. Figure 3-26 illustrates the remaining STO cycle activity for the instruction.

Destructive Load (B0XX through B7XX)

1. During the ROP cycle of the instruction, an inactive FX8-XF signal combines with FXEQBX to enable XRFI21 and XRFI22. These signals gate the operand (obtained from the execution address) from X through I2 to the adder. The operand passes through the adder unchanged.
2. SUMI4 gates the operand from the adder through I4 to I45.
3. 89XB07 enables CLKAR at T00 to gate the operand from I45 to A.
4. 89XB07 causes the subsystem processor to exit to a STO cycle. CLKSR is disabled for STO cycles immediately following ROP cycles, so the execution address for the store is the same as the address the operand was read from.
5. STORE0 disables the A register input to I5. J0TOI5 is inactive, so 0-fill is enabled to I5 for transfer to memory. Figure 3-26 illustrates the remaining STO cycle activity for the instruction.

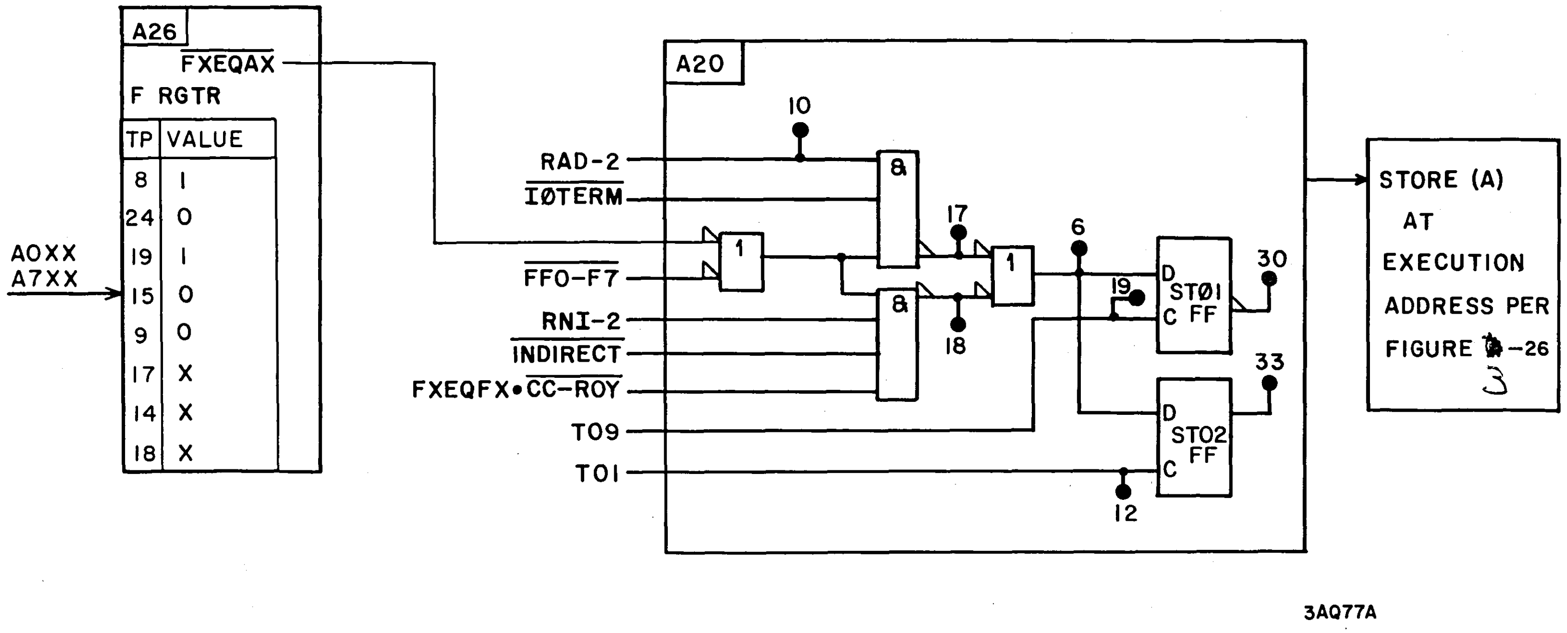
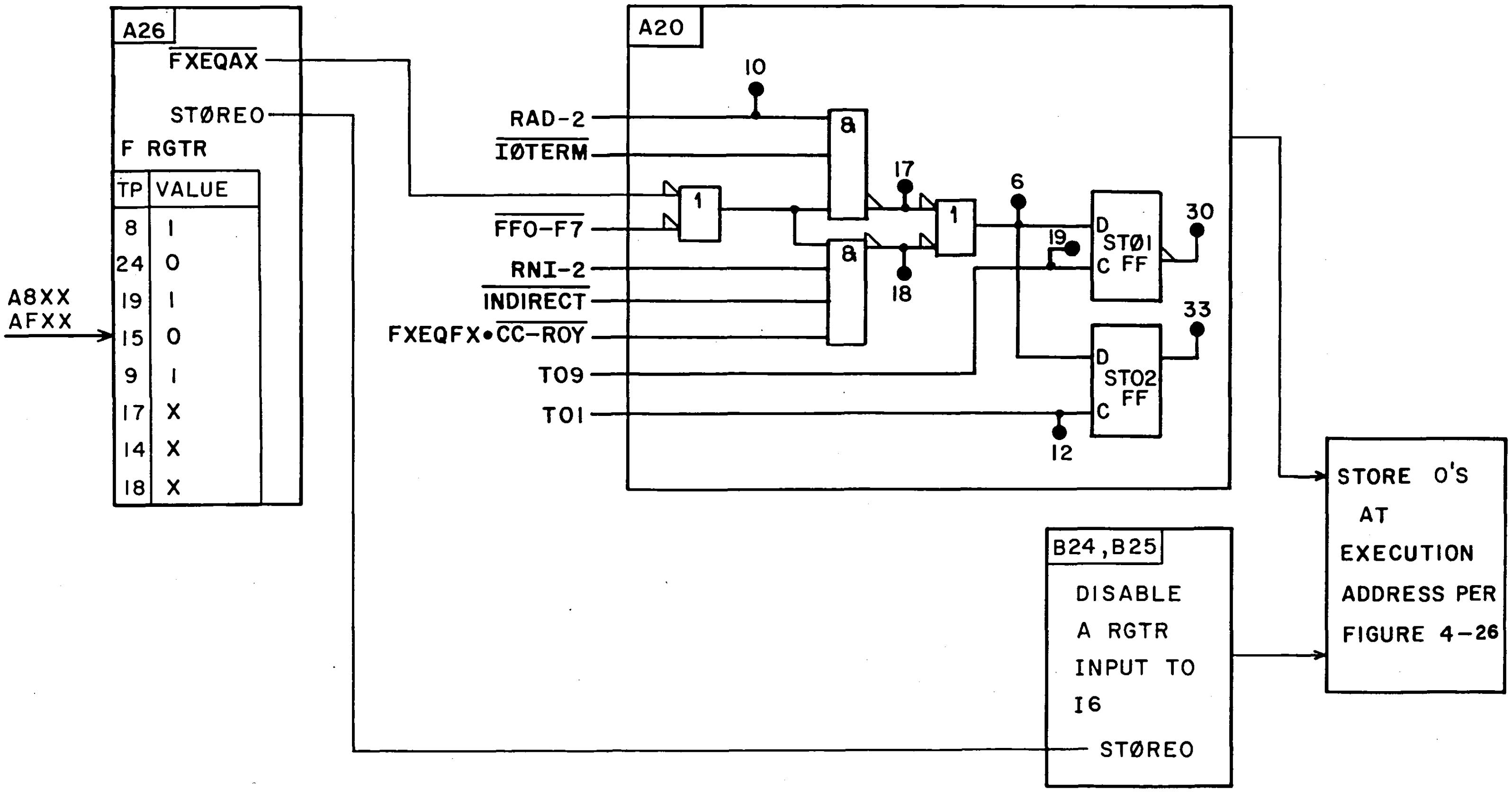
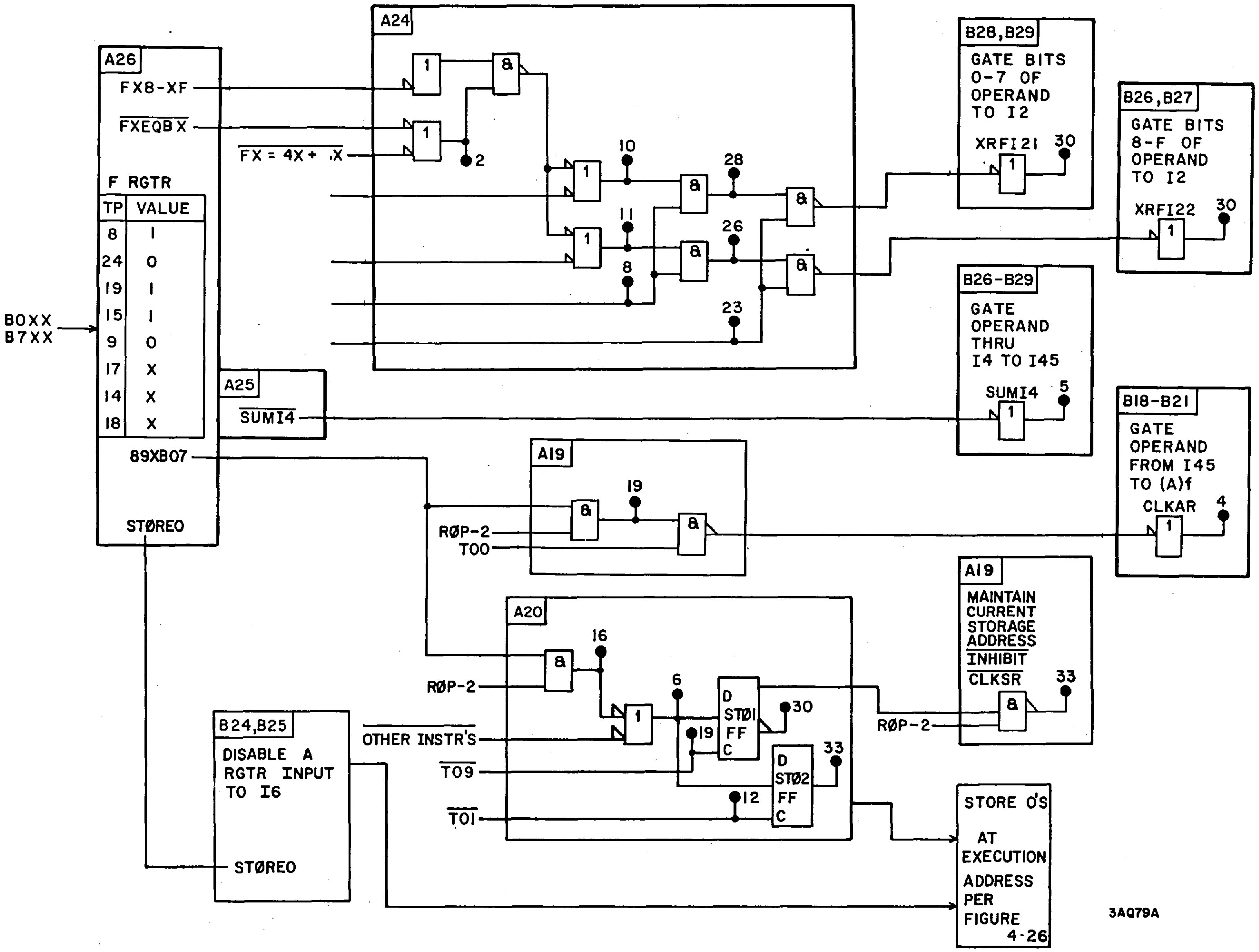

A26
$\overline{\text{FXEQAX}}$
F RGTR
TP VALUE
8 1
24 0
19 1
15 0
9 0
17 X
14 X
18 X
A0XX
A7XX
A20
10
RAD-2
$\overline{\text{IØTERM}}$
$\overline{\text{FF0-F7}}$
RNI-2
$\overline{\text{INDIRECT}}$
FXEQFX•$\overline{\text{CC-ROY}}$
T09
T01
&
&
1
1
17
18
6
19
D STØ1 C FF
D STO2 C FF
30
33
12
STORE (A) AT EXECUTION ADDRESS PER FIGURE 3-26
3AQ77A

Figure 3-78. Store (A0XX-A7XX)



TP	VALUE
8	1
24	0
19	1
15	0
9	1
17	X
14	X
18	X

Figure 3-79. Store Zeros (A8XX-AFXX)


A26
FX8-XF
FXEQBX
FX = 4X + X
F RGTR
B0XX
B7XX
89XB07
STØREO
A24
1
&
2
10
11
8
28
26
23
A25
SUMI4
A19
19
RØP-2
T00
A20
16
6
D
STØ1
FF
C
30
19
D
STØ2
FF
C
33
12
RØP-2
OTHER INSTR'S
T09
T01
B24,B25
DISABLE A RGTR INPUT TO I6
STØREO
B28,B29
GATE BITS 0-7 OF OPERAND TO I2
XRFI21
30
B26,B27
GATE BITS 8-F OF OPERAND TO I2
XRFI22
30
B26-B29
GATE OPERAND THRU I4 TO I45
SUMI4
5
B18-B21
GATE OPERAND FROM I45 TO (A)f
CLKAR
4
A19
MAINTAIN CURRENT STORAGE ADDRESS INHIBIT CLKSR
33
RØP-2
STORE O'S AT EXECUTION ADDRESS PER FIGURE 4-26
3AQ79A

TP	VALUE
8	I
24	O
19	I
15	I
9	O
17	X
14	X
18	X

Figure 3-80. Destructive Load (B0XX-B7XX)

Unconditional Jump (B8XX through BFXX)

1. During the RNI cycle of the instruction, a B8-BF function decode generates JMPMET and ENRADJ.
2. For BCXX, BDXX, BEXX codes, ENRADJ enables a RAD cycle for execution address formation.
3. JMPMET forces PRTOI3 to 0 so that CLKSR loads the execution address into S at T00 of the RNI cycle following address formation. Program execution resumes at the execution address.

A Zero Jump (C0XX through C7XX)

1. When AZERO is inactive during the RNI cycle of the instruction, JMPMET is also inactive to force PRTOI3 to 1. This causes CLKSR to gate the (P) [which equals the current (S) + 1] to S at T00. Program execution resumes at the address immediately following the jump instruction.
2. When AZERO is active during the RNI cycle of the instruction, a C0-C7 function decode generates JMPMET and ENRADJ.
3. For C4XX, C5XX, C6XX codes, ENRADJ enables a RAD cycle for execution address formation.
4. JMPMET forces PRTOI3 to 0 so that CLKSR loads the execution address into S at T00 of the RNI cycle following address formation. Program execution resumes at the execution address.

A Nonzero Jump (C8XX through CFXX)

1. When AZERO is active during the RNI cycle of the instruction, JMPMET is inactive to force PRTOI3 to 1. This causes CLKSR to gate the (P) [which equal the current (S) + 1] to S at T00. Program execution resumes at the address immediately following the jump instruction.
2. When AZERO is inactive during the RNI cycle of the instruction, a C8-CF function decode generates JMPMET and ENRADJ.
3. For CCXX, CDXX, CEXX codes, ENRADJ enables a RAD cycle for execution address formation.
4. JMPMET forces PRTOI3 to 0 so that CLKSR loads the execution address into S at T00 of the RNI cycle following address formation. Program execution resumes at the execution address.

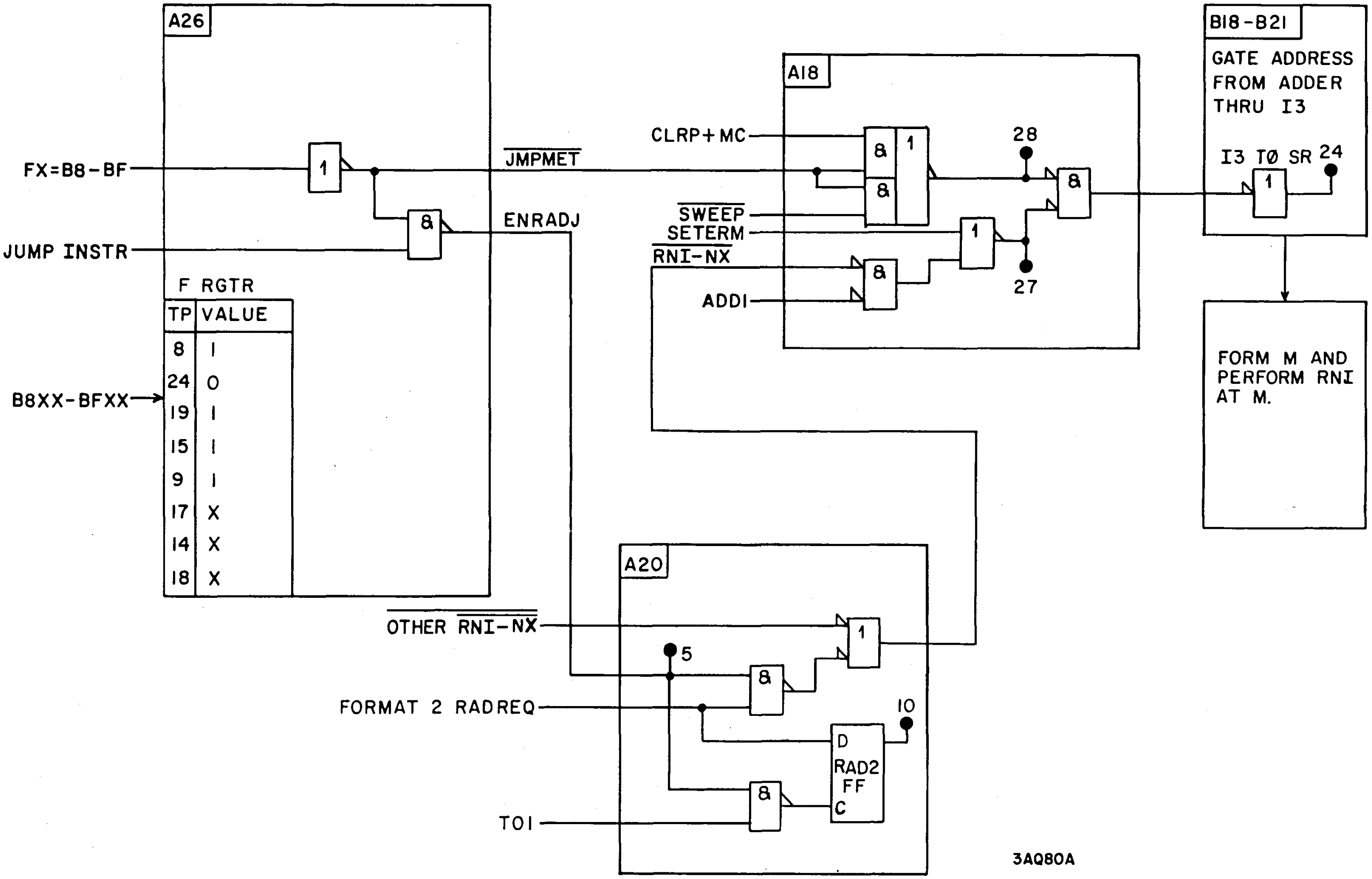


Figure 3-81. Unconditional Jump (B8XX-BFXX)

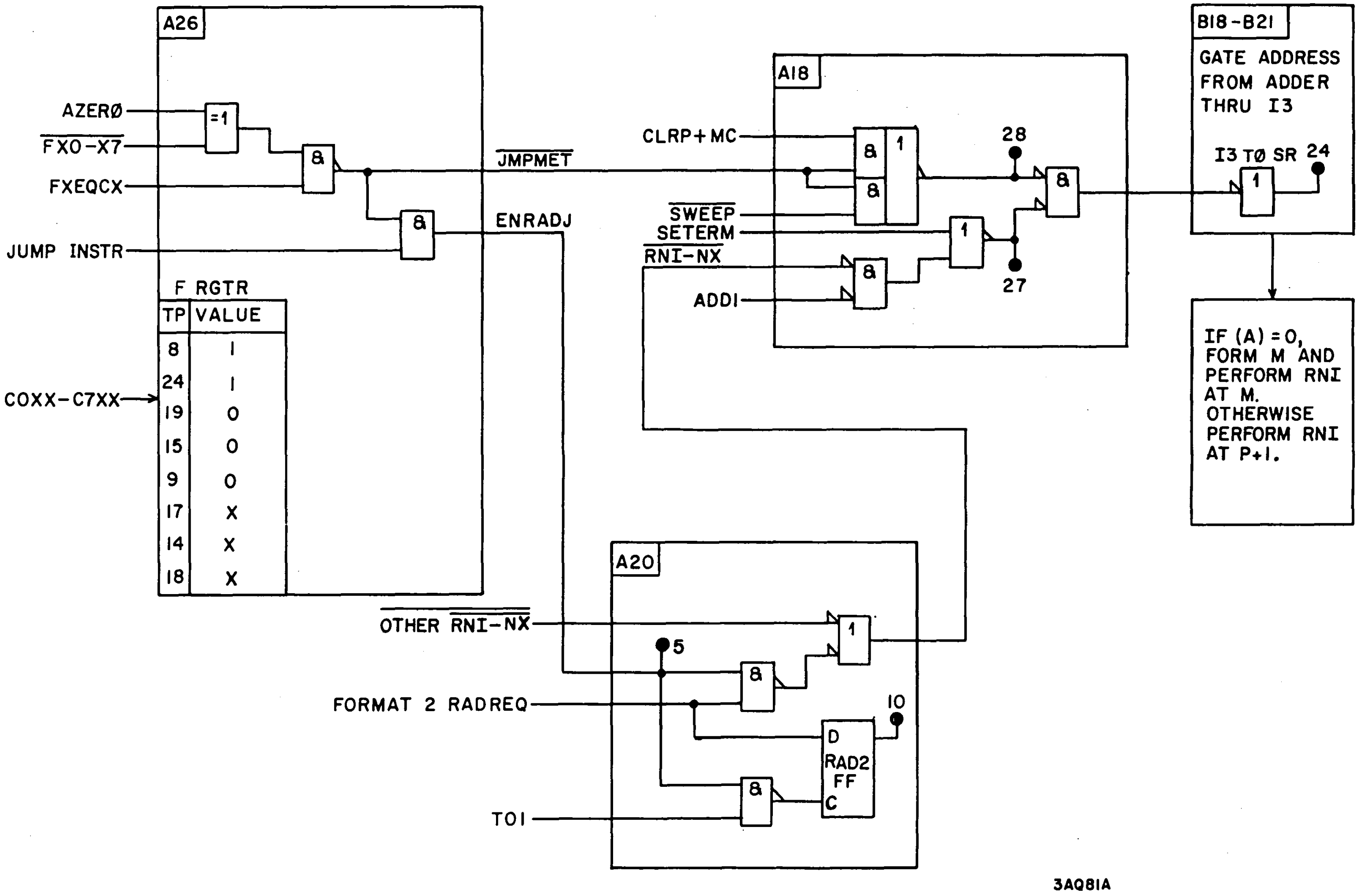


Figure 3-82. A Zero Jump (C0XX-C7XX)

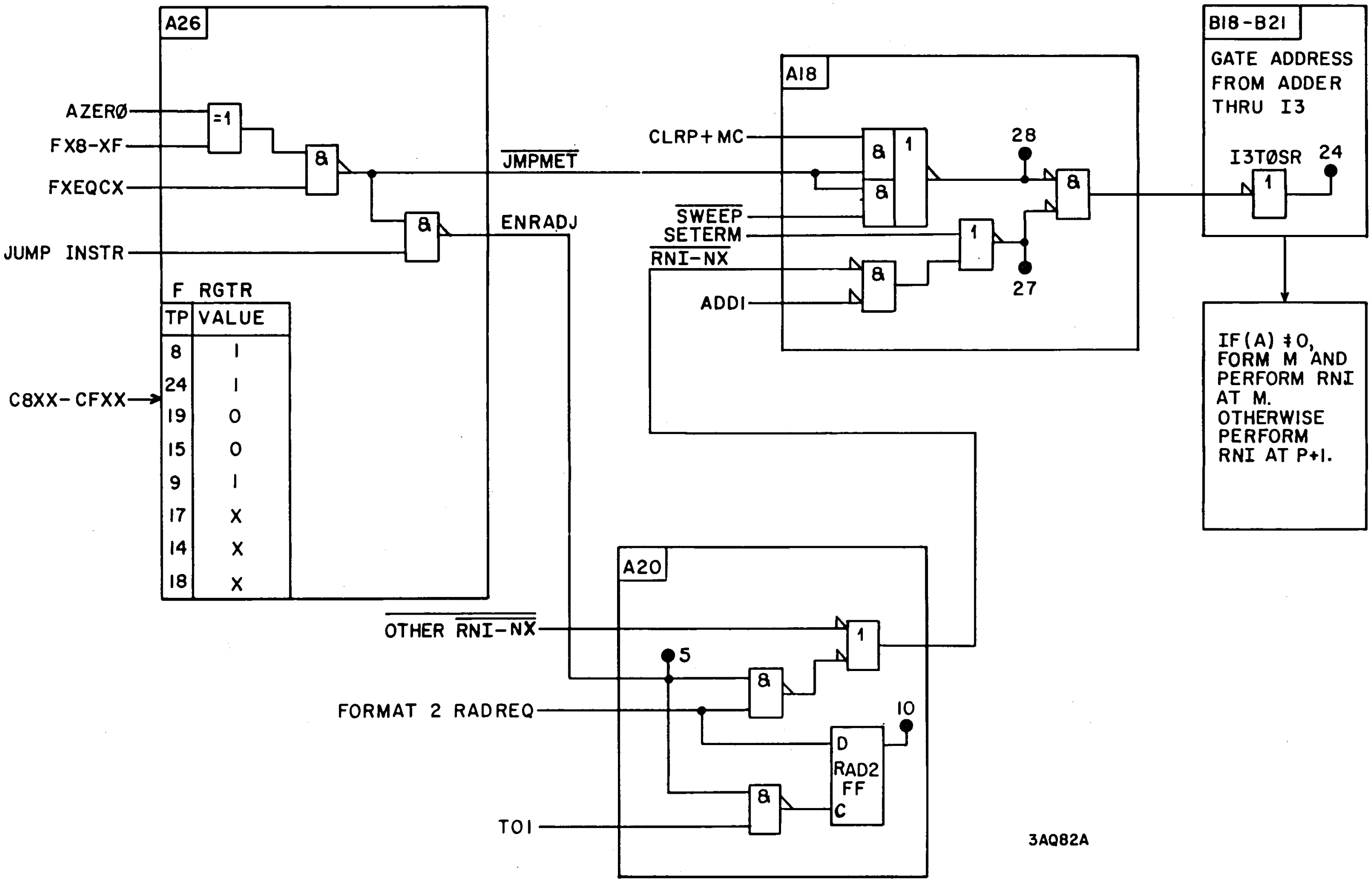


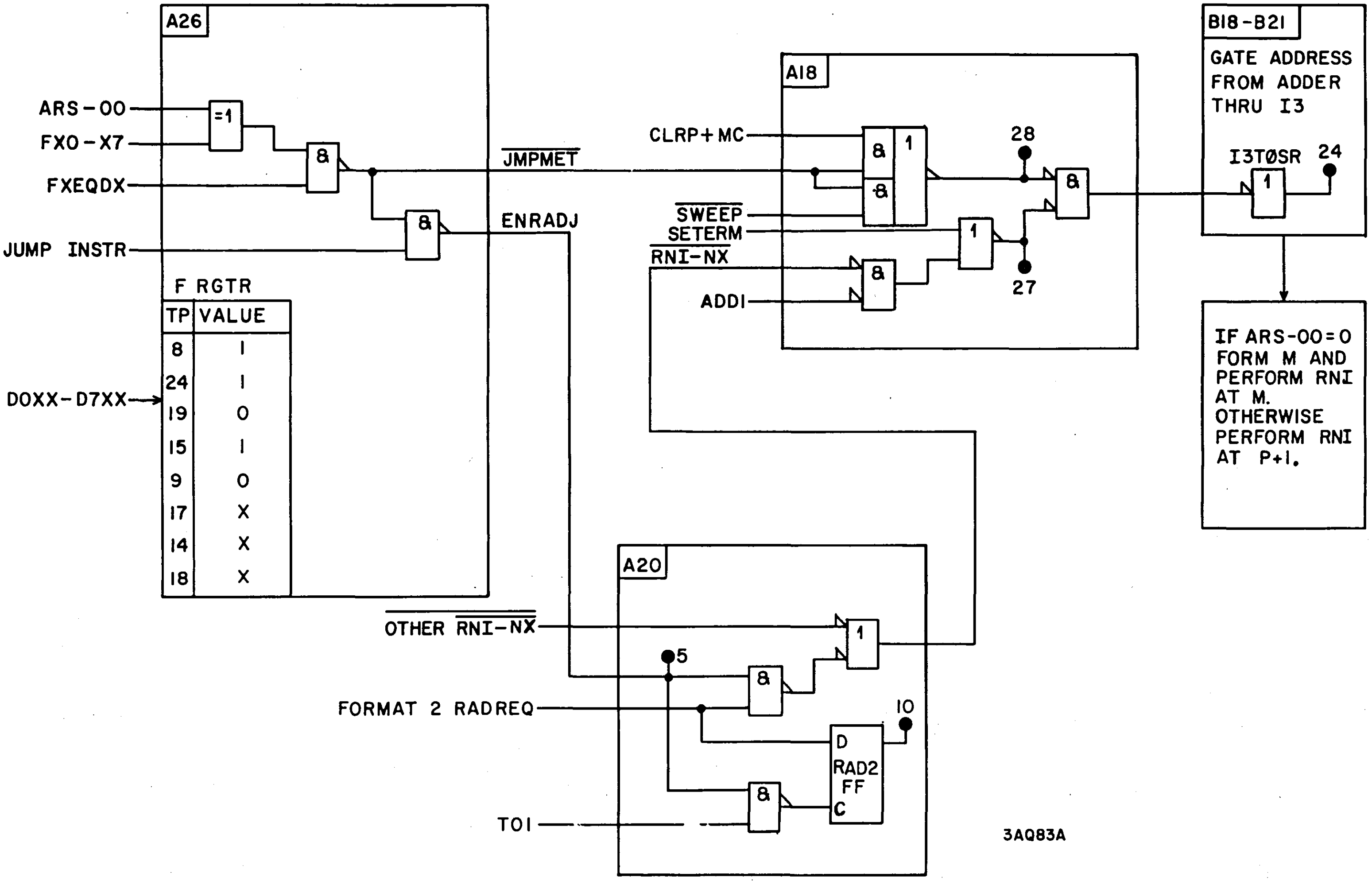
Figure 3-83. A Nonzero Jump (C8XX-CFXX)

A Positive Jump (D0XX through D7XX)

1. When ARS-00 is 1 (indicating a negative quantity in A) during the RNI cycle of the instruction, JMPMET is inactive to force PRTOI3 to 1. This causes CLKSR to gate the (P) [which equal the current (S) + 1] to S at T00. Program execution resumes at the address immediately following the jump instruction.
2. When ARS-00 is 0 (indicating a positive quantity in A) during the RNI cycle of the instruction, a D0-D7 function decode generates JMPMET and ENRADJ.
3. For D4XX, D5XX, D6XX codes, ENRADJ enables a RAD cycle for execution address formation.
4. JMPMET forces PRTOI3 to 0 so that CLKSR loads the execution address into S at T00 of the RNI cycle following address formation. Program execution resumes at the execution address.

A Negative Jump (D8XX through DFXX)

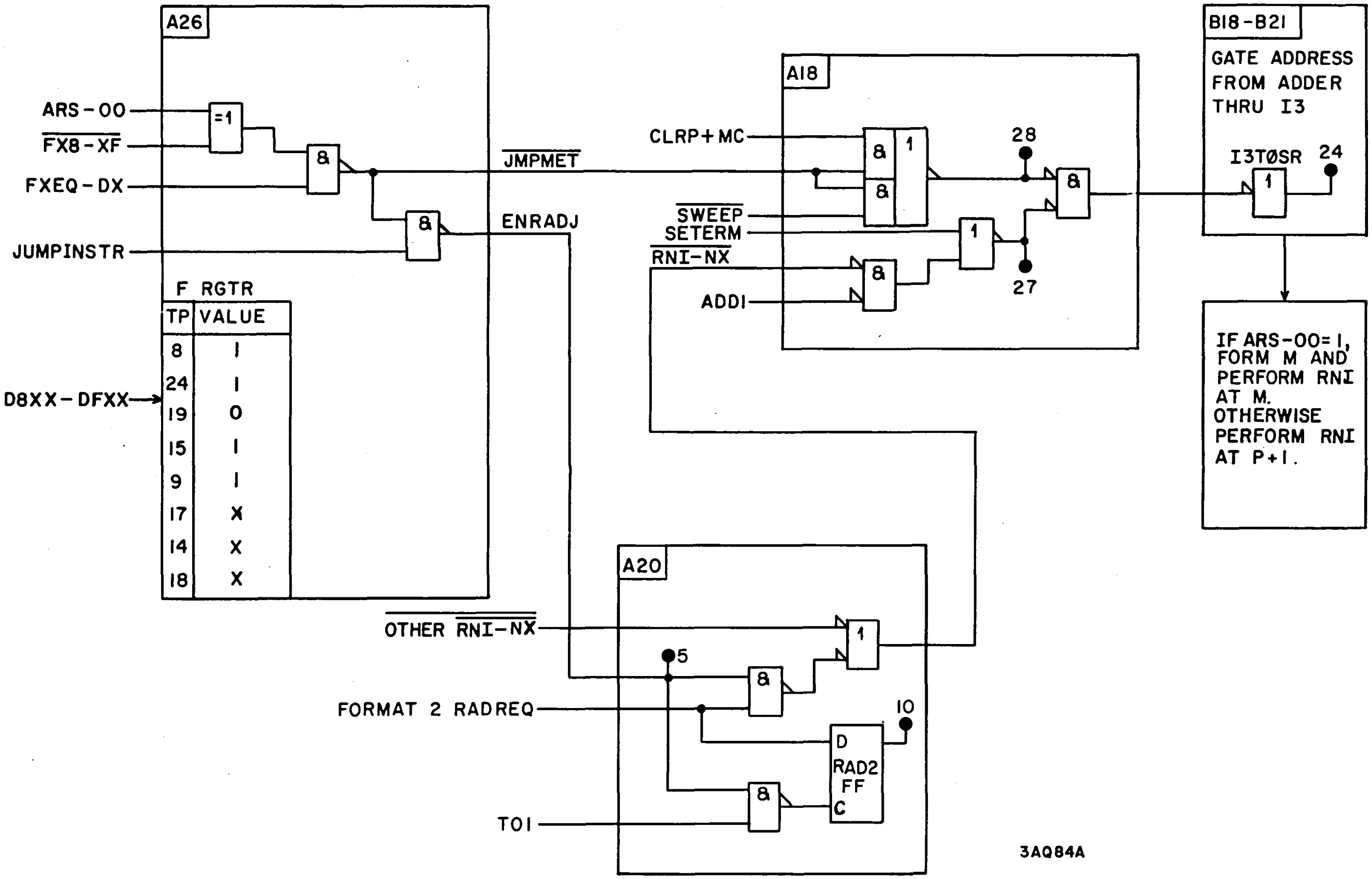
1. When ARS-00 is 0 (indicating a positive quantity in A) during the RNI cycle of the instruction, JMPMET is inactive to force PRTOI3 to 1. This causes CLKSR to gate the (P) [which equal the current (S) + 1] to S at T00. Program execution resumes at the address immediately following the jump instruction.
2. When ARS-00 is 1 (indicating a negative quantity in A) during the RNI cycle of the instruction, a D8-DF function decode generates JMPMET and ENRADJ.
3. For DCXX, DDXX, DEXX codes, ENRADJ enables a RAD cycle for execution address formation.
4. JMPMET forces PRTOI3 to 0 so that CLKSR loads the execution address into S at T00 of the RNI cycle following address formation. Program execution resumes at the execution address.



F RGTR

TP	VALUE
8	1
24	1
19	0
15	1
9	0
17	X
14	X
18	X

Figure 3-84. A Positive Jump (D0XX-D7XX)



F RGTR

TP	VALUE
8	1
24	1
19	0
15	1
9	1
17	X
14	X
18	X

Figure 3-85. A Negative Jump (D8XX-DFXX)

Condition True Jump (E0XX through E7XX)

1. When COND-FF is inactive during the RNI cycle of the instruction, JMPMET is also inactive to force PRTOI3 to 1. This causes CLKSR to gate the (P) [which equal the current (S) + 1] to S at T00. Program execution resumes at the address immediately following the jump instruction.
2. When COND-FF is active during the RNI cycle of the instruction, an E0-E7 function decode generates JMPMET and ENRADJ.
3. For E4XX, E5XX, E6XX codes, ENRADJ enables a RAD cycle for execution address formation.
4. JMPMET forces PRTOI3 to 0 so that CLKSR loads the execution address into S at T00 of the RNI cycle following address formation. Program execution resumes at the execution address.

Condition False Jump (E8XX through EFXX)

1. When COND-FF is active during the RNI cycle of the instruction, JMPMET is inactive to force PRTOI3 to 1. This causes CLKSR to gate the (P) [which equal the current (S) + 1] to S at T00. Program execution resumes at the address immediately following the jump instruction.
2. When COND-FF is inactive during the RNI cycle of the instruction, an E8-EF function decode generates JMPMET and ENRADJ.
3. For ECXX, EDXX, EEXX codes, ENRADJ enables a RAD cycle for execution address formation.
4. JMPMET forces PRTOI3 to 0 so that CLKSR loads the execution address into S at T00 of the RNI cycle following address formation. Program execution resumes at the execution address.

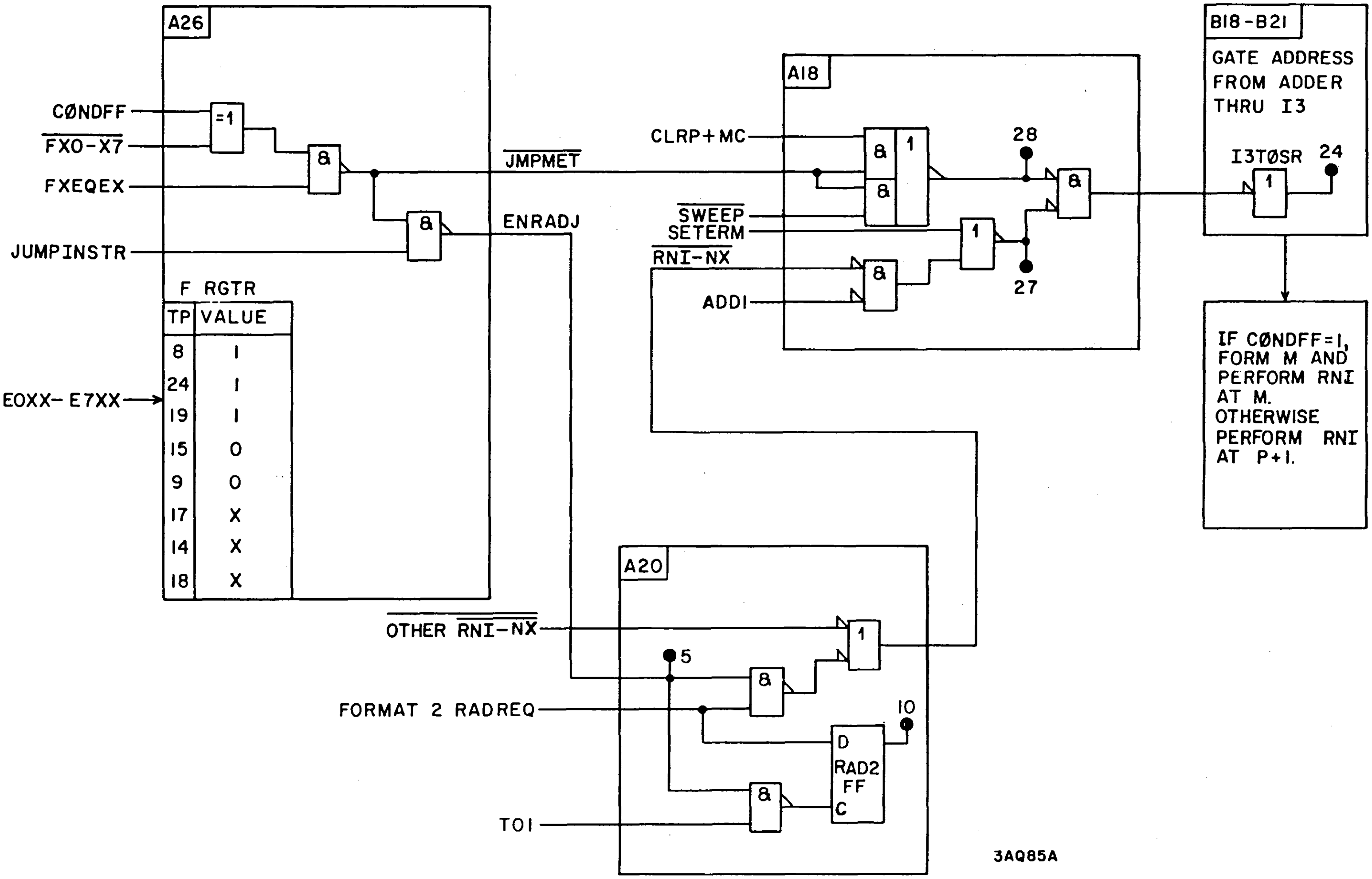
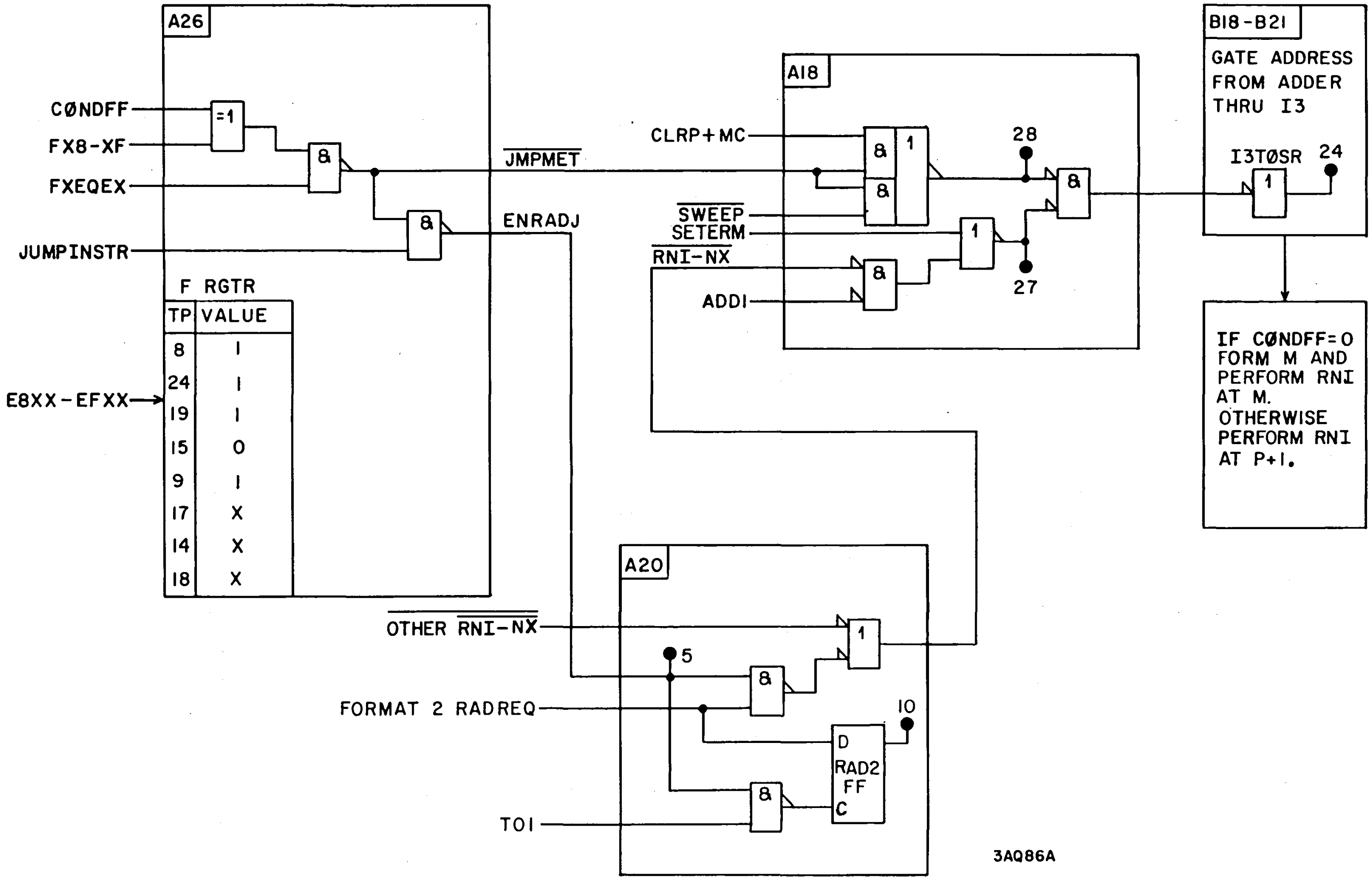


Figure 3-86. Condition True Jump (E0XX-E7XX)


A26
CØNDFF
FX8-XF
FXEQEX
JUMPINSTR
=1
&
&
$\overline{\text{JMPMET}}$
ENRADJ
E8XX-EFXX
A18
CLRP+MC
$\overline{\text{SWEEP}}$
SETERM
$\overline{\text{RNI-NX}}$
ADDI
&
&
1
&
1
&
28
27
B18-B21
GATE ADDRESS FROM ADDER THRU I3
I3TØSR
24
1
IF CØNDFF=O FORM M AND PERFORM RNI AT M. OTHERWISE PERFORM RNI AT P+1.
A20
$\overline{\text{OTHER } \overline{\text{RNI-NX}}}$
FORMAT 2 RADREQ
TOI
5
&
&
1
10
D
RAD2 FF
C
3AQ86A

F RGTR

TP	VALUE
8	I
24	I
19	I
15	O
9	I
17	X
14	X
18	X

Figure 3-87. Condition False Jump (E8XX-EFXX)

Input Block Transfer (F0XX through F7XX)

1. When CREADY (A22) is inactive at T02 during the RNI cycle of the instruction, the block transfer latch clears to force CC-RDY inactive. This generates RNI-NX (A20) which drives PRTOI3 (A18) to 1 so that program execution resumes at P + 1. Since no words are transferred, IORS-2 (A20) remains inactive and the COND flip-flop (A19) is cleared at T00 of the next RNI cycle.

