

# TABLE OF CONTENTS

## ARITHMETIC UNIT 4022A

### INTRODUCTION

INT

### SPECIFICATIONS

### ARITHMETIC UNIT OPTIONS

### ARITHMETIC UNIT DESCRIPTION

DESC

A Register (Accumulator)

B Register (Buffer)

P Register (Program)

J Register (Counter)

I Register (Instruction)

Parallel Full Adder

Q Register (Multiplier/Quotient)

Memory Address Buffer

Automatic Program Interrupt Control

Overflow Flip-Flop

Test Flip-Flop

Operators Console

### LOGIC SYMBOLS AND NOMENCLATURE

### OPTIONS

OPT

#### MEMORY PROTECT

Protect Status

Memory Protect Rules

Theory of Operation

#### STALL ALARM

#### API WATCHDOG

#### LINE FREQUENCY TIMER

#### COMPUTER RESTART

### CONSOLE

CON

#### INTRODUCTION

#### DESCRIPTION

Register Select - P, I, B, A

Console to A, B

CLEAR Register

MANual STEP Operation

#### CONSOLE PROCEDURES

Load A Bootstrap Routine

### COMMANDS

CMD

#### COMMAND FORMATS

Full Operand

GEN 1

GEN 2

GEN 3

Quasi

#### BASIC TIMING OF FULL OPERAND COMMANDS

Instruction Sequencing

Sequence Control State 1

#### INDEXING

#### RELATIVE ADDRESSING

#### MEMORY WRAP AROUND

#### COMMAND DESCRIPTION CONVENTIONS

**ADD** - ADD Z TO A ADD

**ANA** - LOGICAL AND TO A ANA

**BRU** - BRANCH UNCONDITIONALLY BRU

**BTR** - BRANCH IF TEST FLIP-FLOP RESET BTR

**BTS** - BRANCH IF TEST FLIP-FLOP SET BTS

**DMT** - DECREMENT MEMORY AND TEST DMT

**DVD** - DIVIDE DVD

**ERA** - EXCLUSIVE OR TO A ERA

**GEN 1 COMMANDS** GN1

ADO - Add One To Bit K

CBK - Change Bit K

CLO - Count Least Significant Ones

CLZ - Count Least Significant Zeros

CMO - Count Most Significant Ones

CMZ - Count Most Significant Zeros

CPL - Complement A

IBK - Isolate Bit K

LBM - Load Bit Mask

LDO - Load One Into Bit K

LDZ - Load Zeros Into A

LMO - Load Minus One Into A

NEG - Negate

RBK - Reset Bit K

REV - Reset Test Flip-Flop If Bit K is Even

RNZ - Reset Test Flip-Flop If A Is Non-Zero

ROD - Reset Test Flip-Flop If K Is Odd

RST - Reset The Test Flip-Flop

SBK - Set Bit K

SET - Set Test Flip-Flop

SEV - Set Test Flip-Flop If Bit K Is Even

SNZ - Set Test Flip-Flop If A Is Non-Zero

SOD - Set Test Flip-Flop If Bit K Is Odd

SRA - Shift A Right Arithmetic

SRC - Shift Right Circular

SRL - Shift Right Logical

TER - Test Even And Reset Bit K  
 TES - Test Even And Set Bit K  
 TEV - Test Bit K Even  
 TNM - Test Not Minus One  
 TNZ - Test A Non-Zero  
 TOD - Test Bit K Odd  
 TOR - Test Odd And Reset Bit K  
 TOS - Test Odd And Set Bit K  
 TSC - Test And Shift Circular  
 TZC - Test Zero And Complement  
 TZE - Test A Zero

**GEN 2 INPUT/OUTPUT COMMANDS**

GN2

ABT - Abort Device D's Operation  
 ACT - Activate Device D's Interrupt  
 IAI<sub>1</sub> - Inhibit Automatic Interrupt  
 IAI<sub>2</sub> - Inhibit Automatic Interrupt (Optional)  
 IN - Input From Device D  
 JCB - Jump If Channel Busy  
 JDR - Jump If Data Ready  
 JND - Jump If No Demand  
 JNE - Jump If Device D Not In Error  
 JNO - Jump If No Overflow  
 JNP - Jump If No Parity Error  
 JNR - Jump If Device D Not Ready  
 LMR, LMR<sub>2</sub> - Load Mask Register (Optional)  
 OPR - Operate  
 OUT - Output To Device D  
 PAI - Permit Automatic Interrupt  
 RCS - Read Console Switches  
 SEL - Select Device D  
 SSA - Set Atall Alarm (Optional)  
 STM - Select Trapping Mode (Optional)  
 PROGRAMMABLE ALARM