2. When CREADY (A22) is active at T02 during the RNI cycle of the instruction, the subsystem processor forms the execution address, raises STO status (A20), sends an INPREQ pulse (A22) to the coupler, and sets the I/O operation latch which generates IO-OP (A22). IO-OP disables synchronous operation of the timing chain (A21) so that a CREPLY pulse from the coupler is required for each pass down the timing chain.

3. CREPLY from the coupler sets the coupler reply latch and starts the timing chain. At T00, CLKJ0 (A19) gates data from the coupler into J0.

4. STO-1B (A20) generates CK-PAR (A22) which indicates to the coupler that the data has been received.

5. J0 feeds I5, and the subsystem processor performs a storage cycle as indicated in Figure 4-21. A 26

6. IORS-1 (A20) enables ARTOI1 to gate the contents of A (which is the two's complement of the transfer length) through I1 to the adder.

7. IORS-2 (A20) enables FORCE (A19) which generates CARRY4 (A24) to decrement the word count.

8. SUMI4 gates the adder output through I4 to I45.

9. IO+LBY (A20) generates CLKAR (A19) at T05 to gate the decremented word count from I45 to A.

10. When the block transfer is to continue, the subsystem processor sends another INPREQ pulse (A22) to the coupler at T07, and the transfer proceeds as described in step 3 of this sequence.

11. The transfer continues either until the A satisfied latch (A22) sets (indicating the contents of A are about to be decremented to 0) or until the coupler sends CCTERM (A21) in response to INPREQ (A22).

11. a. When the A satisfied latch (A22) sets, it forces IO-OP inactive and also clears the external terminate flip-flop so that the COND flip-flop (A19) is set at T00 of the next RNI cycle. With IO-OP inactive, the timing chain restarts itself after each iteration. The end of operation pulse, EOP-CC (A22), is sent to the coupler at T07.

    b. When the coupler terminates the transfer, CCTERM (A21) generates SETERM (A21) which sets the external terminate flip-flop (A22) so that the COND flip-flop (A19) is cleared at T00 of the next RNI cycle. EXT-TR drops ROP status, raises RNI status, and forces IO-OP inactive so the timing chain can again restart itself after each iteration. The end of operation pulse, EOP-CC (A22), is sent to the coupler at T07.

Output Block Transfer (F8XX through FFXX)

1. When CREADY (A22) is inactive at T02 during the RNI cycle of the instruction, the block transfer latch clears to force CC-RDY inactive. This generates RNI-NX (A20) which drives PRTOI3 (A18) to 1 so that program execution resumes at P+1. Since no words are transferred, IORS-2 (A20) remains inactive and the COND flip-flop (A19) is cleared at T00 of the next RNI cycle.

2. When CREADY (A22) is active at T02 during the RNI cycle of the instruction, the subsystem processor forms the execution address and raised ROP status (A20).

3. The subsystem processor performs a ROP cycle as illustrated in Figure 4-20. The X register provides data to the coupler (CCO-00 through CCO-15).

4. IORS-1 (A20) enables ARTOI1 (A24) to gate the contents of A (which is the two's complement of the transfer length) through I1 to the adder.

5. IORS-2 (A20) generates CARRY4 (A24) to decrement the word count.

6. SUMI4 (A25) gates the adder output through I4 to I45.

7. IO+LBY (A20) generates CLKAR (A19) at T05 to gate the decremented word count from I45 to A.

8. FF8-FF (A26) and ROS-2 (A20) combine at T07 to send an OUTRDY A22 pulse (A22) to the coupler and set the I/O operation latch (A22) which generates IO-OP. IO-OP disables synchronous operation of the timing chain (A21) so that a CREPLY pulse from the coupler is required for each pass down the timing chain.

9. When the coupler has accepted the data word, it sends CREPLY (A21) which sets the coupler reply latch to enable the timing chain. Data passes from the subsystem processor to the coupler as described in step 3 of this sequence.

10. The transfer continues either until the A satisfied flip-flop (A22) sets (indicating the contents of A are about to be decremented to 0) or until the coupler sends CCTERM (A21) in response to an OUTRDY pulse (A22).

    a. When the A satisfied flip-flop (A22) sets, it forces IO-OP inactive and also sets the external terminate flip-flop so that the COND flip-flop (A19) is set at T00 of the next RNI cycle. With IO-OP inactive, the timing chain (A21) restarts itself after each iteration. The end of operation pulse EOP-CC (A22) is sent to the coupler at T07.

    b. When the coupler terminates the transfer, CCTERM (A21) generates SETERM which clears the external terminate flip-flop (A22) so that the COND flip-flop (A19) is cleared at T00 of the next RNI cycle. EXT-TR drops STO status, raises RNI status, and forces IO-OP inactive so the timing chain can again restart itself after each iteration. The end of operation pulse, EOP-CC, is sent to the coupler at T07.

Forced Block Transfers

NOTE

The previous two instructions (F0XX through F7XX, F8XX through FFXX) were described as they occur when read from a program resident in core memory. The block transfer interface (Figure 3-94) provides signals which permit the coupler to force block transfer activity in the subsystem processor. Forced block transfers differ from instruction caused transfers as follows:

1. Before initiating a forced block transfer, the coupler stops and master clears the subsystem processor by sending CYCLES (A18) and MC (A18).

2. FORCEF (A22) enables CLKFXX which generates FXINPA (A18) to set bits 00 through 03 of A. This enables a block transfer word count of 4096_{10}. The coupler can terminate the block transfer before the count is satisfied by sending CCTERM (A21).

3. CLKFXX (A22) generates FXINP (A18) which disables any X register transfers to I2. FXINP also forces the F register (A26) to an FX value. EX-OUT (A26) determines whether the F register is forced to F0 or F8.

4. Because of the master clear from the coupler, the starting address for the block transfer is memory address 0000.

5. The coupler initiates the forced block transfer by sending G0 (A21).

6. When the block transfer terminates, the COND flip-flop (A19) is set or cleared (depending upon whether the 4096_{10} word count is satisfied), and program execution resumes at memory address 0001.

CONTROL LOGIC THEORY

The control logic is a special purpose processor which controls disk storage units and performs other special purpose functions related to disk storage. It uses instructions which vary in length from 16 to 64 bits. These instructions are called directors to distinguish them from instructions which execute in the subsystem processor. Subsystem processor instructions and directors make up the controlware which is stored in core memory. Director sequences are selected by the subsystem processor depending on the operation to be performed. The directors are then loaded into the control logic where they are executed to control disk storage unit operations and to perform data handling and processing. Data processing includes error detection and correction as specified by the directors.

Figure 3-88 shows the basic data paths and major components of the control logic. The major components are described under component theory. Timing relationships for complete sequences such as normal channel functions and director execution are described under operational sequences.

COMPONENT THEORY

The following paragraphs describe the purpose and the operating characteristics of the major components of the control logic. Timing diagrams and block diagrams are included where necessary to highlight specific operations. All references to card locations and test points refer to locations on chassis A11 unless otherwise specified.

Memory Scanner

The core memory is shared by the control logic and the subsystem processor. The control logic accesses memory for data transfers and to obtain directors. The subsystem processor accesses memory to perform its operations. Since these accesses can occur simultaneously, a memory scanner is used to avoid memory conflicts and to assign priorities. The priorities are based on real-time constraints and are as follows:

First priority	Data references from control logic
Second priority	Director references from control logic
Third priority	Subsystem processor references

The memory scanner is essentially a three-section oscillator with each section containing a wired flip-flop and associated logic. The flip-flops provide for blocking and retriggering the oscillator and are used to honor the three types of memory requests.

When the scanner is idling (no memory references), the time required for one complete scanner loop is about 400 nanoseconds. During the first half of the idle loop, each of the scan flip-flops is set in succession. During the second half of the idle loop, each of the scan flip-flops is cleared. This action generates a 200-nanosecond symmetrical scan pulse which loops through the scanner until blocked by a request.

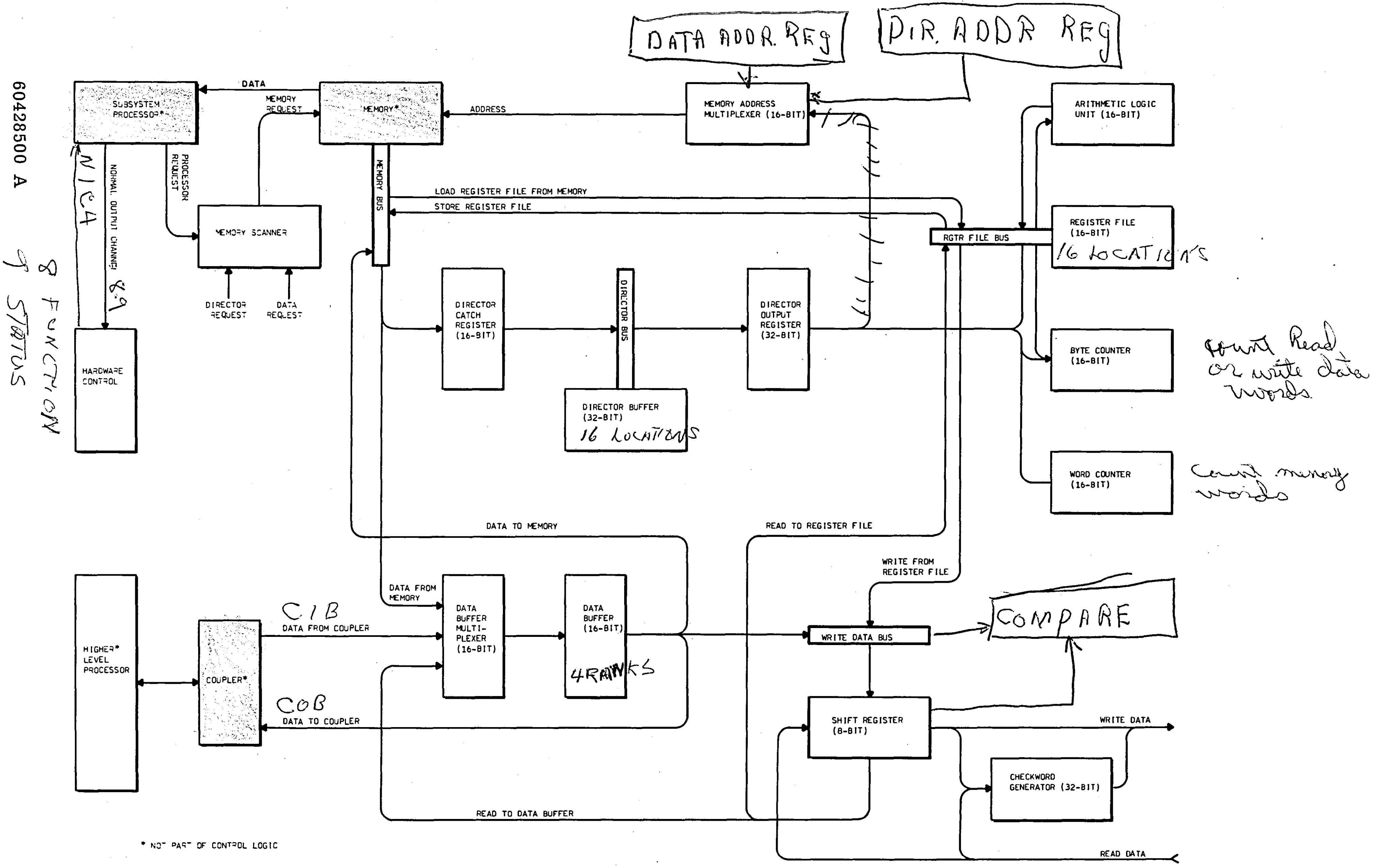


Figure 3-88. Control Logic Block Diagram - Basic Data Path

Figure 3-89 shows the effect of this scan pulse on one section of the scanner. With the scan flip-flop clear (TP1=0) and all stages stable, the output of inverter C provides one enable to the scan gate (inverter D). The leading edge of a scan pulse provides a second enable to the scan gate and also sets the scan flip-flop through inverter A. About 60 nanoseconds after the scan flip-flop sets, the output of inverter C disables the scan gate. If a request comes up during the 60-nanosecond window, the request flip-flop will set (TP2=1). Its set output provides one enable to inverter E while its clear output disables inverter B.

On the trailing edge of the scan pulse, inverter A provides an enable to both inverters B and E. If the request flip-flop is clear, inverter B is enabled and clears the scan flip-flop. This unblocks the scanner. If the request flip-flop is set, inverter E is enabled and generates a request to memory. Since inverter B is disabled, the scanner will remain blocked until the request flip-flop is cleared. This will occur after memory has replied to the request. The memory reply is delayed to allow the data to be sampled and then clears the request flip-flop. As the flip-flop clears, inverter B is enabled, the scan flip-flop is cleared, and the scanner is released to continue its scan loop.

Each scanner section operates basically as described previously. Figure 3-90 shows operation of the scanner with various types of accesses. The following discussions describe these accesses and explain some of the differences in their operation.

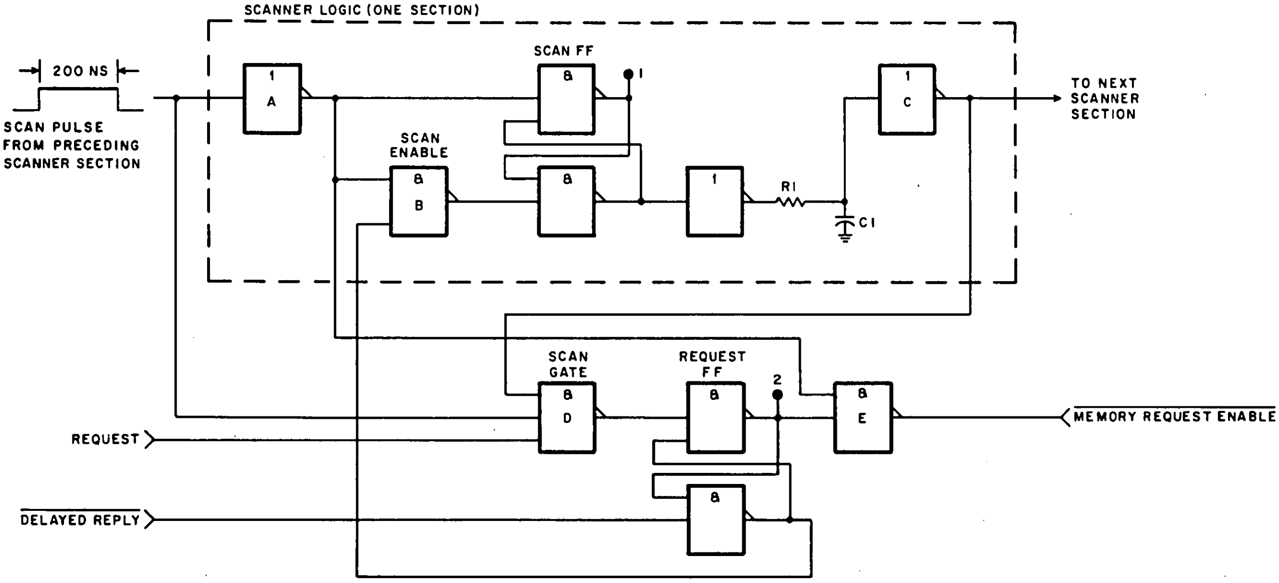
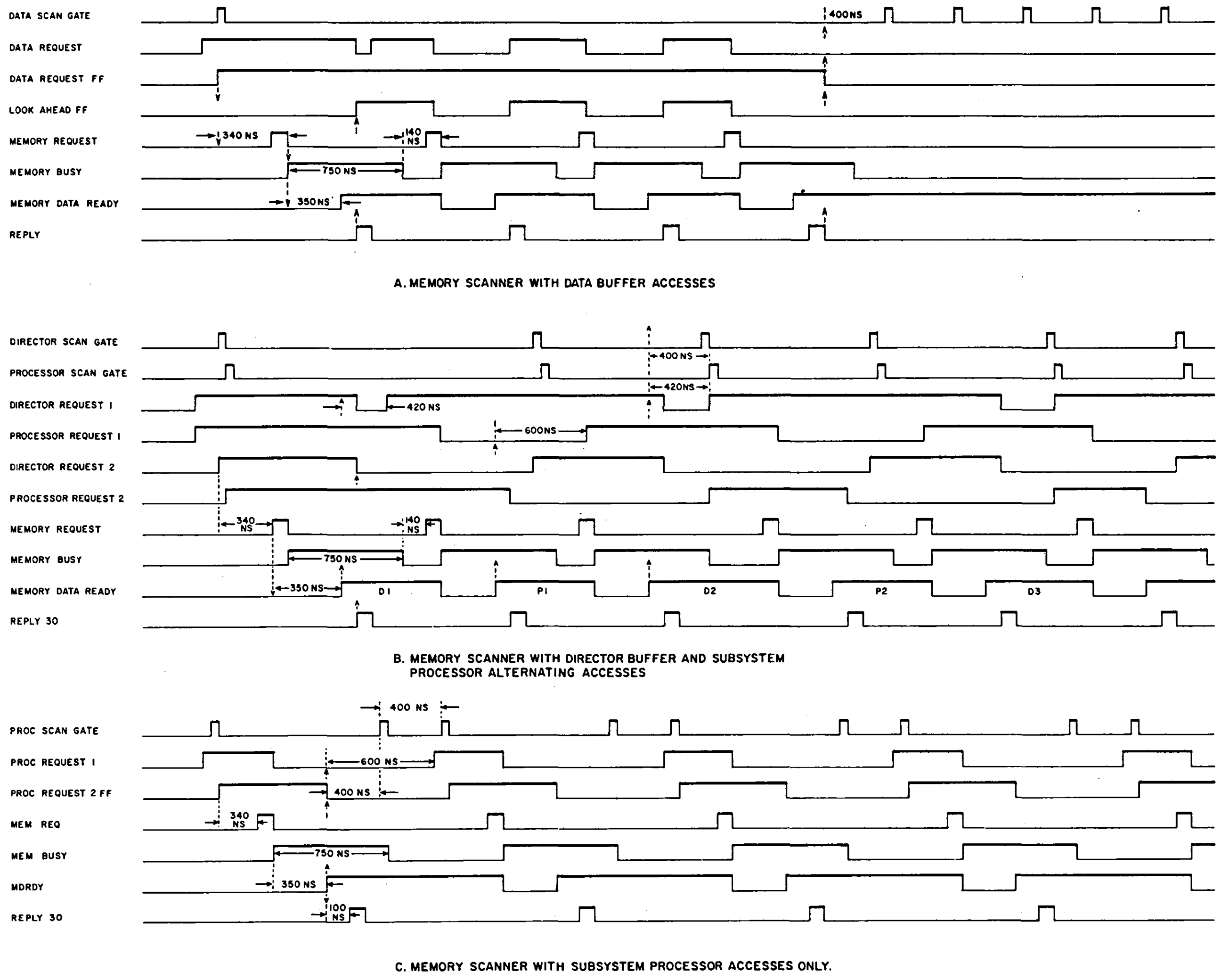


Figure 3-89. Simplified Logic Diagram - Memory Scanner



A. MEMORY SCANNER WITH DATA BUFFER ACCESSES

B. MEMORY SCANNER WITH DIRECTOR BUFFER AND SUBSYSTEM PROCESSOR ALTERNATING ACCESSES

C. MEMORY SCANNER WITH SUBSYSTEM PROCESSOR ACCESSES ONLY.

Figure 3-90. Memory Scanner Access Timing Diagram

Data References From Control Logic (Part A of Figure 3-90)

Data references from the control logic originate in either the register file or the data buffer. These requests are given top priority and the memory scanner is blocked until all data associated with the request has been transferred. Since register file requests are always one-word transfers, the memory scanner is unblocked as soon as a reply is received from memory. Requests from the data buffer, however, usually involve more than one word. To prevent the memory reply for the first transfer from unblocking the scanner, a look-ahead scheme is used.

During output transfers, the look-ahead logic checks the third stage of the data buffer and looks at the word counter. If the third stage is empty or if the word counter has decremented to a count of one, the leading edge of the memory reply clears the look-ahead flip-flop. The trailing edge of the memory reply then clears the data request flip-flop which releases the scanner. If the third stage of the data buffer has additional data to be transferred, the look-ahead flip-flop sets and prevents the data request flip-flop from clearing. The scanner remains blocked and as soon as memory busy drops, a new request is sent to memory.

During input transfers the look-ahead logic checks the second stage of the data buffer and looks at the word counter. If the second stage is full or if the word counter has decremented to a count of one, the leading edge of the memory reply clears the look-ahead flip-flop. The trailing edge of the memory reply then clears the data request flip-flop which releases the scanner. If the second stage of the data buffer is empty, the look-ahead flip-flop sets and prevents the data request flip-flop from clearing. The scanner remains blocked and as soon as memory busy drops, a new request is sent to memory.

The effective memory cycle time for data transfers is about 900 nanoseconds. This includes 750 nanoseconds for the actual memory cycle and about 150 nanoseconds of internal delay in the control logic.

Director/Subsystem Processor References (Part B of Figure 3-90)

When there are no data transfers to memory in progress, director references alternate with subsystem processor references. The director reference is given priority since its scan gate is first in the scanner loop. Neither reference blocks the scanner for more than one reference. As soon as a memory reply is received for the current request, the scanner is unblocked to honor the request from the other access. The memory reply for a director request initiates a timing chain which writes the data into the director buffer. If directors are not being executed, the timing chain responds immediately, and another director request may come up before the director scan gate. In this event, the second director request is honored before the subsystem processor request, and accesses alternate as follows:

Director Director Subsystem processor Director Director Subsystem processor	Configuration 1

When directors are being executed, the write timing chain is delayed until the read timing chain is complete. This causes the second director request to miss the director scan gate as shown in Part B of Figure 3-90. This is the normal operating configuration and causes the accesses to alternate as follows:

Director Subsystem processor Director Subsystem processor Director Subsystem processor	Configuration 2

Configuration 1 allows a complete director (two 16-bit words) to be transferred in a minimum of 1.8 microseconds or a maximum of 3.0 microseconds. The maximum figure assumes an intervening subsystem processor reference.

Configuration 2 provides an average access time of 1.0 microseconds for each access. This allows a complete director to be transferred in a minimum of 3.0 microseconds (one intervening subsystem processor reference) or a maximum of 4.0 microseconds (two intervening subsystem processor references).

The times given for both configurations 1 and 2 are typical and will change depending on the switching times of the chips involved and with the number of memory accesses for data transfers.

Subsystem Processor References (Part C of Figure 3-90)

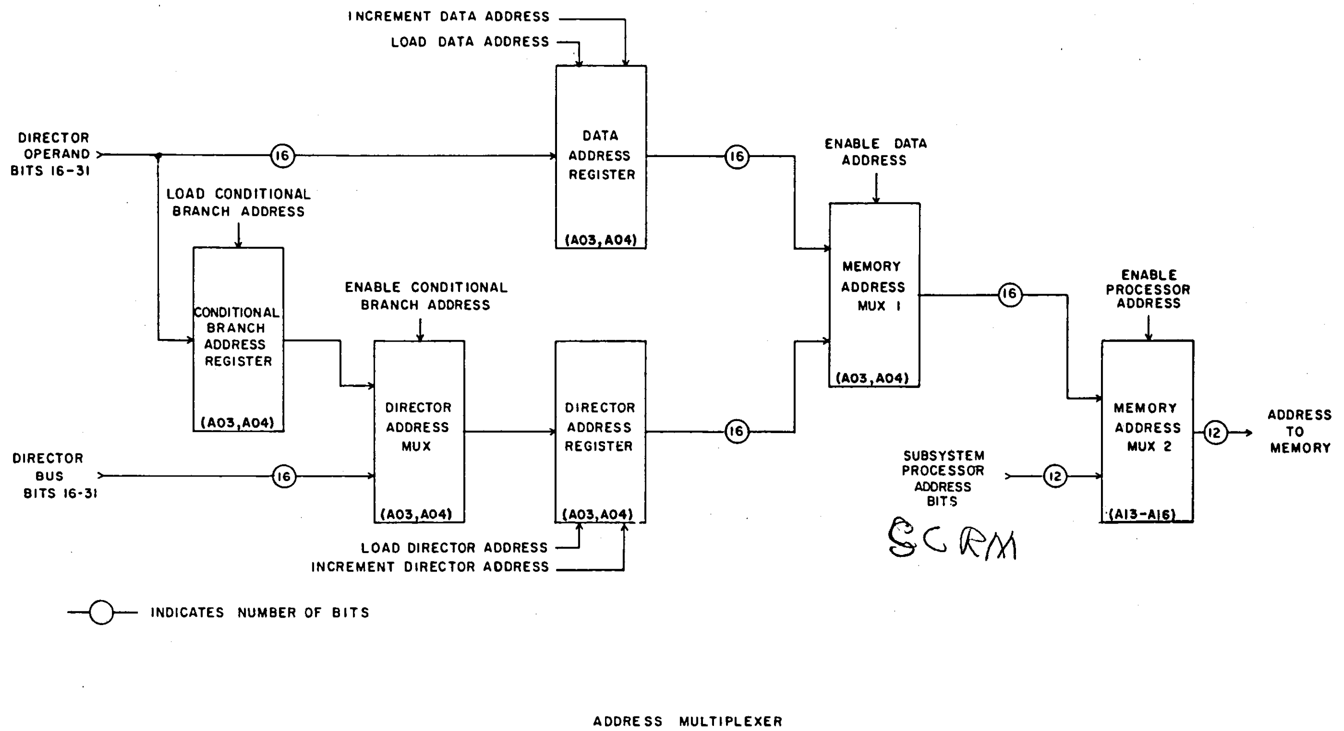
As noted previously, the subsystem processor normally alternates its accesses with those of the director buffer. However, when the director buffer is full or if director loading has been stopped, the subsystem processor has complete control of the memory scanner. Each access averages about 1.5 microseconds but may vary with chip switching times.

Address Multiplexer

The control logic accesses memory for three purposes. It obtains directors from memory, it transfers data to and from memory, and it controls the subsystem processor access to memory. Each of these three accesses reference different locations in memory. The address multiplexer routes the correct address to memory at the proper time. Figure 3-91 is a block diagram of the address multiplexer. Although each of the addressing elements is 16 bits wide, only 12 address bits are sent to memory.

During normal operation (no special enables), address bits are routed from the director address register through the memory address multiplexer 1 and the memory address multiplexer 2 to memory. After each access, the director address register is incremented so that directors are obtained from sequential locations in memory. The director address multiplexer provides for changing the director address in one of two ways. During execution of an unconditional branch director, the new address is taken directly from the director bus, routed through the director address multiplexer, and loaded into the director address register. Future director references are made starting at that memory location. During execution of a conditional branch director, the conditional branch address is held in the conditional branch address register. If the branch conditions are met, the branch address is routed through the director address multiplexer and loaded into the director address register. Future director references are made starting at that memory location. If the branch conditions are not met, the conditional branch address will remain in its register until a new branch address is loaded.

During data transfer operations, the data address is loaded into the data address register from the address field of an address director. When the data transfer is executed, the data address is routed from the data address register through the memory address multiplexer 1 and the memory address multiplexer 2 to memory. The data address register is incremented by one after each transfer until all data has been transferred.



SCRM

Figure 3-91. Address Multiplexer Block Diagram

The memory address multiplexer 2 is controlled by the memory scanner. When a subsystem processor reference is honored by the memory scanner, the address bits from the subsystem processor are enabled through the memory address multiplexer to memory. All other memory accesses are from the controller address multiplexer.

Director Buffer

The director buffer is a circular buffer which will hold 16 32-bit words. The discussion of its operation is divided into the following areas.

1. Director buffer control
2. Director buffer loading
3. Director buffer look-ahead decodes
4. Director buffer execution

Director Buffer Control

The director buffer is loaded and unloaded under control of input and output pointers and flags which set if the buffer is empty or full. The buffer is considered to be full when the input and output pointers become equal when the input pointer is incremented by one. The buffer is considered empty if the input and output pointers become equal when the output pointer is incremented by one. From a full condition, only the output pointer can be moved. From an empty condition, only the input pointer can be moved. The director buffer is also controlled via the normal channel interface from the subsystem processor. This control is via functions received over normal output channel 8. These functions and their effects on the director buffer are listed below.

- 0020 Master clear - This function code clears the director buffer and places it in an empty condition. It also clears the director address register so that the first director reference is made from location 0 in memory.
- 0018 Load director buffer - This function code causes the director buffer to start loading directors. The director buffer references memory through the memory scanner. The memory location referenced will be that address which is in the director address register. This will be address 0 following a master clear function (0020).

- 0022 Execute directors - This function code causes the director buffer to begin execution. If the buffer is not empty, the first director is extracted and placed in the director output register. The output pointer is then incremented by one. The next director will be extracted as soon as the previous one has completed execution. If the buffer is empty when this function is received, the next director loaded into the buffer will execute immediately.

Director Buffer Loading

Upon receipt of an 0018 function code (load director buffer), the director buffer accesses sequential locations in memory through the memory scanner. After each 16-bit access, the director address register is incremented by one. Since these are only 16-bit references and the director buffer is 32 bits wide, two references are required to fill one position in the buffer. The control logic keeps track of the upper and lower director buffer accesses. As soon as the lower bits (16 through 31) are loaded into the buffer, the position is considered full and the input pointer is moved (incremented by one). For certain special directors, the loading pattern is altered. These directors are discussed below under director buffer look-ahead decodes.

Director Buffer Look-Ahead Decodes

The first 15 bits of a director contain the director function code. As these bits are brought in from memory, they are decoded to determine if one of the following directors is involved.

- Unconditional branch - This director is not loaded into the director buffer. Instead, the second 16 bits are obtained from memory and immediately placed in the director address register. Succeeding director references are then obtained starting at the new address location.
- Stop loading directors - This director is not loaded into the director buffer. It is only 16 bits in length and will stop director loading as soon as it is decoded. It increments the director address register by one. An 0018 function code (load director buffer) is required to restart director loading. Upon receipt of this function, director loading will resume from the address in the director address register.
- Terminate - This director is loaded into the director buffer. However, since it is only 16 bits in length, the control logic increments the input pointer immediately. The next memory access is then treated as the first 16 bits of the next director.

- Address director - This director is 64 bits in length and requires two locations in the director buffer. Special treatment is required so that the first 16 bits of the second location are not decoded as a function code. When an address director is decoded, the control logic places the next three memory references into the director buffer without examining them in the look-ahead decode.

If none of the above conditions exist, the first 16 bits of the director are placed into the director buffer and another memory reference is made. The next 16 bits are not examined by the look-ahead logic since these bits will not contain a function code. As soon as the second 16 bits have been placed in the director buffer, the input pointer is incremented and another memory reference is made.

Director Execution

As soon as an 0022 function code (execute directors) is received, (assuming the director buffer is not empty) the first director is transferred from the director buffer to the director output register and the output pointer is incremented by one. The director is immediately decoded and starts an execution cycle which strobes the director information out of the director output register. The next director is immediately transferred from the director buffer into the director output register. It is decoded while the first director is completing its execution. When execution of the first director is completed, the second begins execution and the next director is transferred into the director output register for decoding. Execution of data directors is considered to be complete as soon as all director parameters have been established, even though the complete data transfer is not yet done. This allows support directors to be decoded and executed while data is being transferred. Another data director, however, cannot be executed until a data transfer for a previous data director is complete.

Two directors require special attention when they are decoded in the director output register. These directors and their treatment are discussed below.

- Address director - This director requires two director buffer locations. When the first half of the director is decoded and executed, it establishes conditions which prevent the second half from being decoded as a separate director.
- Terminate director - If a terminate director is decoded, no further directors are extracted from the director buffer. The output pointer is incremented by one and director execution resumes with the next director buffer location as soon as an 0022 function code (execute directors) is received.

Register File

The register file is a buffer which will hold sixteen 16-bit words. It can be used as either an addressable register or as a circular buffer. It is loaded and unloaded under control of input and output pointers. These are counters which may be directly loaded with the desired register file address or which are incremented after each access for circular buffer operation.

Information is stored in the register file from several sources. These sources and the purpose of the data are as follows:

Data Source	Purpose
Core memory	Data is loaded directly from memory via a load register file from memory director. The data is used as operands for various directors.
Director operand	Data is loaded into a register file location from the operand field of a load register file director.
Data shift register	Data is placed in sequential register file locations directly from the data shift register.
Arithmetic logic unit	Results of an arithmetic operation are stored in the register file.
Status bus	Status from the selected disk storage unit is stored in the register file.

Information is taken from the register file and sent to various destinations. These destinations and the purpose of the data are as follows:

Data Destination	Purpose
Core memory	Data is stored in memory as a result of a store register file director.
Byte counter	The I/O length for data directors is taken from the register file if specified by the director.
Data shift register	Data is taken from the register file and sent to the shift register for writing on the disk.

Data Destination	Purpose
Status bus	Data is taken from the register file and sent to the subsystem processor over normal input channel 04.
Arithmetic logic unit	For arithmetic or logical operations in the arithmetic logic unit, one operand is always taken from the register file.

Arithmetic Logic Unit

The arithmetic logic unit performs three arithmetic operations, two logical operations, and a bit compare operation. The arithmetic operations are add (F=A+B), subtract (F=A-B), and decrement (F=A-1). The logical operations are true data (F=A) and complement data (F=$\overline{A}$). For each of these operations, operand A is taken from the register file and operand B is taken from the director operand field. In the bit compare operation, data from the register file is compared with the data from the director operand field. A flag is set if a compare occurs in any bit position. The results of any ALU operation can be stored in the register file.

Data Shift Register

The data shift register assembles or disassembles data going to or coming from the selected disk storage unit. During write operations (data to disk) the shift register accepts data from either the register file or the data buffer. This data is in parallel form and can be either 6 or 8 bits in length. The shift register converts the parallel data to serial data for writing on the disk. During read operations the shift register assembles either 6-or 8-bit bytes and transfers the data in parallel form to either the data buffer or the register file.

The shift register also contains a compare circuit which compares data from the register file with data being read from the disk. When this function is used, the data read from the disk cannot be stored in the register file.

Data Buffer

The data buffer is a four-stage 16-bit buffer which handles both input and output data. During input operations, data enters the buffer from either the system coupler or from core memory and is then transferred to the data shift register. During output operations, data enters the buffer from the data shift register and is transferred to either the system coupler or core memory. During both input and output operations, the output of the data buffer can be sampled as status.

All data transfers with either the system coupler or core memory are 16-bit transfers. Data transfers with the data shift register are handled as two 8-bit bytes. When the second 8-bit byte enters or leaves the data buffer, the transfer is considered complete. Transfers between stages of the data buffer are always in 16-bit parallel transfers and require 50 nanoseconds per stage.

Clock Operations

Figure 3-92 is a simplified logic diagram which shows all clocks generated within the control logic. A 27.2-MHz oscillator provides three clock outputs: WCLK1, WCLK2, and 13-6 (13.6 MHZ). The cycle times and relationships of these signals are shown on the figure. The WCLK1 signal generates the internal clocks which are used for all timing not directly related to reading and writing data. The WCLK2 and 13-6 signals are used in conjunction with the write clock to send write data to the disk.

The shift clock signals, SRCL1 and SRCL2, are used for clocking data related operations such as the data shift register, bit counter, and the checkword generator. During write operations, the shift clock is generated by the WCLK1 signal. During read operations, it is generated by the read clock. Generation of the read clock is also shown on the figure.

During clock step mode operations both the shift clock and the internal clock are generated by the step clock. This clock is a 100-nanosecond one-shot pulse triggered by a normal channel function.

Data Coding Principles

The method used to record data on the disk is referred to as modified frequency modulation (MFM). With this technique, a 1 is defined by the leading edge of the pulse occurring at the center of the cell and the trailing edge of the pulse occurring at the end of the cell. A 0 is defined by the leading edge of the pulse occurring at the start of the cell and the trailing edge occurring at the center of the cell, except that the pulse is omitted for the first 0 following a 1 bit.

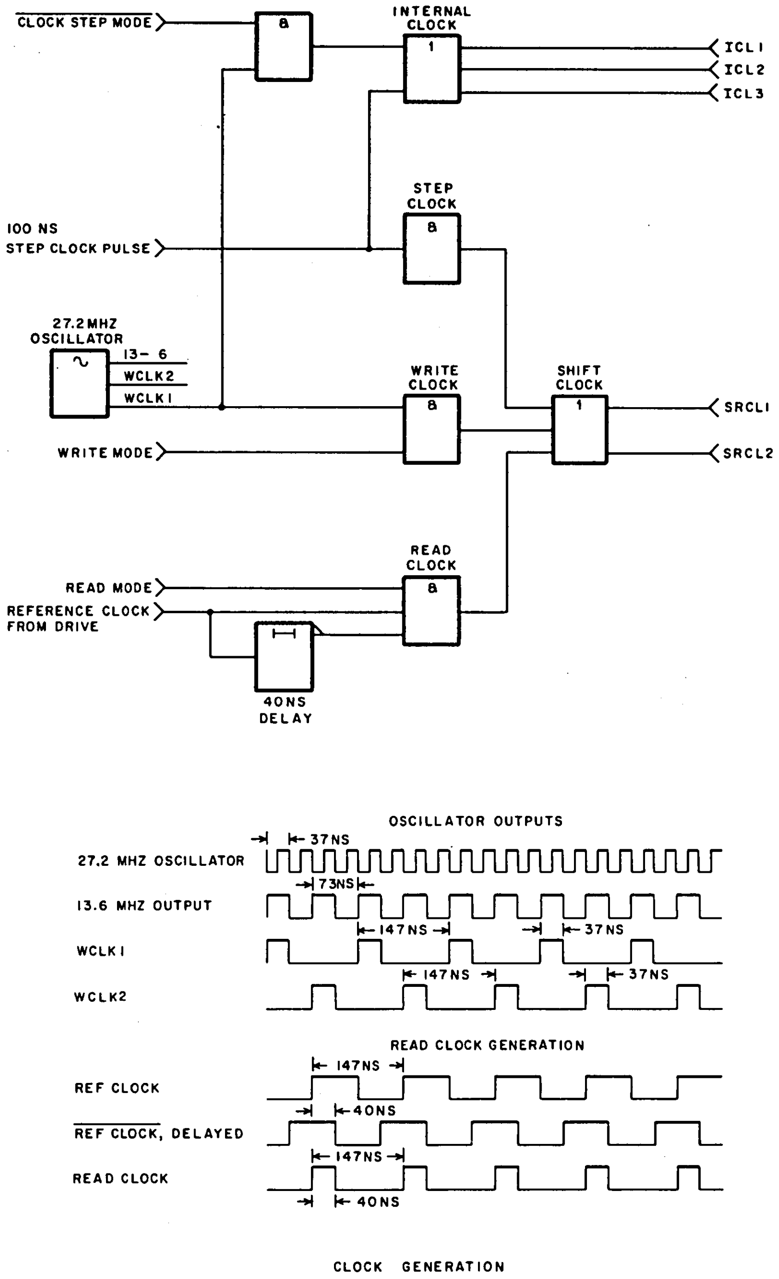
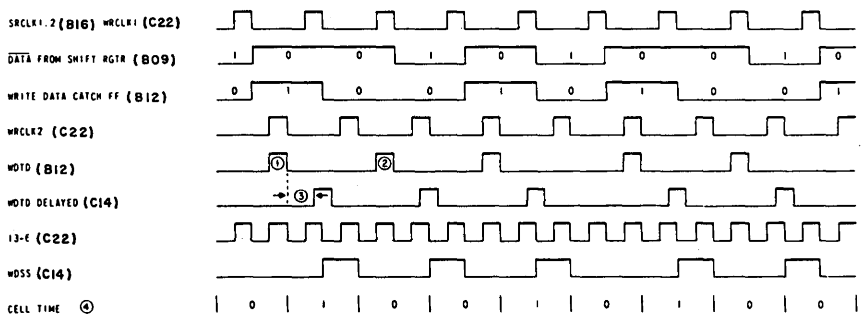


Figure 3-92. Clock Generation

Figure 3-93 illustrates the method of coding the data pulses which are transmitted to the disk. The write data catch flip-flop catches the data as it is shifted out of the shift register. The output of the write data catch flip-flop is ANDed with the output of the last stage of the shift register to provide the look-ahead necessary for proper coding of the 0 bits.



NOTES: ① WDTD = WRCLK2 • WRITE DATA CATCH SET.

② WDTD = WRCLK1 • WRITE DATA CATCH CLEAR • $\overline{\text{DATA}}$ FROM SHIFT REGISTER.

③ DELAYED ABOUT 55 NS TO CENTER DATA PULSE AT NEGATIVE EDGE OF 13-6 CLOCK.

④ MFM DATA CODING: "1" LEADING EDGE AT CENTER OF CELL.
"0" LEADING EDGE AT START OF CELL UNLESS PREVIOUS BIT WAS A "1".

Figure 3-93. Data Coding - Modified Frequency Modulation

The MFM encoded data is annotated as WDSS on Figure 3-93. The cell times are also shown to illustrate the relationship of the data pulses to the cell times. In practice, the initial data is transmitted as all 0 bits to allow the disk storage unit to achieve synchronization.

When the data is read from the disk, the disk storage unit discriminates the recorded information into data pulses and clock pulses and transmits this information to the disk controller. The read data catch flip-flop catches the data pulses. The clock pulses are used to control all clocking operations within the disk controller to ensure proper synchronization.

Checkword Generator

The checkword generator is a programmable 36-bit cyclic encoder/decoder. It is comprised of six printed circuit boards with each board containing a 6-bit shift register and catch registers for 6 bits of both a divide and multiply polynomial. The checkword generator is programmed both by backpanel wiring and through selection of divide and multiply polynomials.

Backpanel wiring in the disk controller is such that only 32 bits of the checkword generator are active. This, together with the divide polynomial, provides for a natural code length (maximum data field length) of 42,987 bits. The backpanel wiring also is used to interconnect the six boards of the generator and to provide feedback loops within individual boards.

The divide polynomial is a 32-bit quantity which is established via software and is loaded into the divide polynomial generator by the load polynomial director. It determines the following characteristics for the checkword generator.

1. Error detection capability
2. Error correction capability
3. Error location capability
4. Natural code length (maximum data field length)

The divide polynomial used in the disk controller is always hexadecimal 00A0 0805. The divide polynomial is used during both read and write operations to affect the transfer of data between stages of the checkword generator.

The multiply polynomial is also a 32-bit quantity which is established via software and is loaded into the multiply polynomial catch register by the load polynomial director. The multiply polynomial is used only during read operations and is selected to reduce the time required to perform error correction for data lengths which are less than the natural code length of 42, 987 bits. The multiply polynomials used in the disk controller and their associated data field lengths are listed below.

Multiply Polynomial (Hexadecimal)	Data Field Length (Decimal)
3DC4 01EE	24 bits
E360 0713	3840 bits
6920 0348	3864 bits
61E0 230F	32, 768 bits
00A0 0805†	42, 987 bits
0000 0000††	

During write operations only the divide polynomial is used. The multiply polynomial must be set to zero. Each bit which is set in the divide polynomial will cause the data from the preceding stage of the checkword generator to be exclusively ORed with the feed-back bit. The feed-back bit is an exclusive OR of the incoming data bit and the bit in the last stage of the generator. If the polynomial bit is not set, data is shifted directly between stages. This operation results in the generation of a checkword residue which remains in the generator after the last data bit has been transferred. The control logic then converts the checkword generator to a normal shift register and writes the checkword residue on the disk.

During read operations, both the divide and multiply polynomials are used. The read data plus the checkword are routed through the checkword generator. The divide polynomial performs the same as during a write operation. The multiply polynomial forces a zero's transfer equal to the difference between the natural code length and the

†For the natural code length of 42, 987 bits, the multiplier polynomial equals the divide polynomial.