**GEN 3 COMMANDS**

GN3

DLA - (Shift) Double Left Arithmetic  
 DLL - (Shift) Double Left Logical  
 DRA - (Shift) Double Right Arithmetic  
 DRC - (Shift) Double Right Circular  
 DRL - (Shift) Double Right Logical  
 MAQ - Move A To Q  
 SLA - Shift Left Arithmetic  
 SLL - Shift Left Logical

**INDEXING ADDRESS MODIFICATION**

INDEX

INX - INCREMENT X INX  
 LDA - LOAD THE A REGISTER LDA  
 LDP - LOAD PLACE LDP  
 LDQ - LOAD THE Q REGISTER LDQ  
 LDX - LOAD X LOCATION FROM Z LDX  
 LPR - LOAD PLACE AND RESTORE LPR  
 LXC - LOAD X WITH COUNT LXC  
 L XK - LOAD X WITH K L XK  
 MPY - MULTIPLY MPY  
 NOP - NO OPERATION NOP  
 ORA - LOGICAL OR TO A ORA  
 QSI - QUASI COMMANDS QSI

**RELATIVE ADDRESSING**

RLT

SPB - SAVE PLACE AND BRANCH SPB  
 STA - STORE CONTENTS OF A STA  
 STQ - STORE CONTENTS OF Q STQ  
 STX - STORE X LOCATION INTO Z STX  
 SUB - SUBTRACT Z FROM A SUB

TIM/TOM TABLE INPUT TO  
 MEMORY/TABLE OUTPUT  
 FROM MEMORY

TIM/TOM

TXH - TEST X HIGH OR EQUAL TXH  
 XEC - EXECUTE XEC

# INTRODUCTION

This section describes the functional operation of the GE-PAC\* 4022A Arithmetic Unit (GE Drawing Number 68C972367) which is the computational and control center for the GE-PAC 4020 Central Processing Unit.

The 4022A Arithmetic Unit performs calculations, a wide range of logical operations, and sequences and distributes data throughout the computer system. It supplies and receives information to/from the Core Memory, Automatic Program Interrupt Module, and Input/Output Peripheral Devices.

The Arithmetic Unit addresses sequential programmed commands stored in Core Memory. Each command addressed is transferred from the memory location to the Arithmetic Unit where the command is executed. Since execution of the command may require the transfer of data to or from one of the other modules of the system, the Arithmetic Unit becomes the communications hub of the system.

The arithmetic operations performed by the Arithmetic Unit include Add, Subtract, Multiply, and Divide. Addition and Subtraction are performed in 3.2 microseconds, while Multiply requires between 8.9 and 12.1 microseconds and Divide requires 13.6 microseconds. Add and Subtract use single length 24-bit numbers. Multiply operates on two 24-bit numbers to produce a double length (46 bits plus sign) product. Divide operates on a double length dividend and a single length divisor to produce a single length (23 bits plus sign) quotient and a single length remainder.

## SPECIFICATIONS

- Electrical Requirements
  - Voltage: 115/230  $\pm$  10%
  - Phase: Single
  - Frequency: 50 or 60 cps  $\pm$  3%
  - Central Processor Power: 5.8 KW Maximum.
- Environment
  - Temperature: 32° - 131° F
  - Relative Humidity: 5 - 95%.
- Circuit Type
  - Monolithic Integrated.
- Man/Machine Communication
  - Programming and Maintenance Console.

- Basic Clock Frequency
  - 10 Megacycles
  - Pulse Width - 25 nanoseconds  $\pm$  10%.
- Addressable Core Memory
  - 4096 locations minimum, 32,768 locations maximum; directly 16,384, relative ( $\pm$  8K) and indexing to 32,768.
- Operation
  - Parallel and serial internal; parallel external.
- Arithmetic
  - Digital, binary, fixed point, and 2's complement; floating point by subroutine.
- Word Size
  - 24-bits (23 through 00); bit 23 is sign and most significant bit.
- Index Words
  - Total of 7 words; 5 are unrestricted and 2 (1 and 2) are restricted by Quasi and subroutine linkage.