††This polynomial inhibits the input to the checkword generator during a read operation. If generator was precleared, no checkword error can result. If generator was preset, a checkword error is forced.

actual data transfer length. The combination of multiply and divide polynomial bits determines the manner in which data is passed between stages. These combinations are:

1. If multiply and divide bits are both set, exclusive OR the feedback bit with the data from the previous stage.
2. If multiply bit only is set, exclusive OR the incoming data bit with the data from the previous stage.
3. If divide bit only is set, exclusive OR the bit in the last stage of the generator with the data from the previous stage.
4. If neither bit is set, perform a normal shift.

If no errors occurred during the data transfer, the net result of the operation will be a zero residue in the checkword generator. If an error has occurred, the contents of the generator will be nonzero. Controlware will sense this condition via a status bit and will initiate error correction if desired. Prior to performing error correction, the multiply polynomial must be set to zero. Error correction consists of loading the byte counter with a shift count equal to the number of bits in the data field. The checkword generator is then shifted once for each bit in the shift count while using the divide polynomial. The shift count is decremented after each shift. The operation continues until bits 11 through 31 of the checkword generator are all zero or until the shift count becomes zero. In the first case the error is correctable. The contents of stages 00 through 10 of the checkword generator are the 11-bit correction vector. The shift count remaining in the byte counter must be subtracted from the initial shift count. This will provide a bit displacement from the leading edge of the data buffer at which the correction vector must be exclusively ORed to correct the data. If the shift count decrements to zero, the error is considered to be uncorrectable. In this case, the HLP must request a reread of the data if it wants to attempt additional error recovery.

Open Cable Detector Current Supply

Each disk storage unit is connected to the control logic through two cables. Each cable contains a differential signal which feeds a line receiver. If the differential signal is present, it indicates that the cable is connected to the control logic and that the control logic is operating at normal voltage levels. If the cable is open or if the control logic is operating at below normal voltages, the line receivers are disabled and the disk storage unit will not respond to control signals from the control logic.

The differential reference signal is supplied by the open cable detector current supply (Figure 3-94). This circuit monitors the ±5-volt supply in the control logic and outputs a constant 800-millivolt reference voltage unless the supply voltage falls below 4.5 volts. At this point, the circuit turns off the 800 -millivolt reference signal. This allows the disk storage unit to disregard any spurious signals which could be generated as the supply voltage falls below 3.2 volts which is the minimum operating voltage for the TTL logic chips.

The open cable detector current supply also adjusts for the number of disk storage units which are on-line. Up to eight drives (16 cables) can be driven by the circuit. Each cable requires about 6 milliamperes of current from the current supply. As disk storage units are switched in and out of operation, the current supply adjusts its output voltage to maintain a constant current to the unit cables. The current supply operates as follows:

The current supply consists of a voltage comparator, a reference voltage source, a differential amplifier, and an output regulator. The voltage comparator monitors the +5-volt supply voltage. If this voltage falls below 4.5 volts, the comparator disables the differential amplifier which turns off the output regulator and drops the reference voltage to the unit cables. If the supply voltage is above 4.5 volts, the differential amplifier is enabled and controls the output of the output regulator.

The reference voltage source consists of two transistors, a zener diode, and a voltage divider. It provides a constant 800-millivolt reference signal to the differential amplifier.

The differential amplifier compares the 800-millivolt reference voltage to the output voltage of the output regulator. If the output voltage drops below 800 millivolts, the differential amplifier adjusts the bias of the output regulator which then increases the output voltage. Conversely, if the output voltage rises above 800 millivolts, the differential amplifier changes the bias voltage to reduce the output voltage.

The output regulator tracks the bias input of the differential amplifier to maintain a constant 800-millivolt output voltage.

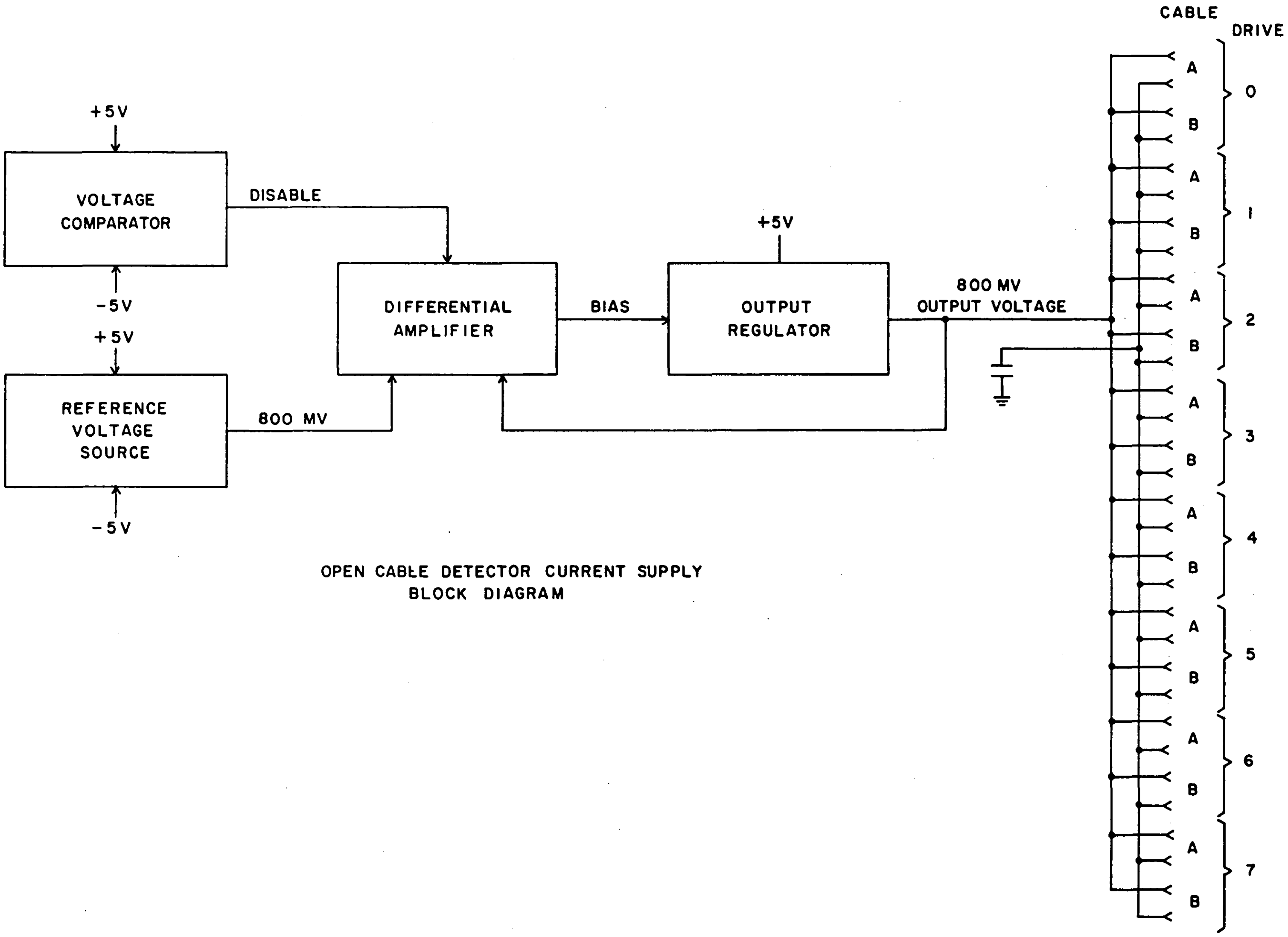


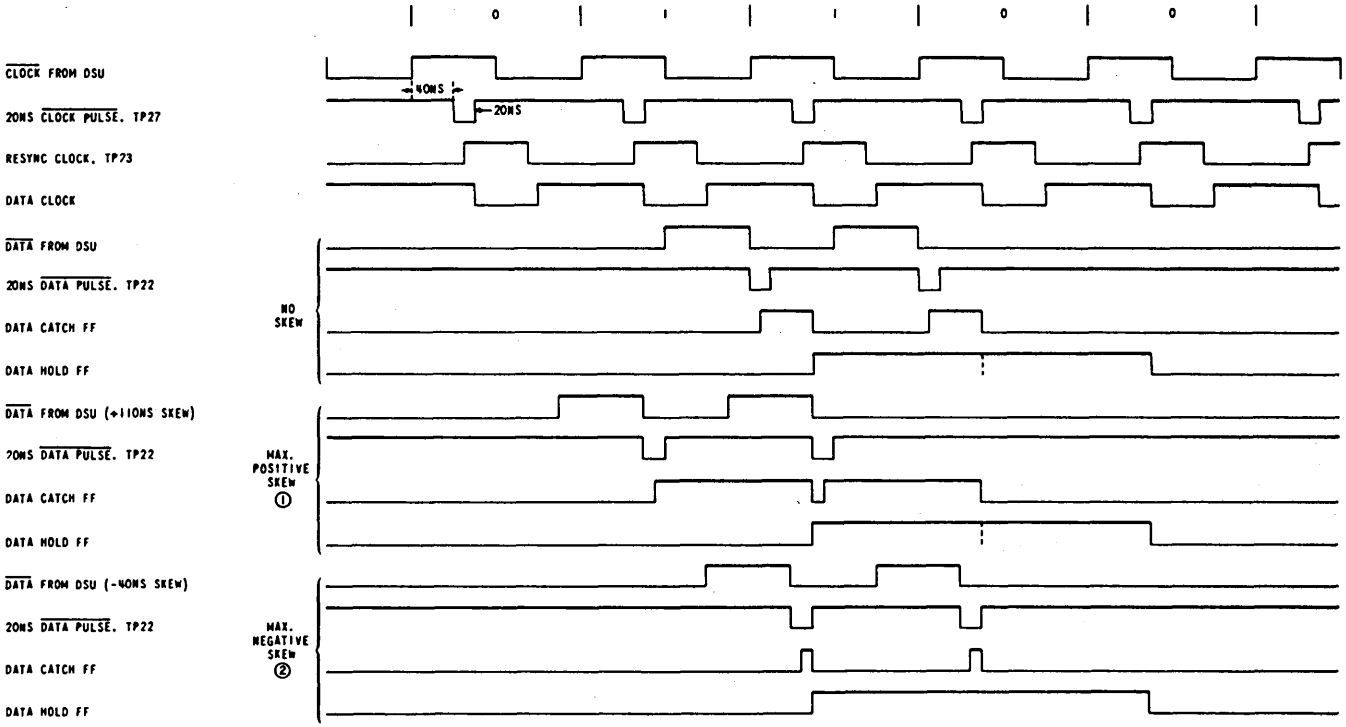
Figure 3-94. Open Cable Detector Current Supply Block Diagram

Data Resynchronization

Skew errors may occur during transmission of clock and data signals from a disk storage unit to the control logic. The resync logic resynchronizes these inputs to reduce these errors. The clock pulses are routed through a delay line and a pulse forming network which generates a negative going 20-nanosecond pulse 40 nanoseconds after the trailing edge of the clock pulse (Figure 4-17). This pulse sets the resync clock FF which is then clocked to a clear state by the 60 nanosecond tap of another delay line. The result is a clock pulse with a pulse width of 73.5 nanoseconds nominal. The data pulses are also routed through a delay line and pulse forming network to generate a 20 nanosecond pulse on the leading edge of the data pulse. This pulse presets the data catch FF. The generated clock pulse is used to toggle the data catch FF to a clear state while simultaneously transferring its contents to the data hold FF. The output of the data hold FF is thus in sync with the clock pulses. Both TTL outputs and differential outputs are provided for the resynchronized clock and data pulses. The TTL data output has its trailing edge delayed 20 nanoseconds to provide a wider data pulse for use within the control logic.

If no clock pulses are being received and gated through the clock delay line, the lock out gate is made. This gate holds the data and resync clock flip-flops in a clear state and sets the lockout FF. The first clock pulse will remove the lockout signal and allow the resync clock FF to set. When the resync clock FF is cleared via its delay line, the trailing edge of the clock pulse will toggle the lockout FF to a clearstate and the circuit will again be operative.

The resync circuit will handle maximum skew errors of about 110 nanoseconds positive (data leading clock) and about 40 nanoseconds negative (data trailing clock).



NOTES: ① MAXIMUM PERMISSIBLE POSITIVE SKEW ≈ 110NS.

② MAXIMUM PERMISSIBLE NEGATIVE SKEW ≈ 40NS.

Figure 3-95. Data Resynchronization Timing Diagram

Data Path Switching

Data is routed throughout the control logic via various data buses. These buses are comprised of open collector AND/OR invert gates which have their outputs tied together to form a wired AND gate. If data is enabled through any one of the input gates, the AND gate is broken and the data bus is enabled. The output of the data bus is inverted to provide true data for use in the control logic or routing to another data bus.

Figure 3-96 illustrates the four data buses used in the control logic and their interrelationships. Each data bus is described separately in the following paragraphs.

Register File Bus

The register file bus is located on data paths 1 and 2 (diagrams A19 through A24 and A07 through A12). It receives inputs from the disk storage unit, the memory bus, the data catch register, the arithmetic logic unit, and from the director bus. It provides inputs to the byte counter, word counter, arithmetic logic unit, memory bus, and write data bus. The contents of the register file bus can be written into the register file or obtained from the register file, depending on how the register file is enabled.

Memory Bus

The memory bus is located on data path 1 (diagrams A19 through A24) and on the memory interface boards (diagrams A13 through A16). It receives inputs from the memory, data buffer, and register file. It provides inputs to the data buffer, the director bus, and to memory.

Director Bus

The director bus is located on data path 1 (diagrams A19 through A24). It receives inputs from the memory bus. Since the director bus is 32 bits wide, inputs are provided as two 16-bit words. The director bus provides inputs to the director output register which then fans the director bits throughout the control logic. The contents of the director bus can be written into the director buffer or can be obtained from the director buffer depending on how the director buffer is enabled.

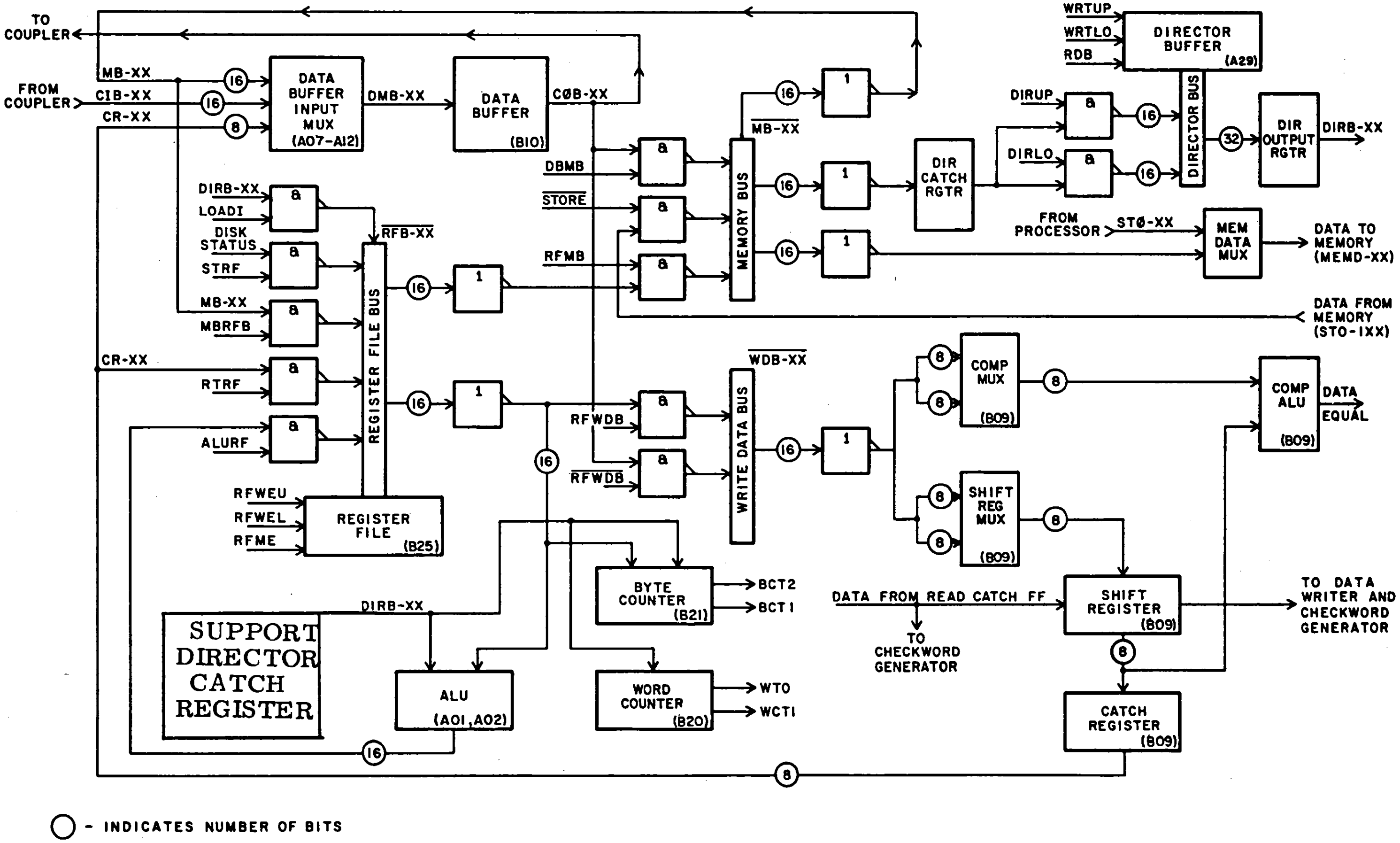


Figure 3-96. Data Path Switching Block Diagram

Write Data Bus

The write data bus is located on data path 2 (diagrams A07 through A12). It receives inputs from either the register file or the data buffer and provides inputs to the compare multiplexer and the shift register multiplexer on the data shift register board.

Data Buffer

The data buffer does not use any bus logic but is included on Figure 3-96 to show its relationship to the other data paths. It can receive inputs from the memory, coupler, or data catch register. It provides inputs to the memory bus, coupler, and write data bus.

OPERATIONAL SEQUENCES

The following paragraphs provide step-by-step procedures for execution of primary sequences within the control logic. The discussion is divided as follows:

1. Hardware functions - These paragraphs cover communication between the subsystem processor and the control logic over the normal channels.
2. Common sequences - These paragraphs cover sequences which are common to many of the directors. These sequences are detailed only once and then referred to where required.
3. Director sequences - These paragraphs describe the execution of each of the directors. Support directors are described first in the order of their function codes followed by the data directors, also in the order of their function codes. Table 3-4 shows the order in which the directors are discussed.
4. Subsystem processor memory requests

All references to card location and test points refer to locations on chassis A11 unless otherwise specified.

TABLE 3-4. DIRECTOR FUNCTION CODES

Function Code (Base 16)	Support Directors
00	Load register file from memory
02	Address
04	Conditional branch
08	Terminate
10	Subtract
12	Add
14	Test
16	Complement
80	Load register file direct
82	Keypoint
84	Load polynomial
86	Load hardware conditions
88	Copy disk status
92	Stop loading directors
94	Unconditional branch
98	Store register file
C0	Enable disk functions
C8	Disable control select
E0	Enable/disable disk status
F0	Initiate error correction

Function Code (Base 2)	Data Directors
00100	Read normal
00101	Read skip
00110	Read checkword
00111	Read address pattern
01000	Write normal
01001	Write Nx
01010	Write checkword
01011	Write address pattern
01100	Compare normal
01111	Delay

Normal Channel 8 Functions

Normal channel 8 is used to pass hardware functions from the subsystem processor to the control logic. The applicable function codes are listed and defined in the following table. The codes are flowcharted on Figures 3-97 through 3-105. The flow charts indicate the test point and chassis location on chassis A11 for each signal.

All functions originate on card B27. The SNOC8 signal generates a one-shot pulse which then enables the applicable function decode. A 200-nanosecond one-shot is used for all codes except those related to clock step operations. Clock step operations use a 100-nanosecond one-shot. The function codes and their definitions are:

Function Code	Definition
0018	Load director buffer - This code causes the control logic to start loading the director buffer from memory.
0020	Master clear - This code generates a 15-microsecond pulse which causes the control logic to halt operations. It is followed by a two microsecond pulse which clears or voids all registers and drops all tag and select lines to the disk storage units.
0022	Execute directors - This code causes the control logic to begin executing the directors in its director buffer.
0024	Stop execution - This code causes the control logic to stop executing directors. Any director in progress will complete its execution. When the current director is complete, the control logic turns off the write and erase heads in the disk storage unit. This function does not master clear the control logic. An 0022 code will cause the control logic to resume execution from the current address in the director address register.
0025	Clear parity error - This code clears the parity error flip-flop and drops the composite status bit if no other error conditions exist. A parity error causes a fault condition which clears the director execution flip-flop. When the parity error condition has been cleared, an 0022 code (execute directors) is required to resume operation.
0028	Enter clock step mode - This code inhibits the internal clock and puts the control logic under program control.
0029	Exit clock step mode - This code disables the clock step mode and enables the normal internal clock.
002A	Step clock - This code causes a 100-nanosecond clock pulse during the clock step mode of operation.

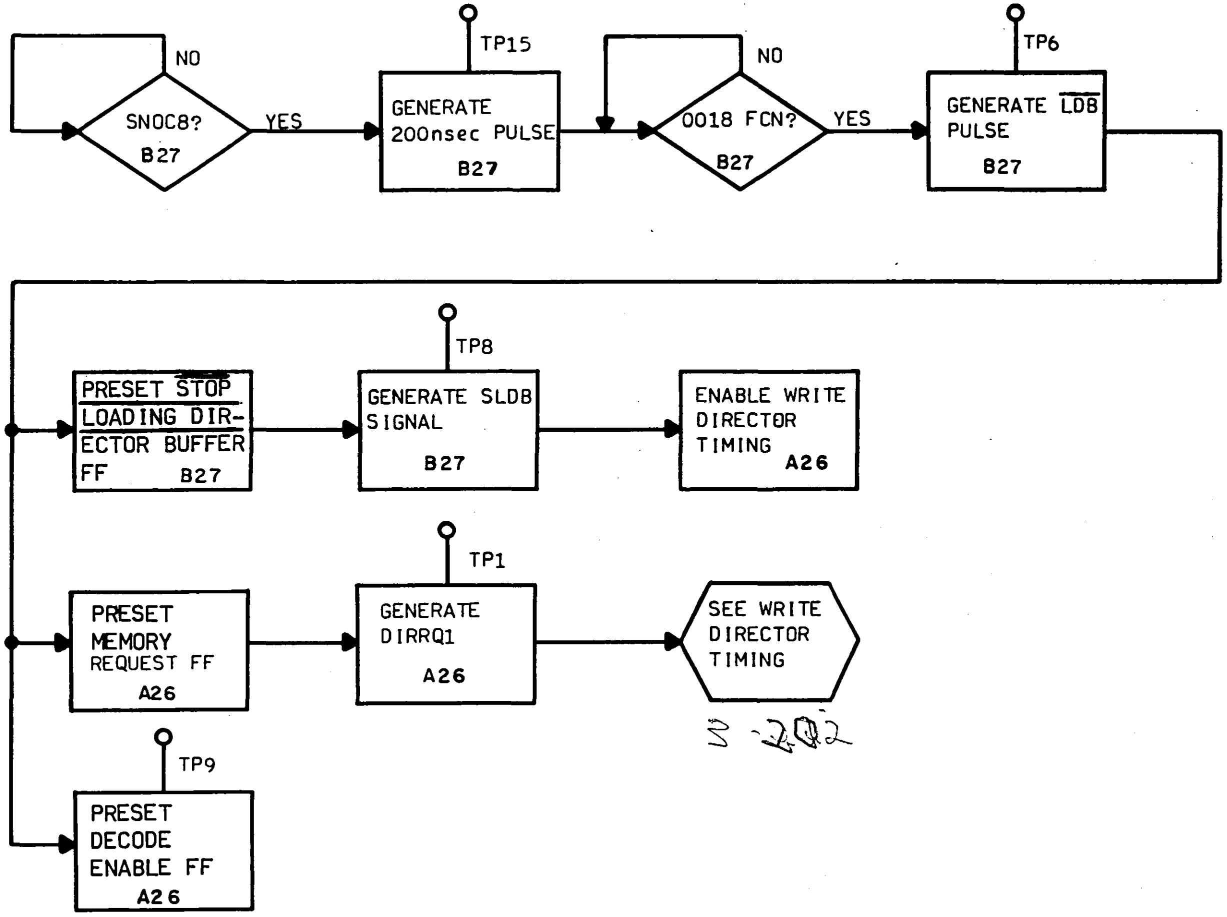


Figure 3-97. Load Director Buffer - 0018

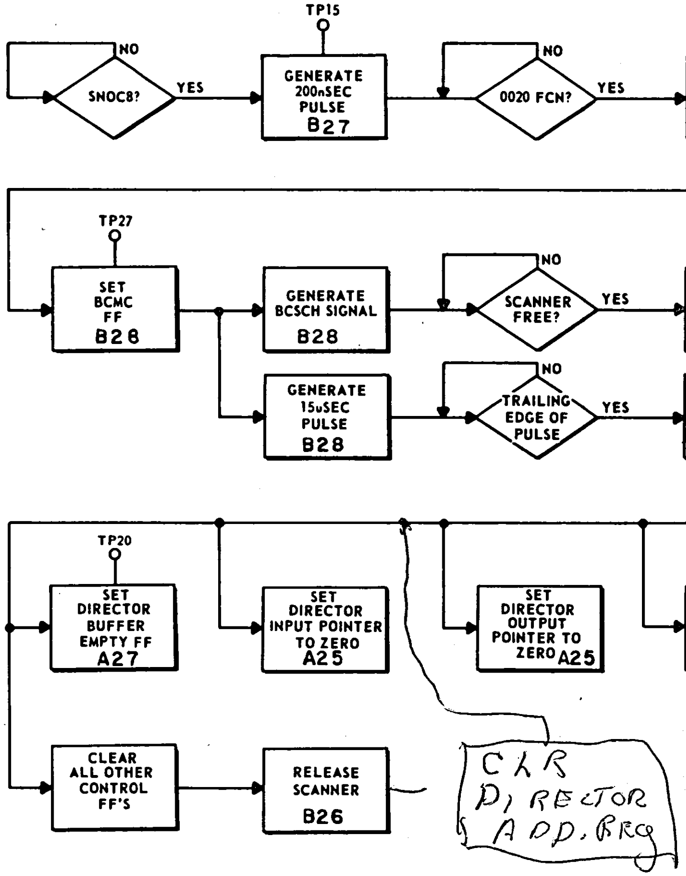
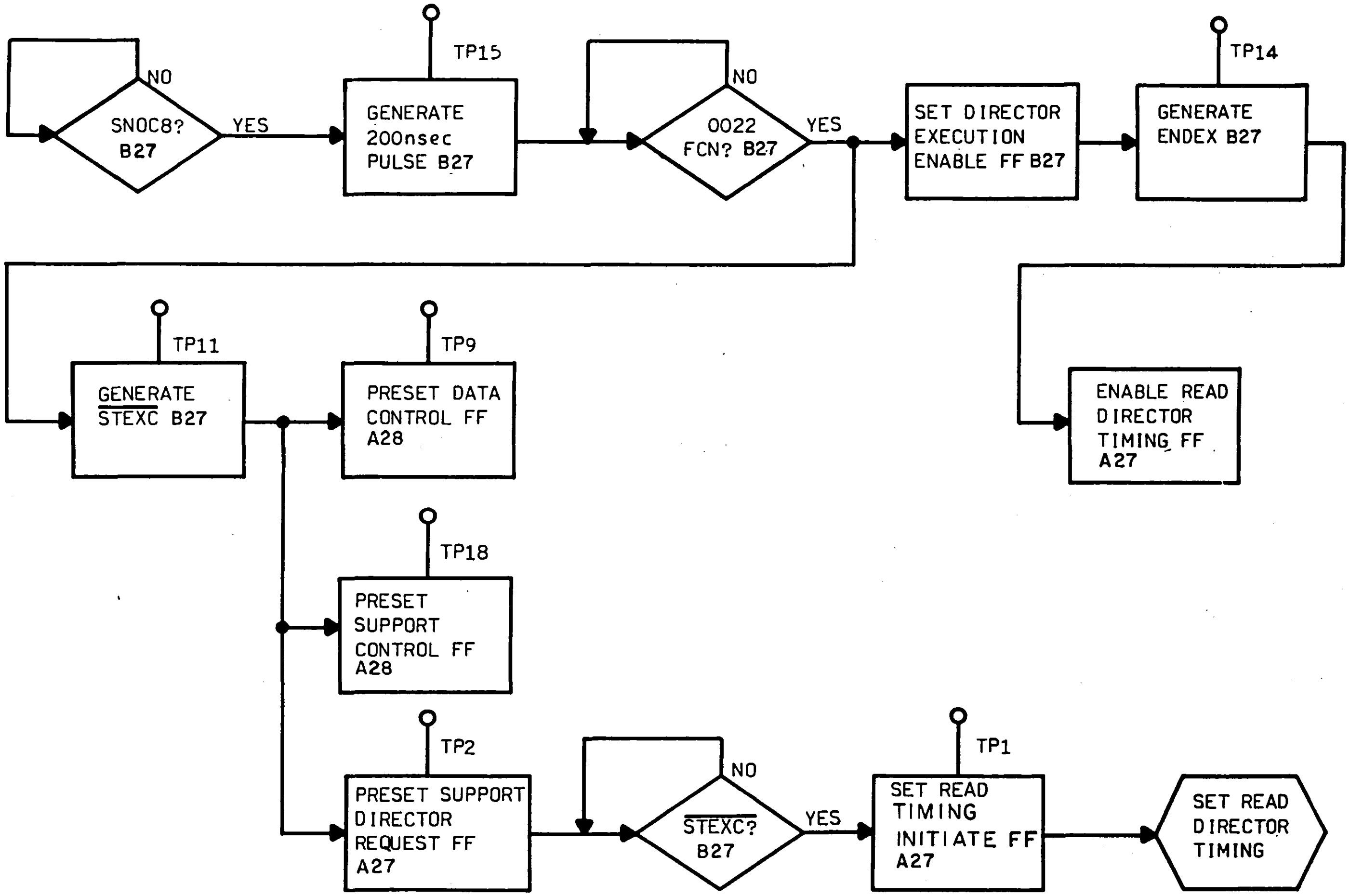

TP15
SNOC8?
NO
YES
GENERATE 200nSEC PULSE B27
0020 FCN?
NO
YES
GENERATE BCMC PULSE B27
TP27
SET BCMC FF B28
GENERATE BCSCH SIGNAL B28
SCANNER FREE?
NO
YES
BLOCK SCANNER B26
TP26
GENERATE 15uSEC PULSE B28
TRAILING EDGE OF PULSE
NO
YES
GENERATE 2uSEC MCI PULSE B27
TP20
SET DIRECTOR BUFFER EMPTY FF A27
SET DIRECTOR INPUT POINTER TO ZERO A25
SET DIRECTOR OUTPUT POINTER TO ZERO A25
DROP ALL SELECT LINES TO DISK UNIT B14, C12
CLEAR ALL OTHER CONTROL FF'S
RELEASE SCANNER B26
CLR DIRECTOR ADD. REG

Figure 3-98. Function Master Clear - 0020


NO
SNOC8? B27
YES
TP15
GENERATE 200nsec PULSE B27
NO
0022 FCN? B27
YES
SET DIRECTOR EXECUTION ENABLE FF B27
TP14
GENERATE ENDEX B27
TP11
GENERATE STEXC B27
TP9
PRESET DATA CONTROL FF A28
ENABLE READ DIRECTOR TIMING FF A27
TP18
PRESET SUPPORT CONTROL FF A28
TP2
PRESET SUPPORT DIRECTOR REQUEST FF A27
NO
STEXC? B27
YES
TP1
SET READ TIMING INITIATE FF A27
SET READ DIRECTOR TIMING

3-210

Figure 3-99. Start Director Execution - 0022

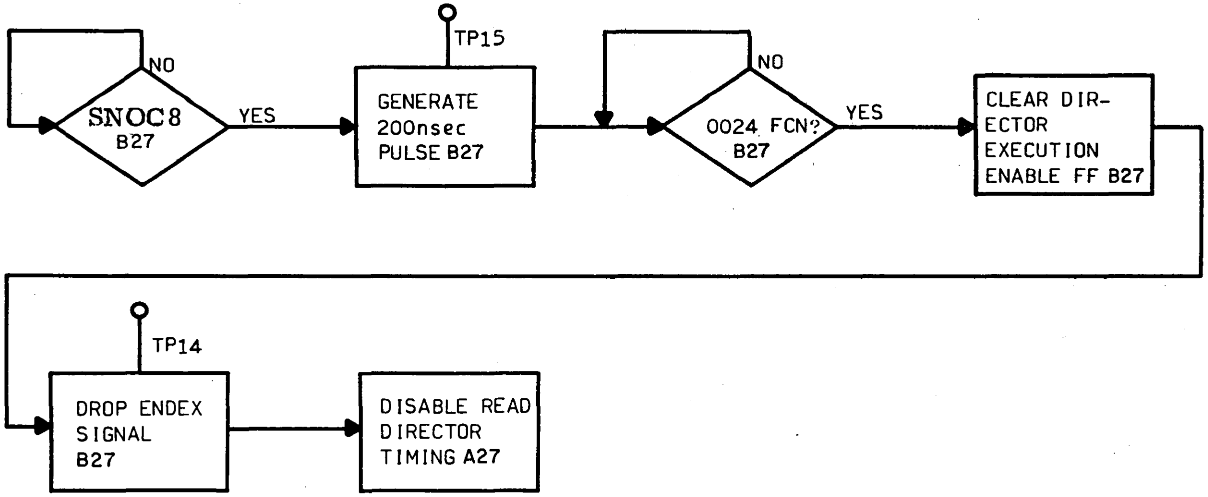


Figure 3-100. Stop Execution - 0024

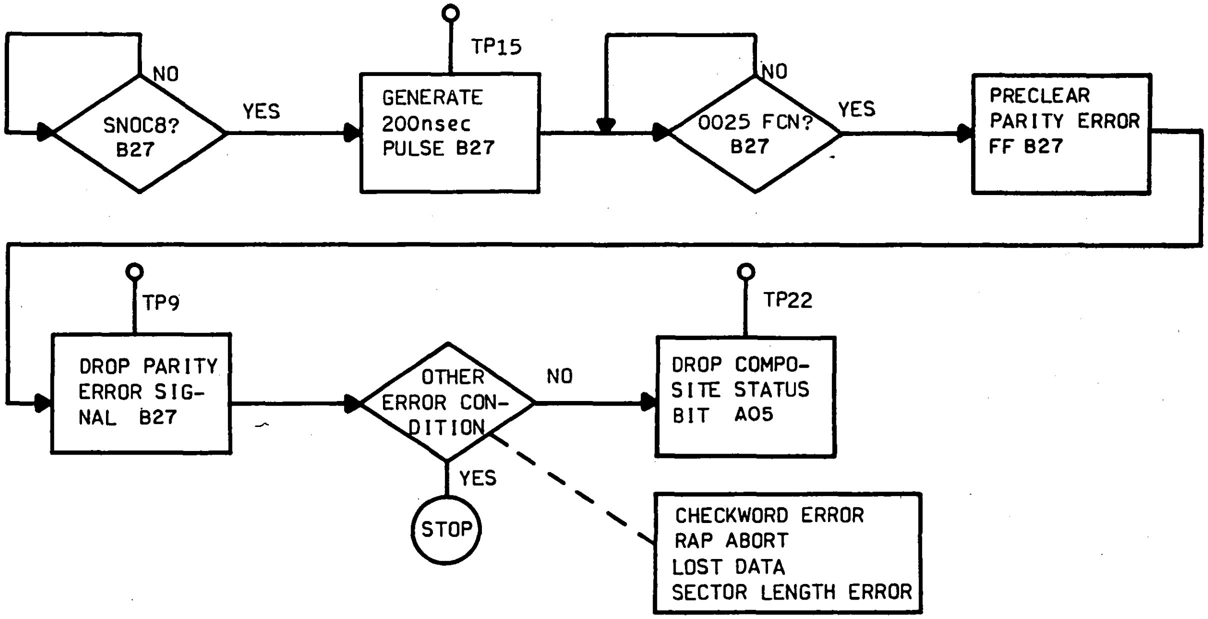


Figure 3-101. Clear Parity Error - 0025

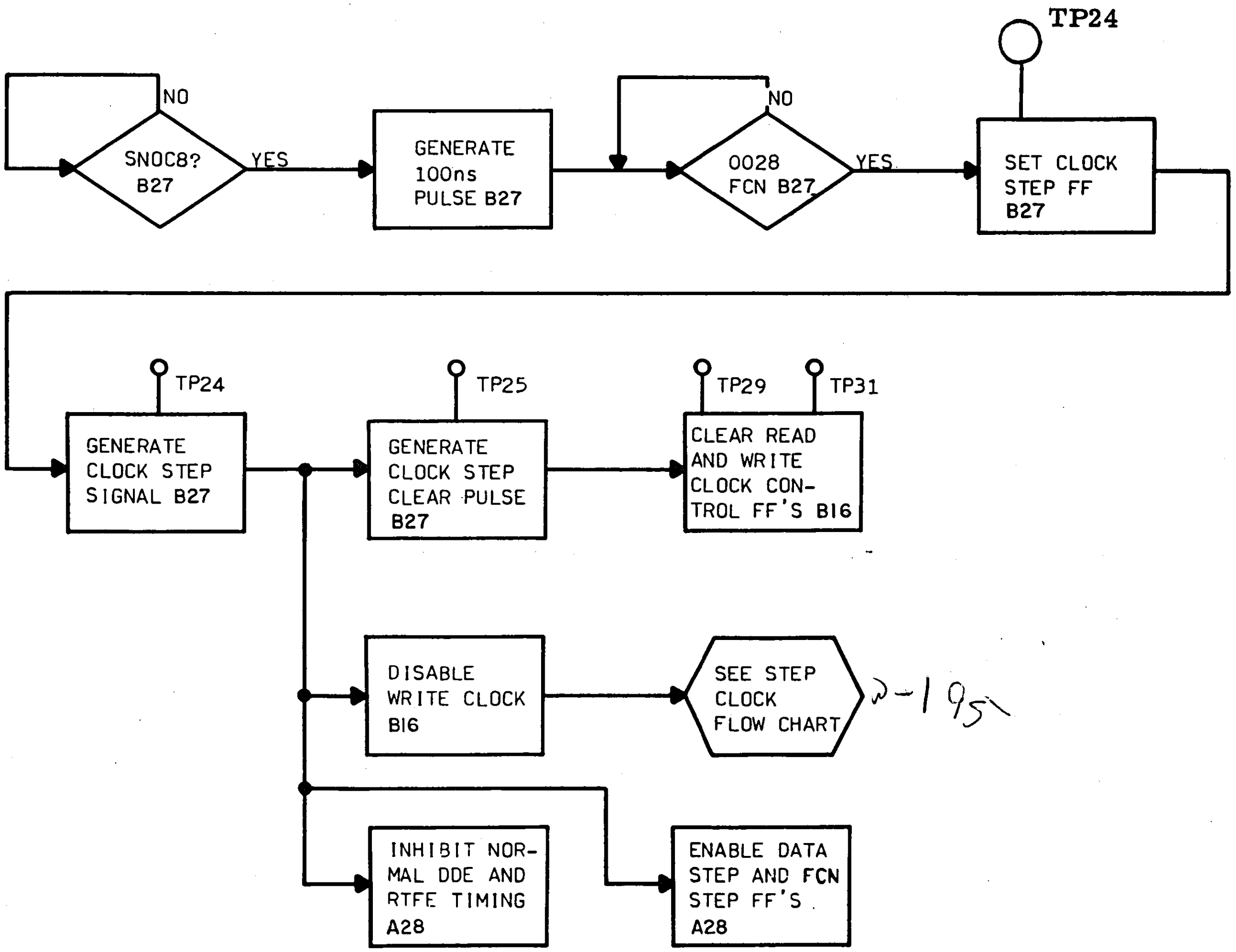

TP24
NO
SNOC8? B27
YES
GENERATE 100ns PULSE B27
NO
0028 FCN B27
YES
SET CLOCK STEP FF B27
TP24
GENERATE CLOCK STEP SIGNAL B27
TP25
GENERATE CLOCK STEP CLEAR PULSE B27
TP29
TP31
CLEAR READ AND WRITE CLOCK CON- TROL FF'S B16
DISABLE WRITE CLOCK B16
SEE STEP CLOCK FLOW CHART
2-195
INHIBIT NOR- MAL DDE AND RTFE TIMING A28
ENABLE DATA STEP AND FCN STEP FF'S A28

Figure 3-102. Enter Clock Step Mode - 0028

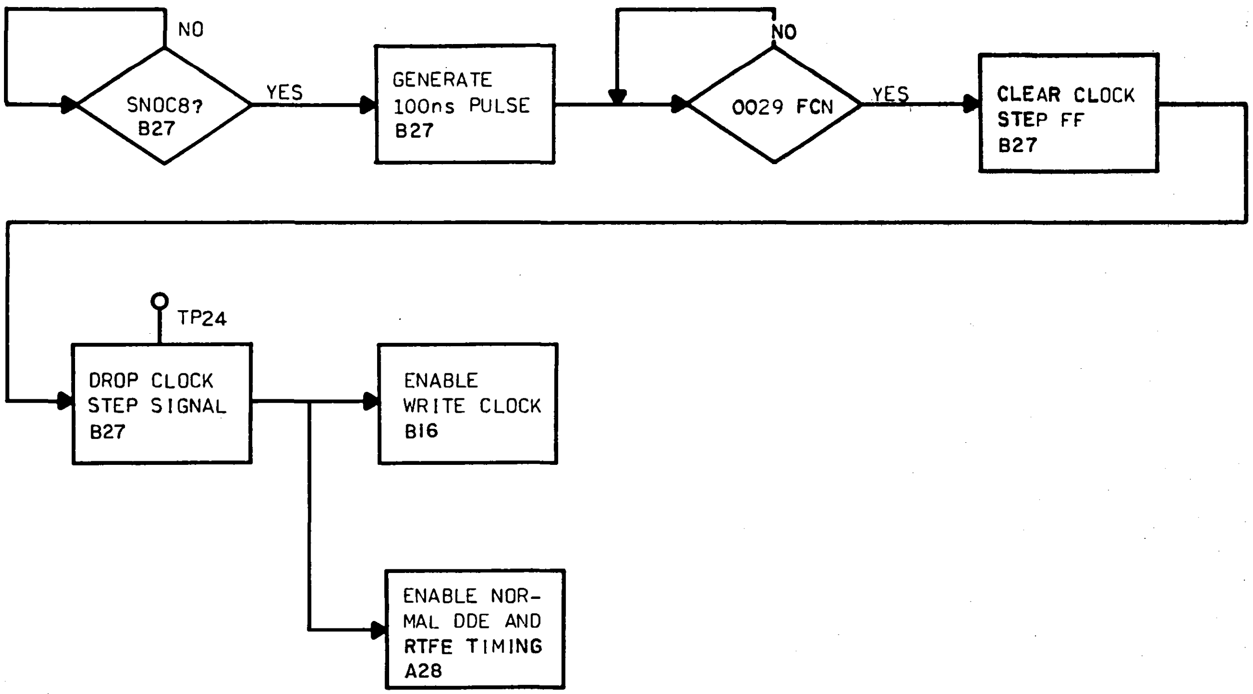


Figure 3-103. Exit Clock Step Mode - 0029

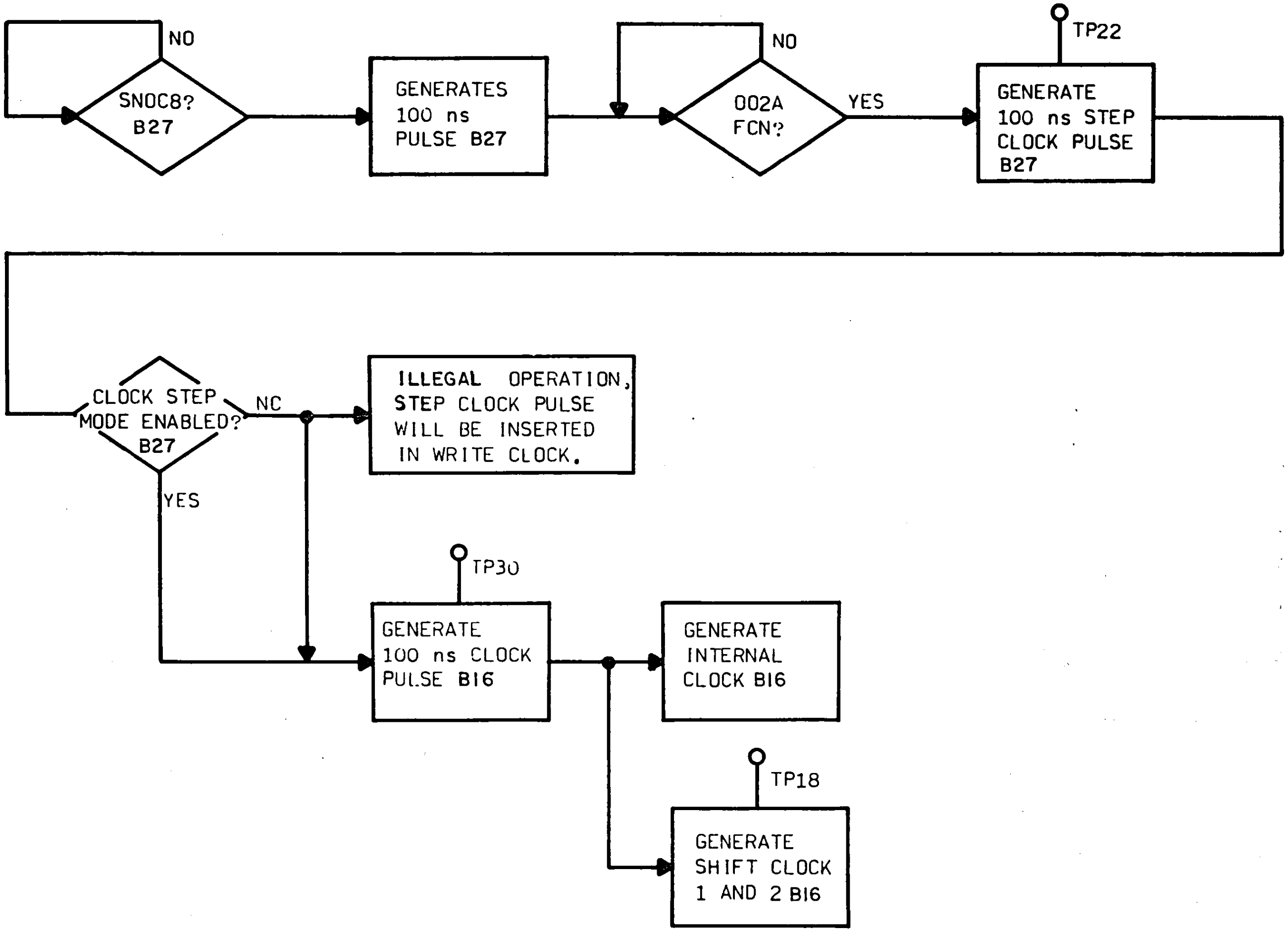


Figure 3-104. Step Clock - 002A

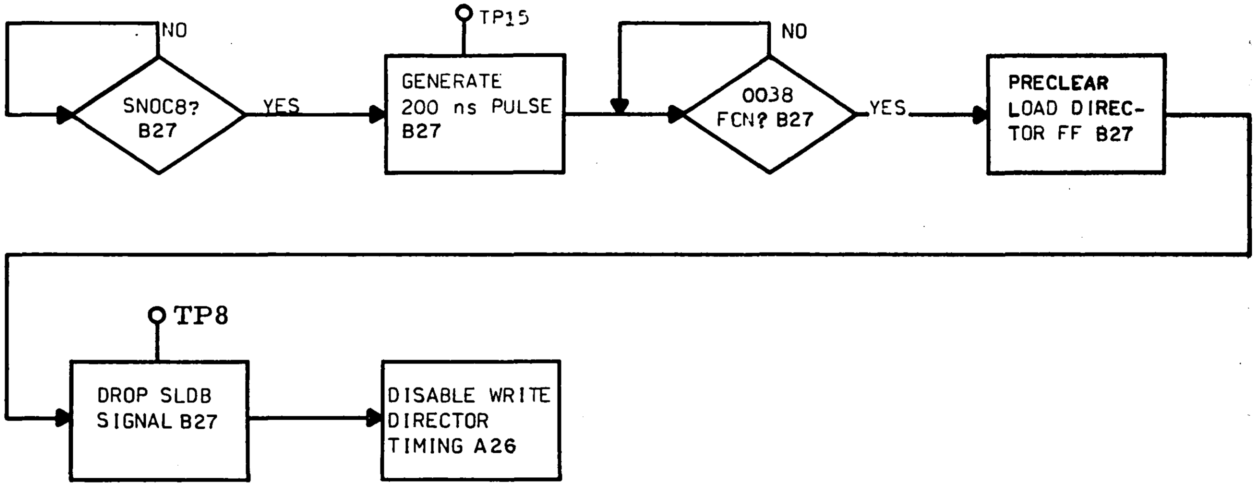

NO
SNOC8? B27
YES
TP15
GENERATE 200 ns PULSE B27
NO
0038 FCN? B27
YES
PRECLEAR LOAD DIRECTOR FF B27
TP8
DROP SLDB SIGNAL B27
DISABLE WRITE DIRECTOR TIMING A26

Figure 3-105. Stop Loading Directors - 0038

Function Code	Definition
0038	Stop loading directors - This code causes the control logic to stop loading the director buffer. A 0018 code is required to resume director loading.

Normal Channel 9 Functions

Normal channel 9 enables two special functions and also preconditions the hardware for a status input. Figures 3-106 through 3-108 contain flowcharts for each of the channel 9 functions. To enable a desired function, the function must be placed on the normal output channel lines. The SNOC9 signal must then be toggled (output and then dropped) to enable the function. Each of the normal channel 9 functions is discussed in detail in the following paragraphs.

Status Selection

Figure 3-106 indicates the general action which occurs when status is selected on normal channel 9. The codes which are output on normal channel 9 are:

Hexadecimal Code	Description
0000	Select normal operating status 1
1000	Select register file bus status
2000	Not used
3000	Select word counter status
4000	Select normal operating status 2
5000	Select data buffer output status
6000	Not used
7000	Select bits 16 through 31 of checkword generator
8000	Select bits 00 through 15 of checkword generator
9000	Select disk status
A000	Select director register bits 00 through 07 and 24 through 31
B000	Select byte counter status

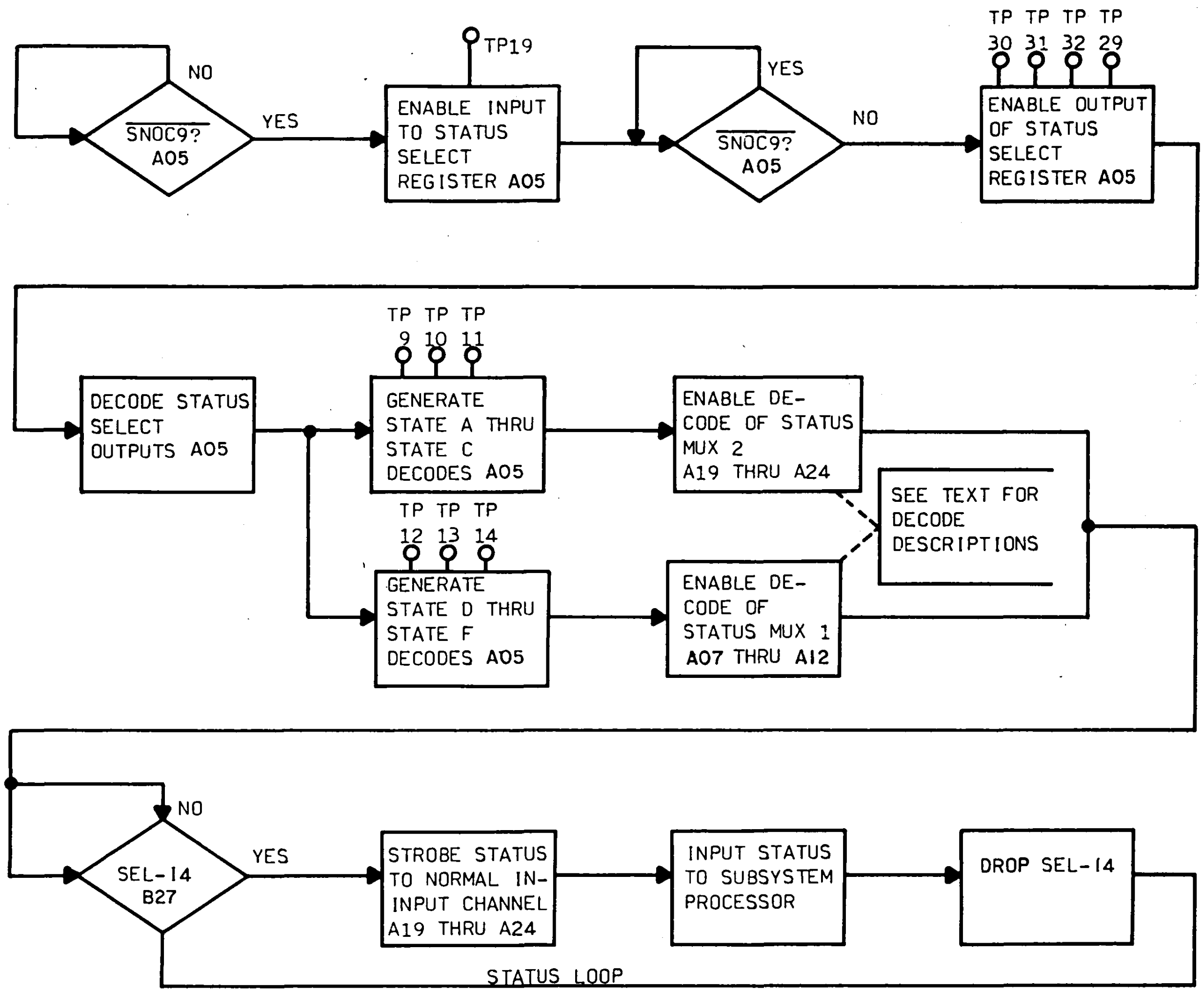

NO
SNOC9?
A05
YES
TP19
ENABLE INPUT TO STATUS SELECT REGISTER A05
YES
SNOC9?
A05
NO
TP TP TP TP
30 31 32 29
ENABLE OUTPUT OF STATUS SELECT REGISTER A05
DECODE STATUS SELECT OUTPUTS A05
TP TP TP
9 10 11
GENERATE STATE A THRU STATE C DECODES A05
ENABLE DE-CODE OF STATUS MUX 2 A19 THRU A24
SEE TEXT FOR DECODE DESCRIPTIONS
TP TP TP
12 13 14
GENERATE STATE D THRU STATE F DECODES A05
ENABLE DE-CODE OF STATUS MUX 1 A07 THRU A12
NO
SEL-14
B27
YES
STROBE STATUS TO NORMAL IN-INPUT CHANNEL A19 THRU A24
INPUT STATUS TO SUBSYSTEM PROCESSOR
DROP SEL-14
STATUS LOOP

Figure 3-106. Status Selection Flowchart

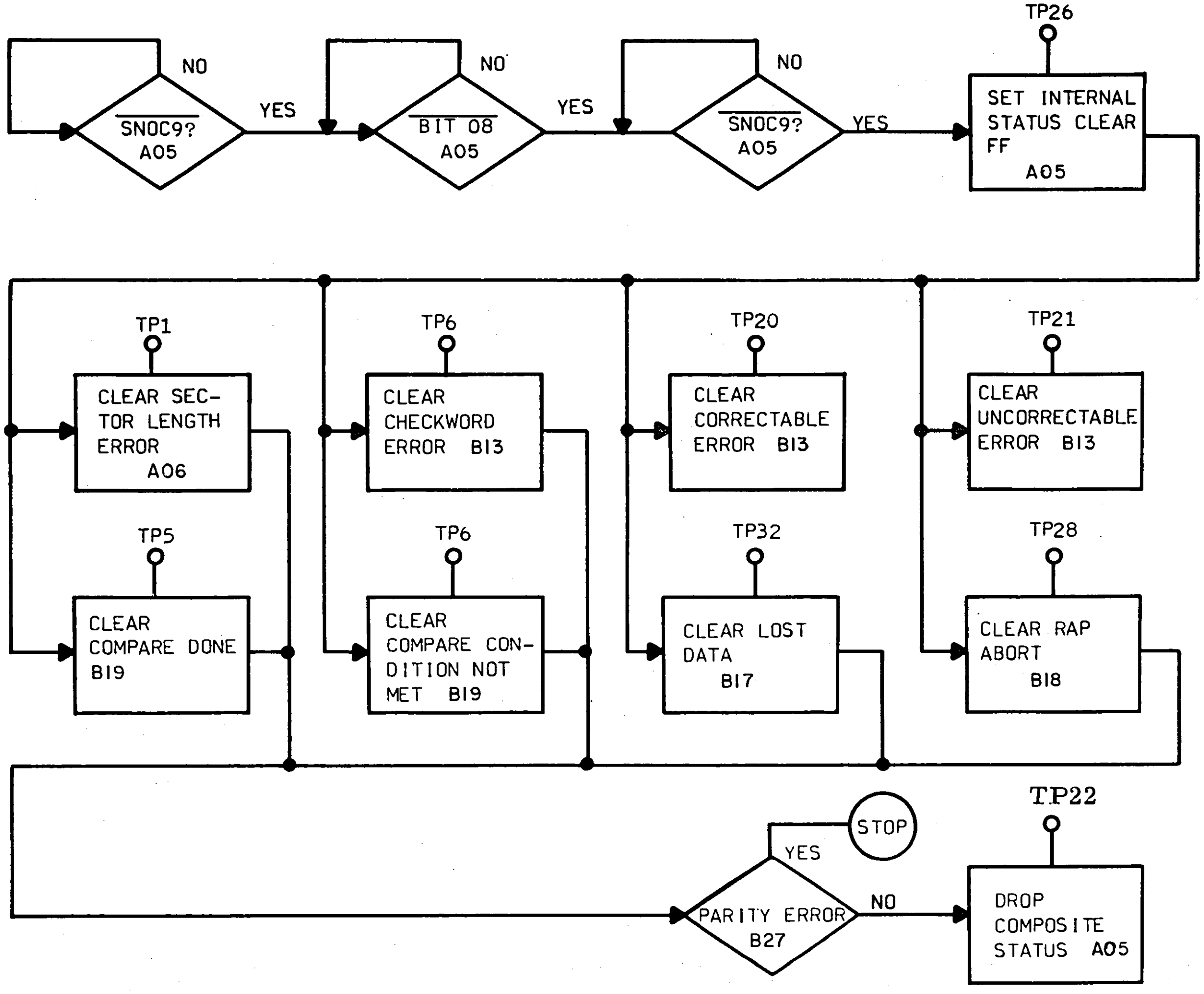

SNOC9?
A05
NO
YES
BIT 08
A05
NO
YES
SNOC9?
A05
NO
YES
TP26
SET INTERNAL STATUS CLEAR FF
A05
TP1
CLEAR SECTOR LENGTH ERROR
A06
TP6
CLEAR CHECKWORD ERROR B13
TP20
CLEAR CORRECTABLE ERROR B13
TP21
CLEAR UNCORRECTABLE ERROR B13
TP5
CLEAR COMPARE DONE
B19
TP6
CLEAR COMPARE CONDITION NOT MET B19
TP32
CLEAR LOST DATA
B17
TP28
CLEAR RAP ABORT
B18
STOP
YES
PARITY ERROR
B27
NO
TP22
DROP COMPOSITE STATUS A05

Figure 3-107. Clear Internal Status Flowchart

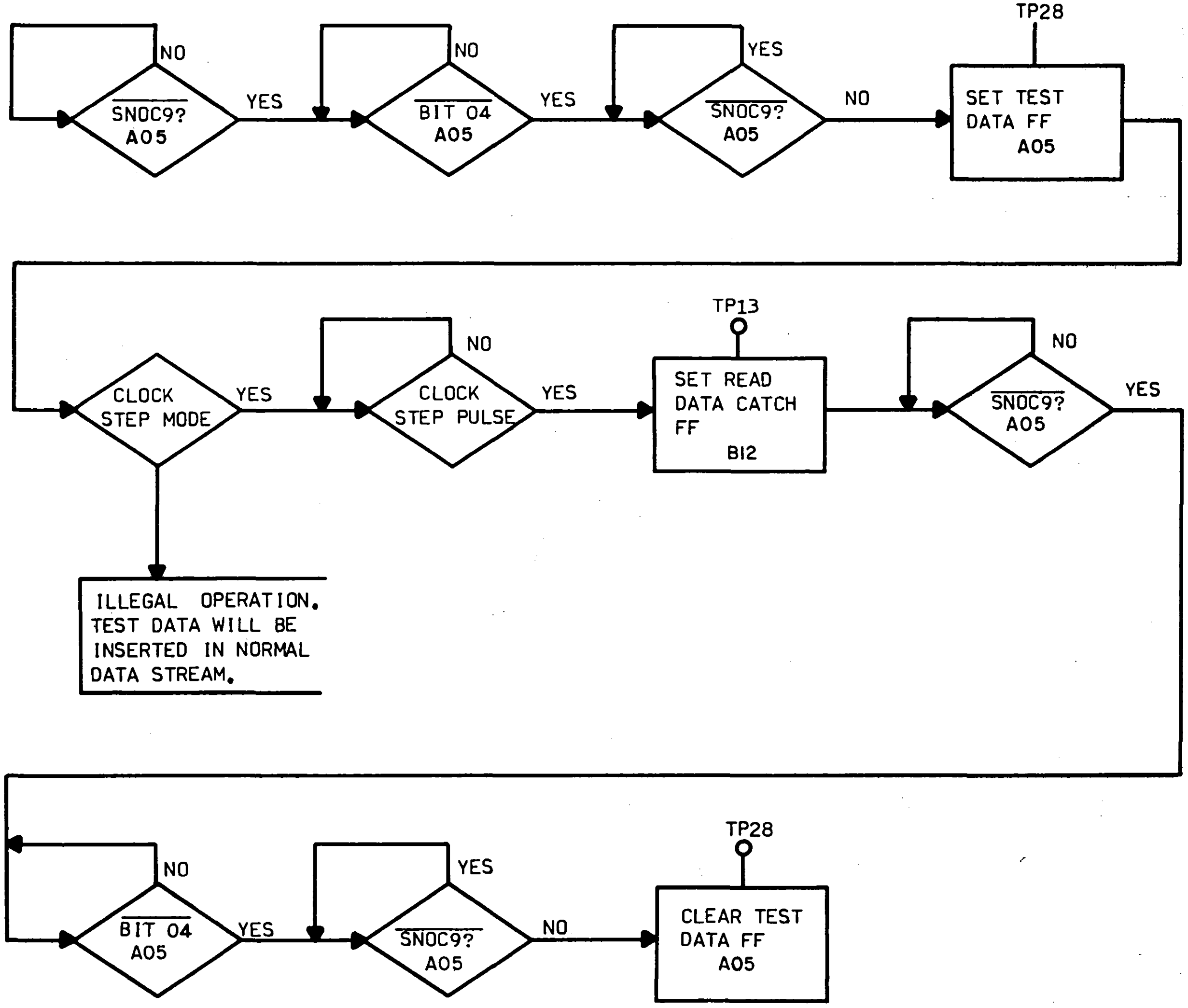


Figure 3-108. Test Data Flowchart

Hexadecimal Code	Description
C000	Select bit counter, director buffer input address register, and director buffer output address register
D000	Select director buffer address register
E000	Not used
F000	Select director bits 08 through 23

The status select codes are partially translated on logic card A07. The partial translations are state A, state B, state C, state D, state E, and state F. States A, B, and C are further decoded on the data path one logic cards (A19 through A24). The codes are structured such that all even status (0000, 2000, 4000, etc.) is routed directly through data path 1 to the normal input channel lines. States D, E, and F are further decoded on data path 2 logic cards (A07 through A12) to enable all odd status (1000, 3000, 5000, etc.) onto the status bus to data path 1. If an odd status is selected, states A, B, and C decodes route the status bus onto the normal input channel lines.

Once the status has been selected, it is input on normal input channel 4. The SEL-I4 signal (enable normal input channel) strobes the status through the multiplexer on data path 1 logic cards and then enables the output of the multiplexer through an exclusive OR chip to the normal input channel lines. True data is presented to the subsystem processor.

Clear Internal Status

When hexadecimal code 0080 is output on normal channel 9, it sets the internal status clear flip-flop. Figure 3-107 shows the resultant action. Note that the parity error flip-flop is not cleared by this function.

Test Data

The test data function (0800) allows simulated data transitions during clock step operations. When bit 04 of normal channel 9 is set and the SNOC9 signal is toggled, it sets the test data flip-flop. The read data catch flip-flop will set when the clock is stepped. If the test data flip-flop is clear, the read data catch flip-flop will clear when the clock is stepped. This function must only be used with clock step operations. If the test data flip-flop remains set during normal clock operations, the test data will be inserted into the data stream from the disk storage unit.

Normal Channel 9 Special Considerations

Although the functions on normal channel 9 are considered as 16-bit hexadecimal codes, they are actually bit-oriented. The status select codes use only bits 00 through 03, the test data code uses only bit 04, and the clear internal status code uses only bit 08. This causes an interaction between these functions. For example, when the test data code (0800) or the clear internal status code (0080) are used, the upper 4 bits are all zero and normal operating status 1 is automatically selected. If a specific status is desired during test data operations, the function codes must be combined. For example, code 5800 will select the data buffer output status during test data operations.

Director Buffer Write Timing

Upon receipt of a 0018 function code (load director buffer), the control logic begins making memory requests to obtain directors. The 0018 function code sets the memory request flip-flop which generates the initial request to the memory scanner. When the scanner has honored the director request and the data is available on the memory bus, the MDRDY signal comes up. This signal is routed through the memory reply timing chain on B26 and generates the DIRSB signal on B27. This signal initiates the director buffer write timing sequence. The sequence of operation is discussed below and the timing relationships are shown on Figure 3-109. The sequence assumes that the director buffer is not full and that no directors are being executed. Either of these conditions will halt director loading until the inhibiting condition is removed.

Since a memory word is 16 bits in length and a director is normally 32 bits in length, two memory accesses are required to obtain one director. The sequence shows the handling of both accesses and the enables required to route the two memory words into the upper and lower director buffer locations. Certain directors require special handling either because they are not 32 bits in length or because they are not loaded into the director buffer. These directors are discussed separately. The normal sequence for 32-bit directors is as follows:

1. DIRSB signal is generated on B27 (TP12).

   A. Gate data from memory bus into director catch register (A19 through A24).

   B. Clear memory request flip-flop (A26, TP1).

   C. Set memory reply flip-flop (A26, TP2).

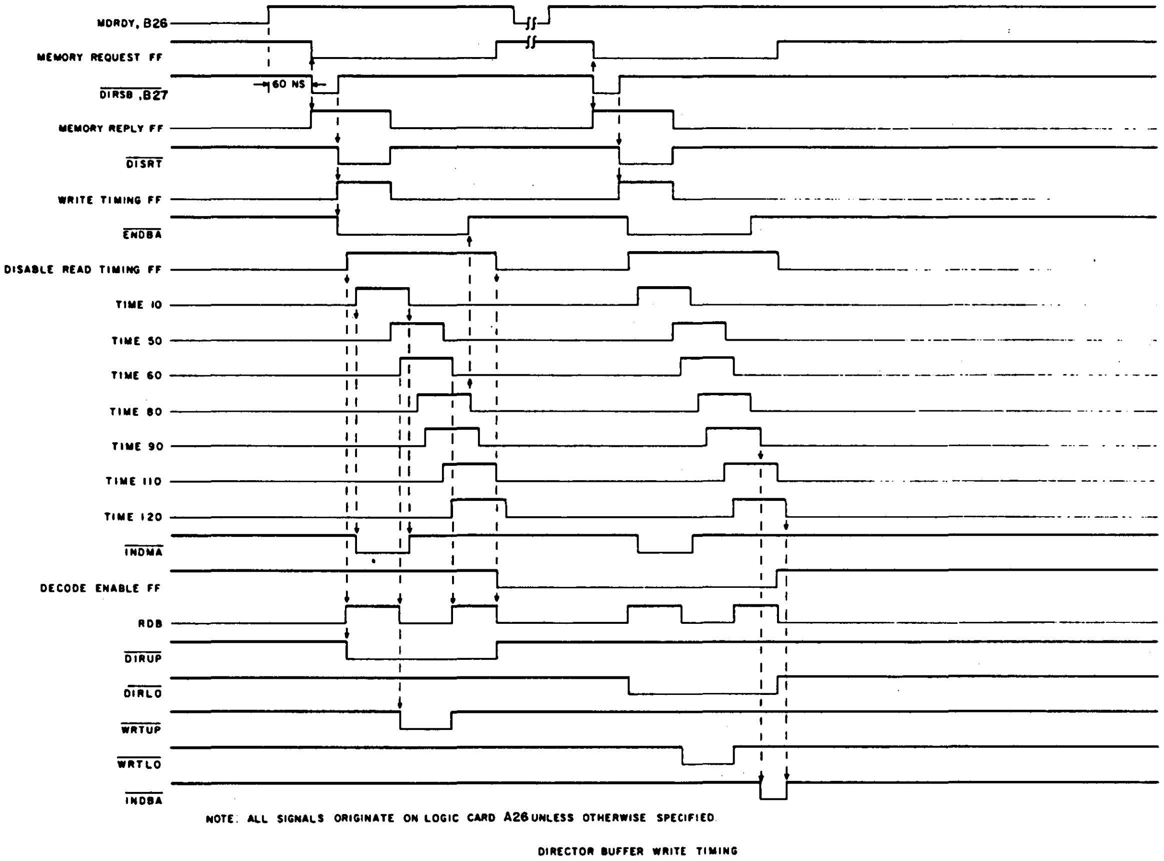


Figure 3-109. Director Buffer Write Timing

II. Memory reply flip-flop and trailing edges of DIRSB generate DISRT and set write timing initiate flip-flop (A26, TP3).

A. Generate ENDBA (A26, TP26). This signal gates the director buffer input address (A25) to the director buffer (A29) via the director buffer address multiplexer (A29).

B. Set disable read timing flip-flop (A26, TP19).

C. Start director write timing chain.

III. Disable read timing flip-flop generates DSRT.

A. Block read director timing chain (A27).

B. Generate RDB signal (A26, TP24). This allows the director bus (A29) to float.

C. Generate DIRUP (A26, TP20). This signal gates the contents of the director catch register to the upper portion of the addressed director buffer location. It also gates the upper 8 bits (00 through 07) to the special director look-ahead logic on A25.

IV. Generate write time 10 (A26).

A. Generate INDMA (A26, TP18). This signal increments the director memory address (A03, A04).

V. Generate write time 50 (A26).

A. Maintain ENDBA signal.

B. Clear memory reply flip-flop.

C. Drop DISRT signal.

D. Clear write timing initiate signal.

E. Clock special director flip-flops (A26, TP5, 7, 8, and 11). One of these flip-flops will set if a special director is decoded. These directors cause a special timing sequence which is discussed under the particular director.

VI. Generate write time 60 (A26).

A. Generate WRTUP (A26).

B. Drop RDB. Steps A and B write the memory data into the upper director buffer.

VII. Generate write time 80 (A26).

A. Maintain ENDBA signal.

VIII. Generate write time 110 (A26).

A. Clock the disable read timing flip-flop.

IX. Generate write time 120 (A26).

A. Enable INDBA (A26, TP15).

B. Enable memory request flip-flop.

X. Write time 90 drops.

XI. Write time 110 drops.

A. Set the memory request flip-flop (A26, TP1). This generates another director request, DIRRQ1.

B. Clear the decode enable flip-flop (A26, TP9). This inhibits decoding the lower part of the director as a special director.

C. Drop RDB and DIRUP.

XII. Memory scanner honors director request. Generates DIRSB on B27.

A. Gate data from memory bus into director catch register (A19 through A24).

B. Clear memory request flip-flop (A26).

C. Set memory reply flip-flop (A26).

XIII. Generate DISRT and set write timing initiate flip-flop (A26).

A. Generate ENDBA (A26).

B. Set disable read timing flip-flop (A26).

C. Start director write timing chain (A26).

XIV. Generate DSRT (A26).

A. Block read director timing chain (A27).

B. Generate RDB.

C. Generate DIRLO (A26, TP21). This signal gates the contents of the director catch register to the lower portion of the addressed director buffer location.

XV. Generate write time 10 (A26).

A. Generate INDMA (A26).

XVI. Generate write time 50 (A26).

A. Maintain ENDBA (A26).

B. Clear memory reply flip-flop (A26).

C. Drop DISRT (A26).

D. Clear write timing initiate flip-flop (A26).

XVII. Generate write time 60 (A26).

A. Generate WRTLO (A26).

B. Drop RDB. Steps A and B write the memory data into the lower director buffer.

XVIII. Generate write time 80 (A26).

A. Maintain ENDBA (A26).

XIX. Generate write time 110 (A26).

A. Clock the disable read timing flip-flop (A26).

XX. Generate write time 120 (A26).

A. Enable INDBA (A26).

B. Enable memory request flip-flop (A26).

XXI. Write time 90 drops (A26).

A. Generate INDBA (A26, TP15). This signal increments the director buffer input address on A25.

B. Clear the empty flip-flop (A25, TP20).

XXII. Write time 110 drops (A26).

A. Set the memory request flip-flop. Generate DIRRQ1 (A26, TP1).

B. Set the decode enable flip-flop (A26, TP9). This enables the decode for the upper part of the next director.

C. Drop RDB and DIRUP (A26).

Director Buffer Write Timing (Special Directors)

Four directors require special handling when they are obtained from memory. These directors are detected in the look-ahead decode logic on A25. If the decode is in the first 16 bits of a director, the decode enable flip-flop on A26 will be set, and at write time 50, the applicable special director flip-flop on A26 will set. The discussion which follows shows the deviations from normal write timing for each of these special directors.

Unconditional Branch Director Loading 94

The unconditional branch director causes an immediate branch to a new memory address location. The director is not loaded into the director buffer. Deviations from normal timing are:

I. Generate DIRUP (A26, TP20).

   A. Gates memory data to director bus and to look-ahead decode logic.

II. Generate UNCBRH (A25, TP6) and UNCBH (A25).

   A. Enable set input of unconditional branch flip-flop (A26).

III. Generate write time 50 (A26).

   A. Set unconditional branch flip-flop (A26, TP7). This flip-flop disables the INDBA gate and the WRTLO gate.

IV. Generate write time 60 (A26).

   A. Generate WRTUP. Director bits 00 through 15 are written into the director buffer.

NOTE

Although these bits are written into the director buffer, the director buffer input address will not be advanced. The next director will be written over these bits.

V. Generate write time 120 and 110 (A26).

   A. Generate DIRRQ1 to obtain second 16 bits of director.

   B. Clear decode enable FF to inhibit decode of second 16 bits. This prevents branch address from being decoded as a special director and setting a special director flip-flop.

VI. Generate DIRLO. This signal gates the second 16 bits onto the director bus.

VII. Generate write time 110 and 80.

A. Generate LDDMA (**A26**, TP13). This signal gates the branch address (bits 16 through 31) into the director memory address register on **A03** and **A04**. Next director reference will be made from this memory address.

VIII. WRTLO and INDBA were both inhibited by unconditional branch flip-flop so that lower bits were not placed in director buffer and director buffer input address was not incremented.

IX. If the next director is not another unconditional branch director, the UNCBH signal will remain high. At write time 10, the unconditional branch flip-flop will clear.

Address Director Loading

The address director is 64 bits in length. Special handling is required to prevent the second, third, and fourth 16-bit bytes from being decoded as special directors. Deviations from normal timing are:

1. Generate DIRUP (**A26**, TP20).

   a. Gates memory data to director bus and to look-ahead decode logic.

2. Generate ADDRESS (**A25**, TP3).

3. Generate write time 50 (**A26**).

   a. Set address 1 flip-flop (**A26**, TP11). The clear output of this flip-flop inhibits the special director clock.

4. First byte of director is loaded normally.

5. Second byte of director is loaded normally.

6. Generate write time 10 during loading of third byte.

   A. Set address II flip-flop (**A26**, TP12). This flip-flop also inhibits the special director clock.

7. Generate write time 70 during third byte.

   a. Clear address I flip-flop.

8. Third byte is loaded normally.

9. Fourth byte is loaded normally.

10. Generate time 110 during fourth byte.
    a. Clear address II flip-flop.
11. Continue normal director loading.

Stop Loading Director Loading

This director causes the control logic to stop loading directors. It is a 16-bit director buffer. Deviations from normal timing are:

1. Generate DIRUP (A26, TP20).
    a. Gates memory data to director bus and to look-ahead decode logic.
2. Generate SLDDIR (A25, TP5).
3. Generate write time 50 (A26).
    a. Set stop loading flip-flop (A26, TP5). This flip-flop disables the INDBA gate. The director buffer input address will not be incremented.
4. Director loading continues normally. The first byte is written into the director buffer, but will be destroyed when director loading resumes.
5. Generate write time 120.
    a. Inhibit setting of memory request flip-flop.
6. An 0018 function ~~is~~ is required to resume director loading. This function will generate an LDB signal to preset both the memory request flip-flop and the decode enable flip-flop.

Terminate Director Loading

This is a 16-bit director. Special handling is required to increment the director buffer input address after one memory access and to leave the decode enable flip-flop set for the next director. Deviations from normal are:

1. Generate DIRUP (A26, TP20).
    a. Gates memory data to director bus and to look-ahead decode logic.
2. Generate TERM (A25, TP4).
3. Generate write time 50 (A26).
    a. Set terminate flip-flop (A26, TP8). This flip-flop enables the INDBA gate and disables the clock to the decode enable flip-flop.

4. Director loading continues normally except that director buffer input address is incremented.

5. The memory request for the next memory word is inhibited from clearing the decode enable flip-flop.

6. Write time 10 of the next write timing chain clears the terminate flip-flop. Since this occurs after the memory request flip-flop has cleared, the decode enable flip-flop does not toggle. A master clear will also clear the terminate flip-flop.

Director Buffer Read Timing

The director buffer read timing chain is initiated by a 0022 function (execute directors). Once initiated, directors will continue to be read from the director buffer as needed until the director buffer becomes empty. The read timing chain can be temporarily interrupted by the write director timing chain since the read and write processes share the director bus. When the 0022 function is received, the control logic generates an ENDEX signal which enables the read timing gate. It also generates an STEXC pulse which initiates the first read timing chain. Figure 3-110 illustrates the normal read timing. Additional information is presented below.

I. Generate STEXC (B27).

   A. Preset support director request flip-flop (A27, TP2).

   B. Trailing edge of STEXC sets read timing initiate flip-flop (A27, TP1). This assumes the buffer is not empty and no write timing.

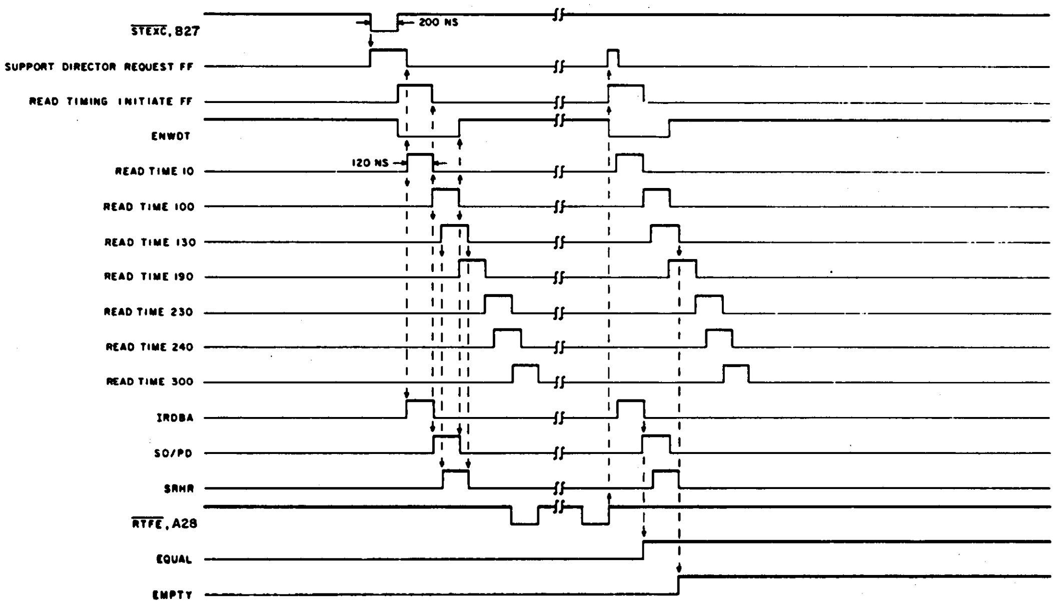
II. Drop ENWDT to disable write timing chain (A27, TP19).

III. Start read timing chain. Set inhibit write timing flip-flop.

IV. Generate read time 10 (A27).

   A. Generate IRDBA (A27, TP12). This signal increments the director buffer output address on A25.

   B. Preclear support director request flip-flop (A27, TP2) and data director request flip-flop (A27, TP3). This disables the read enable gate.



DIRECTOR READ TIMING

NOTE. ALL SIGNALS ORIGINATE ON A27UNLESS OTHERWISE SPECIFIED.

Figure 3-110. Director Buffer Read Timing

V. Generate read time 100 (A27).

   A. Preclear read timing initiate flip-flop.

   B. Generate SO/PD pulse (A27, TP23). This pulse strobes the contents of the director bus into the director output register (A25, A27, and A28).

   C. Clock inhibit write timing flip-flop.

   D. Decode director function (A25, B23).

VI. Generate read time 130 (A27).

   A. Clock director buffer empty flip-flop. If input and output pointers are now equal, the flip-flop will set.

   B. Generate SRHR pulse (A27, TP13). This pulse strobes the director buffer output address to the output address holding register (A25). is the read address for the next director.

VII. Generate read time 190 (A27). This signal clocks the support director ready flip-flop on A28. If a support director has been decoded, the flip-flop will set (A28, TP20). It will also preset the address director flip-flop (A27, TP22) if an address director has been decoded. These functions are discussed in detail under specific director execution.

VIII. Generate read time 230 (A27). This pulse is discussed under support director control.

IX. Generate read time 240 (A27). This signal generates a DDE pulse if a data director has been decoded. The DDE pulse (A28, TP8) initiates execution of the data director.

X. Generate read time 300 (A27). This signal generates an RTFE pulse (A28, TP24) if a support director has been decoded. This pulse initiates the execution of support directors.

XI. Generate RTFE or DDE.

A. RTFE will set support director request flip-flop (A27).

B. DDE will set the data director request flip-flop (A27). Either A or B above will initiate another read timing chain.

Director Control Logic

The director control logic determines the type of director which is present in the director output register and then generates the signal which causes execution of that director. It contains separate logic for control of data directors and support directors so that these directors may be executed in parallel. The logic is initially enabled by a 0022 function code over normal channel 08. When the function code is decoded, it generates STEXC. This is the start execution pulse which enables the director read timing chain. The pulse enables the director control logic for handling the first director. Subsequent directors are then handled as they are decoded. Execution of a terminate director will disable the director control logic and another 0022 function code is required to resume operation. The director control logic operates as follows:

I. 0022 function code generates STEXC (B27, TP11).

A. Preset data control flip-flop (A28, TP 9). This flip-flop enables the DDE gate.

B. Preset support control flip-flop (A28, TP18). This flip-flop enables the RTFE gate.

II. Generate read time 100.

A. Generate SO/PD (A27, TP23). This signal transfers the director from the director buffer to the director output register. The director output register feeds the decode logic on A25.

1. Decode logic generates FUNDIR (A25, TP18). Go to steps III and IV.

2. Decode logic generates DATDIR (A25, TP23). Go to steps V and VI.

III. If FUNDIR is decoded:

A. Set support director ready flip-flop (A28, TP20) at read time 190.

B. At read time 300, generate RTFE (A28,TP24). This signal causes execution of the support director, clears the support control flip-flop, clears the support director ready flip-flop, and starts the read chain for the next director.

IV. Next director is decoded at read time 100.

A. Generate DATDIR (A25,TP23). Go to steps V and VI.

B. Generate FUNDIR (A25,TP18).

1. At read time 190, set support director ready flip-flop.

a. If RTFD is generated during read time 230, set support control flip-flop and generate RTFE at read time 300.

b. If RTFD is generated after read time 230, generate RTFE immediately.

C. If RTFD is generated before the next FUNDIR director is decoded, set the support control flip-flop and generate CLRF. The next FUNDIR signal will repeat steps III and IV.

V. If DATDIR is decoded:

A. Generate DDE at read time 240. This signal causes execution of the data director, clears the data control flip-flop, and starts the read chain for the next director.

VI. Next director is decoded at read time 100.

A. Generate FUNDIR (A25,TP18). Go to steps III and IV.

B. Generate DATDIR (A25,TP23).

1. Upon receipt of DD (B19) indicating previous data director is complete, generate DDE and DTON. These signals cause execution of the decoded director and start the read chain for the next director.

C. If DD is generated before the next data director is decoded, set the data control flip-flop. The next DATDIR signal will repeat steps V and VI.

Load Register File From Memory Director Execution

00 XX 5-19

This director transfers the contents of the memory location specified by director bits 16 through 31 into the register file location specified by director bits 12 through 15. The sequence of operation is:

I. Read timing chain generates SO/PD (A27, TP23).

   A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and director decode logic on B23.

II. Director is decoded on A25 and B23.

   A. Generate FUNDIR (A25, TP18). This signal causes the support director control logic to generate RTFE.

   B. Generate LIMDE and FLDIA (B23).

III. Support director control logic generates RTFE (A28, TP24).

   A. LIMDE enables .director bits 12 through 15 into register file address mux 1 (B24).

   B. RTFE and LIMDE generate LDAD2 (B24) which generates LDADR (A06) which then loads director bits 16 through 31 into the data address register on A03 and A04.

   C. RTFE and LIMDE set the register file input request flip-flop (B24). a data transfer to memory is not in progress, the request flip-flop generates MBRFB which enables the memory bus onto the register file bus. The request flip-flop also generates RFIPRQ which generates a memory request through the memory scanner (B27, B26).

IV. RFIPRQ sets input request flip-flop (B27) and stops memory scanner at the data gate (B26).

   A. Generate SCAND (B26). This enables the data address through the memory address mux 1 (A03, A04).

   B. Generate MREQ (B26). This is the request to memory.

V. Memory returns MDRDY and starts reply timing chain on **B26.**

A. Generate REPLY **(B26).** This generates MI/PRY on **B27** which drops DRQT on **B28** to unblock the scanner, and sets the request-honored flip-flop on **B24.**

B. Generate RFME, RFWEU, and RFWEL on **B24.** These signals write the memory data into the register file.

C. Clear the input request flip-flop **(B24).**

VI. Generate ICL2 **B16.**

A. Set the memory operation done flip-flop**(B24).**

B. Clear the request honored flip-flop **(B24).**

C. Generate IDDC **(B24).** This signal enables the RTFD flip-flops on **B23.**

VII. Generate ICL2**(B16).**

A. Clear memory operation done flip-flop (B32).

B. Set the RTFD flip-flops on **B23.** This causes execution of the next director.

VIII. Generate ICL1 **(B16).**

A. Clear the RTFD flip-flops.

Address Director Execution

This is a 64-bit director which establishes the path for a data transfer operation. It also determines the direction of data flow, the data port to be used, and whether or not buffering will be required. The field entries for the director are:

- Bits 00 through 07 contain the function code.
- Bits 08 through 23 contain the length of the data transfer in 16-bit words.
- Bit 24 specifies the data port. A 0 designates the coupler, and a 1 designates core memory.

- Bit 26 specifies data flow direction. A 0 designates data flow from the data buffer to the shift register (write or compare operation), and a 1 designates data flow from the shift register to the data buffer (read operations).
- Bit 27, when set, specifies that no end of record pulse will be sent or is expected. When clear, this bit specifies that an EOR must be sent or received after the data transfer. This bit applies to coupler transfers only and has no effect on transfers to or from memory.
- Bits 32 through 63 contain the starting address of the data operation.

The sequence of operation is illustrated in Figure 3-111 and described below.

I. Read director timing time 100 generates SO/PD signal (A27, TP23).

A. Gate the contents of the addressed director buffer memory location (A29) to the director register (A25, A27, A28). This is the first 32 bits of the address director.

B. The director register (A25, A27, A28) and director register fanout (A17) feed the director translators (A25, B23).

1. Generate ADDIR (A25, TP2).

2. Generate FUNDIR (A25, TP18).

II. Read director timing time 190 occurs (A27).

A. Set the ADR flip-flop (A27, TP22).

1. Block the RTFE start read director timing path (A27).

III. Director execution sequence generates RTFE (A28, TP24).

A. Clock the address director control flip-flop IA (A05, TP25). It will set.

B. Gate director bits 8 through 23 (I/O length) into director bit holding register (A01, A02).

IV. When next internal clock occurs with address director control complete, set address director control flip-flop IB (A05, TP23).