## ARITHMETIC UNIT OPTIONS

The GE-PAC 4022A Arithmetic Unit may contain, as options, Memory Protect, Expanded Console, TIM/TOM, Program Load, I/O Capability (future), and Stall/Watchdog/Timer. These options are described under the mnemonic OPT in this section. The options for a particular system are contained in a special drawing for each system titled "Option List". The model numbers for each of the above options are listed below:

Basic Arithmetic Unit	-	4DP4022A0100
Memory Protect Option	-	4DP4022A0001
TIM/TOM Option	-	4DP4022A0002
Console Option	-	4DP4022A0003
Program Load Option	-	4DP4022A0004
I/O Capability (future)	-	4DP4022A0005
Stall Alarm/Timer/Watchdog Option	-	4DP4022A0006.

\* Registered Trademark of General Electric Company

# ARITHMETIC UNIT DESCRIPTION

The following discussion describes the functional block diagram of the 4022A Arithmetic Unit. Fig. DESC. 1 contains a block diagram of the Arithmetic Unit illustrating the interconnection of the various registers, parallel adder unit, serial adder, and the associated core memory module. This diagram does not attempt to define timing, sequencing, or control organization.

## A Register (Accumulator)

The A Register is the accumulator for arithmetic and bit manipulation operation and temporary storage for data coming from or going to the registers of the I/O equipment. It is comprised of 24 high-speed flip-flops in a bit configuration numbered 00-23 with bit 23 being the most significant.

The A Register may be shifted right or left via the micro-coded GEN 1 command. However, the left shift (when using GEN 1) is accomplished by shifting right. A true left shift may be accomplished with a GEN 3 instruction where data from either B23 or B22 enters A0 and is shifted left. On a typical right shift, A0 passes through the serial full adder (where it may be altered) and back into A23. Position A23 may also enter the serial full adder, possibly modified, and routed back to A23 and at the same shift A23 may be applied to A22.

When A is shifted left, bit position 23 is lost. The contents of the A and B Registers may be shifted right

where the contents of A0 enter either B23 or B22 as specified by the shift instruction.

The contents of the A Register may be gated in parallel to the Parallel Full Adder and the A Register may receive data in parallel from the Adder. The contents of A may be selectively displayed on the optional console.

## B Register (Buffer)

The B Register is a 24-bit register used to hold all instructions and AU data going to or coming from core memory. During the fetch portion of an instruction, the 10 most significant bits of the B Register are transferred in parallel to the I Register for decoding. The contents of the B Register are routed in parallel to the Parallel Full Adder in two fields: B13-0 for address modification, and B23-0 for full register operations. The contents of the Parallel Full Adder may be routed to the B Register in parallel.

The B Register may act as an extension to the A Register when performing right and left shifts (MPY, DVD, GEN 3, TIM/TOM). During a left shift, the contents of either B23 or B22 as specified by the instruction, enter position A0 of the A Register. During a right shift, the contents of A0 enter either B23 or B22 as specified by the instruction. The contents of B may be displayed on the optional console.