A. Generate ADRC signal (A05).

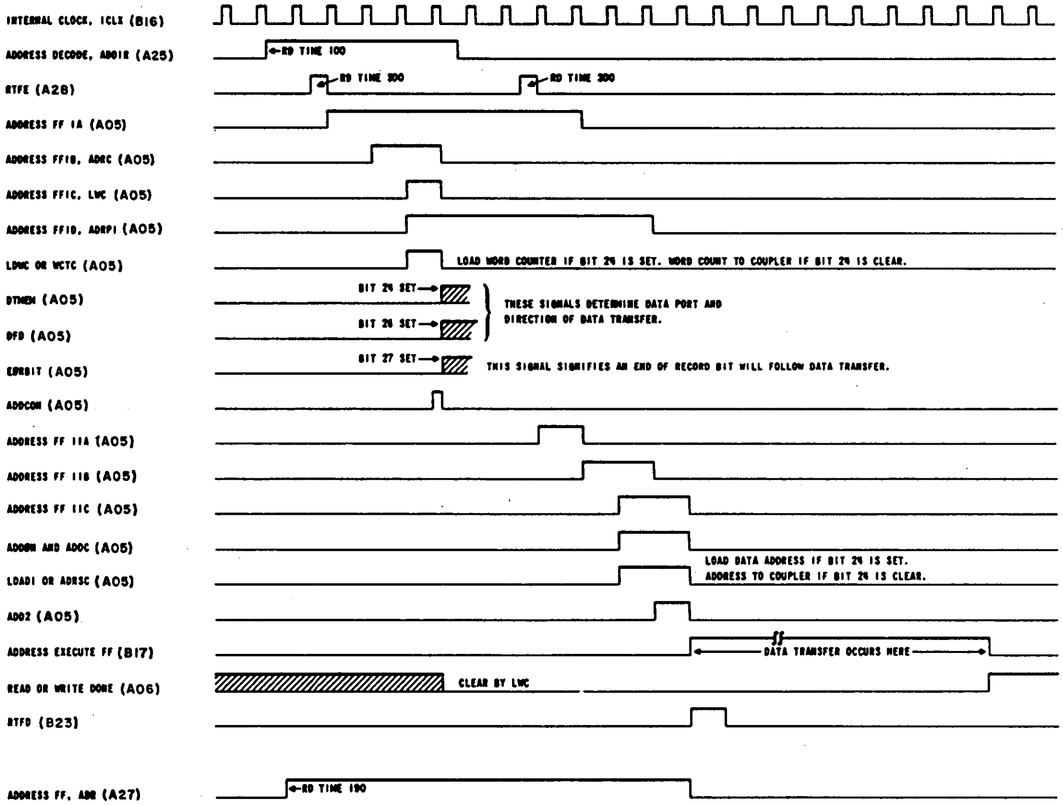


Figure 3-111. Address Director Execution

V. When conditions are not BCCHAN and not CBF and $\overline{\text{DIRB24}}$ or word count 0 and DIRB24, and an internal clock occurs, set address director control flip-flop IC (A05, TP21) and flip-flop ID (A05).

   A. If DIRB24 is present, generate LDWC signal (A05).

      1. Load I/O length into the word counter (B20).

   B. Generate ADRPI signal (A05).

      1. Block FUNDIR and DATDIR translations (A25).

      2. Clear support director ready flip-flop (A28, TP20).

      3. Enable address director RTFE signal (A28).

   C. Generate LWC signal (A05).

      1. Gate director bits 24, 26, and 27 into holding flip-flops (A05). Enable signals DTMEM, DFD, and EORBIT which will affect this and coming directors.

         a. If DFD and DTMEM signals (A05) are 0, generate DMUXE signal (B19). This is a write from the coupler.

            Gate coupler data, CIBXX signals, through the data buffer mux (A07 through A12) to the data buffer (B10).

         b. If DFD signal (A05) is a 0 and DTMEM signal (A05) is a 1, gate the output of the memory data bus (A19 through A24), MBXX signals, through the data buffer mux (A07 through A12) to the data buffer (B10). This is a write from memory.

         c. If DFD signal (A05) is a 1 and DTMEM signal (A07) is a 0, gate the output from the shift register catching register (B09) CRXX signals, through the data buffer mux (A07 through A12) to the data buffer (B10). This is a read to the coupler.

      2. Clear the WDONE (A06, TP32), RDONE (A06, TP8), and EORIN (A06, TP29) flip-flops.

VI. When next internal clock occurs, generate ADDCON signal (A05).

   A. Set support request flip-flop (A27, TP2) to request remaining 32 bits from the director buffer.

   B. Clear address director control flip-flops IB and IC.

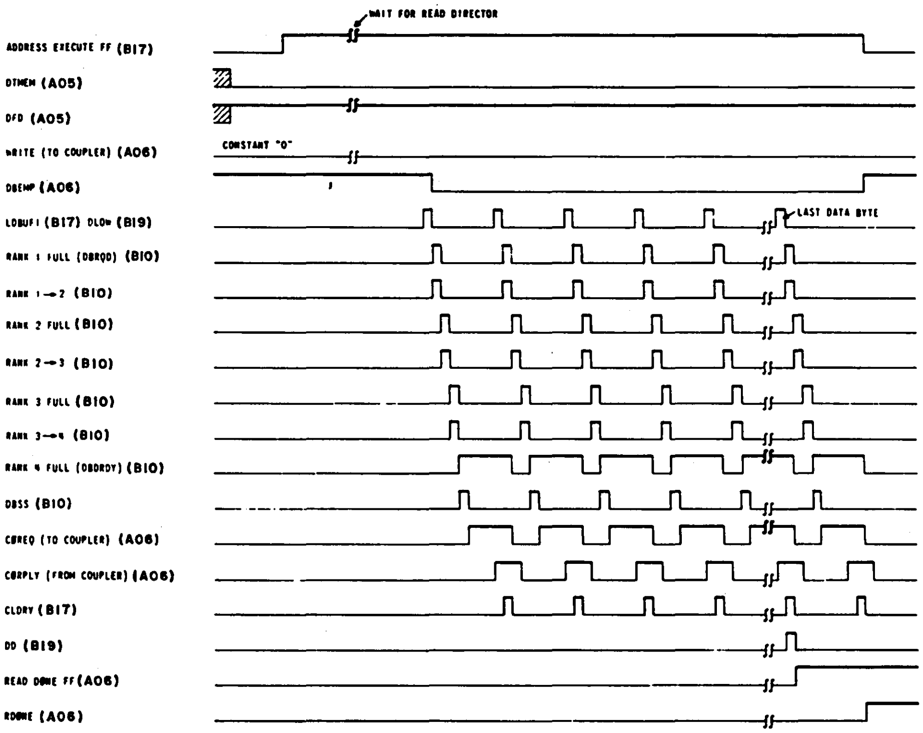
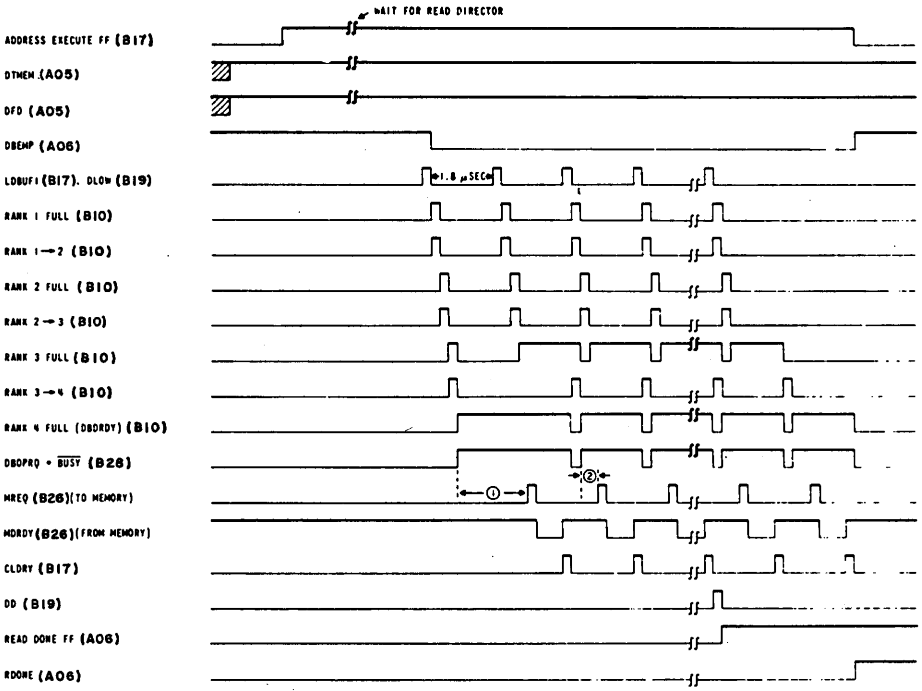


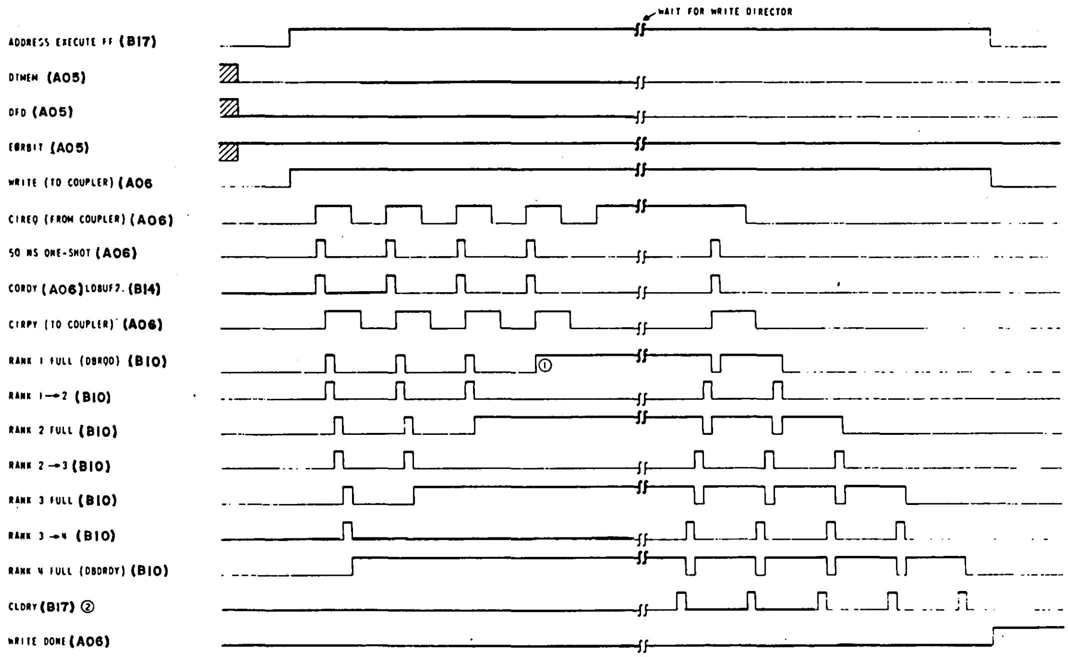
Figure 3-112. Data Transfer - Data Buffer to ~~Memory~~ COUPLER



NOTES: ① 2 μSEC MAXIMUM TO OBTAIN MEMORY SCANNER

② ABOUT 400 NS ONCE SCANNER HAS HONORED FIRST DATA REQUEST.

Figure 3-113. Data Transfer - Data Buffer to ~~Coupler~~ MEMORY



NOTES: ① DATA BUFFER IS FULL AND WAITING FOR A WRITE DIRECTOR.

② THIS PULSE IS GENERATED WHEN LOWER 8-BIT BYTE IS TRANSFERRED TO SHIFT REGISTER DURING EXECUTION OF WRITE DIRECTOR.

Figure 3-114. Data Transfer - Coupler to Data Buffer

VII. Read director timing time 100 generates SO/PD signal (A27, TP32).

A. Gate the contents of the addressed director buffer memory location (B15) to the director register (A25, A27, A28). This is the address portion of the address director.

VIII. Read director timing time 300 generates address director RTFE (A28, TP24).

A. Clock address director control flip-flop IIA (A05, TP16). It will set.

IX. When the next internal clock occurs, set address director control flip-flop IIB (A05, TP18).

A. Clear address director control flip-flops IIA and IA.

X. When the next internal clock occurs, set address director control flip-flop IIC (A05, TP20).

A. If DIRB24 holding flip-flop (A05, TP1) is set, generate LDAD signal (A05).

1. Generate LDADR signal (A06).

a. Gate director bits 16 through 31 into the data address counter (A03, A04).

B. Generate ADDON signal (A05, TP20).

1. Clock the address execute flip-flop (B17, TP5). It will set, generating ADEX signal (B17).

a. If the following operation is to be a read from the selected drive, DFD signal (A05) is a 1. Enable the transfer signals (B17) and wait for the coming read director to start the data transfer. Figures 3-112 and 3-113 illustrate the read data transfers to memory and the data buffer.

b. If the following operation is to be a write or compare operation (DFD signal (A05) is a 0) and the data transfer is to be from the coupler (DTMEM signal (A05) is a 0), the transfer from the coupler occurs as follows (refer to Figure 3-114):

Generate WRITE signal (A06, TP12). This signal goes to coupler as request for the first byte of data.

Wait for the coupler to return CIREQ signal (A06) and byte of data (A07 through A12).

Data goes through mux to data buffer (B10).

CIREQ generates CORDY pulse (A06, TP19) if EORIN flip-flop (A06, TP29) is clear and DBRQD (B10, TP23) is a 0.

Clock the CIRPLY flip-flop (A06, TP6). It will set, sending CIRPLY signal to the coupler requesting the next byte of data.

Bring LDBUF2 signal (B14, TP27) up to a one. Gate the byte of data into the data buffer (B10).

Return to the second step under b. above, and wait for next CIREQ from the coupler.

The above sequence will continue until the data buffer (B10) gets full, DBRQD signal (B10) is a constant 1, or the coupler sends EORIN (A06) and sets the EORIN flip-flop (A06, TP19).

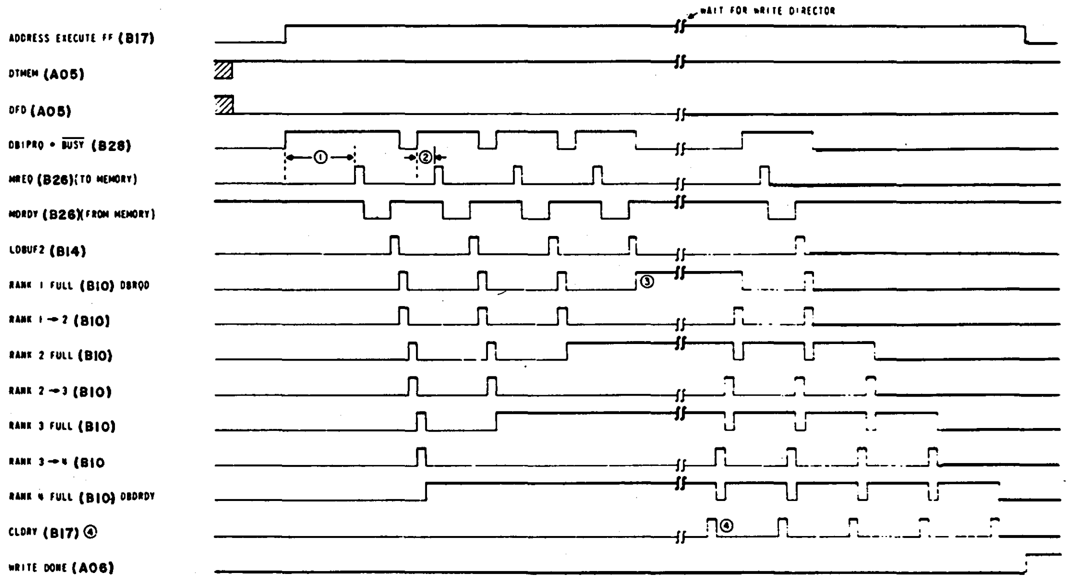
If the EORIN flip-flop has set, the data buffer is empty [signal DBEMP (A06, TP20) is 1], and zero fill is selected (signal ZFILL B12 is a 1), generate ZROFL signal (A06) which blocks lost data (B17), and DBCLR signal (A06) which puts a clear on data buffer rank 4 (B10). This allows zero filling the remaining portion of any data field.

c. If the following operation is to be a write or compare operation [DFD (A05) is a 0] and the data transfer is to be from memory [DTMEM (A05) is a 1], the transfer from memory will occur as follows (refer to Figure 3-115):

If data buffer rank 1 is empty [DBRQD (B10) is a 0, making signal BRQ (B28) a 0], and the word counter (B20) is not zero [signal WTO (B20, TP6) is a 0], generate DTMIP (B17) and DBIPRQ (B17, TP4) signals.

DTMIP blocks all register file memory requests (B24).

DBIPRQ enables the MRLY1 signal (B14, TP26) and the input from memory flip-flop (B27, TP31), and requests the scanner (B26).



NOTES: ① 2 μSEC MAXIMUM TO OBTAIN MEMORY SCANNER.

② ABOUT 400 NS ONCE SCANNER HAS HONORED DATA REQUEST.

③ DATA BUFFER IS FULL AND WAITING FOR A WRITE DIRECTOR.

④ THIS PULSE IS GENERATED WHEN LOWER 8-BIT BYTE IS TRANSFERRED TO SHIFT REGISTER DURING EXECUTION OF WRITE DIRECTOR.

Figure 3-115. Data Transfer - Memory to Data Buffer

When the scanner reaches the correct position, generate ENDRQ signal (B26, TP29) which sets the DRQT flip-flop (B28, TP19) generating DRQT (B28).

DRQT blocks the scanner (B26) at the data position.

DRQT enables the memory request logic (B26).

When the scanner (B26) locks up on data position, generate DATARQ (B26) and SCAND (B26, TP16). When BUSY (B28) drops, start memory request timing (B26).

SCAND gates the output of the data memory address counter (A03, A04) through the controller address mux (A03, A04) to the memory address mux (A13 through A16). The address then goes out to memory.

Memory request timing (B26) generates MRQ1 signal (B26, TP23).

Clear the input and output memory flip-flops (B27, TP31, and 23) and the director request 3 flip-flop (B27, TP29).

Memory request timing (B26) generates MREQT signal (B26).

Set the reply timing enable flip-flop (B26, TP27).

Generate DATARQ signal (B27), which clocks the input and output memory flip-flops (B27, TP31, and 23). The input memory flip-flop will set enabling the memory parity error circuit (B27) and MI/PRY signal (B27). In addition, DATARQ clears the BRY (B28, TP21) and BRQ (B28, TP24) flip-flops.

Start the memory busy delay line (B28) and clear the memory not busy flip-flop (B28, TP1) dropping BUSY (B28) to a zero disabling any memory requests (B26).

Memory request timing (B26) generates MREQ signal (B26, TP20). This goes to memory and is the memory request signal.

MSBY1 signal (B28) arrives from memory and holds BUSY signal (B28) to a zero.

MDRDY signal (B26) arrives from memory along with the data byte, STO 100 through STO 116 signals (A13 through A16).

MDRDY starts reply timing (B26).

The data byte goes to the parity checker (A17). If an error is detected, MPERR (A17, TP7) comes up a 1.

The data byte comes off the memory data bus [signals MB00 through MB15 (A13 through A16)] and is gated through the data buffer mux (A07 through A12) to the data buffer (B10). Signals DMB00 through DMB15 (A07 through A12) to (B10).

Reply timing generates REPLY signal (B26), which in turn generates MI/PRY signal (B27, TP32). This generates MRLY1 signal (B14, TP26) which, in turn, generates DWC signal (B14, TP28) and LDBUF2 signal (B14, TP27).

MRLY1 clocks the BRQ flip-flop (B28, TP24). If the word count is not one, signal WCT1 (B20, TP31) is a 0, and the data buffer RK2 is not full [signal DBR2 (B10) is a 1], the flip-flop will set, maintaining signal BRQ (B28, TP25) at a 0 and disabling clearing of DRQT flip-flop (B28, TP19).

LDBUF2 gates the data byte into the data buffer (B10). When data enters rank one, generate DBRQD (B10). If BRQ flip-flop is clear, generate BRQ (B28, TP25) dropping the DBI PRQ signal (B17, TP4) which in turn drops ENDRQ (B26).

DWC increments the data memory address counter (A03, A04) and decrements the word counter (B20).

Reply timing generates MRP60 signal (B26).

Clear the reply timing enable and initiate flip-flops (B26, TP27).

Clock the memory parity error flip-flop (B27, TP9). If MPERR signal (A17) is a 1, the flip-flop will set generating PARERR signal (B27) which is a fault condition and will stop director loading execution.

Reply timing drops the REPLY signal (B26) which, in turn, drops MI/PRY (B27, TP32). Clock the DRQT flip-flop (B28, TP19). If the BRQ flip-flop (B28, TP24) is clear, DRQT will clear; otherwise, it remains set.

If it clears, DRQT (B28) drops releasing the scanner (B26). Data loading stops.

If it remains set, the scanner stays locked on its data position.

The memory busy timing chain times out (B28, TP3).

Clock the memory not busy flip-flop (B28, TP1). It will set dropping the BUSY signal (B28).

The MBSY1 signal (B28) from memory drops causing BUSY signal (B28) to drop.

Enable memory request timing (B26). If the scanner (B26) is still locked in data position, start timing to request next byte of data from memory (return to the fourth main step under X. B. 5. c to continue the loading sequence). Otherwise, enable next memory request.

C. Generate ADDC signal (A05).

1. Generate RTFD signal (B23, TP1).

a. Enable execution of next director (A28).

XI. When next internal clock occurs, clear address director control flip-flop IIB (A05, TP18) and address director control flip-flop ID (A05).

A. Generate ADD2 signal (A05).

1. Clock the ADR flip-flop (A27, TP22). It will clear the address director enables.

2. Clock the support request flip-flop (A27, TP2). It will set requesting the next director from the director buffer.

XII. Next internal clock clears address director control flip-flop IIC (A05, TP20).

XIII.. Director execution continues with next director.

T1Ø-50

0 4 XX

Conditional Branch Director Execution 5-21

This director loads the address specified by director bits 16 through 31 into the director address register if the flag bit indicated by bits 08 through 14 is set. This results in a branch operation which will cause subsequent directors to be loaded from the new memory location. The flags, which are checked by this director, must be preconditioned by a test director, a compare director, or a read checkword director. More than one flag can be checked simultaneously. If the indicated flag is set, the branch address is transferred to the director address register, the director buffer input and output pointers are cleared, and the director buffer empty flip-flop is set. Directors previously loaded into the director buffer will not be executed. If the indicated flag is not set, the branch is not made and director execution continues normally.

Bit 15 of this director will inhibit execution of the director until the read in progress bit drops,indicating the end of a read operation. This option is used to turn on the erase gate when switching from a read to a write operation. It is normally used when reading the address field checkword prior to writing a sector. When the read in progress bit drops, the conditional branch director checks for a compare error or a checkword error. If neither flag is set, the branch is not made, the erase gate is turned on, and the write operation will be performed. If one of the flags is set, the erase gate is not turned on, the branch is made, and the subsequent directors will handle the error condition. (Erase gate is not used in double density disk units.) The following is a list of the optional bits and their purposes.

- Bit 08 Check equal flag.
- Bit 09 Check positive flag.
- Bit 10 Check negative flag.
- Bit 12 Check compare condition not met flag.
- Bit 13 Check checkword not equal to zero flag.
- Bit 14 Check bit compare flag.
- Bit 15 Inhibit execution until read in progress bit drops.

The following is the sequence of operations for this director (refer to Figure 3-116).

I. Read timing chain generates SO/PD (A27, TP23).

   A. Gate the contents of the addressed director buffer location to the director output register (A25, A27, and A28). The director output register feeds the director fan-out (A17) and the director decode logic on B23.

II. Director is decoded on A25 and B23.

   A. Generate FUNDIR (A25, TP18). This signal causes the support director control logic to generate RTFE.

   B. Generate ZDE (A25). This signal is a partial decode which provides enables on B23.

   C. Generate CONDB (B23).

III. Support director control logic generates RTFE (A28, TP24).

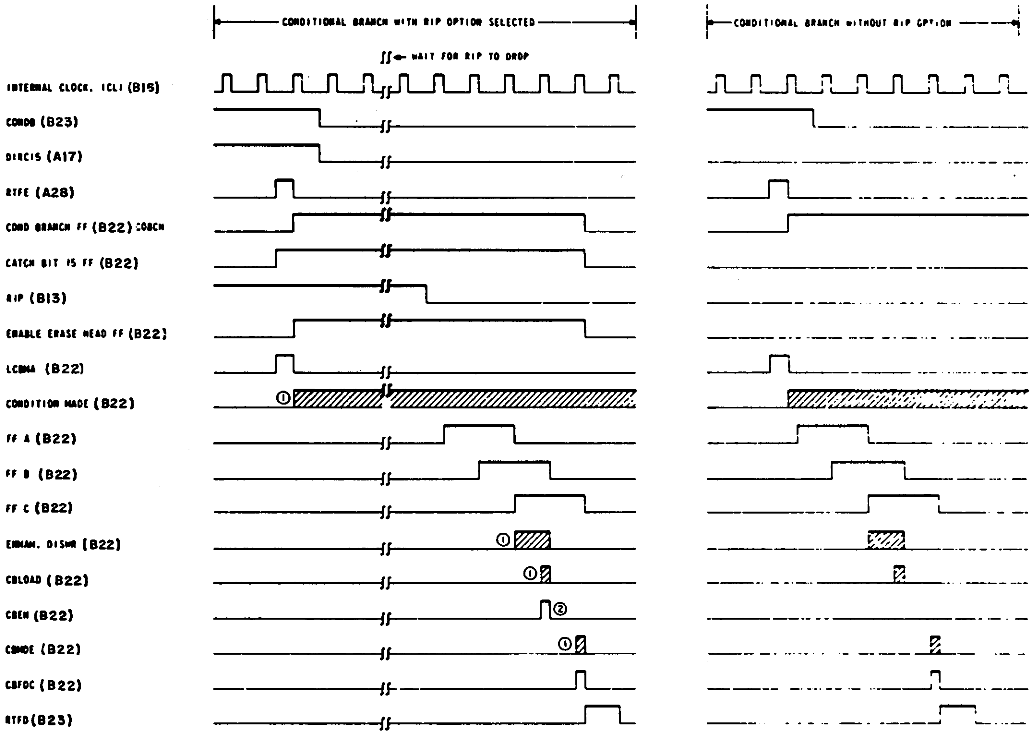
   A. CONDB and RTFE set the conditional branch flip-flop (B22, TP11) and clock the catch bit 15 flip-flop (leading edge) and the erase head enable flip-flop (B22). The catch bit 15 flip-flop sets if director bit 15 is set. The erase head enable flip-flop sets only if the catch bit 15 flip-flop is set and the RIP signal (B13) is up.

   B. Generate LCBMA (B22). This signal gates director bits 16 through 31 into the conditional branch address holding register on A03 and A04.

   C. CONDB inhibits setting of support director request flip-flop (A27, TP2). This prevents another director from being gated onto the director bus until the branch decision is made.

   D. Generate COBCH (B22). This signal preclears the support director ready flip-flop(A28, TP20).

   E. Gate director bits 08 through 14 into support director catch register (A01, A02). These bits are then gated to the branch condition gates on B22. If the indicated condition exists, the condition made signal (B22, TP6) is generated.



NOTES: ① SHADED SIGNALS OCCUR ONLY IF BRANCH CONDITION IS MADE.
② CBEN OCCURS ONLY IF BRANCH CONDITION IS NOT MADE.

Figure 3-116. Conditional Branch Director Execution

NOTE

Conditional branch timing chain is enabled by the conditional branch flip-flop. Flip-flops A, B, and C will set on successive clock pulses (ICL1) unless inhibited by a director request (DIRRQ1) or by the director write timing chain (DSRT and DISRT). If the inhibit is present, the respective flip-flops will set on the clock immediately following removal of the inhibit signal.

IV. Generate ICL1 (B16).

A. Flip-flop A (B22, TP25) will set. If catch bit 15 flip-flop is set, flip-flop A will not set until RIP signal (B13) drops.

~~V. Generate ICL1 (B16).~~

~~A. Flip-flop B (B22, TP9) will set.~~

VI. Generate ICL1 (B16).

A. Flip-flop B (B22, TP17) will set and flip-flop A ~~will clear~~ IS ENABLED. If the condition made signal is present, the following will occur.

1. Generate DISWR (B22, TP1). This signal disables the director write timing chain.

2. Generate ENMAM (B22, TP20). This signal gates the branch address from the memory address multiplexer on A03 and A04.

VII. Generate ICL1 (B16). FF "C" (B22, TP17) will set. FF "A" ~~will clear.~~ WILL CLEAR FF"B" is ENABLED

A. If the condition made signal is present, generate CBLOAD (B25, TP21). This signal generates the LDDMA signal on A26 which then gates the branch address from the memory address multiplexer to the director register on A03 and A04. It also generates BRACH which clears the director buffer input and output address registers on A25, disables any director write timing which is in progress and generates a new director request on A26, and presets the director buffer empty flip-flop on A27.

B. If the condition made signal is not present and the enable erase head flip-flop is set, generate CBEH (B22). This turns on the erase head in the selected disk storage unit.

VIII. Generate ICL1 (B16). Clear FF "B" (B22).

A. Preclear the conditional branch flip-flop (B22, TP11).

B. Generate CBFDC (B22, TP7). This signal presets the support director request flip-flop (A27) and generates RTFD on B23. The next director, either branch or not branch, can now be executed.

C. If condition made is present generate "CBMDE" signal (B22) which sets Decode Enable FF (A26, TP9)

IX. Generate ICLL clear FF "C" (B22).

Terminate Director Execution 08

This director causes the control logic to stop executing directors. The sequence of operation is:

I. Read timing chain generates SO/PD (A27, TP23).

A. Director is gated to director output register on A25.

II. Director is decoded on A25.

A. Generate TERMD (A25, TP22).

1. Preclear support director ready flip-flop on A28.

2. Enable END gate on A28.

III. Generate read time 300.

A. Generate END (A28, TP14).

1. Preclear director execution enable flip-flop (B27, TP14). This drops the ENDEX signal and disables the read director timing chain on A27.

IV. No further directors will be executed until a 0022 function code is received on normal channel 8.

Subtract Director Execution 10 5-23

This director subtracts the operand in director bits 08 through 23 from the contents of the register file location specified by director bits 28 through 31. The difference is then placed back into the specified register file location. The sequence of operation is:

I. Read timing chain generates SO/PD (A27, TP23).

A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and the director decode logic on B23.

II. Director is decoded on A25 and B23.

   A. Generate FUNDIR (A25, TP18). This signal causes support director control logic to generate RTFE.

   B. Generate ZDE (A25). This is a partial decode which enables decode logic on B23.

   C. Generate SUB and FLDIA (B23).

III. Support director control logic generates RTFE (A28, TP24).

   A. FLDIA and RTFE load director bits 28 through 31 into the register file address counter (B24). This enables the register file address through the register file address multiplexer 2 on B25. Since the register file is normally in a read mode, the addressed location is placed on the register file bus and becomes operand A of the ALU (A01, A02).

   B. Load director bits 08 through 23 into the support director catch register (A01, A02). These bits become operand B of the ALU.

IV. RTFE and SUB set the subtract flip-flop (B11, TP7).

   A. Generate ALUCN (B11). This provides a carry input to the ALU.

   B. Generate ALUS1 (B11). This enables the ALU for a subtract operation.

   C. Generate ALUMC (B11). This signal enables the output of the ALU to the ALU catching register (A01, A02).

   D. Start ALU timing chain (B11). The timing chain is strobed by successive internal clock pulses, ICL1. These pulses are considered CLK1 through CLK8 on B11.

V. Generate CLK5 (B11).

   A. Generate CALCR (B11). This strobes the output of the ALU into the ALU catching register (A01, A02). This is the result of the subtract operation.

VI. Generate CLK6 (B11).

   A. Generate ALURF (B11, TP17). This enables the ALU catching register onto the register file bus (A07 through A12) and drops RFME (B24, TP9).

   B. Clear the subtract flip-flop.

VII. Generate CLK7 (B11).

A. Generate ALUWT (B11, TP19). This signal generates RFWEL, RFWEU, and RFME on B24. These signals write the contents of the register file bus into the register file.

B. Generate ALUD (B11, TP20). This signal sets the RTFD flip-flop (B23, TP1) and the clear RTFD flip-flop (B23, TP2). RTFE initiates the next director.

VIII. Generate CLK8 (B11).

A. Clear last flip-flop in ALU timing chain (B11).

B. Preclear the RTFD flip-flop. The clear RTFD flip-flop will clock itself to a clear state.

5-23

<u>Add Director Execution</u> 12

This director adds the operand in director bits 08 through 23 to the contents of the register file location specified by director bits 28 through 31. The sum is then placed back into the specified register file location. The sequence of operations is:

I. Read timing chain generates SO/PD (A27, TP23).

A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and director decode logic on B23.

II. Director is decoded on A25 and B23.

A. Generate FUNDIR (A25, TP18). This signal causes support director control logic to generate RTFE.

B. Generate ZDE (A25). This is a partial decode which enables decode logic on B23.

C. Generate ADD and FLDIA (B23).

III. Support director control logic generates RTFE (A28, TP24).

A. FLDIA and RTFE load director bits 28 through 31 into register file address counter (B24). This enables the register file address through address multiplexer 2 on B25. Since register file is normally in a read mode, the addressed location is placed on the register file bus and becomes operand A of the ALU (A01, A02).

B. Load director bits 08 through 23 into support director catch register (A01, A02). These bits become operand B of the ALU.

IV. RTFE and ADD set add flip-flop (B11, TP5).

A. Generate ALUS0 and ALUS3 (B11). These signals enable the ALU for an add operation (A01, A02).

B. Generate ALUMC (B11). This signal enables the output of the ALU to the ALU catching register (A01, A02).

C. Start ALU timing chain (B11). The timing chain is strobed by successive internal clock pulses, ICL1. These pulses are considered CLK1 through CLK8.

V. Generate CLK5 (B11).

A. Generate CALCR (B11). This strobes the output of the ALU into the ALU catching register (A01, A02). This is the result of the add operation.

VI. Generate CLK6 (B11).

A. Generate ALURF (B11, TP17). This enables the ALU catching register onto the register file bus (A07 through A12) and raises RFME (B24, TP9).

B. Clear the add flip-flop.

VII. Generate CLK7 (B11).

A. Generate ALUWT (B11, TP19). This signal generates RFWEL, RFWEU, and RFME on B24. These signals write the contents of the register file bus into the register file.

B. Generate ALUD (B11, TP20). This signal sets the RTFD flip-flop (B23, TP1) and the clear RTFD flip-flop (B23, TP2). RTFE initiates the next director.

VIII. Generate CLK8 (B11).

A. Clear last flip-flop in ALU timing chain (B11).

B. Preclear the RTFD flip-flop. The clear RTFD flip-flop will clock itself to a clear state.

Test Director Execution 14 5-24

This director performs a test for relative magnitude or a test for a bit compare. The test for relative magnitude can be modified to decrement the contents of the register file if the test results are not equal. The desired test function is selected by setting one of the following bits.

- Bit 24 - Test for relative magnitude. With this bit set, the director operand (bits 08 through 23) is subtracted from the contents of the register file location specified by director bits 28 through 31. Following the subtract operation, a flag is set to specify whether the results were equal, positive, or negative. The original contents of the register file are restored in the specified register file location.
- Bit 26 - Test and decrement. With this bit set, the operand is tested the same as with bit 24 set. However, if the result is not equal, the contents of the register file are decremented by one and then stored in the specified register file location.
- Bit 25 - Bit compare. With this bit set, the director operand is compared with the contents of the specified register file location. If a bit compare occurs in any location, the bit compare flag is set.

The sequence of operation for all test functions is:

I. Read timing chain generates SO/PD (A27, TP23).

A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and director decode logic on B23.

II. Director is decoded on A25 and B23.

A. Generate FUNDIR (A25, TP18). This signal causes support director control logic to generate RTFE.

B. Generate ZDE (A25). This is a partial decode which enables decode logic on B23.

C. Generate TEST and FLDIA (B23).

III. Support director control logic generates RTFE (A28, TP24).

A. FLDIA and RTFE load director bits 28 through 31 into register file address counter (B24). This enables the register file address through register address multiplexer 2 on B25. Since register file is normally in a read mode, the addressed location is placed on the register file bus and becomes operand A of the ALU.

B. Load director bits 08 through 23 into the support director catch register (A01, A02). These bits become operand B of the ALU.

C. Load director bits 24 through 26 into the test condition catch register (B11).

D. RTFE and TEST preclear all test flag flip-flops (equal, positive, negative, and bit compare).

IV. RTFE and TEST set the test flip-flop (B11, TP6)

A. Decode the output of the test condition catch register (B11). Bits 24 or 26 will enable the set input gates of the equal, positive, and negative test flags (B11). Bit 25 enables the set input gate of the bit compare test flag.

B. The test flip-flop and the test condition decodes enable various gates for performing the ALU operations. These gates will be specified in subsequent steps.

C. Generate ALUMC (B11). Γhis signal enables the output of the ALU to the ALU catching register on A01 and A02.

D. Start ALU timing chain (B11). The timing chain is strobed by successive internal clock pulses, ICL1. These pulses are considered CLK1 through CLK8.

E. Generate ALUM, ALUSO, ALUS1, and ALUS3 if this is a bit compare operation. This gates the contents of the register file through the ALU to the bit compare logic.

V. Generate CLK1 (B11).

A. Generate ALUS1 if this is a magnitude compare operation. This will enable the ALU for a subtract operation without a carry input.

VI. ALU performs subtract operation or a bit compare operation (A01, A02).

A. Generate ALUEQ if quantities are equal.

B. Generate ALUC4U (A02) if result is negative.

C. Generate neither of the above if result is positive.

D. Generate BITL (A01) or BITU (A02) for a bit compare.

VII. Generate CLK3 (B11).

A. Clock all test flag flip-flops (equal, positive, negative, and bit compare). Applicable flip-flop will set if condition is present.

B. If results are not equal and this is a test and decrement operation, set the test and decrement flip-flop (B11). This flip-flop disables the set input gate to the not store ALU flip-flop. It also generates ALUS0, ALUS1, and ALUS3 which enable the ALU for a decrement operation.

VIII. Generate CLK5 (B11).

A. Generate CALCR (B11). This strobes the output of the ALU into the ALU catching register (A01, A02). This is the result of the decrement operation.

B. Set the not store ALU flip-flop unless the test and decrement flip-flop is set. Setting the not store ALU flip-flop inhibits the register file write cycle and the original contents remain in the register file.

IX. Generate CLK6 (B11).

A. Generate ALURF (B11, TP17). This enables the ALU catching register onto the register file bus (A07 through A12) and drops RFME (B24, TP9).

B. Clear the test flip-flop (B11).

X. Generate CLK7 (B11).

A. Generate ALUWT (B11, TP19). This signal generates RFWEL, RFWEU, and RFME on B24. These signals write the contents of the register file bus into the register file.

B. Generate ALUD (B11, TP20). This signal sets the RTFD flip-flop (B23, TP1) and the clear RTFD flip-flop (B23, TP2). RTFD clears the test and decrement flip-flop and the not store ALU flip-flop on ~~C07~~ B11 and initiates the next director.

XI. Generate CLK8 (B11).

A. Clear last flip-flop in ALU timing chain (B11).

B. Preclear the RTFD FF. The clear RTFD flip-flop will clock itself to a clear state.

5-25

Complement Director Execution

This director complements the contents of the register file location specified by director bits 28 through 31. The result of the operation is then placed back into the specified register file location. The sequence of operations is:

I. Read timing chain generates SO/PD (A27, TP23).

   A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and director decode logic on B23.

II. Director is decoded on A25 and B23.

   A. Generate FUNDIR (A25, TP18). This signal causes support director control logic to generate RTFE.

   B. Generate ZDE (A25). This is a partial decode which enables decode logic on B23.

   C. Generate COMP and FLDIA (B23).

III. Support director control logic generates RTFE (A28, TP24).

   A. FLDIA and RTFE load director bits 28 through 31 into register file address counter (B24). This enables the register address through address multiplexer 2 on B25. Since register file is normally in a read mode the addressed location is placed on the register file bus and becomes operand A of the ALU (A01, A02).

IV. RTFE and COMP set complement flip-flop (B11, TP8).

   A. Generate ALUM (B11). This enables the ALU for a complement operation.

   B. Generate ALUMC (B11). This signal enables the output of the ALU to the ALU catching register (A01, A02).

   C. Start ALU timing chain (B11). The timing chain is strobed by successive internal clock pulses, ICL1. These pulses are considered CLK1 through CLK8.

V. Generate CLK5 (B11).

   A. Generate CALCR (B11). This strobes the output of the ALU into the ALU catching register (A01, A02). This is the result of the complement operation.

VI. Generate CLK6 (B11).

A. Generate ALURF (B11, TP17). This enables the ALU catching register onto the register file bus (A07 through A12) and drops RFME (B24, TP9).

B. Clear the complement flip-flop.

VII. Generate CLK7 (B11).

A. Generate ALUWT (B11, TP19). This signal generates RFWEL, RFWEU, and RFME on B24. These signals write the contents of the register file bus into the register file.

B. Generate ALUD (B11, TP20). This signal sets the RTFD flip-flop (C09, TP1) and the clear RTFD flip-flop (B23, TP2). RTFE initiates the next director.

VIII. Generate CLK8 (B11).

A. Clear last flip-flop in ALU timing chain (B11).

B. Preclear the RTFD flip-flop. The clear RTFD flip-flop will clock itself to a clear state.

Load Register File Director Execution 80 5-25

This director loads the operand in director bits 08 through 23 into the register file location specified by director bits 28 through 31. The sequence of operations is:

I. Read timing chain generates SO/PD (A27, TP23).

A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and director decode logic on B23.

II. Director is decoded on A25 and B23.

A. Generate FUNDIR (A25, TP18). This signal causes the support director control logic to generate RTFE.

B. Generate LIDE and FLDIA (B23)

III. Support director control logic generates RTFE (A28, TP24).

A. LIDE and RTFE set load register file flip-flop (B24, TP4).

B. LIDE inhibits setting of support director request flip-flop (A27, TP2). This prevents another director from being gated onto the director bus before the present director bits have been loaded into the register file.

C. FLDIA and RTFE load director bits 28 through 31 into register file address counter (B24). This enables the register file address through address multiplexer 2 on B34. Since register file is normally in a read mode, the addressed location is placed on the register file bus.

IV. Generate ICL2 (B16).

A. Set the enable director to register file flip-flop (B24, TP2). This flip-flop drops the RFME signal (B24) causing the register file bus to float. It also generates LOADI (B24) which enables the output of the director output register onto the register file bus.

V. Generate ICL2 (B16).

A. Set the director operation done flip-flop (B24, TP1).

1. Generate RFME, RFWEU, and RFWEL. These signals write the director operand into the register file.

2. Generate LDICONA (B24, TP1). This signal sets the support director request flip-flop (B27, TP2) and initiates the read timing chain for the next director.

3. Generate IDDC (B24). This signal indicates that the register file operation is done and enables the RTFD flip-flop on B23.

4. Preclear the load register file flip-flop (B24).

VI. Generate ICL1 and ICL2 (B16).

A. The enable director to register file flip-flop and the director operation done flip-flop both are clocked to a clear state. Register file is returned to a normal read mode.

B. Set the RTFD flip-flop (B23, TP1). This enables execution of the next director.

C. Set the clear RTFD flip-flop (B23, TP2).

VII. Generate ICL1 (B16).

A. Preclear the RTFD flip-flop (B23).

B. The clear RTFD flip-flop will clock itself to a clear state (B23).

Keypoint Director Execution

This director leads the support data catch register with director operand bits 08 through 23 and provides a negative going pulse on B23, TP32. It provides a keypoint in a program loop for maintenance purposes. The sequence of operation is:

I. Read director timing chain generates SO/PD (A27, TP23).

   A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and director decode logic on B23.

II. Director is decoded on A25 and B23.

III. A. Generate FUNDIR (A25, TP18). This signal causes support director control logic to generate RTFE.

   B. Keypoint instruction is decoded on B23.

IV. Support director control logic generates RTFE (A28, TP24).

   A. Gate director bits 08 through 23 into support director catch register on A01 and A02.

   B. Generate negative going pulse on B23, TP32. This is the RTFE pulse and is 100 to 140 nanoseconds wide. The operand bits are stable on the trailing edge of this pusle.

   C. Generate LPCD (B23).

      1. Set request honored flip-flop (B24).

V. Generate ICL2 (B16).

   A. Set memory operation done flip-flop (B24).

      1. Clear request honored flip-flop.

      2. Generate IDDC (B24).

VI. Set RTFD flip-flops on B23.

   A. Generate RTFD (B23, TP1). This terminates the sequence. RTFD flip-flops clear on next ICL1 pulse.

Load Polynomial Director 84

This director loads operand bits 08 through 23 into designated polynomial holding register for use during operations using the checkword generator. Bits 25 through 28 specify the holding registers as follows:

• Bit 25	Load positions 00 through 15 of the divide polynomial.
• Bit 26	Load positions 16 through 31 of the divide polynomial.
• Bit 27	Load positions 00 through 15 of the multiply polynomial.
• Bit 28	Load positions 16 through 31 of the multiply polynomial.

If the polynomial quantities are identical, more than one of these bits may be set at a time. For example, if zeros are to be loaded into all positions of both the divide and multiply polynomial registers, bits 25 through 28 are all set and one director will load both registers. In the worst case, four directors would be required to load the two registers. The sequence of operation is:

I. Read timing chain generates SO/PD (A27, TP23).

   A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on C01 A17 and director decode logic on B23.

II. Director is decoded on B23 and A25.

   A. Generate FUNDIR (A25, TP18). This signal causes the support director control logic to generate RTFE.

   B. Function code 84 is decoded on B23.

III. Support director control logic generates RTFE (A28, TP24).

   A. Generate LECD1, LECD2, LECM1, and LECM2 depending on setting of bits 25 through 28. These signals load the director operand into the applicable registers on the checkword generator (B03 through B08).

   B. Generate LPCD (B23). This sets the request-honored flip-flop on B24.

IV. Generate ICL2 (B16).

   A. Set memory operation done flip-flop (B24).

      1. Clear request honored flip-flop.

      2. Generate IDDC (B24).

V. Set RTFD flip-flops on B23. Generate RTFD to terminate the sequence. RTFD flip-flops will clear on next ICL1 pulse.

Load Hardware Conditions Director

This director preconditions the hardware for conditions which will remain relatively static for all hardware operations. The selectable conditions are:

- Bit 11 - If set, this bit will preset every stage of the checkword generator prior to each checkword operation. If clear, this bit will preclear the checkword generator.
- Bit 12 - When set, this bit selects 8-bit mode of operation. If clear, this bits selects 6-bit mode.
- Bit 15 - When set, this bit will cause the control logic to zero fill to the end of a sector upon receipt of an end of record signal. If this bit is clear, no zero fill will be performed.
- Bits 16 through 20 - These bits contain a sector notch count which is one less than the number of physical notches required per logical sector. A zero count equals one physical sector notch per logical sector.

The sequence of operation is:

I. Read director timing chain generates SO/PD (A27, TP23).

A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and director decode logic on B23.

II. Director is decoded on A25 and B23.

A. Generate FUNDIR (A25, TP18). This signal causes support director control logic to generate RTFE.

B. Function code 86 is decoded on B23.

III. Support director control logic generates RTFE (A28, TP24).

A. Generate LHDWC (B23).

1. Load director bits 16 through 20 into the sector notch count register on A06.

2. Load director bits 11, 12, and 15 into the hardware conditions register on B12.

3. Set request honored flip-flop (B24).

IV. Generate ICL2 (B16).

A. Set memory operation done flip-flop (B24).

1. Clear request honored flip-flop.

2. Generate IDDC (B24).

V. Set RTFD flip-flops on B23. Generate RTFD to terminate the sequence. RTFD flip-flops will clear on next ICL1 pulse.

Copy Disk Status Director 88 5-28

This director copies status from the selected disk drive and places it in the register file location designated by bits 28 through 31. The status word will always contain the status previously selected by an enable disk status director. If bit 24 is clear, bits 00 through 05 of the status word will be zero.

If bit 24 is set, bits 00 through 05 of the status word will contain the following unit status.

- Bit 0 - Sector alert
- Bit 1 - Seek error
- Bit 2 - Unit busy
- Bit 3 - Unit selected
- Bit 4 - Unit ready
- Bit 5 - Unit on-line

Bits 07 through 15 of the status word contain the selected status.

If no unit is selected, the status word will be all zeros. The sequence of operation is:

I. Read director timing time 100 generates SO/PD signal (A27, TP23).

A. Gate the contents of the addressed director buffer memory location (A29) to the director register (A25, A27, A28). This is the copy disk status director.

B. The director register (A25, A27, A28) and director register fanout (A17) feed the director translators (A25, B23).

1. Generate FUNDIR (A25, TP18).

2. Generate CDSDE (B23) and FLDIA (B23).

II. Director execution sequence generates next RTFE signal (A28, TP24).

   A. Load director bits 28 through 31 into the register file address counter (B24). This is the register file address.

      1. The outputs of the register file counter (B24) feed through the register file address mux (B25) and address a register file location.

   B. Clock the enable unselected status flip-flop (C12, TP16). If bit 24 is a one, it will set.

   C. Set the copy disk status flip-flop (C12, TP12).

      1. Start the status control timing chain (C12).

III. After seven internal clock pulses, status control timing generates STTRF signal (C12).

   A. Generate RFME signal (B24, TP9).

      1. Stop referencing the addressed register file location (B25).

   B. Gate the selected status from the selected drive onto the register file bus (A09 through A12).

      1. The status comes up on the input to the addressed register file location (B25).

IV. If director bit 24 holding flip-flop (C12, TP16) is set, status control timing generates STRF signal (C12).

   A. Gate the unselected status bits from the selected drive onto the register file bus (A07, A08).

      1. The status comes up on the input to the addressed register file location (B25).

V. After eight internal clock pulses, status control timing generates DSW signal (C12).

   A. RFWEL (B24), RFWEU (B24), and RFME (B24, TP9) signals go to zeros.

      1. Write the status into the addressed register file location (B25).

   B. Clear the copy disk status flip-flop (C12, TP12) stopping the timing chain.

   C. Generate RTFD signal (B23, TP1).

      1. Enable execution of next director (B28).

VI. Director execution continues with next director.

Stop Loading Directors Director Execution 92

This director is not loaded into the director buffer, but is executed immediately when it is obtained from memory. Refer to the paragraphs on director write timing for special directors for the execution sequence for this director.

Unconditional Branch Director Execution 94

This director is not loaded into the director buffer, but is executed immediately when it is obtained from memory. Refer to the paragraphs on director write timing for special directors for the execution sequence for this director.

Store Register File Director Execution 98 5-36

This director stores the contents of the register file location specified by bits 12 through 15 in the memory location specified by bits 16 through 31. The sequence of operation is:

I. Read timing chain generates SO/PD (A27, TP23).

   A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and the director decode logic on B23.

II. Director is decoded on A25 and B23.

   A. Generate FUNDIR (A25, TP18). This signal causes support director control logic to generate RTFE.

   B. Generate 4DE (A25). This is a partial decode which enables decode logic on B23.

   C. Generate STIDE and FLDIA (B23).

III. Support director control logic generates RTFE (A28, TP24).

   A. STIDE enables director bits 12 through 15 into the register file address mux (B24).

   B. FLDIA and RTFE load director bits 12 through 15 into the register file address counter (B24). This enables the register file address through the register file address multiplexer 2 on B25. Since the register file is normally in a read mode, the addressed location is placed on the register file bus.

C. STIDE and RTFE set the register file output request flip-flop (B24). If there is no data transfer to memory in progress, the flip-flop generates a register file output request, RFOPRQ, and enables the register file to the memory bus, RFBMB.

D. STIDE and RTFE generate LDAD2 (B24) which generates LDADR (A06) which then loads director bits 16 through 31 into the data address mux.

IV. The RFOPRQ signal generates a memory request through the memory scanner (B26, B27).

A. Generate SCAND (B26) which gates the data address through memory address mux 1 (A03, A04) to memory address mux 2 (A14 through A16).

B. Generate O/PRQ (B27) which generates STORE (B26). This inhibits the memory read data access to the memory bus (A13 through A16) and informs the memory that the coming operation is a store operation.

C. Generate MREQ (B26). This is the access request to memory.

V. The memory accepts the data on the data lines and returns the MDRDY signal.

A. Generate REPLY (B26). This releases the scanner on B26 and generates a memory output reply, MO/PRY (B27).

VI. Set the request honored flip-flop (B24).

A. Clear the register file output request flip-flop (B24).

VII. Generate ICL2 (B16).

A. Set the memory operation done flip-flop (B24). This generates IDDC which enables the RTFD flip-flops on B23.

VIII. Generate ICL2 (B16).

A. Clear the memory operation done flip-flop (B24). Set the RTFD flip-flops on B23. This enables execution of the next director.

IX. Generate ICL1 (C16).

A. Clear the RTFD flip-flops (B23).

S-37

Enable Disk Function Director CO

This director sends control information to the disk interface. Bits 08 through 14 set tag lines which specify how the information is to be used. The required information can either be supplied from the operand field (bits 15 through 23) or from bits 07 through 15 of the specified register file location. (For double density option, register file bits are 06 through 15.) The field entries and their purposes are:

• Bit 08	Enable unit select
• Bit 09	Disable unit select
• Bit 10	Load difference register
• Bit 11	Load cylinder register
• Bit 12	Load sector register
• Bit 13	Load head register
• Bit 14	Enable control select line
• Bit 24	Disable reserve on opposite channel
• Bit 25	Use register file location specified by bits 28 through 31

The sequence of operation is illustrated in Figure 3-117 and described below.

I. Read director timing time 100 generates SO/PD signal (A27, TP23).

   A. Gate the contents of the addressed director buffer memory location (B15) to the director register (A25, A27, A28).

   B. The director register and director register fanout A17 feed the director translators (A25 and B23).

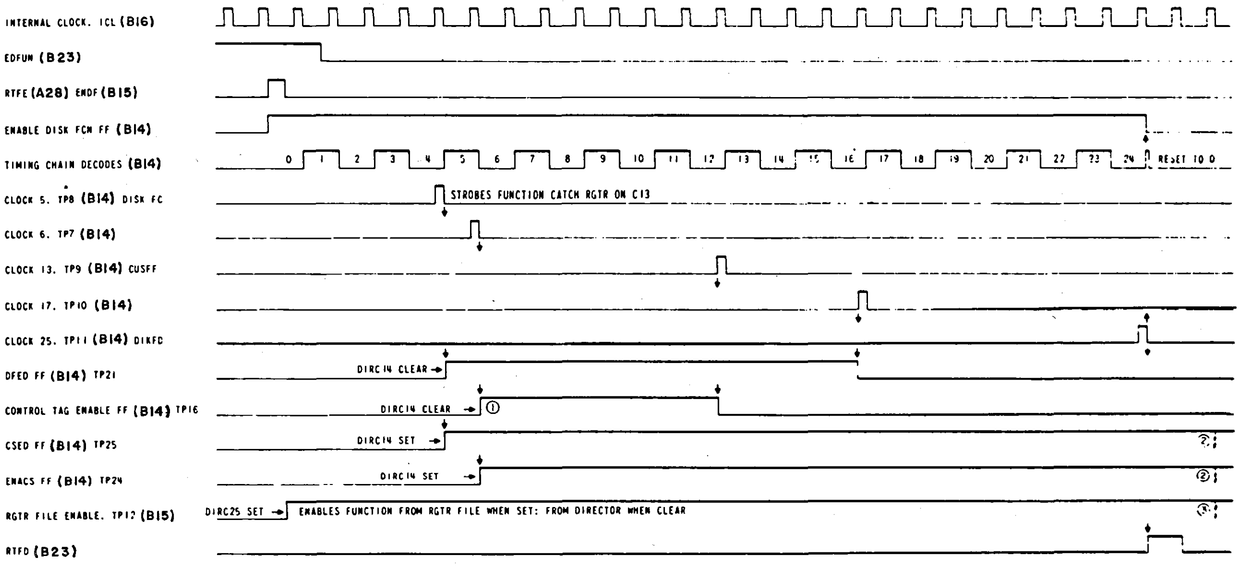
      1. Generate FUNDIR (A25, TP18).

      2. Generate EDFUN and FLDIA (B23).

II. Director execute sequence generates RTFE (A28, TP24).

   A. Load the register file address counter (B24, TP11) with director bits 28 through 31.

      1. The counter output (B24) feeds through the register file address mux (B25) and addresses a register file memory location.


INTERNAL CLOCK. ICL (B16)
EDFUN (B23)
RTFE (A28) ENDF (B15)
ENABLE DISK FCN FF (B14)
TIMING CHAIN DECODES (B14)
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
RESET TO 0
CLOCK 5. TP8 (B14) DISK FC
STROBES FUNCTION CATCH RGTR ON C13
CLOCK 6. TP7 (B14)
CLOCK 13. TP9 (B14) CUSFF
CLOCK 17. TP10 (B14)
CLOCK 25. TP11 (B14) DIKFD
DFED FF (B14) TP21
DIRC14 CLEAR
CONTROL TAG ENABLE FF (B14) TP16
DIRC14 CLEAR
①
CSED FF (B14) TP25
DIRC14 SET
②
ENACS FF (B14) TP24
DIRC14 SET
②
RGTR FILE ENABLE. TP12 (B15)
DIRC25 SET
ENABLES FUNCTION FROM RGTR FILE WHEN SET: FROM DIRECTOR WHEN CLEAR
③
RTFD (B23)

NOTES: ① ENABLES EQPSL + LDDIF + LDCYL + LDSEC + LDHED.

② CLEARED BY DISABLE CONTROL SELECT DIRECTOR.

③ CLEARED BY NEXT ENABLE DISK FUNCTION DIRECTOR IF DIRC25 IS CLEAR.

Figure 3-117. Enable Disk Function Director Execution

B. Generate ENDF signal (B14, TP6).

1. Set enable disk function flip-flop (B14, TP12) and start the disk interface timing chain (B14).

2. Clock the enable disk function from register file flip-flop (B15, TP12). If bit 25 is present, the flip-flop will set enabling the output from the register file (B25) via data path 2 (A07 through A12) into the address and control inverters (B15). If bit 25 is zero, the flip-flop will clear and the operand portion of the director feeds the inverters.

C. Clock the disable reserve on opposite channel flip-flop (C12, TP5).

1. If bit 24 is present, it will set and enable the disable reserve signal (C12).

2. If bit 24 is a zero, it will clear.

III. Disk interface timing chain in proper condition.

A. Generate DISKFC (B14, TP8) on next ICL1 pulse.

1. Clock the address and control register to gate either the operand from the director register or register file into the register (B15).

2. Clock the disk function enable flip-flop (B14, TP21) and the control select address enable flip-flop (B14, TP25).

B. Next ICL1 signal generates signal from B14, TP7.

1. Clock the control tag enable flip-flop (B14, TP16) and the control select tag flip-flop (B14, TP24).

2. If bit 14 is present, set the control select flip-flop (B14, TP24) and the control select enable flip-flop (B14, TP25).

   a. Generate ENACS signal (B14, TP24) and send it out to the selected drive (C05, TP31).

   b. Generate CSED signal (B14, TP25). Gate control information out of the address and control register (B15) to the selected drive (C01, C02, C03).

3. If bit 14 is not present, set the control tag enable flip-flop (B14, TP16) and the disk function enable flip-flop (B14, TP21).

a. Send an address tag (B14, TP17 through TP20) signal to the selected drive (C04, C05).

b. Gate the address from the address and control register (B15) to the selected drive (C01, C02, C03).

C. If director bit 8 is present, gate bit 0, 1, 2, 4, 5, and $\overline{3}$ of the operand into the unit select register (C12).

IV. Disk interface timing generates CUSFF (B14, TP9).

A. Clear the control tag enable flip-flop (B14, TP16) and drop any existing address tags (B14, TP17 through TP20).

1. Set the drive pulse flip-flop (C12, TP7).

a. If the disable reserve flip-flop (C12, TP5) is set, generate DISRE signal (C12) and send it to the selected drive (C09).

b. If bit 8 is present, enable CREL and CFAUT signals (C12) out to selected drive (C09, C10). Bit 4 or 5 in the unit select register (C12) must be set to send either of these signals.

B. Clock the unit select enable flip-flop (C12, TP8).

1. If bit 8 is present, it will set.

a. Bring up unit select decode (C12).

b. Generate USELT signal (C12) and send a unit select signal to the selected unit (C14).

c. Generate USELR signal (C12) and enable unit receivers from selected drive (C16 through C21).

2. If bit 9 is present, it will clear dropping the present drive selection.

V. Disk interface timing brings up the inverter to TP10 (B14).

A. Clear the disk function enable flip-flop (B14, TP21), if set. Drop the address from the lines to the selected drive (B15).

VI. Disk interface timing generator DIKFD (B14, TP11).

A. Clock the enable disk function flip-flop (B14, TP12). It will clear stopping the timing chain.

B. Clock the drive pulse flip-flop (C12, TP7). It will clear dropping CREL, CFAUT, or DISRE signals, if present.

C. Generate RTFD signal (B23, TP1) which indicates the end of this director.

1. Enable execution of the next director (A28).

Disable Control Select Director Execution

This director drops the control select line previously enabled by the enable disk function director. The sequence of operation is:

I. Read director timing time 100 generates SO/PD signal (A27, TP23).

A. Gate the contents of the addressed director buffer memory location (A29) to the director register (A25, A27, A28). This is the disable control select director.

B. The director register (A25, A27, A28) and director register fanout (A17) feed the director translators (A25, B23).

1. Generate FUNDIR (A25, TP18).

2. Generate DDFUN (B23).

II. Director execution sequence generates RTFE (A28, TP24).

A. Set the disable control select flip-flop (B14, TP13).

1. Block setting of DFED flip-flop (B14, TP21) and control tag enable flip-flop (B14, TP16).

2. Start the disk interface timing chain (B14).

B. Clear the control select tag flip-flop (B14, TP24).

1. Drop the ENACS signal (B14).

III. Disk interface timing generates DISKFC signal (B14, TP8).

A. Clear the CSED flip-flop (B14, TP25) dropping the CSED signal.

1. Drop any existing control signals to units (B15).

IV. Disk interface timing generates DIKFD signal (B14, TP11).

A. Clock the disable control select flip-flop (B14, TP13). It will clear.

1. Stop the disk interface timing chain (B14).

B. Generate RTFD signal (B23, TP1).

1. Enable execution of next director (A28).

V. Director execution continues with next director.

E0 E8

Enable/Disable Disk Status Director

This director enables a specific disk drive status to the hardware. The desired status is specified by setting one of the following bits.

- Bit 11 Read cylinder register
- Bit 12 Read difference register
- Bit 13 Read head register
- Bit 14 Read sector counter
- Bit 15 Read sector register
- Bit 16 Read interlock
- Bit 17 Read position
- Bit 18 Read fault
- Bit 19 Read control

Any clear bits disable the associated status. The sequence of operation is:

I. Read director timing time 100 generates SO/PD signal (A27, TP23).

   A. Gate the contents of the addressed director buffer memory location (A29) to the director register (A25, A27, A28). This is the enable/disable disk status director.

   B. The director register (A25, A27, A28) and director register fanout (A17) feed the director translators (A25, C09).

      1. Generate FUNDIR (A25, TP18).

II. Director execution sequence generates RTFE signal (A29, TP24).

   A. Gate director bits 11 through 19 into the status select register (C12).

      1. Gate proper status select signal to selected drive (C05, C06, C07, C08).

      2. If bit 19 flip-flop in the register (C12, TP30) sets, set the RDCT2 flip-flop (C12). Generate RDCT2.

         a. Enable sector mark and index mark signals from the selected drive to come into the controller (B15).

         b. Enable EOC signal from the selected drive into the controller (B12).

III. Generate EDSD signal (C12).

A. Generate LPCD signal (B23).

1. Generate IDDC signal (B24).
Generate RTFD signal (B23, TP1) which enables execution to next director (A28).

IV. Director execution continues with next director.

Initiate Error Correction Director FO 541

This director is used to determine if a checkword error is correctable. The director contains a shift count in director bits 16 through 31. This count is representative of the bit length of the data field in which the error occurred. Upon initiation, the director checks for all zeros in bit positions 11 through 31 of the checkword generator. If these bits are not all zeros, the checkword generator is shifted end-around one position, the shift count is decremented, and the check is repeated. This action is repeated until bits 11 through 31 are all in a zero state (correctable error) or until the shift count has decremented to zero (uncorrectable error). The sequence of operation is:

I. Read timing chain generates SO/PD (A27, TP23). Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and the director decode logic on B23.

II. Director is decoded on A25 and B23.

A. Generate FUNDIR (A25, TP18). This signal causes support director control logic to generate RTFE.

B. Decode director on B23.

III. Support director control logic generates RTFE (A28, TP24).

IV. Generate SECOR (B23).

A. Set error correction flip-flop (B13, TP14).

B. Clear correctable error and uncorrectable flip-flops (B13).

C. Generate BCLCW (B13). This generates LBCD (B21, TP11) which loads director bits 16 through 31 into the byte counter on B21.

V. Generate ICL2 (B16).

A. Set error correction resync flip-flop (B13, TP17). This enables the error correction clock gate, the correctable error gate, and the uncorrectable error gate. Each of these gates is clocked by the ICL internal clock.

VI. Generate ICL2 (B16).

A. If bits 11 through 31 equal zero, set correctable error flip-flop (B13, TP20).

B. If BCTZ is present (byte count equals zero), set uncorrectable flip-flop (B13, TP21).

C. If neither of the above conditions is present, generate CORCK (B13, TP19). This is the correction clock. It shifts the checkword generator (B03 through B08) one position. It also generates DDBC on B17 (TP15) which decrements the byte counter (B21).

NOTE

Byte counter is actually incremented. However, the complement of the shift count was loaded into the counter. Incrementing the complement is decrementing the byte count.

VII. Continue step VI until either correctable error flip-flop sets or uncorrectable error flip-flop sets.

A. Generate ECDON (B13). This sets the RTFD flip-flops on B23.

B. Generate either CORERR (B13, TP20) or UNCERR (B13, TP21). These are available as status bits 13 and 14, respectively, on A23.

VIII. Generate ICL1 (B16).

A. RTFD flip-flops will clear.

Read Normal Director Execution

This director transfers data from the disk to the destination designated by the associated address director. The I/O length is determined by bits 16 through 31 or is obtained from the register file location designated by bits 28 through 31. Figures 3-118 and 3-119 illustrate the two operations. The sequence of operation is:

I. Read director timing time 100 generates SO/PD signal (A27, TP23).

A. Gate the contents of the addressed director buffer memory location (A29) to the director register (A25, A27, A28).

B. The director register (A25, A27, A28) and director register fanout (A17) feed the director translators (A25, C09).

1. Generate DATDIR (A25, TP23) and RDD (A25).

a. Generate read or write decode (RDWTD, B17).

Generate RWCDE signal (B19).

Enable loading of register file counter (B24).

If director bit 10 is a one, generate CRFAD signal (B19).

Enable clearing the register file counter (B24).

II. Director execution sequence generates DDE (A28, TP8).

A. If CRFAD signal (B19) is a one, clear the register file address counter (B24).

B. Director bit 15 controls the selecting of the byte count (input length) when DDE occurs.

1. Read or write decode has the RWCDE signal (B19) at a one and if bit 15 is a one, generate a load the register file counter pulse (B24, TP11) which gates director bits 28 through 31 into the register file counter (B24).

a. The outputs from the counter (B24) feed through the register file address mux (B25) and address a register file location (B25).

The location (B25) is read out on the register file bus to data path 2 (A07 through A12) where it is inverted (signals BYI00 through BY15) and feeds the byte counter mux (B21).

2. Signals BCT4 and BYIE (B19) control the gating on the byte counter mux (B21). BCT4 is a constant zero during this operation. If BYIE is a one, director bits 16 through 31 (B21) go through the mux (B21). If it is a zero, the BYIXX signals go through.

a. If bit 15 is a one, set the load byte counter from RF flip-flop (B19, TP24).

Generate BYIE (B19) making the mux selection (B21).

Enable LBCRF (B19).

When DDE (B19) drops, generate LBCRF (B19) which loads the byte counter (B21).

b. If bit 15 is a zero, generate LBCD (B19).

Load the byte counter (B21) with the count from the director.

C. Clock the read flip-flop (B17, TP7). It will set.

1. Generate read signal (B17).

a. Generate RIP (B13).

Enable sector length error circuit (A06) to start operating.

Enable bit 0 of status word 0 (A19).

Generate RIP2 (B15) which enables DTEC (B12) and generates RIP2 and RIP2A (B13). These enable the checkword generator (B03 through B08) to begin operating.

2. Enable TOGUL signal (B17).

3. Enable load buffer and lost data logic (B17).

4. Enable read done flip-flop (A06, TP8).

D. Clock the read to register file flip-flop (B19, TP2). If director bit 10 is a one, it will set and enable data transfer to the register file.

1. Generate DATRF (B19, TP15) and RTRF (B19).

a. Enable increment of the register file counter (B24).

b. Enable generation of RFWEU and RFWEL (B24).

c. Drop RFME (B24, TP9) which stops referencing the addressed register file location (B25).

d. Enable the output of the shift network catching register (B09) onto the register file bus (A07 through A12) and to the addressed register file location (B25).

2. Drop the BUFOP signal (B19) to a 0. The BUFOP signal (B19), when a 1, enables the data buffer control (B17 and lost data flip-flop (B17). When it is a 0, it disables these same elements.

E. Preset or preclear the byte control flip-flop (B19) according to the state of director bit 9. If bit 9 is a 1, the flip-flop will set. If bit 9 is a 0, it will clear.

III. The read operation set up by the previous director (RAP) is still active. The selected unit is reading data, the read select tag (B15) is still up, the read clock is generating shift clocks (B16), data is being clocked into the shift register(B09) and checkword generator (B03 through B08), and the bit counter (B16) is running in step with the data field.

IV. If director bit 10 was set initially, a read to the register file operation is occurring (refer to Figure 3-118).

A. When the bit counter (B16) reaches a count of one [DCT1 (B16) is a 1] and SRCL1 (B16) occurs, clock the upper/lower byte flip-flop (B19, TP10). It will go to the same state as the byte control flip-flop (B19) generating either DLOW or DUP (B19).

1. DUP enables generation of RFWEU (B24).

2. DLOW enables generation of RFWEL (B24), and increments the register file counter (B24).

B. When the bit counter (B16) reaches a count of three [signal DCT3 (B16) is a 1] and SRCL2 (B16) occurs, transfer the shift register contents into the shift register catching register (B09).

C. When the bit counter (B16) reaches a count of four [DCT4 (B16) is a 1] and SRCL1 occurs, the following things happen.

1. Clock the byte ready flip-flop (B17, TP9). It will set, generating RTRFP signal (B17).

a. If DUP (B19) is a 1, generate RFWEU and RFME (B24 which transfer the contents of the shift register catching register (B09) into the upper portion of the addressed register file location (B25).

b. If DLOW (B19 is a 1, generate RFWEL and RFME (B24) which transfer the contents of the shift register catching register (B09) into the lower portion of the addressed register file location (B25).

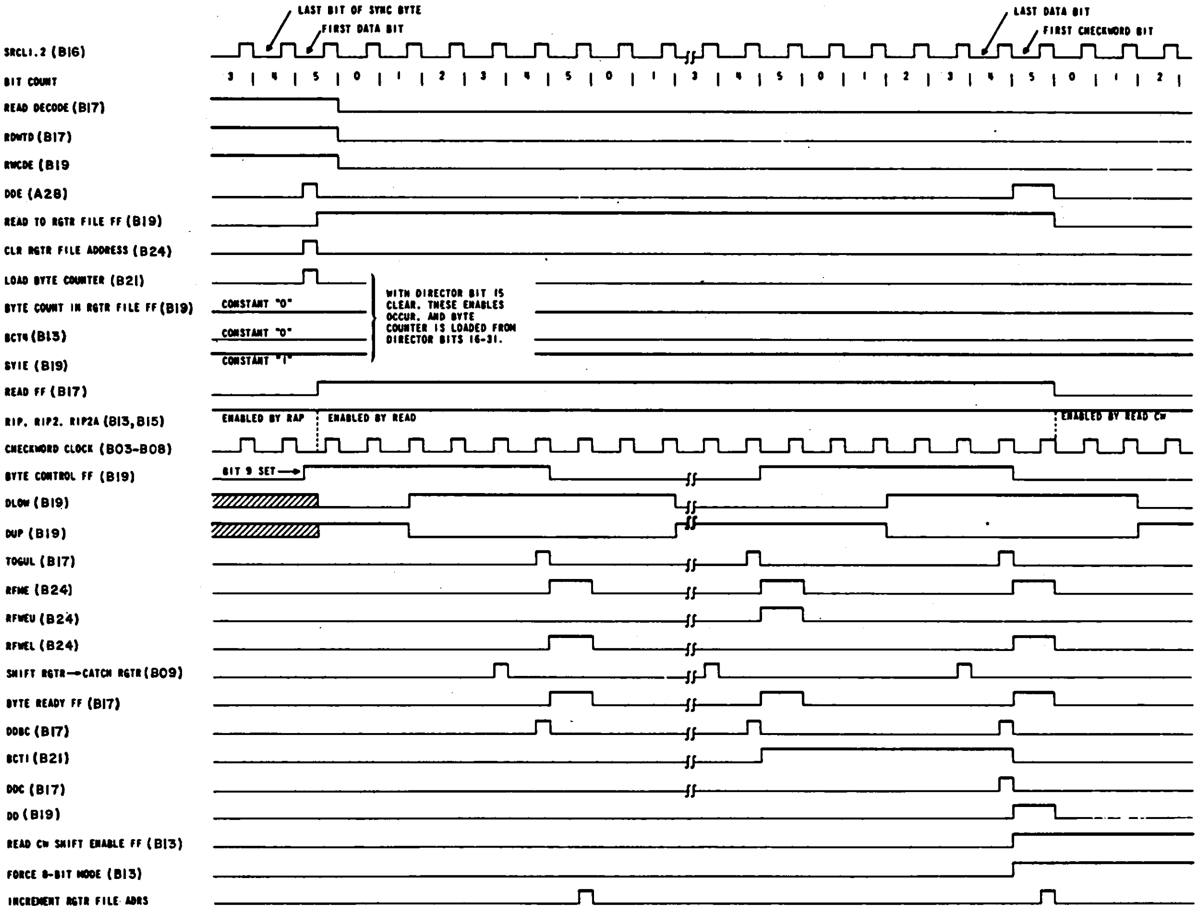


Figure 3-118. Read to Register File Director Execution

2. Generate TOGUL (B17, TP18).

   a. Clock the byte control flip-flop (B19). It will go to its opposite state.

3. Generate DDBC (B17, TP15).

   a. Decrement the count in the byte counter (B21) by one count.

4. If the byte counter contains a count of one, [BCT1 (B21) is a 1] prior to step 3, generate DDC (B17) which causes the following steps to occur.

   a. Generate DD signal (B19).

      Enable execution of next director (A28).

   b. If the next director is a support director, generate CLRD, clear the read (B17) and read to RGTR file (B19) flip-flops ending this read operation.

      Drop READ (B17) which drops RIP (B13).

         Turn off sector length error circuit (A06).

         Drop bit 0 of status word 0.

         Drop RIP2 (B15) and generate RIPP (B15). RIP 2 turns off the checkword generator (B03 through B08).
         RIP drops read tag (B15) and stops shift register clock generation (B16).

   c. Clock the force 8-bit mode flip-flop (B13, TP1), setting it.

      Generate F8BM (B13).

         Bring up 8BM signal (B12) enabling the shift register (B09) and the bit counter (B16) to operate in 8-bit mode.

D. When the bit counter (B16) reaches a count of five [DCT5 (B16) is a 1] , the following steps will occur.

1. Clear the byte ready flip-flop (B17, TP9).

   a. Drop RTRFP (B17). This disables loading the register file (B24).

2. If DLOW (B19) is a 1, generate an increment the register file counter signal (B24, TP5).

   a. Increase the count in the register file counter (B24) by one, selecting the next sequential register file location (B25).

E. If byte counter was not at a count of one when step IV C. 4 occurred, return to step IV. A. and continue the read.

V. If director bit 10 was clear initially, a read using the data buffer is occurring (refer to Figure 3-119).

A. When the bit counter (B16) reaches a count of one, [DCT1 (B16 is a 1] and SRCL1 (B16) occurs, clock the U/L byte flip-flop (B19) TP10). It will go to the same state as the byte control flip-flop (B19) generating either DLOW or DUP (B19).

1. The DUP and DLOW signals control the loading of data into rank 1 of the data buffer (B10).

a. If DUP is a 1, the 6- or 8-bit byte of data will be routed to rank 1 upper.

b. If DLOW is a 1, the 6- or 8-bit byte of data will be routed to rank 1 lower.

B. When the bit counter (B16) reaches a count of three, [DCT3 (B16) is a 1] and SRCL1 or 2 (B16) occur, the following things will happen.

1. Transfer the shift register contents into the shift register catching register (B09).

a. The output of the shift register catching register feeds through data path 2 (A07 through A12) to the data buffer (B10).

2. Clock the lost data flip-flop (B17, TP32). If BRQ signal (B28) is a 1, the flip-flop will set.

a. Generate COMST and fault (A05, TP22, and TP17).

Enable bit 3 of status word 0 (A20).

Block director loading sequence (B26).

Clear enable director execution flip-flop (B27).

b. Enable bit 6 of status word 0 (A21).

C. When the bit counter (B16) reaches a count of four, [DCT4 (B16) is a 1] and SRCL1 occurs, the following things happen.

1. Generate TOGUL (B17, TP18).

a. Clock the byte control flip-flop (B19). It will go to its opposite state.

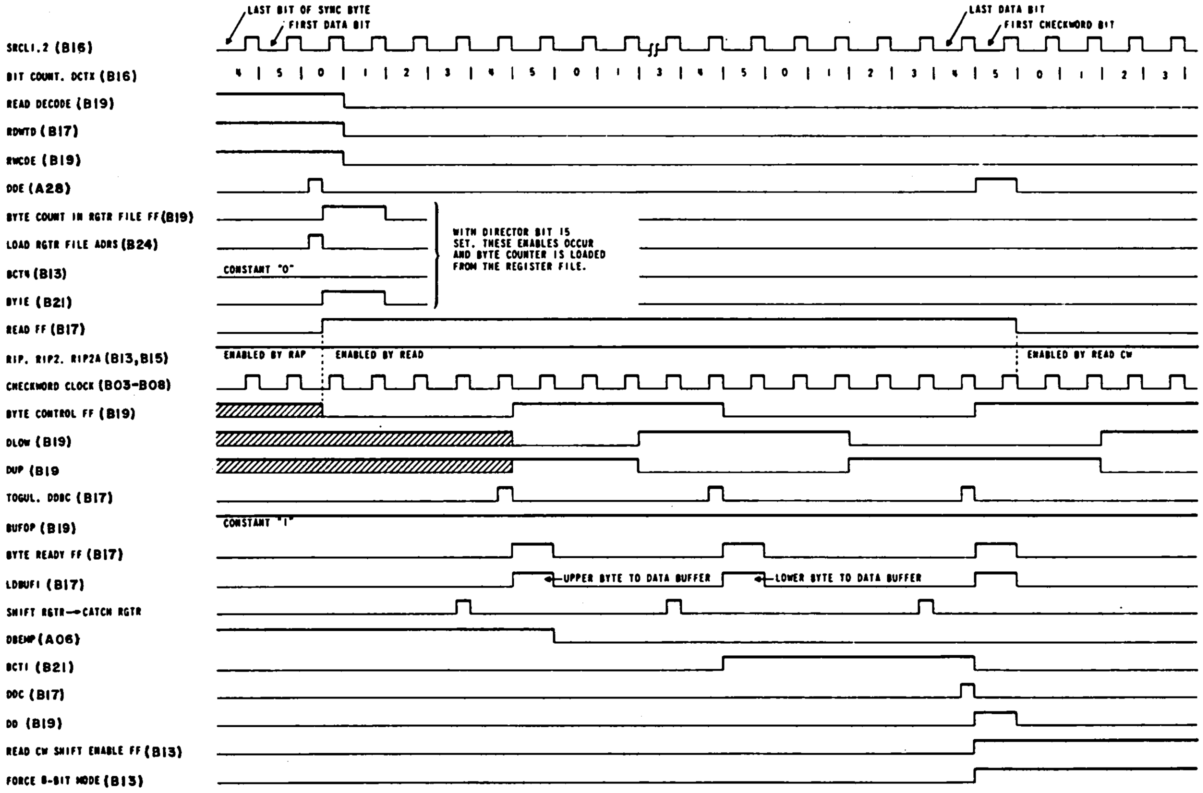


Figure 3-119. Read to Data Buffer Director Execution

2. Generate DDBC (B17, TP15).

   a. Decrement the count in the byte counter (B21) by one count.

3. Clock the byte ready flip-flop (B16, TP9). It will set.

   a. Generate LDBUF1 (B17, TP6).

      If DUP (B19) is a 1, load the byte of data from the shift register catching register (B09) into the upper portion of rank 1 of the data buffer (B10) and generate LDBUF (B10).

         Set RK1 loading flip-flop (A06) and drop data buffer empty (DBEMP, A06, TP20).

            Drop bit 1 of status word 2 (A19).

      If DLOW (B19) is a 1, load the byte of data from the shift register catching register (B09) into the lower portion of rank 1 of the data buffer (B10) and set the rank 1 full flip-flop (B10, TP23).

         Drop DBRQD (B10). Clear RK1 loading flip-flop (A06) and hold DBEMP (A06, TP20) at zero. Enable BRQ (B28). If the input look ahead flip-flop (B28, TP24) is clear, BRQ comes up a one enabling lost data (B17).

      The 16-bit byte of data from rank 1 will move through the data buffer (B10) to rank 4 as room for it becomes available.

4. If the byte counter contains a count of one [BCT1 (B21) is a 1] prior to step V. C. 2. generate DDC signal (B17) which causes the following sequence.

   a. Generate DD signal (B19).

      Enable execution of next director (A28).

      Clock the read done flip-flop (A06, TP8). It will set.

         Enable EOR OUT and RDONE signals (A06).

   b. If the next director is support director generator CLRD (A28), clear the read flip-flop (B17) ending this read operation.

Drop READ (B17) which drops RIP (B13).

Turn off sector length error circuit (A06).

Drop bit 0 of status word 0.

Drop RIP2 (B15) and generate RIPP (B15). RIP 2 turns off the checkword generator (B03 through B08).
RIPP drops read tag (B15) and stops shift register clock generation (B16).

c. Clock the force 8-bit mode flip-flop (B13, TP1), setting it.

Generate F8BM (B13).

Bring up 8BM signal (B12) enabling the shift register (B09) and the bit counter (B16) to operate in 8-bit mode.

D. When the bit counter (B16) reaches a count of five [DCT5 (B16) is a 1] , the following steps will occur.

1. Clear the byte ready flip-flop (B17, TP9).

a. Drop LDBUF1 (B17).

E. If the byte counter was not at a count of one when step V. C. 4. occurred, return to step V. A. and continue the read.

VI. The previous address director has set up the unloading of the data buffer. Signal DFD (A05, TP5) is a 1 enabling the data transfer from the disk. Signal DTMEM (A05, TP1) decodes whether the data is to be sent to the system coupler or to the processor.

A. If DTMEM (A05, TP1) is a 0, the data buffer will be emptied to the system coupler as follows:

1. A 16-bit byte of data from the disk arrives in rank 4 of the data buffer (B10). The rank 4 full flip-flop (B10, TP22) sets.

a. If CORPLY signal (B10) from the coupler is a 0, generate DBSS (B10, TP31).

Clock the coupler output request flip-flop (A06, TP10). It will set sending COREQ (A06) to the coupler.

b. The 16-bit byte of data goes directly from rank 4 to the coupler.

2. The coupler accepts the byte of data and returns CORPLY (A06, TP18, and B10).

   a. Block generation of DBSS (B10).

   b. Clock the coupler output request flip-flop (A06, TP10). It will clear.

      Drop the COREQ signal (A06) to the coupler.

      Generate coupler reply (CORP, A06) signal.

         Generate CLDRY signal (B17) which clocks the rank 4 full flip-flop (B10, TP22) clearing it and enabling a rank 3 to 4 transfer.

3. When the coupler senses the dropping of the COREQ signal (A06), it will drop the CORPLY signal (A06, TP18, and B10).

   a. Enable generation of DBSS signal (B10).

4. If data buffer empty signal (A06, TP20) did not occur when the last 16-bit byte was transferred to the coupler, return to step VI. A. 1. and continue the data transfer. If it did occur, continue below.

   a. The read done flip-flop (A06, TP8) will have been set by the read from disk sequence. Generate RDONE (A06).

      Clear the address execute flip-flop (B17, TP5) preventing any further data transfer.

   b. If EORBIT (A05, TP7) is present, selected by preceding address director, generate end of record signal (EOROUT, A06) and send it to the coupler.

      The coupler terminates its data transfer operation.

B. If DTMEM signal (A05, TP1) is a 1, the data buffer will be emptied to the subsystem memory as follows:

   1. A 16-bit byte of data enters rank 4 of the data buffer (B10). The rank 4 full flip-flop (B10, TP22) sets.

      a. Generate DBDRDY (B10) which brings buffer ready signal (BRY, B28, TP22) up to a 1.

         Block read lost data (B17).

         Generate data buffer output request signal (DBOPRQ, B17, TP3).

2. If fault (A05, TP17) is a 1 and the scanner (B26) arrives at the data request position, generate enable data request (ENDRQ, B26, TP29).

a. Set the data request 2 flip-flop (B28, TP19) generating DRQT (B28).

Block the scanner at the data request position (B26).

3. The processor request 3 flip-flop (B26, TP14) is set and will stay set throughout this sequence, inhibiting MEMGO (B26).

a. Enable the output of the controller memory address mux (A03, A04) through the memory address mux (A13 through A16) to memory.

b. Enable the memory bus output (A19 through A24) through the memory data mux (A13 through A16) to memory.

4. When the scanner (B26) locks up on the data position, the following things will happen.

a. Generate DATARQ (B26) which in turn generates SCAND (B26, TP16).

Enable the output of the data address register (A03, A04) through the controller address mux (A03, A04).

Generate DBMB (B17).

Enable the output from rank 4 of the data buffer (B10) onto the memory bus (A19 through A24).

b. Bring up request gate inverter (B26). If memory is not busy, start memory request timing (B26).

5. Memory request timing generates MRQ1 (B26, TP23).

a. Clear input request flip-flop (B27, TP31).

b. Clear output request flip-flop (B27, TP28).

c. Clear director request 3 flip-flop (B27, TP29).

6. Memory request timing generates MREQT (B26).

a. Clock the memory request in progress flip-flop (B26, TP27). It will set enabling reply timing.

b. Generate DATRQ (B27).

Clock the output request flip-flop (B27, TP28). It will set generating O/PRQ (B27) and enabling memory output reply (B27).

Generate STORE (B26, TP5) and send it to memory.

Block the data from memory inputs to the memory bus (A13 through A16).

Clear the data request look-ahead flip-flops (B28, TP21, and TP24).

c. Start the memory busy delay line (B28).

d. Clock the internal busy flip-flop (B28, TP1). It will clear.

Generate BUSY (B28), disabling memory request timing (B26).

7. Memory request timing generates MREQ (B26, TP20) and sends it to memory starting the storage reference cycle.

8. The memory returns MBSY1 (B28, TP2) when it goes busy.

a. Maintain BUSY (B28).

9. The memory returns MDRDY (B26) when it has accepted the byte of data.

a. Clock the reply flip-flop (B26). It will set.

Start the reply timing chain (B26).

10. The reply timing chain generates REPLY (B26).

a. Generate MO/PRY (B27, TP27).

Generate CLDRY (B17).

Clock the rank 4 full flip-flop (B10, TP22). It will clear enabling a rank 3 to rank 4 transfer in the data buffer (B10). Drop DBDRDY (B10, TP22) which enables buffer ready signal (B28, TP22).

Generate MRLY (B17).

Clock the output data request look-ahead flip-flop (B28, TP21). If data buffer rank 3 is full, [DBR3 (B10) is a 1] and count in the word counter is not one [WCT1 (B20, TP31) is a 0] , the flip-flop will set.

Generate DWC (B14, TP28). Increment the data address register (A03, A04) by one. Decrement the count in the word counter (B20) by one.

11. The reply timing chain drops REPLY (B26) which, in turn, drops MO/PRY (B27, TP27).

    a. Clock the data request 2 flip-flop (B28, TP19). If the output data request look ahead flip-flop (B28, TP21) is set, the flip-flop remains set. If output look ahead is clear, the flip-flop clears.

        If data request 2 clears, the following things occur.

        DRQT (B28) drops releasing the scanner (B26) and disabling the data gate to request timing (B26).

        The word count will be decremented to zero [WTO (B20, TP6)]. Disable or drop DBOPRQ, DTMIP, DBMB, and MRLY signals (B17).

        When data buffer empty signal (A06, TP20) comes up, generate RDONE (A06). Clear the address execute flip-flop (B17, TP5), preventing any further data transfer.

        If data request 2 remains set, the scanner (B26) remains locked on the data position, buffer ready (B28, TP22) remains a 0, and the data transfer stays active.

        When BUSY (B28) drops, restart the request timing chain (B26) to transfer the next byte of data.

VII. End of read normal director.

Read Skip Director Execution

This director is identical to the read normal director except that no data is transferred. Although no data is transferred, a checkword is generated. This director can thus be used to verify a sector of data without requiring the HLP to retransmit the data as is done during a compare normal operation.

The sequence of operation is the same as that for a read normal director except that the LDBUF1 signal is inhibited. This prevents any data from being placed in the data buffer.

Read Checkword Director Execution

This director causes the hardware to treat the next 32 bits from the disk as a checkword and to compare it with the checkword generated by the hardware. The results of the compare are saved in a flip-flop for use by subsequent directors. The contents of the checkword generator will be retained until execution of the next RAP director. If bit 15 is set, the controller will generate a head advance pulse to the selected drive, providing the function tag specifying control select has been enabled via an enable disk function director. A 0.5-microsecond head advance pulse is sent to the drive after the completion of the read checkword director and causes the head counter to be incremented by one. This allows rapid switching of heads when reading from one track to the next. The sequence of operation is illustrated in Figure 3-120 and described below.