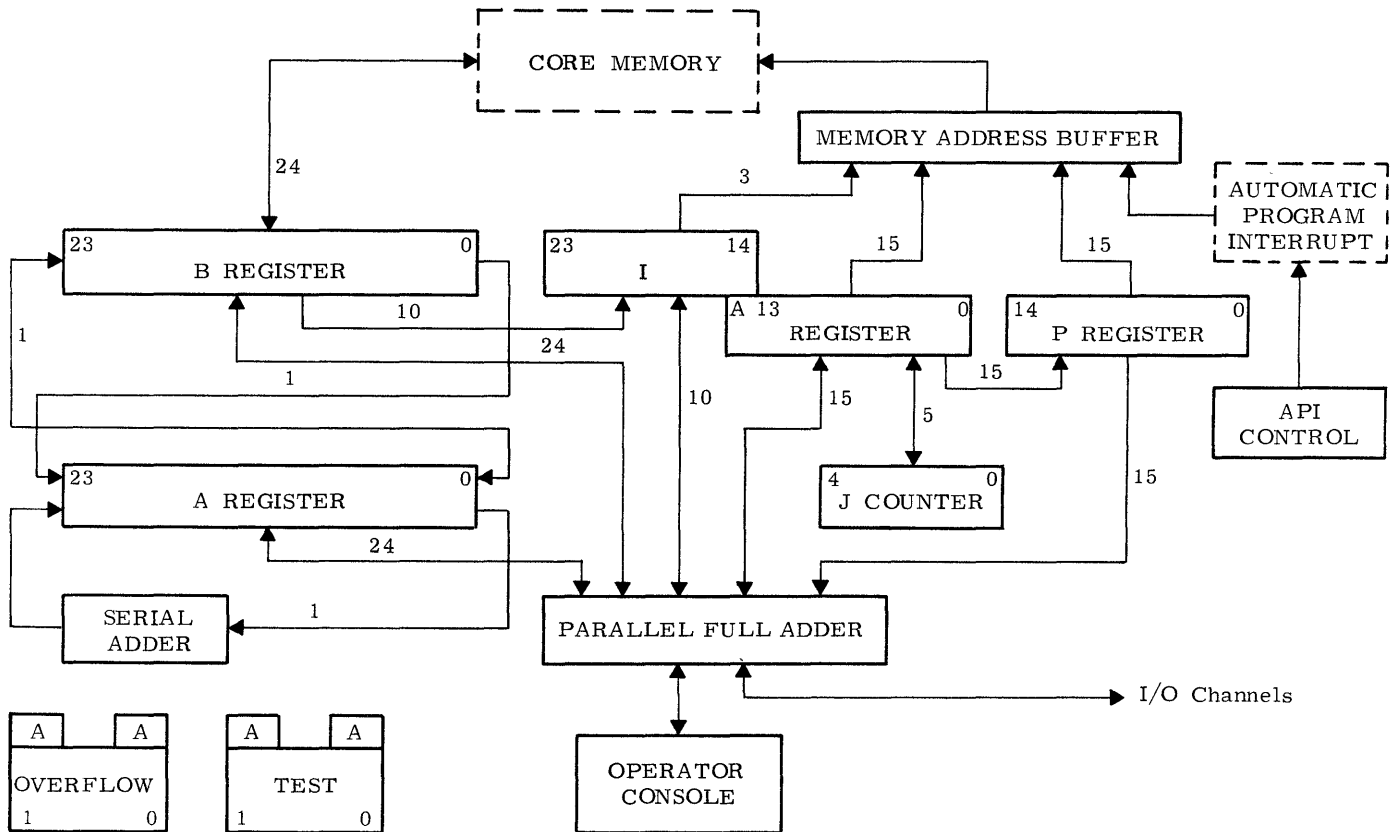


Fig. DESC. 1. GE/PAC 4022A Arithmetic Unit Block Diagram

## P Register (Program)

The P Register is a 15-bit binary counter register whose primary function is to control the addressing of the next instruction. Depending upon the instruction being executed, the P Register may be incremented by one or two. Information from the I Register (address) is transferred to the P Register during the operation of Branch commands. The contents of P may be selectively displayed on the console.

## J Register (Counter)

The J Register is a 5-bit binary counter used to count shifts when executing shift instructions. Incrementation of J is controlled by the micro-coded GEN 1, GEN 3 instructions or Multiply, Divide, or TIM/TOM. The contents of J may be transferred to the I Register or the complement of the 5 least significant bits of I may be transferred to J. The contents of J are displayed on the optional console.

## I Register (Instruction)

The I Register is a 24-bit register used to hold the instruction while it is being decoded, interpreted, and executed. An additional bit (14) position is used in the I Register to provide for relative addressing. In addition to holding the instruction to be executed, the I Register is also used to hold the divisor during Divide, the multiplicand during Multiply, and the TIM/TOM control during TIM/TOM operations.

The I Register may receive data in parallel from the Parallel Full Adder, B Register, and J Counter. Data is transferred from the I Register in parallel to the J Counter, P Register, Memory Address Buffer, and to the Parallel Full Adder. The contents of the I Register may be selected for display on the optional console.

## Parallel Full Adder

The Parallel Full Adder is a 24-bit full adder. Most arithmetic operations within the arithmetic and control unit are accomplished through this adder. In addition to the normal add and subtract operations, the adder has the capability of accomplishing the following logical combinations:

Exclusive "OR" (if A = B, 0 out; if A ≠ B, 1 out).

Logical "AND" (if A = 1 and B = 1, 1 out; if A ≠ 1 or B ≠ 1, 0 out).

Logical "OR" (if A = 0 and B = 0, 0 out; if A ≠ 0 or B ≠ 0, 1 out).

The adder may receive data in parallel from A, B, I, or the P Register. The adder also acts as a buffer for Input/Output operations. Data transferred into the central processor unit from Input/Output devices is channeled through the adder for distribution within the arithmetic and control unit. Data transmitted from memory to specified Input/Output devices is routed through the B Register to the Adder for distribution outside the arithmetic and control unit. Data transmitted from the A Register is routed through the adder for distribution outside the arithmetic and control unit.

## Q Register (Multiplier/Quotient)

The Q Register is a pseudo register (located in core memory address 10g) used as an auxiliary accumulator (MPY and DVD). The contents of Q are also used to define fields of the A and/or B Register during GEN 3 commands. The Q Register is addressed and gated to the B Register during the command execution.