I. Read director timing time 100 generates SO/PD signal (A27, TP23).

  A. Gate the contents of the addressed director buffer memory location (A29) to the director register (A25, A27, A28).

  B. The director register (A25, A27, A28) and director register fanout (A17) feed the director translators (A25, B23).

    1. Generate DATDIR (A25, TP23) and RDD (A25).

      a. Generate BCT4 signal (B13).

        Force count of 4 in byte counter mux (B21).

        Generate BCT4 and BYIE signals (B19).

          Gate count of 4 through byte counter mux (B21) to byte counter (B21).

II. Director execution sequence generates DDE (A28, TP8).

  A. Generate BCLCW signal (B13).

    1. Generate LBCD signal (B19) which loads the byte counter (B21).

    2. Clock the director bit 15 holding flip-flop (B12). If director bit 15 is a 1, it will set.

  B. Clock the read checkword flip-flop (B13, TP2). It will set.

    1. Enable CLRDC and RCWB1 signals (B13).

    2. Generate RIP signal (B13).

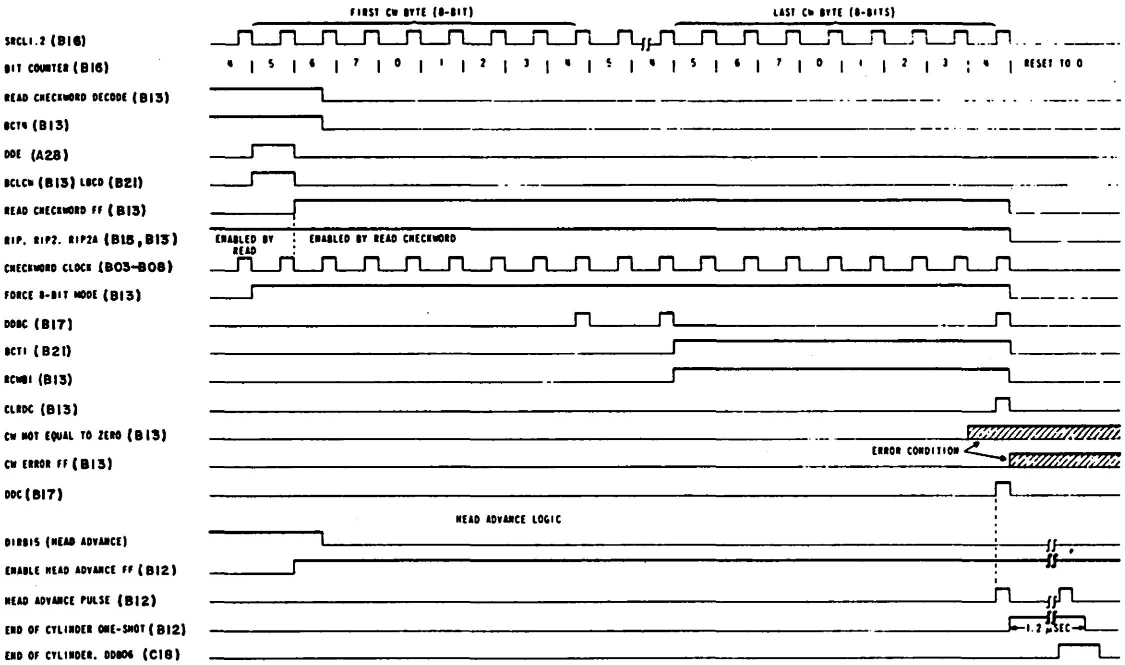


Figure 3-120. Read Checkword Director Execution

a. Enable sector length error circuit (A06) to start operating.

b. Generate RIP 2 signal (B15) which enables the DTEC signal (B12) and generates RIP2 and RIP2A signals (B13) These enable the checkword generator (B03 through B08) to begin operating.

III. The read operation from the preceding director, read or compare, is still active. The selected unit is reading data, the read select tag (B15) is still up, the read clock is generating shift clocks (B16), data is being clocked into the shift register (B09) and checkword generator (B03 through B08), and the bit counter (B16) is running in step with the data field.

IV. Each time the bit counter reaches a count of four, DCT4 (B16) and SRCL1 (B16) occur, generating DDBC (B17, TP15).

A. Decrement the count in the byte counter (B21) by one count.

V. When the byte counter is decremented to a count of one, generate BCT1 (B21, TP31) and RCWB1 (B13).

A. When the bit counter reaches the next count of three, generate DCT3 (B16), clock the checkword generator initiate flip-flop (B12, TP1). It will clear.

1. Drop CWCE (B12) and CWGCE (B13).

a. Turn off the checkword generator (B03 through B08).

VI. With the byte counter (B21) at a count of one, when the bit counter reaches a count of four [DCT4 (B16) is a one], generate SRCL1 (C27). The following steps will occur.

A. Generate CLRDC (B13).

1. Clock the checkword error flip-flop (B13, TP6). If any position in the checkword generator (B03 through B08:) is a 1, the checkword error signal (C17, TP13) will be a 1, and the flip-flop will set.

a. Generate COMST signal (A05, TP22).

Enable bit 3 of status word 0 (A20).

b. Enable bit 2 of status word 0 (A19).

c. Enable conditional branch condition met logic (B22).

2. If the director bit 15 holding flip-flop (B12) is set, generate HAP signal (B12).

   a. Generate HAP1 signal (B15, TP18).

      Bring up bit 3 control tag (B15, TP22), which is head advance to selected drive (C02).

      Hold a clear on the HAP generate flip-flop (B12).

3. The trailing edge of the HAP generate pulse (B12) fires the 1.2-microsecond end of cylinder check one-shot (B12).

   a. If end of cylinder [signal DDB06 (C18, TP29)] comes up before the end of cylinder check pulse drops, set the HAP generate flip-flop (B12) when the check pulse drops.

      NOTE

      At the end of a cylinder, two head advance pulses are required.

      Generate HAP signal (B12) which, in turn, generates HAP1 signal (B15, TP18).

         Bring up bit 3 control tag (B15, TP22) which is head advance to the selected drive (C02).

         Clear the HAP generate flip-flop (B12).

B. Clock the force 8-bit mode flip-flop (B13, TP1). It will clear.

   1. Drop F8BM signal (B13).

      a. Shift register and bit counter return to selected mode of operation.

C. Generate DDC signal (B17).

   1. Generate DD signal (B19).

      a. Enable execution of next director (A28).

VII. If the next director is a support director, generate CLRD (A28). Clear the read checkword flip-flop (B13, TP2) ending the read checkword operation.

A. Drop the RIP signal (B13).

1. Turn off sector length error circuit (A06).

2. Drop RIP2 (B15) and generate RIPP (B15). RIP 2 turns off the checkword generator (B03 through B08). RIPP drops read tag (B15) and stops shift register clock generation (B16).

VIII. Director execution continues with the next director.

Read Address Pattern Director

This director establishes bit and byte synchronization with serial information received from the selected disk storage unit. It is actually a compare operation which compares data from the disk with patterns previously stored in the register file. With bit 15 of the director set, the disk data is compared with the pattern stored in bits 08 through 15 of the specified register file location. With bit 15 clear, the disk data is compared with the pattern stored in bits 00 through 07 of the specified register file location. The register file location is specified by bits 28 through 31. Other bit positions and their use in the director are described as follows:

- Bit 16 - This bit is always used in the first RAP director to turn on the read gate in the selected disk storage unit. The control select line to the unit must always be selected to allow head switching during a data operation.
- Bit 21 - This bit determines the point in the bit stream at which a compare is made. If this bit is set, a compare is made at every bit position (RAP anyplace). If this bit is clear, a compare is made at the immediately-following full byte position (RAP specific).
- Bit 22 - This bit is used in a RAP anyplace director. It will set the bit counter to the position indicated by bits 25 through 27 of the director.
- Bit 23 - This bit determines the type of compare. When the bit is clear, the compare is satisfied when the contents of the shift register exactly match the specified pattern. When the bit is set, the compare is satisfied when the contents of the shift register do not match the specified pattern.
- Bit 24 - This bit causes the logic to wait at least one full byte time before attempting a compare. It is used in the first RAP in a series to ensure that the shift register is emptied of any residue which might produce a false compare.

Bit and byte synchronization is established by reading the address pattern from the disk drive and making compares at the proper time. The address pattern consists of a sync pattern which is 300 zero bits, and a sync byte which is 110011. Three RAP directors are required to definitely locate the sync byte. The first is a RAP anyplace which looks for all zeros. The second is a RAP anyplace with bit 23 set. This director looks for the first nonzero bit. If no errors have occurred, this should be the first bit of the sync byte. The third director is a RAP specific which looks for the sync byte. The sync byte must be found at the next byte time. If not, the RAP abort flip-flop will set. The RAP abort flip-flop also sets if a sector mark pulse is encountered during any RAP sequence.

Prior to starting a RAP sequence, the necessary patterns (000000 and 110011) must be placed in the register file. The RAP directors will specify that register file location and bit ~~23~~ 15 will specify upper or lower byte position. The control select line to the selected disk storage unit must also be enabled so that the read gate can be turned on by the first RAP director.

Following is a complete sequence of operations for three RAP directors. The sequence assumes that the necessary patterns have been loaded and that the control select line is enabled. Figure 3-121 illustrates the sequence.

I. Read timing chain generates SO/PD (A27, TP23).

   A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and director decode logic on A25.

II. Director is decoded on A25.

   A. Generate DATDIR (A25, TP23). This signal causes data director control logic to generate DDE.

   B. Generate RDD (A25). This is a partial decode for read directors.

   C. RDD, DIRC03, and DIRC04 generate RAPDE (B18). This is the read address pattern decode.

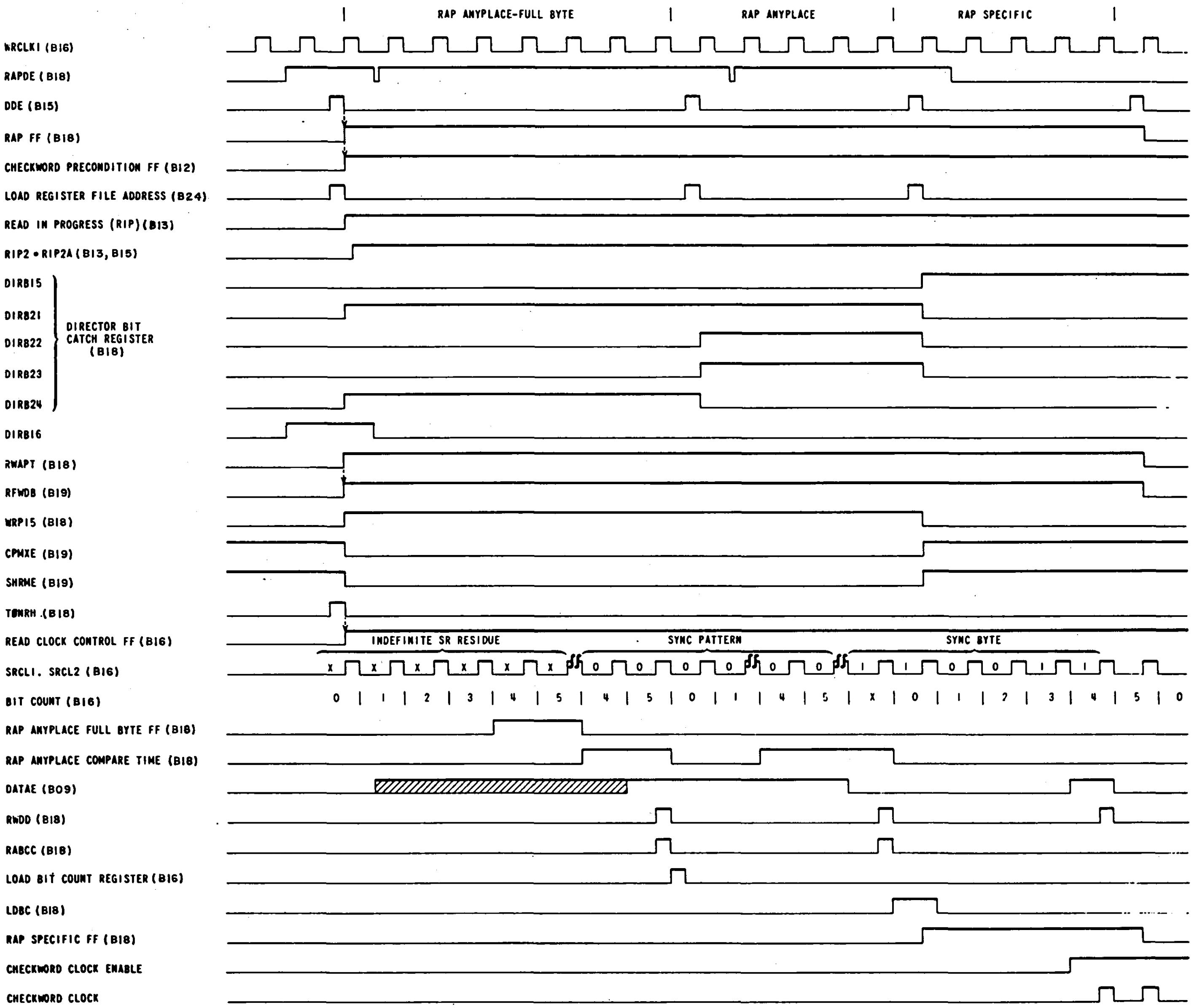


Figure 3-121. Read Address Pattern Sequence

III. Data director control logic generates DDE (A28, TP8).

A. Set checkword precondition flip-flop (B12, TP2).

B. Load director bits 15 and 21 through 24 into the catch register on B18. Since this is first RAP director, the bits will be defined as follows:

- Bit 15 = 0 - This assumes the zeros pattern was stored in bits 00 through 07 of the register file
- Bit 21 = 1 - RAP anyplace
- Bit 22 = 0 - Do not set bit counter
- Bit 23 = 0 - Check for actual pattern
- Bit 24 = 1 - Wait one full byte

C. Load director bits 28 through 31 into the register file address counter on B24. This enables the register file address through register file address multiplexer 2 on A29. Since the register file is normally in a read mode, the addressed location is placed on the register file bus (A29).

D. Set read address pattern flip-flop (B18, TP1).

1. Generate RAP (B18). This brings up the read in progress bit (RIP) on B13.

   a. Enable sector length error circuit on A06.

   b. Generate RIP2 on B15. This enables the read data gate on B12 and arms the RIPP one-shot.

   c. Generate RIP2 and RIP2A on B13. These signals enable the checkword generator (B03 through B08) for a read operation.

2. Generate RWAPT (B18). This generates RFWDB (B19) which enables the contents of the register file bus onto the write data bus (A07 through A12).

3. Since bit 15 is a 0, generate WRP15 (B18). This drops CPMXE and SHRME on B19. This enables bits 00 through 07 of the write data bus into the compare mux and the shift register mux on B09.

4. Since this is the first RAP director, bit 16 must be set. This generates TONRH (B18) which sets the applicable control flip-flop on B15 (TP25) and sets the read clock control flip-flop (B16, TP31). Since the control select line to the disk storage unit was previously enabled, the read head is turned on and the unit sends RDCLK (C21) and DATAIN (C21) signals as soon as data is detected.

   a. The RDCLK generates SRCL1 and SRCL2 on B16.

   b. The bit counter begins to count cell times (B16).

   c. The read data catch flip-flop (B12, TP13) reflects the condition of the DATAIN signal. The RDCFF signal is gated into the shift register on B09 by the SRCL2 clock pulse.

E. Start read timing chain for next director. The director will be read up and decoded but will not be executed until the first director generates a data done (DD).

IV. Generate SRCL1 and DCT3 (B16). This is the shift clock and a bit count of three.

A. Set the RAP anyplace full byte flip-flop (B18, TP22).

V. Generate SRCL1 and DCT3 (B16).

A. Set the RAP anyplace compare time flip-flop (B18, TP21). This flip-flop enables the compare gate. At this time, the shift register has shifted one full byte and should be full of zeros from the sync pattern. These are compared to the zeros pattern from bits 00 through 07 of the register file. If the compare is successful, the compare network on B16 generates DATAE.

B. Clear the RAP anyplace full byte flip-flop.

VI. Generate SRCL1 (B16).

A. If DATAE is present, generate RWDD (B18, TP20) and RABCC (B18). Set the clear RAP anyplace flip-flop (B18, TP28).

1. RWDD sets the data done flip-flops to generate DD (B19). This generates the DDE (A28, TP8) for the next director.

2. RABCC clears the bit counter on B16.

3. Clear the RAP anyplace flip-flop.

VII. Generate SRCL1 (B16).

A. Preclear the clear RAP anyplace flip-flop (B18).

VIII. Generate DDE (A28, TP8).

A. Load director bits 15 and 21 through 24 into the catch register on B18. Since this is the second RAP director, the bits will be defined as follows:

- Bit 15 = 0 - Zeros pattern in bits 00 through 07 of register file
- Bit 21 = 1 - RAP anyplace
- Bit 22 = 1 - Load bit counter to count specified by bits 25 through 27 if RAP is successful
- Bit 23 = 1 - Look for data that does not match the pattern
- Bit 24 = 0 - Do not wait a full byte time

B. Load bits 25 through 27 into bit count catch register on B16.

C. The read address pattern flip-flop (B18, TP1) remains set from first RAP director. This flip-flop will clear if a DDE is generated and a read address pattern decode is not present. See steps 3C1 through 3C4 for the enables generated by the RAP flip-flop.

D. Start read timing chain for next director.

IX. Generate SRCL1 and DCT3 (B16).

A. Set the RAP anyplace compare time flip-flop (B18, TP21). This flip-flop enables the compare gate. At this time, the shift register will still contain all zeros from the sync pattern and DATAE will still be present. The shift clock (SRCL1) continues shifting data into the shift register until the first bit on the sync byte (110011) enters the shift register. This will cause DATAE to drop.

X. Generate SRCL1 (B16).

A. If DATAE has dropped, generate RWDD (B18, TP20) and RABCC (B18). Set the clear RAP anyplace flip-flop (B18, TP28).

1. RWDD sets the data done flip-flops to generate DD (B19). This generates the DDE (A28, TP8) for the next director.
2. RABCC clears the bit counter on B16.

3. Set the load bit counter flip-flop (B18, TP29). This generates LDBC which transfers the bit count from the holding register to the bit counter on B16.

4. Set the clear RAP anyplace flip-flop (B18, TP28). This clears the RAP anyplace compare time flip-flop.

B. Second bit of sync byte enters shift register.

XI. Generate SRCL1 (B16).

A. Preclear the clear RAP anyplace flip-flop (B18).

B. Clear the load bit counter flip-flop (B18). Bit counter now contains a count of one.

C. Third bit of sync byte enters shift register.

XII. Generate DDE (A28, TP8).

A. Load director bits 15 and 21 through 24 into the catch register on B18. Since this is the third RAP director, the bits will be defined as follows:

- Bit 15 = 1 - Sync pattern is in bits 08 through 15 of register file
- Bit 21 = 0 - RAP specific
- Bit 22 = 0 - Do not set bit counter
- Bit 23 = 0 - Check for actual pattern
- Bit 24 = 0 - Do not wait a full byte time

B. The read address pattern flip-flop (B18, TP1) remains set from the first RAP director. The enables are the same as for steps 3C1 through 3C4 except that register file bits 08 through 15 are enabled into the compare logic on B09.

C. Start read timing chain for next director.

D. Set the RAP specific flip-flop (B18, TP24).

1. Generate RAPS (B18). This enables the set input gate of the checkword clock enable flip-flop on B12.

XIII. Generate SRCL1 and DCT3 (B16).

A. Set checkword clock enable flip-flop (B12, TP1).

B. On trailing edge of SRCL1, last bit of sync byte enters shift register and bit counter increments to DCT4.

XIV. Generate DCT4 (B16).

A. If DATAE is present, generate RWDD (B18, TP20). This will generate DD from B19 to terminate the RAP sequence and generate the DDE for the next director.

B. If DATAE is not present, set the RAP abort flip-flop (B18, TP28). This will generate RAPAB which will clear the checkword precondition flip-flop and checkword clock enable flip-flop on B12. RAPAB also becomes a status bit on A20 and sets the composite status bit on A05. It generates the FAULT signal on A05 which stops the memory scanner on B26 and clears the director execution flip-flop on B27 to stop director execution.

XV. The next DDE will clear both the read address pattern flip-flop and the RAP specific flip-flop on B18.

Write Normal Director

This director transfers data to the disk from the data port designated by the associated address director. The I/O length is determined by bits 16 through 31 or is obtained from the register file location designated by bits 28 through 31 if bit 15 is set. Director bit 9 controls byte handling. Director bit 10 specifies a write from the register file. Figures 3-122 and 3-123 illustrate the data buffer operation and register file operation, respectively.

I. Read director timing time 100 generates SO/PD signal (A27, TP23).

A. Gate the contents of the addressed director buffer memory location (A29) to the director register (A25, A27, A28). This is the write normal director.

B. The director register (A25, A27, A28) and director register fanout (A17) feed the director translators (A25, B23).

1. Generate DATDIR (A25, TP23) and WRD (A25).

a. Enable bit 4 of status word 2.

b. Generate RDWTD signal (B17) which, in turn, generates RWCDE signal (B19).

c. If director bit 10 is a 1, generate CRFAD signal (B19).

d. Generate WRDXT signal (B13).

If director bit 7 is a 1, enable the checkword precondition flip-flop (B12, TP2).

II. Director execution sequence generates DDE signal (A28, TP8).

A. If CRFAD signal (B19 is a 1, clear the register file address counter (B24).

B. Director bit 15 controls the selecting of the byte count (write length) when DDE occurs.

1. If bit 15 is a 1, generate a load the register file counter pulse (B24, TP11) which gates director bits 28 through 31 into the register file counter (B24). The outputs from the counter (B24) feed through the register file address mux (B25) and address a register file location (B25). The location (B25) is read out on the register file bus to data path 2 (A07 through A12) where it is inverted, signals BYI00 through BY15, and feeds the byte counter mux (B21).

2. Signals BCT4 and BYIE (B19) control the gating on the byte counter mux (B21). BCT4 is not generated during this operation. If BYIE is not generated, director bits 16 through 31 (B21) go through the mux (B21). If it is generated, the BYIXX signals go through.

a. If bit 15 is a 1, set the load byte counter from RF flip-flop (B19, TP24).

Generate BYIE (B19) making the mux selection (B21).

Enable LBCRF signal (B19).

When DDE (B19) drops, generate LBCRF signal (B19) which loads the byte counter (B21).

b. If bit 15 is a 0, generate LBCD signal (B19).

Load the byte counter (B21) with the count from the director.

C. Clock the write from register file flip-flop (B19, TP1). If director bit 10 is a 1, it will set. If it sets, drop BUFOP (B19) and generate RFWDB (B19) and DATRF (B19, TP15).

1. RFWDB (B19) controls the gating on the write data mux (A07 through A12). The output of the write data mux feeds the shift register mux (B09).

a. If RFWDB is not generated, the output of the data buffer (B10) is gated through the write data mux.

b. If RFWDB is generated, the register file bus (BYIXX) signals (A07 through A12) are gated through the write data mux.

2. BUFOP (B19) enables the data buffer control (B17), write done flip-flop (A06), and the lost data circuit (B17). When it is a 0, it disables these same elements.

3. DATRF (B19, TP15) enables advancing the register file counter (B24). When it is a 0, it blocks the advance.

D. Preset or preclear the byte control flip-flop (B19) according to the state of director bit 9. If bit 9 is a 1, the flip-flop will set. If bit 9 is a 0, it will clear.

E. Clock the write flip-flop (B17, TP20). It will set.

1. Enable the TOGUL signal (B17, TP18).

2. Generate WTWNX signal (B17).

a. Enable the write done flip-flop (A06).

b. Enable the SHRME signal (B19, TP13).

3. Generate WRITEF signal (B17).

a. Enable the write data enable flip-flop (B13, TP7).

b. Enable the write in progress flip-flop (B13).

c. Enable the WOWCD signal (B13).

4. Enable SRLE signal (B17, TP10).

5. Enable the write data control logic (B17).

F. Clock the checkword precondition flip-flop (B12, TP2). If director bit 7 was present initially, it will set.

1. If hardware condition register bit 11 (B12) is set, generate PSCWG signal (B12).

a. Set all flip-flops in the checkword generator (B03 through B08).

2. If hardware condition register bit 11 (B12 is clear, generate PCCWG signal (B12).

a. Clear all flip-flops in the checkword generator (B03 through B08).

III. The write operation set up by the previous director (WAP) is still active. Data is being shifted from the shift register (B09) through the write data generator (B12 and C14) to the selected drive (C01 through C08). The shift register output (B12) is feeding into the checkword generator (B03 through B08). The write oscillator (C22) is generating shift register clocks (B16) to operate the shift register (B09) and the checkword generator B03 through B08), and run the bit counter (B16) in step with the data field.

IV. If director bit 10 was 0 initially, a write operation using the data buffer as a source for data will occur (refer to Figure 3-122).

A. When the bit counter (B16) reaches a count of one [DCT1 (B16)] and SRCL1 (B16) occurs, clock the upper/lower byte flip-flop (B19, TP10). It will go to the same state as the byte control flip-flop (B19), generating either a DLOW or a DUP signal (B19).

1. The upper/lower byte flip-flop controls SHRME signal (B19, TP13) generation. If the flip-flop is clear, SHRME will be a 0; if set, it will be a 1.

a. SHRME selects the upper or lower portion of the data word to the shift register (B09). If SHRME is a 1, transfer lower.

2. DLOW enables generation of the CLDRY signal (B17).

3. DCT1 and SRCL1 will also clock the byte count in register file flip-flop (B19, TP24). If it was set previously, it will not clear.

a. Signals BCT4 and BYIE (B19) go back to their original configuration.

B. When the bit counter (B16) reaches a count of three (DCT3) and SRCL1 (B16) occurs, clock the lost data flip-flop (B17, TP32). If BRY signal (B28) is a 0, the flip-flop will set.

1. Generate COMST and FAULT signals (A05, TP22 and 17).

a. Stop director loading (B26) and execution (B27).

b. Enable status bit 3 of word 0 (A20).

2. Enable status bit 6 of word 0 (A21).

3. Generate ERCLR signal (B17).

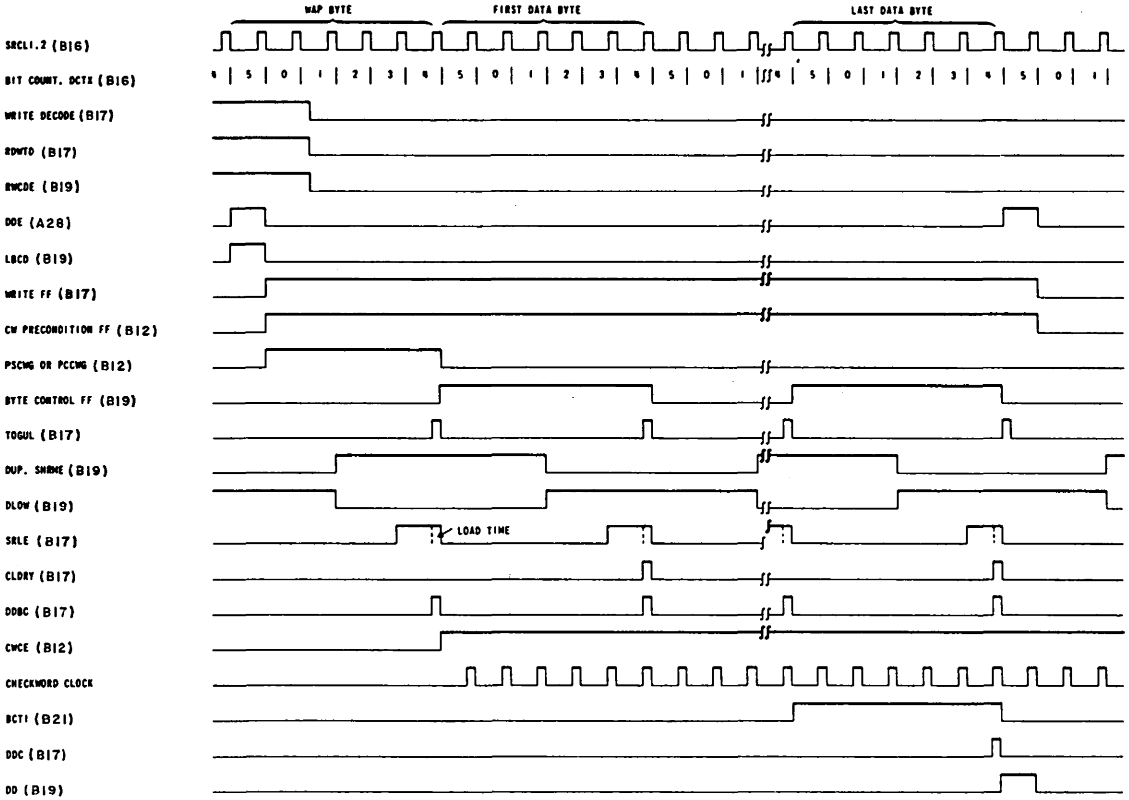


Figure 3-122. Write from Data Buffer Director Execution

a. Clear the write flip-flop (B17, TP20).

Drop WTWNX signal (B17).

Disable the control logic (B17).

Drop WRITEF signal (B17).

b. Clear the write data enable (B13, TP7), and write in progress flip-flops.

Drop WRTG signal (B13) which disables write data generation (B12).

Drop WIP signal (B13).

Disable the sector length error circuit (A06).

Drop status bit 1 in word 0 (A19).

Disable write data to the disk (B12).

Generate WIPP signal (B15, TP17) which clocks the write clock control I flip-flop (B16), clearing it, and generates TOFWH and TOFEH signals (B18).

Clear the write and erase control flip-flops in the address and control register (B15), dropping the write and erase signals to the selected drive (C01 and C02). (Erase function is not used in double density disk units.)

When the next write oscillator pulse (C22) occurs, clock the write clock control II flip-flop (B16, TP29). It will clear, stopping the generation of shift register clocks (B16).

C. When the bit counter (B16) reaches a count of four, [DCT4 (B16)], generate SRLE signal (B17, TP10).

1. When the next SRCL2 signal occurs, load a new byte of data into the shift register (B09).

D. When the bit counter (B16) reaches the next count of four [DCT4 (B16)] and SRCL1 occurs, the following things happen.

1. Generate DDBC signal (B17, TP15).

a. Decrement the count in the byte counter (B21) by one count.

2. Generate WOWCD signal (B13).

   a. Clock the checkword clock enable flip-flop (B12, TP1). If checkword precondition flip-flop was set previously, it will set.

      Drop the PSCWG or the PCCWG signal (B12).

      Generate CWCE signal (B12).

         Generate CWGCE signal (B13) allowing the checkword generator (B03 through B08) to start operating. SRCL2 signals will clock it.

3. Clock the write data enable flip-flop (B13, TP7) and write in progress flip-flop (B13). These flip-flops were set by the previous WAP director and with WRITEF signal present, they remain set. This enables the data transfer to the drive to continue.

4. Generate TOGUL signal (B17, TP18).

   a. Clock the byte control flip-flop (B19). It will go to its opposite state.

5. If DLOW signal (B19) is a 1, data buffer rank 4 is full. Generate CLDRY signal (B17).

   a. Clock the data buffer RK4 full flip-flop (B20, TP22). It will clear enabling a rank 3 to rank 4 transfer of the next data byte.

6. If the byte counter contains a count of one, [BCT1 (B21)] prior to step IV. D. 1., generate DDC signal (B17) which causes the following actions.

   a. Generate DD signal (B19).

      Enable execution of next director (A28).

      Clock the write done flip-flop (A06, TP32). It will set.

         Clear the address execute flip-flop (B17, TP5) blocking the data transfer logic.

   b. When the execution of the next director is enabled (A28) and it is not a write director [no write decode (B17)], or it is a support director [signal CLRD (A28)], clear the write flip-flop (B17, TP20) ending this write operation.

      Drop WTWNX signal (B17).

      Disable the write control logic (B17).

Drop WRITEF signal (B17) up to a 1.

Clear the checkword precondition flip-flop (B12, TP2).

c. When bit counter (B16) reaches a count of four (DCT4) and SRCL1 (B16) occurs, clock the write data enable (B13, TP7) and write in progress (B13) flip-flops. They will clear.

Drop WRTG signal (B13) which disables write data generation (B12).

Drop WIP signal (B13).

Disable the sector length error circuit (A06).

Drop status bit 1 in word 0 (A19).

Disable write data to the disk (B12).

Generate WIPP signal (B15, TP17) which clocks the write clock control I flip-flop (B16), clearing it and generating TOFWH and TOFEH signals (B18).

Clear the write and erase control flip-flops in the address and control register (B15), dropping the write and erase signals to the selected drive (C01 and C02). (Erase function is not used in double density disk units.)

When the next write oscillator pulse (C22) occurs, clock the write clock control II flip-flop (B16, TP29). It will clear, stopping the generation of shift register clock (B16).

Generate WOWCD signal (B13) which clocks the checkword clock enable flip-flop (B12, TP1). If checkword precondition flip-flop is clear, this flip-flop will clear.

Drop CWGCE signal (B13) which disables the checkword generator (B03 through B08).

7. If the byte counter was not at a count of one when step IV. D. 6. occurred, return to step IV. A. and continue the write operation.

V. If director bit 10 was a 1 initially, a write operation using the register file as a source for data will occur (refer to Figure 3-123).

A. When the bit counter (B16) reaches a count of one [DCT1 (B16)], and a SRCL1 (B16) occurs, clock the upper/lower byte flip-flop (B19, TP10). It will go to the same state as the byte control flip-flop (B19), generating either a DLOW or a DUP signal (B19).

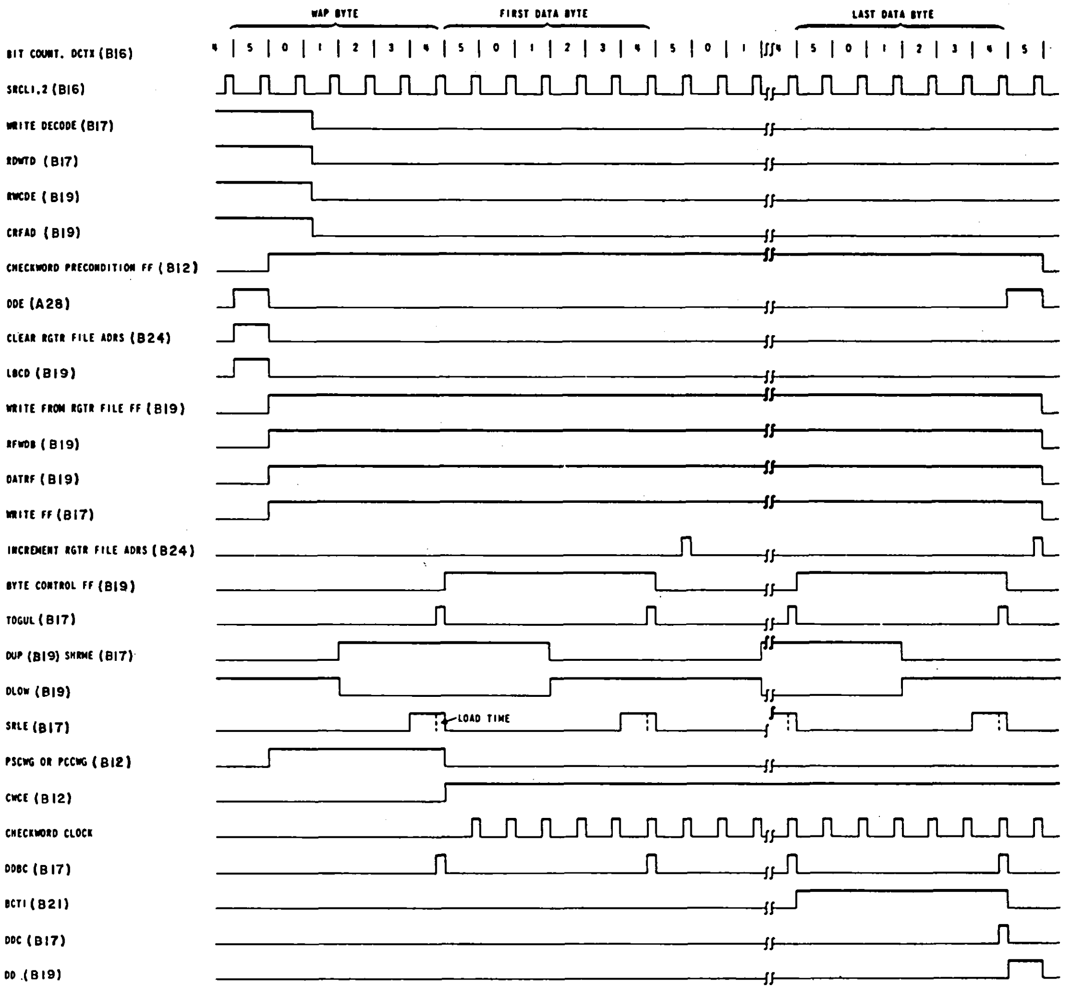


Figure 3-123. Write from Register File Director Execution

1. The upper/lower byte flip-flop controls SHRME signal (B19, TP13) generation. If the flip-flop is clear, SHRME will be a 0; if set, it will be a 1.

   a. SHRME selects the upper or lower portion of the data word to the shift register (B09). SHRME a one says transfer lower.

B. When the bit counter (B16) reaches a count of four [DCT4 (B16)], generate SRLE signal (B17, TP10).

1. When the next SRCL2 signal occurs, load a new byte of data into the shift register (B09).

C. When the bit counter (B16) reaches a count of four [DCT4 (B16)], is a 1, and SRCL1 signal occurs, the following things happen.

1. Generate DDBC signal (B17, TP15).

   a. Decrement the count in the byte counter (B21) by one count.

2. Generate TOGUL (B17, TP18) signal.

   a. Clock the byte control flip-flop (B19). It will go to its opposite state.

3. Generate WOWCD signal (B13).

   a. Clock the checkword clock enable flip-flop (B12, TP1). If checkword precondition flip-flop was set previously, it will set.

      Drop the PSCWG or the PCCWG signal (B12).

      Generate CWCE signal (B12).

         Generate CWGCE signal (B13) allowing the checkword generator (B03 through B08) to start operating.
         SRCL2 signals will clock it.

4. Clock the write data enable flip-flop (B13, TP7) and write in progress flip-flop (B13). These flip-flops were set by the previous WAP director and with WRITEF signal present, they remain set. This enables the data transfer to the drive to continue.

5. If the byte counter contains a count of one [BCT1 (B21)] prior to step V. C. 1. generate DDC signal (B17) which causes the following actions.

a. If BUFOP signal (B19) is a 1, generate CLDRY signal (B17).

Clock the rank 4 full flip-flop (B10, TP22). It will clear, enabling a rank 3 to 4 transfer of the next data byte.

b. Generate DD signal (B19).

Enable execution of next director (A28).

c. When the execution of the next director is enabled (A28) and it is not a write director [no write decode (B17)], or it is a support director [signal CLRD (A28)], clear the write flip-flop (B17), ending this write operation.

Drop WTWNX signal (B17).

Disable the control logic (B17).

Drop WRITEF signal (B17).

Clear the checkword precondition flip-flop (B12, TP2).

Clear the write from register file flip-flop (B19, TP1). Generate BUFOP and drop DATRF and RFWOB.

d. When bit counter (B16) reaches the next count of 4 (DCT4) and a SRCL1 signal (B16) occurs, clock the write data enable (B13, TP7) and write in progress (B13) flip-flops. They will clear.

Drop WRTG signal (B13) which disables write data generation (B12).

Drop WIP signal (B13).

Disable the sector length error circuit (A06).

Drop status bit 1 in word 0 (A19).

Disable write data to the disk (B12).

Generate WIPP signal (B15, TP17) which clocks the write clock control I flip-flop (B16), clearing it and generating TOFEH signals (B18).

Clear the write and erase control flip-flops in the address and control register (B15), dropping the write and erase signals to the selected drive (C01 and C02). (Erase function is not used in double density disk units.)

When the next write oscillator pulse (C22) occurs, clock the write clock control II flip-flop (B16, TP29). It will clear, stopping the generation of shift register clocks (B16).

Generate WOWCD signal (B13) which clocks the checkword clock enable flip-flop (B12, TP1). If checkword precondition flip-flop is clear, this flip-flop will clear.

Drop CWGCE signal (B13) which disables the checkword generator (B03 through B08).

D. When the bit counter (B16) reaches a count of five [DCT5 (B16)], and if DLOW (B19) is a 1, increment the register file counter (B24, TP5).

1. Increase the count in the register file counter (B24) by one, selecting the next sequential register file location (B25).

If the byte counter was not at a count of one when step V. C. 5. occurred, return to step V. A. and continue the write.

VI. Director execution continues with the next director.

Write Nx Director Execution

This director is identical to a write normal director except that it causes one 6-bit character or 8-bit byte to be written consecutively on the disk surface. It is used to write long fields of redundant information such as a field of zeros preceding a data field for synchronization purposes. If bit 10 is set, the write data is taken from location 0 of the register file. If bit 10 is clear, the write data is taken from the data buffer. Bit 9 determines if the data is taken from the upper or lower byte position of the designated source register. If bit 9 is set, the data is taken from the lower byte position (bits 08 through 15).

The sequence of operation is identical to the write normal director except that the write Nx flip-flop sets and inhibits the following signals.

1. TOGUL is inhibited so that the byte position remains constant.
2. DATRF is inhibited so that the register file counter cannot be incremented during register file operations.
3. CLDRY is inhibited so that rank 4 of the data buffer cannot be cleared during data buffer operations.

Write Checkword Director

This director will transfer the contents of the 32-bit checkword generator to the disk. This will be the checkword generated during the immediately preceding write operation. Bit 15 provides an optional head advance pulse. Bit 07 must always be set.

If another write type director does not follow the write checkword director, the following actions will occur.

1. Drop erase gate (Not used in double density disk units.)
2. Write the checkword
3. Write pad byte
4. Drop write in progress bit
5. Advance head (only if selected)

The sequence of operation is illustrated in Figure 3-124 and described below.

I. Read director timing time 100 generates SO/PD signal (A27, TP23).

   A. Gate the contents of the addressed director buffer memory location (A29) to the director register (A25, A27, A28). This is the write checkword director.

   B. The director register (A25, A27, A28) and director register fan-out (A17) feed the director translators (A25, C09).

      1. Generate DATDIR (A25, TP23) and WRD* (A25).

         a. Enable bit 4 of status word 2.

         b. Enable write pad byte flip-flop (B12, TP4).

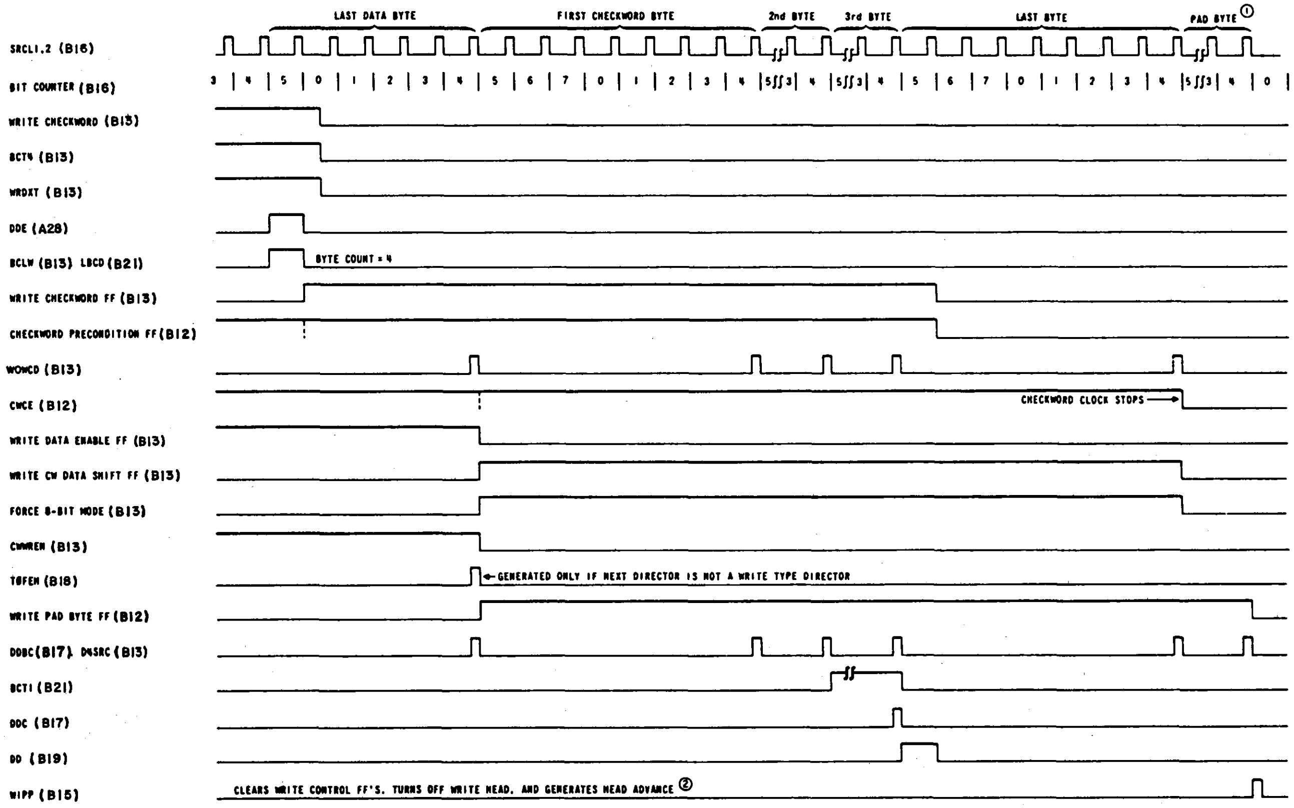
         c. Generate BCT4 signal (B13).

            Set up count of four on byte counter mux (B21).

               Gate count of four through byte counter mux (B21) to byte counter (B21).

         d. Generate WRDXT signal (B13).

            With director bit 7 a 1, enable the checkword precondition flip-flop (B12, TP2).



NOTES: ① WRITES ONE 6-BIT BYTE OF ZEROS.

② SEE READ CHECKWORD FOR HEAD ADVANCE TIMING.

Figure 3-124. Write Checkword Director Execution

II. Director execution sequence generates next DDE signal (A28, TP8).

A. Generate BCLCW signal (B13).

1. Generate LBCD signal (B19) which loads the byte counter (B21).

2. Clock the enable head advance flip-flop (B12). If director bit 15 is a 1, it will set.

B. Clock the write checkword flip-flop (B13, TP9). It will set.

1. Enable the write checkword data shift enable flip-flop (B13, TP8).

2. Generate WCWD signal (B13).

a. Block setting of clear write pad byte flip-flop (B12, TP5).

3. Enable the write in progress flip-flop (B13).

4. Enable WOWCD signal (B13).

5. .Enable CWD4S signal (B13).

C. Clock the checkword precondition flip-flop (B12, TP2). It will remain set. (It was set originally by preceding write or write NX director.)

1. Enable checkword clock enable flip-flop (B12, TP1).

III. The write operation set up by the previous director (write or write NX) is still active. Data is being shifted from the shift register (B09) through the write data generator (B12 and C14) to the selected drive (C01 through C08). The shift register output (B12) is feeding into the checkword generator (B03 through B08). The write oscillator C22 is generating shift register clocks (B16) to operate the shift register and (B09) the checkword generator (B03 through B08), and run the bit counter (B16) in step with the data field.

A. When the bit counter (B16) reaches a count of four, signal DCT4 (B16) is a 1, and a SRCL1 signal occurs, the following things happen.

1. Generate WOWCD signal B13).

a. Clock the checkword clock enable flip-flop (B12, TP1). It will remain set (it was set by the preceding director) enabling the checkword generator to continue operating

2. Clock the write data enable flip-flop (B13, TP7). It will clear, dropping the WRTG signal (B13).

a. Block the shift register to write data generator path (B12).

3. Clock the write in progress flip-flop (B13). It will stay set, enabling the data transfer to the drive to continue.

4. Clock the write checkword data shift enable flip-flop (B13, TP8). It will set.

   a. Generate F8BM signal (B13) selecting 8-bit mode (B12).

   b. Generate WCWDSE signal (B13) which enables the checkword generator to write data generator path (B12).

   c. Drop CWWREN signal (B13) which stops the generation function of the checkword generator (B03 through B08).

5. Generate CWD4S signal (B13).

   a. If a new director, which is not another write director, has replaced the write checkword director in the director output register (A25, A27, A28), WRD signal (A25) drops generating CEHCW signal (B12).

      Generate TOFEH signal (B18) which clears the control bit 4 flip-flop (B15, TP19), dropping the erase tag to the selected unit (C02).

      Clock the write pad byte flip-flop (B12, TP4). It will set, generating WPBYT signal (B12).

         Enable write in progress flip-flop (B13).

         Enable WOWCD signal (B13).

         Enable clear write pad byte flip-flop (B12, TP5).

6. Generate DDBC signal (B17, TP15).

   a. Decrement the count in the byte counter (B21) by one count.

IV. The write checkword operation is now in progress. The write oscillator (C22) is generating shift register clocks (B16). The contents of the checkword generator (B03 through B08) is being shifted out through the write data generator (B12 and C14) to the selected drive (C01 through C08). The bit counter (B16) is counting in step with the data field.

A. Each time the bit counter reaches a count of four, generate DCT4 signal (B16); when a SRCL1 signal (B16) occurs, generate DDBC signal (B17, TP15).

   1. Decrement the count in the byte counter (B21) by one count.

B. If byte counter was not at a count of one when step IV. A. 1. occurred, return to step IV. A. and continue the write.

C. If the byte counter contains a count of one [signal BCT1 (B21) is a 1], generate DDC signal (B17) which causes the following actions.

1. Generate DD signal (B19).

a. Enable execution of next director (A28).

b. Clock the write done flip-flop (A06, TP32). It will set.

Clear the address execute flip-flop, (B17, TP5) blocking the data transfer logic.

V. When the execution of the next director is enabled (A28), if it is not a write checkword director [no write checkword decode (B13)], or it is a support director [signal CLRD (A28)], clear the write checkword flip-flop (B13) ending this write operation.

A. Drop WCWD signal (B13).

1. Enable clear write pad byte flip-flop (B12, TP5).

B. Enable clearing of write checkword data shift enable flip-flop (B13, TP8).

VI. When the execution of the next director is enabled (A28), if it is not a write director [no write decode (B13)], or director bit 7 is not present (B12) or it is a support director [signal CLRD (A28)], clear the checkword precondition flip-flop (B12, TP2).

VII. When the bit counter (B16) reaches the next count of four [signal DCT4 (B16) is a 1] and SRCL1 signal occurs, the following things happen.

A. Clock the write checkword data shift enable flip-flop (B13, TP8). It will clear dropping WCWDSE signal (B13).

1. Block the checkword generator to write data generator path (B12), stopping the write data transfer to the selected drive.

2. Drop F8BM signal (B13) removing the 8-bit mode force (B12).

B. Generate WOWCD signal (B13) which clocks the checkword clock enable flip-flop (B12, TP1). If checkword precondition flip-flop is clear, this flip-flop will clear.

Drop CWGCE signal (B13) which disables the checkword generator (B03 through B08).

C. Generate D4SRC signal (B13).

1. Clock the clear write pad byte flip-flop (B12, TP5). It will set.

a. Clear write pad byte flip-flop (B12, TP4).

Drop WPBYT signal (B12).

Enable clearing write in progress flip-flop (B13).

Disable WOWCD signal (B13).

b. Enable head advance on write (B12).

VIII. When the bit counter (B16) reaches the next count of four [signal DCT4 (B16) is a 1] and SRCL1 signal occurs, the following things happen.

A. Clock the write in progress flip-flop (B13). It will clear.

1. Hold CWWREN signal (B13) at a 0.

2. Drop WIP signal (B13).

a. Disable the sector length error circuit (A06).

b. Drop status bit 1 in word 0 (A19).

c. Disable write data to the disk (B12).

d. Generate WIPP signal (B15, TP17) which clocks the write clock control I flip-flop (B16), clearing it, and generates T0FWH signal (B18).

e. Clear the write control flip-flop in the address and control register (B15), dropping the write signal to the selected drive (C01 and C02).

f. When the next write oscillator pluse (C22) occurs, clock the write clock control II flip-flop (B16, TP29). It will clear, stopping the generation of shift register clocks (B16).

B. Generate D4SRC signal (B13).

1. Generate head advance on write signal (B12).

a. If the enable head advance flip-flop (B12) is set, generate HAP signal (B12).

Generate HAP1 signal (B15, TP18).

Bring up bit 3 control tag (B15, TP22), which is set head advance to selected drive (C02).

Hold a clear on the end of cylinder flip-flop (B12).

b. The trailing edge of the HAP generate pulse (B12) fires the 1.2-microsecond end of cylinder check one-shot (B12).

If control status from the selected drive is enabled, and end of cylinder [signal DDB06 (C18, TP29)] comes up before the end of cylinder check pulse drops, set the end of cylinder flip-flop (B12) when the check pulse drops.

Generate HAP signal (B12) which, in turn, generates HAP1 signal (B15, TP18). Bring up bit 3 control tag (B15, TP22) which is head advance to the selected drive (C02). Clear the end of cylinder flip-flop (B12).

2. Clock the clear write pad byte flip-flop (B12, TP5). It will clear.

a. Disable head advance (B12).

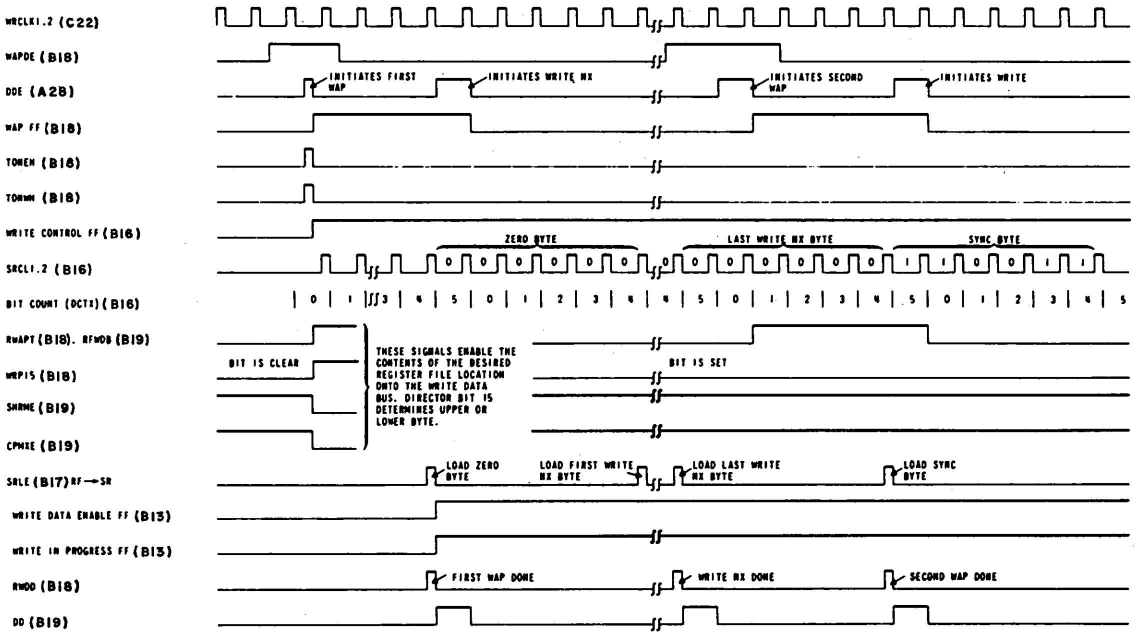
IX. Operation continues with the execution of the next director.

Write Address Pattern Director Execution

This director writes the sync byte on the selected disk storage units and also controls the write and erase heads. The significant bits and their definitions are:

- Bit 15 - If this bit is clear, the required pattern is taken from bits 00 through 07 of the specified register file location. If this bit is set, the required pattern is taken from bits 08 through 15 of the specified register file location.
- Bit 16 - When set, this bit enables head control operations.
- Bit 17 - When set, this bit turns on the write head. When clear, it turns off the write head.
- Bit 18 - When set, this bit turns on the erase head. When clear, it turns off the erase head. (Not used in double density disk units.)
- Bits 28 through 31 - Register file location of required pattern.

The register file must have been previously loaded with the required pattern and the control select line to the disk storage unit must be enabled. The sequence of operation is illustrated in Figure 3-125 and is described as follows:



NOTE: DETAILS OF WRITE NX EXECUTION ARE NOT SHOWN.
FIRST WAP WRITES ONE ZERO BYTE AND TURNS ON WRITE AND ERASE HEADS.
WRITE NX WRITES ZEROS TO FILL SYNC PATTERN.
SECOND WAP WRITES THE SYNC BYTE.

Figure 3-125. Write Address Pattern Sequence

I. Read timing chain generates SO/PD (A27, TP23).

A. Director is gated to director output register on A25, A27, and A28. Output register feeds director fan-out on A17 and director decode logic on A25.

II. Director is decoded on A25.

A. Generate DATDIR (A25, TP23). This signal causes data director control logic to generate DDE.

B. Generate WRD (A25). This is a partial decode for write directors.

C. WRD, DIRC03, and DIRC04 generate WAPDE (B18). This is the write address pattern decode.

III. Data director control logic generates DDE (A28, TP8).

A. Generate TONWH and TONEH on B18. These preset the applicable control flip-flops on B15 and are sent to the selected disk storage unit to turn on the write and erase heads. Director bits 16, 17, and 18 must be set to perform this function. TONWH also presets the write control flip-flop (B16). This enables the shift clock (SRCL1 and 2) and the bit counter. Clock pulses are obtained from the write clock, WCLK1.

B. Load director bits 28 through 31 into the register file address counter on B24. This enables the register file address through register file address multiplexer 2 on A29. Since the register file is normally in a read mode, the addressed location is placed on the register file bus (A29).

C. Set the write address pattern flip-flop (B18, TP2).

1. Generate WAP (B18). This generates WRITEF (B17) and enables the load shift register gate on B17. WRITEF enables the set input to the write data enable flip-flop (B13, TP7) and the write in progress flip-flop (B13).

2. Generate RWAPT (B18). This generates RFWDB (B19) which enables the contents of the register file bus onto the write data bus (A07 through A12).

3. If director bit 15 is a 0, generate WRP15 (B18). This drops CPMXE and SHRME on B19 and enables bits 00 through 07 of the write data bus into the shift register on B09. If director bit 15 is a 1, bits 08 through 15 are enabled.

IV. Bit counter generates DCT4 (B16). This indicates that the last bit of any previous data has been shifted out of the shift register (B09).

   A. Generate SRLE (B17). This loads the enabled bits of the write data bus into the shift register (B09).

   B. Set the write data enable flip-flop (B13, TP7). This generates WRTG which enables the write data gate on B12.

   C. Set the write in progress flip-flop (B13). This generates WIP which enables the sector notch counter on A06, enables the output of the write data flip-flop on B12, and arms the WIPP one-shot circuit on B15. The sync pattern can now be shifted out of the shift register through the write data flip-flop and out to the selected disk storage unit by the shift clock.

   D. Generate RWDD (B18). This sets the data done flip-flops on B19 to terminate the WAP sequence.

V. The write address pattern flip-flop will clear on the next DDE.

Compare Normal Director Execution

This director is similar to a combination read/write operation. Data is read from the disk and compared with data being supplied from the system port, subsystem memory, or from the register file. Bits 11 and 12 of normal operating status word 1 (compare done and compare condition not met) provide the results of the compare operation. The length of the operation is obtained from bits 16 through 31 or from the index register specified by bits 28 through 31.

I. Read director timing time 100 generates SO/PD signal (A27, TP23).

   A. Gate the contents of the addressed director buffer memory location (A29) to the director register (A25, A27, A28).

   B. The director register (A25, A27, A28)and director register fanout (A17)feed the director translators (A25, B23).

      1. Generate DATDIR (A25, TP23) and CPD (A25).

         a. If director bit 6 is a 1, generate CRFAD signal (B19).

         b. Generate RWCDE signal (B19).

         c. If director bit 10 is a 1, generate CRFAD signal (B19).

II. Director execution sequence generates DDE (A28, TP8).

A. If CRFAD signal (B19) is a 1, clear the register file address counter (B24).

B. Director bit 15 controls the selecting of the byte count (compare length) when DDE occurs.

1. Compare director decode has the RWCDE signal (B19) at a 1. If bit 15 is a 1, load the register file counter (B24) with director bits 28 through 31.

a. The outputs from the counter (B24) feed through the register file address mux (B25) and address a register file location (B25).

The location (B25) is read out on the register file bus to data path 2 (A07 through A12) where it is inverted (signals BYI00 through BY15) and feeds the byte counter mux (B21).

2. Signals BCT4 and CYIE (B19) control the gating on the byte counter mux (B21). BCT4 is a constant 1 during this operation. If BYIE is a 1, director bits 16 through 31 (B21) go through the mux (B21). If it is a 0, the BYIXX signals go through.

a. If bit 15 is a 1, set the load byte counter from register file flip-flop (B19, TP24).

Drop BYIE (B19) to a zero making the mux selection (B21).

Generate LBCRF signal (B19).

When DDE (B19) comes up a 1, generate LBCRF signal (B19) which loads the byte counter (B21).

b. If bit 15 is a 0, generate LBCD signal (B19).

Load the byte counter (B21) with the count from the director.

C. Gate director bits 6 and 13 into a holding register (B19).

D. Clock the COMP flip-flop (B29, TP4). It will set.

1. Generate read signal (B17).

a. Generate RIP signal (B1~~6~~).3

Enable sector length error circuit (A06) to start operating.

Generate RIP2 signal (B15) which enables the DTEC signal (B12) and generates RIP2 and RIP2A signals (B13). These enable the checkword generator (B03 through B08) to begin operating.

2. Enable the compare done logic (B19).

3. If director holding register bit 6 flip-flop is a 1, generate DATRF (B19, TP15) and RTRF (B19).

   a. Enable increment of the register file counter (B24).

   b. Enable generation of RFWEU and RFWEL (B24).

   c. Signal RFME (B24, TP9) comes up a 1, which stops referencing the addressed register file location (B25).

   d. Enable the output of the shift network catching register (B09) onto the register file bus (A07 through A12) and to the addressed register file location (B25).

4. Enable CPMXE signal (B19, TP14) generation.

E. Clock the select register file flip-flop (B19). If director bit 10 is a 1, it will set. If it sets, BUFOP (B19) and RFWDB (B19) signals go to 0's. If it remains clear, these signals remain 1's.

1. The RFWDB signal (B19) controls the gating on the write data mux (A07 through A12). The output of the write data mux feeds the compare network (B09).

   a. If RFWDB is a 1, the output of the data buffer (B10) is gated through the write data mux.

   b. If RFWDB is a 0, the register file bus output (BYIXX) signals (A07 through A12) are gated through the write data mux.

2. The BUFOP signal (B19), when a 1,, enables the data buffer control (B17) and write done flip-flop (A06). When it is a 0, it disables these same elements.

F. Preset or preclear the byte control flip-flop (B19) according to the state of director bit 9. If bit 9 is a 1, the flip-flop will set. If bit 9 is a 0, it will clear.

G. Clear the compare done (B19, TP5) and compare condition not met (B19, TP6) flip-flops.

III. The read operation set up by the previous director (RAP) is still active. The selected unit is reading data, the read select tag (B15) is still up, the read clock is generating shift clocks (B16), data is being clocked into the shift register (B09 and checkword generator (B03 through B08), and the bit counter (B16) is running in step with the data field.

IV. If director bit 6 was set initially, a compare and save operation is occurring.

A. When the bit counter (B16) reaches a count of one [DCT1 (B16) is a 1] and a SRCL1 signal (B16) occurs, clock the DLOW/DUP flip-flop (B19, TP10). It will go to the same state as the byte control flip-flop (B19) generating either DLOW or DUP signal (B19).

1. DUP enables generation of RFWEU signal (B24).

2. DLOW enables generation of RFWEL signal (B24), incrementing the register file counter (B24) and CLDRY signal (B17).

3. The DLOW/DUP flip-flop controls CPMXE signal (B19, TP14) generation. If the flip-flop is clear, CPMXE will be a 1; if set, it will be a 0.

a. CPMXE selects the upper or lower portion of the compare word to the compare network (B09). A 1 equals upper.

B. When the bit counter (B16) reaches a count of three [DCT3 (B16) is a 1] and a SRCL2 (B16) occurs, the following steps occur.

1. Transfer the shift register contents into the shift register catching register (B09).

2. Clock the lost data flip-flop (B17, TP32). If BRY signal (B28) is a 0, the flip-flop will set.

a. Generate COMST and fault signals (A05, TP22, and 17).

Stop director loading (B26) and execution (B27) and bring up status bits (A14, A20).

C. When the bit counter (B16) reaches a count of four [DCT4 (B16) is a 1] and SRCL1 occurs, the following things happen.

1. Clock the byte ready flip-flop (B17, TP9). It will set, generating RTRFP signal (B17).

a. If DUP signal (B19) is a 1, generate RFWEU and RFME signals (B24) which transfer the contents of the shift register catching register (B09) into the upper portion of the addressed register file location (B25).

b. If DLOW signal (B19) is a 1, generate RFWEL and RFME signals (B24) which transfer the contents of the shift register catching register (B09) into the lower portion of the addressed register file location (B25).

2. Generate TOGUL (B17, TP18) signal.

   a. Clock the byte control flip-flop (B19). It will go to its opposite state.

3. DATAE signal, from compare network (B09), is ANDed with director bit holding register bit 13 (B19) to produce a compare not met signal (B19, TP16). The compare not met flip-flop (B19, TP6) is clocked by DCT4 and SRCL1. It will set if compare not met signal is a 1.

   a. Enable status word 0 bit 12 (A23).

   b. Enable CDCNM signal (B19).

4. Generate DDBC signal (B17, TP15).

   a. Decrement the count in the byte counter (B21) by one count.

5. If signal DLOW (B19) is a 1 and data buffer RK4 is full [BRY signal (B28) is a 1], generate CLDRY signal (B17).

   a. Clock the data buffer RK4 full flip-flop (B10, TP22). It will clear, enabling a rank 3 to rank 4 transfer (B10).

6. If the byte counter contains a count of one [BCT1 (B21) is a 1] prior to step 4, generate DDC signal (B17) which causes the following steps to occur.

   a. Clock the compare done flip-flop (B19, TP5). It will set.

      Enable bit 11 of status word 0 (A22).

      If compare condition not met flip-flop (B19) is set, generate CDCNM signal (B19).

         Bring up COMST signal (A05, TP22) which enables bit 3 of status word 0 (A20).

b. Generate DD signal (B19).

Enable execution of next director (A28).

Clock the write done flip-flop (A06, TP32). It will set.

Clear the address execute flip-flop (B17, TP5) blocking the data transfer logic.

c. If the next director is a support director, generate CLRD (A28), and clear the compare control flip-flops (B19), ending this compare operation.

Drop read signal (B17) which drops RIP signal (B13).

Turn off sector length error circuit (A06).

Drop RIP2 signal (B15) and generate RIPP signal (B15). RIP 2 turns off the checkword generator (B03 through B08). RIPP drops read tag (B15) and stops shift register clock generation (B16).

d. Clock the force 8-bit mode flip-flop (B13, TP1), setting it.

F8BM signal (B13) comes up to a 1.

Bring up 8BM signal (B12) enabling the shift register (B09) and the bit counter (B16) to operate in 8-bit mode.

D. When the bit counter (B16) reaches a count of five [DCT5 (B16) is a 1] the following steps will occur.

1. Clear the byte ready flip-flop (B17, TP9).

a. Drop RTRFP signal (B17). This disables loading the register file (B24).

2. If signal DLOW (B19) is a 1, generate an increment the register file counter signal (B24, TP5).

a. Increase the count in the register file counter (B24) by one, selecting the next sequential register file location (B25).

E. If byte counter was not at a count of one when step IV. C. 6. occurred, return to step IV. A. and continue the compare.

V. If director bit 10 was set initially, a compare operation using the register file as the source for compare data is occurring.

A. When the bit counter (B16) reaches a count of one [DCT1 (B16) is a 1] and SRCL1 (B16) occurs, clock the DLOW/DUP flip-flop (B19, TP10). It will go to the same state as the DLOW/DUP select flip-flop (B19), generating either a DLOW or a DUP signal (B19).

1. The DLOW/DUP flip-flop controls CPMXE signal (B19, TP14) generation. If the flip-flop is clear, CPMXE will be a 1; if set, it will be a 0.

a. CPMXE selects the upper or lower portion of the compare word to the compare network (B09). If CPMXE is a 1, compare upper.

B. When the bit counter (B16) reaches a count of three, check lost data. Refer to step IV. B. 2.

C. When the bit counter (B16) reaches a count of four [DCT4 (B16) is a 1] and SRCL1 occurs, the following things happen.

1. Generate TOGUL (B17, TP18).

a. Clock the DLOW/DUP select flip-flop (B19). It will go to its opposite state.

2. Clock the compare not met flip-flop (B19, TP6). If compare not met signal (B19, TP16) is a 1, the flip-flop will set.

a. Enable bit 12 of status word 0 (A23).

b. Enable CDCNM (B19).

3. Generate DDBC (B17, TP15).

a. Decrement the count in the byte counter (B21) by one count.

4. If the byte counter contains a count of one [BCT1 (B21) is a 1] prior to the preceding step, generate DDC (B17) and clock the force 8-bit mode flip-flop (B13, TP1), setting it. The remaining steps under this step are the same as those under IV. C. 6. Complete those before continuing.

D. When the bit counter (B16) reaches a count of five [DCT5 (B16) is a 1], the following steps will occur.

1. If DLOW (B19) is a 1, generate an increment the register file counter signal(B24, TP5).

   a. Increase the count in the register file counter (B24) by one, selecting the next sequential register file location (B25). It will be read out and routed to the compare mux.

E. If the byte counter was not at a count of one when step V. C. 4. occurred, return to step V. A. and continue the compare.

VI. If director bits 6 and 10 were clear initially, a compare operation using the data buffer as the source for compare data is occurring.

A. When the bit counter (B16) reaches a count of one [DCT1 (B16) is a 1] and SRCL1 (B16) occurs, clock the DLOW/DUP flip-flop (B19, TP10). It will go to the same state as the DLOW/DUP select flip-flop generating either a DLOW or a DUP signal (B19)

1. DLOW enables generation of the CLDRY signal (B17).

2. The DLOW/DUP flip-flop controls CPMXE signal (B19, TP14) generation. If the flip-flop is clear, CPMXE will be a 1; if set, it will be a 0.

   a. CPMXE selects the upper or lower portion of the compare word to the compare network (B09). If CPMXE is a 1, compare upper.

3. DCT1 and SRCL1 will also clock the load byte counter from RF flip-flop (B19, TP24). If it was set previously (bit 15 was a 1 initially), it will now clear.

   a. Signals BCT4 and BYIE (B19) go back to their original configuration.

B. When the bit counter (B16) reaches a count of four [DCT4 (B16) is a 1] and SRCL1 occurs, the following things happen.

1. Generate TOGUL (B17, TP18).

   a. Clock the DLOW/DUP select flip-flop (B19). It will go to its opposite state.

2. Clock the compare not met flip-flop (B19, TP6). If compare not met signal (B19, TP16) is a 1, the flip-flop will set.

   a. Enable bit 12 of status word 0 (A23).

   b. Enable CDCNM signal (B19).

3. If DLOW (B19) is a 1, and data buffer rank 4 is full, generate CLDRY (B17).

   a. Clock the data buffer RK4 full flip-flop (B20, TP22). It will clear, enabling a rank 3 to rank 4 transfer (B09).

4. Generate DDBC (B17, TP15).

   a. Decrement the count in the byte counter (B21) by one count.

5. If the byte counter contains a count of one [BCT1 (B21) is a 1] prior to the preceding step, generate DDC (B17) and clock the force 8-bit mode flip-flop (B13, TP1), setting it. The remaining steps under this step are the same as those under 4. C. 6. Complete those before continuing.

C. If the byte counter was not at a count of one when step VI. B. 5. occurred, return to step VI. A. and continue the compare.

VII. Director execution continues with the next director.

Delay Director Execution

This director causes a delay which is one byte greater than the byte count (8-bit mode) or character count (6-bit mode) contained in director bits 16 through 31. The following data director must be made available during the delay period so that it can begin execution when the delay period has expired. Support directors may be executed during the delay period.

Programming Considerations

1. If a mark start is specified (bit 8 set), the delay will not start until the mark start condition is satisfied.

2. The unit read control status must be selected before a mark start delay is executed to ensure starting on the leading edge of the disk condition specified by bits 9 and 10. These selectable conditions are:

00	Index mark
01	Sector mark
10	Sector alert
11	Not used

The sequence of operation is:

I. Read director timing time 100 generates SO/PD signal (A27, TP23).

  A. Gate the contents of the addressed director buffer memory location (A29) to the director register (A25, A27, A28).

  B. The director register (A25, A27, A28) and director register fanout (A17)feed director translators (A25, C09).

    1. Generate DATDIR (A25, TP23) and CPD (A25).

II. Director execution sequence generates DDE (A28, TP8).

  A. Set the delay flip-flop (B22, TP3). Generate delay signal.

    1. Block data transfer inverter and normal byte counter operation (B17).

  B. Transfer director bits 9 and 10 into holding flip-flops (B22).

  C. If director bit 8 is present, set the mark start flip-flop (B22, TP2).

    1. Transfer the contents of the sector length holding register into the sector length counter (A06).

    2. Generate IBCC signal (B22) which clears and stops the operation of the delay byte counter (B16).

  D. Generate DBCL signal (B22).

    1. Generate LBCD signal (B19, TP11) which loads director bits 16 through 31, the delay byte count, into the byte counter (B21).

III. If mark start flip-flop (B22, TP2) is set, wait for selected mark start condition [selected by director bits 9 and 10 stored previously in holding flip-flops (B22, TP5 and 4)]; clock the mark start flip-flop. It will clear.

  A. Drop IBCC signal (B22).

    1. Enable delay byte counter (B16).

      a. Generate I5O7 signal on every count of five or seven as selected by 8-bit mode.

IV. If mark start flip-flop (B22, TP2) is not set, I507 signal, BCTZ, and an internal clock generate a DLYCK signal (B22, TP12).

  A. Generate DDBC signal (B17, TP15).

    1. Decrement the byte counter (B21) by one.

V. BCTZ signal (B21, TP6) occurs when byte count reaches zero.

  A. Disable DLYCK signal (B22, TP12) and enable DLYD signal (B22, TP16).

VI. Wait for next I5O7 signal and an internal clock; then generate a DLYD signal (B22, TP16).

  A. Generate DD signal (B19).

    1. Enable execution of next director (A28).

VII. Director execution continues with next director.

Read Sequence

Figure 3-126 shows the timing involved in a complete read sequence. The sequence includes the following directors which are executed in the order listed.

1. RAP anyplace full byte
2. RAP anyplace
3. RAP specific
4. Read normal
5. Read checkword

The sequence illustrates the major actions associated with each director. For detailed information for a particular director, refer to the operational sequence for that director.

Write Sequence

Figure 3-127 shows the timing involved in a complete write sequence. The sequence includes the following directors which are executed in the order listed.

1. WAP
2. Write normal
3. Write checkword

The sequence illustrates the major actions associated with each director. For detailed information for a particular director, refer to the operational sequence for that director.

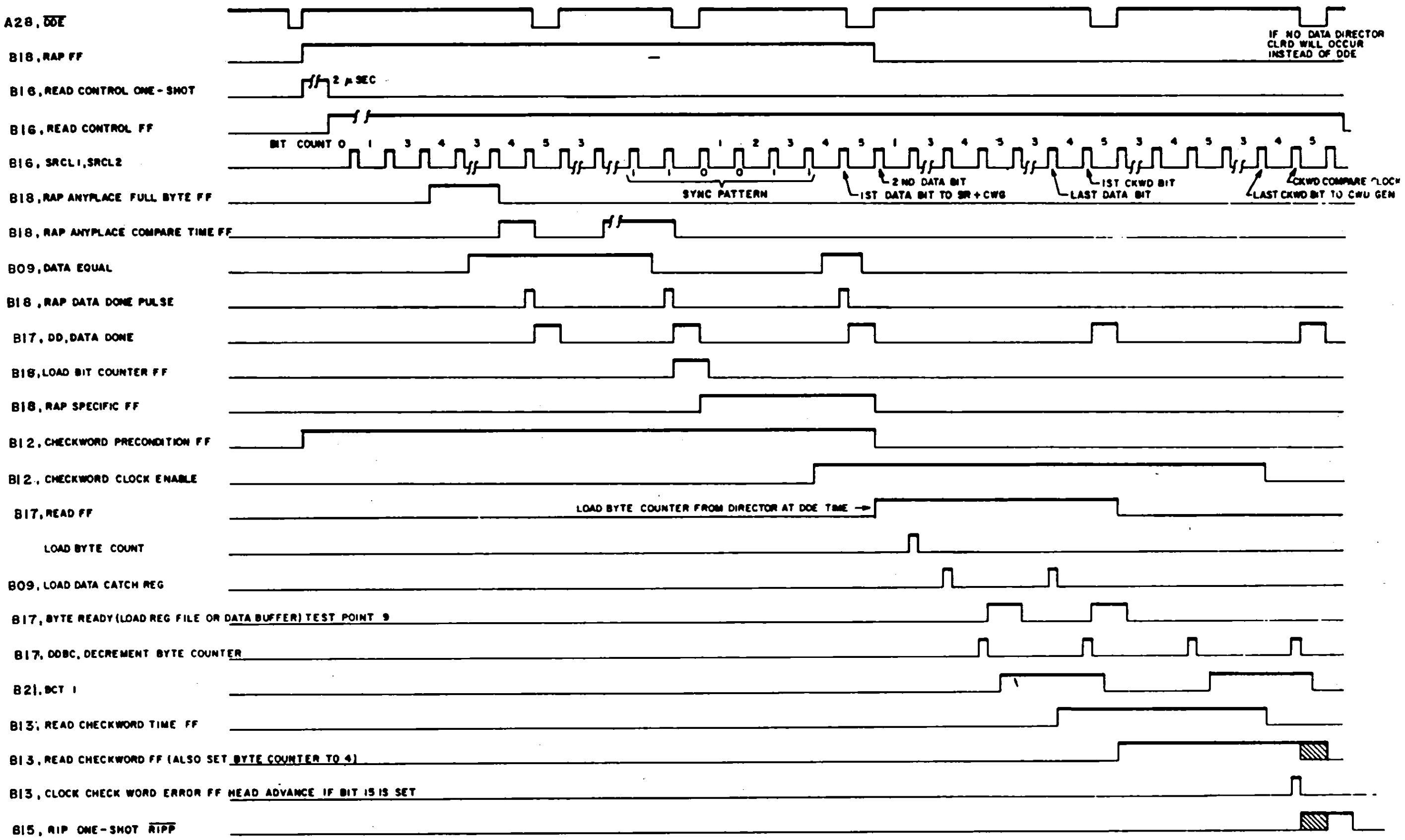

A28, DDE
B18, RAP FF
B16, READ CONTROL ONE-SHOT
2 μSEC
B16, READ CONTROL FF
B16, SRCL1, SRCL2
BIT COUNT 0 1 3 4 3 4 5 3 1 2 3 4 5 1 3 4 5 3 4 5 3 4 5 3 4 5
1 1 0 0 1 1
SYNC PATTERN
1ST DATA BIT TO SR + CWG
2 ND DATA BIT
LAST DATA BIT
1ST CKWD BIT
LAST CKWD BIT TO CWU GEN
CKWD COMPARE CLOCK
IF NO DATA DIRECTOR CLRD WILL OCCUR INSTEAD OF DDE
B18, RAP ANYPLACE FULL BYTE FF
B18, RAP ANYPLACE COMPARE TIME FF
B09, DATA EQUAL
B18, RAP DATA DONE PULSE
B17, DD, DATA DONE
B18, LOAD BIT COUNTER FF
B18, RAP SPECIFIC FF
B12, CHECKWORD PRECONDITION FF
B12, CHECKWORD CLOCK ENABLE
B17, READ FF
LOAD BYTE COUNTER FROM DIRECTOR AT DDE TIME
LOAD BYTE COUNT
B09, LOAD DATA CATCH REG
B17, BYTE READY (LOAD REG FILE OR DATA BUFFER) TEST POINT 9
B17, DDBC, DECREMENT BYTE COUNTER
B21, BCT 1
B13, READ CHECKWORD TIME FF
B13, READ CHECKWORD FF (ALSO SET BYTE COUNTER TO 4)
B13, CLOCK CHECK WORD ERROR FF HEAD ADVANCE IF BIT 15 IS SET
B15, RIP ONE-SHOT RIPP

Figure 3-126. Complete Read Sequence Timing

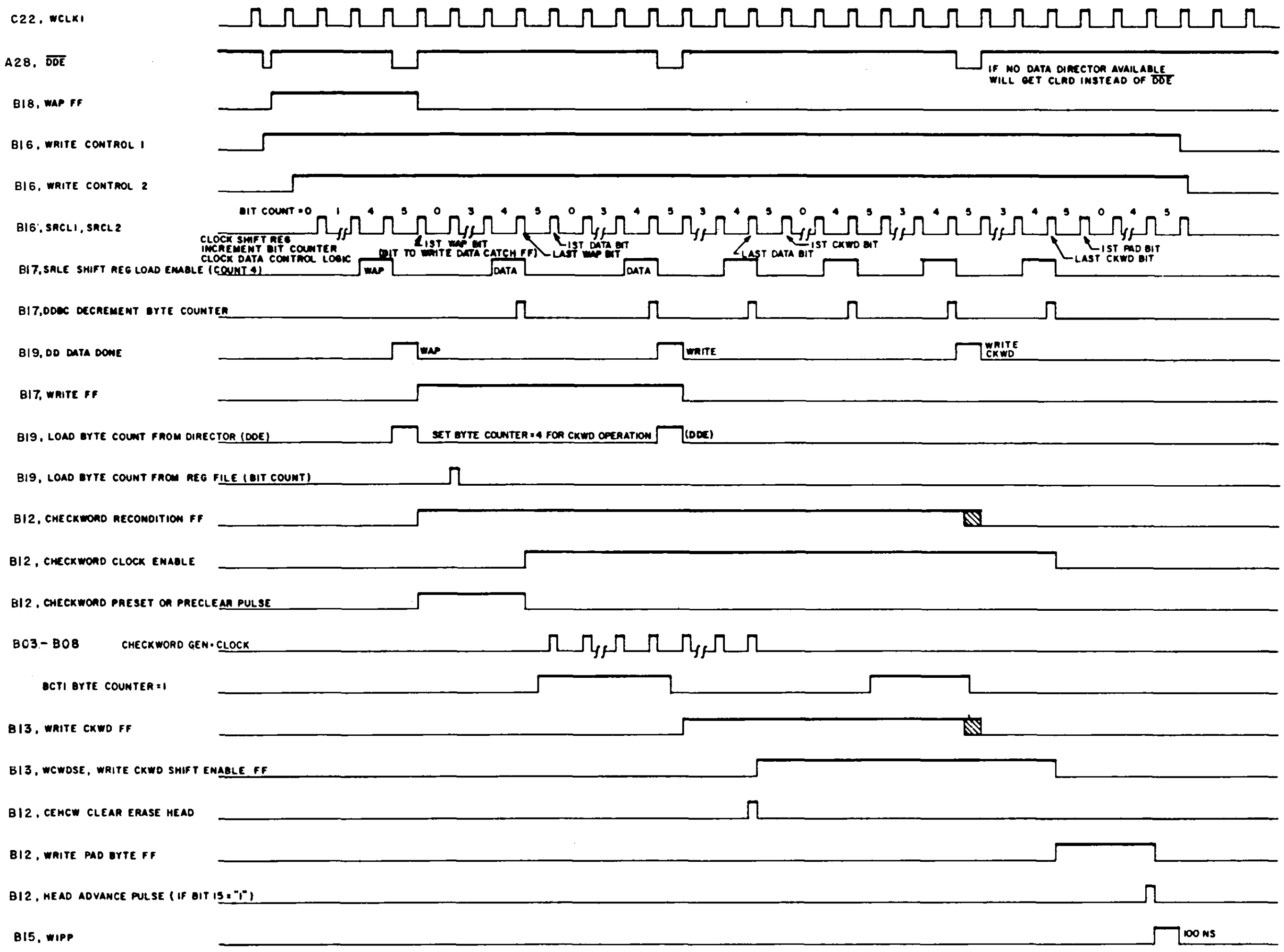

C22, WCLKI
A28, DDE
IF NO DATA DIRECTOR AVAILABLE WILL GET CLRD INSTEAD OF DDE
B18, WAP FF
B16, WRITE CONTROL 1
B16, WRITE CONTROL 2
BIT COUNT = 0 1 4 5 0 3 4 5 0 3 4 5 3 4 5 0 4 5 3 4 5 3 4 5 0 4 5
B16', SRCL1, SRCL2
CLOCK SHIFT REG
INCREMENT BIT COUNTER
CLOCK DATA CONTROL LOGIC
1ST WAP BIT
(BIT TO WRITE DATA CATCH FF)
LAST WAP BIT
1ST DATA BIT
LAST DATA BIT
1ST CKWD BIT
LAST CKWD BIT
1ST PAD BIT
B17, SRLE SHIFT REG LOAD ENABLE (COUNT 4)
WAP
DATA
DATA
B17, DDBC DECREMENT BYTE COUNTER
B19, DD DATA DONE
WAP
WRITE
WRITE CKWD
B17, WRITE FF
B19, LOAD BYTE COUNT FROM DIRECTOR (DDE)
SET BYTE COUNTER = 4 FOR CKWD OPERATION
(DDE)
B19, LOAD BYTE COUNT FROM REG FILE (BIT COUNT)
B12, CHECKWORD RECONDITION FF
B12, CHECKWORD CLOCK ENABLE
B12, CHECKWORD PRESET OR PRECLEAR PULSE
B03-B08 CHECKWORD GEN. CLOCK
BCT1 BYTE COUNTER = 1
B13, WRITE CKWD FF
B13, WCWDSE, WRITE CKWD SHIFT ENABLE FF
B12, CEHCW CLEAR ERASE HEAD
B12, WRITE PAD BYTE FF
B12, HEAD ADVANCE PULSE (IF BIT 15 = "1")
B15, WIPP
100 NS

Figure 3-127. Complete Write Sequence Timing

Subsystem Processor Memory Request

The following sequence describes how the control logic handles a subsystem processor memory request.

I. Signals required from subsystem processor to initiate sequence.

   A. STREQ1

   B. WRITE1 only when requesting a store into memory

   C. SCRM04 to SCRM15 to select memory address

   D. ST0-00 to ST0-16 (data to be stored in memory if WRITE 1 is up)

II. Set processor request 1 flip-flop (B26, TP24) with STREQ 1.

III. Set processor request 2 flip-flop (B26, TP25) with memory scanner (B26, TP10).

   A. Disable clear of processor scanner flip-flop (B26, TP10).

IV. When scanner stops on processor position:

   A. Clear processor request 3 flip-flop (B26, TP14).

      1. Generate MEMGO (B26).

         a. Gate memory address of subsystem processor to memory (A13 to A16).

         b. Enable data output STO-00 to STO-16 of subsystem processor to memory (A13 to A16) (will only be used during a store).

      2. Generate STORE (B26, TP5) if WRITE 1 signal from subsystem processor is present.

         a. Disable read input (A13 to A16) from memory.

   B. Start memory request timing chain (B26) if memory is not busy (B26, TP22).

      1. Generate MRQ1 (B26, TP23).

         a. Clear I/P request flip-flop (B27, TP31)

         b. Clear O/P request flip-flop (B27, TP28)

         c. Clear director request 3 flip-flop (B27, TP29)

2. Generate MREQT (B26).

   a. Clear processor request 1 flip-flop (B26, TP24)

   b. Set the memory request in progress flip-flop (B26, TP27)

   c. Start memory busy delay line (B28, TP4)

   d. Clock the internal busy flip-flop (B28, TP1). It will clear.

   e. Generate BUSY signal (B28).

3. Generate MREQ to memory to start memory cycle (B26, TP20).

V. MBSY1 signal arrives from memory (B28, TP2)

A. Maintain BUSY signal (B28).

VI. MDRDY (B26) arrives from memory to indicate memory data ready.

A. Start reply timing chain (B26, TP26).

1. Generate reply at time 10.

   a. Send MDRDY1 (B26, TP6) to subsystem processor.

2. Clear processor request 2 flip-flop (B26, TP25) at time 30.

   a. Clear processor scanner flip-flop (B26, TP10) and release scanner.

3. Drop reply at time 70

   a. Set processor request 3 flip-flop (B26, TP14)

      Drop MEMG0, STORE.

VII. Drop BUSY signal (B28) when MBSY1 signal from memory drops and memory delay line times out (enables next storage reference).

COMMENT SHEET

MANUAL TITLE CDC FA722-A/B/C, DT220-C, DT716-A Disk Storage Controller

Volume 1 Hardware Maintenance Manual

PUBLICATION NO. 60428500 REVISION M

FROM: NAME:

BUSINESS ADDRESS:

CUT ALONG LINE

PRINTED IN USA

AA3419 REV. 7/78

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Publications and Graphics Division
ARH219
4201 North Lexington Avenue
Saint Paul, Minnesota 55112

CUT ALONG LINE

FOLD

FOLD

CONTROL DATA

1/2

3/4

1-1/4

>>> CUT OUT FOR USE AS LOOSE-LEAF BINDER TITLE TAB

CONTROL DATA
CORPORATION

CORPORATE HEADQUARTERS, 8100 34th AVE. SO., MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.