## Memory Address Buffer

The Memory Address Buffer provides selection of the core memory address bits from the desired register or control circuit.

Certain memory locations are pre-assigned for specific use. These memory locations should be used only for the purpose intended unless hardware is not included to make use of the specific reserved address. The reserved memory locations and the pre-assigned uses are listed below.

Octal Address	Specific Use
00	Primary Bulk Storage Pointer Word (Channel 1)
01 ↓ 07	} Index Register Words
10	
11 ↓ 17	} Additional Bulk Storage Pointer Words (11 = Channel 2) (12 = Channel 3)
20	
21	Memory Protect, Saved I Register Word
22	Not Dedicated
23	Not Dedicated
24	API Stall Alarm Error Exit Location (SPB)
25	Not Dedicated
26	Not Dedicated
27	Not Dedicated
30	Not Dedicated
31 ↓ 37	} Common Peripheral Pointer Words
40 ↓ 77	
100 ↓ 177	} Memory Protect Status Words
200 ↓ 377	

## Automatic Program Interrupt Control

The Automatic Program Interrupt control logic within the Arithmetic Unit provides timing synchronization between the AU and API modules, inhibits interrupts during the execution of specific commands, inhibits inhibitable interrupts under program control, and provides an Echo interrupt signal to the API under certain conditions.

Each of these control functions and the logic elements providing these functions are described below.

- Enable API Gate, G1WENA (logic sheet 122), must be enabled to allow any interrupts to be serviced. This gate is enabled only during Time 3 Envelope (TSCA · TSCC) of Sequence Control State 4. Since some commands do not enter Sequence State 4 for execution, interrupts following these commands are inhibited. These commands include: BRU, BTS, BTR, and TXH.

In addition, the following commands inhibit G1WENA, thereby inhibiting interrupts following the command execution.

Command	Logic Inhibit Signal	Function
CLO CLZ CMO CMZ	DGN1 · $\overline{WI08}$ · WS67	Allow J Count value to be stored following the commands
LPR LDP SPB	XRMF	Allows the P Register to be changed before API
JNO JNR JDR JCB	NGS6	Inhibits a TIM/TOM interrupt from following JDR, JNR, and JCB commands
LDX	WLDX	Allows Trap Error detection
Volume Difference Trap Error	$\overline{MSSI}$	Allows Memory Protect operation

The API Lockout switch on the Console when in the Lockout position inhibits G1WENA.

- The Permit Automatic Interrupt flip-flop, F1WPMT (logic sheet 121), when reset inhibits all inhibitable type interrupts. F1WPMT is reset during Time 3 Envelope (TSCA · TSCB · TSCC) of State 4 when there is not a Volume Difference or Memory Protect Error by an SPB, IAI, or LPR command.

$$\overline{F1WPMT} = TSCC \cdot TSCA \cdot WCPM \cdot ECLK + \text{Initialize.}$$

G1WCPM when enabled to reset F1WPMT, also applies a signal to the API module to clear the priority flip-flops.

F1WPMT is set, allowing inhibitable interrupts, at Last Pulse of State 4 during the execution of a PAI or LPR command having bit 21 a "one".

$$F1WPMT = WSPM \cdot TLPE \cdot ECLK.$$

- G1WEKO applies a signal to the API to generate an "Echo" interrupt when a TIM/TOM operation is in progress and the last character of the last word is being transferred or when a DMT command decrements the memory cell from 0 to -1. Although the Echo signal is generated whenever the DMT decrements a cell from 0 to -1, a new interrupt is generated only if the DMT resulted from a previous interrupt.
- When an interrupt is generated by the API module, memory is addressed from the API module (G0SAMW) during the next Sequence State 1. This interrupt signal inhibits addressing memory from the P Register. G0SAMW enables memory address bit 7 (D1MA07) and in conjunction with the API Address Generator enables D1MA06-00 according to the interrupt level. In this manner an interrupt response address between 200g and 377g will be selected.

A detailed description of the API module is contained in the next section of this book set.

## Overflow Flip-Flop

The Overflow flip-flop, F1UOFL (logic sheet 54), provides arithmetic overflow detection during the execution of ADD, DVD, MPY, and SUB commands. In addition, the Overflow flip-flop is set during the execution of GEN 3 commands when bit 5 of the command word is a "one" if the exclusive "OR" of A23 and A22 is a "one".

Arithmetic overflow occurs when the result of an arithmetic operation provides, or will provide, a result whose magnitude exceeds the capacity of 23 bits for single word length register (i.e., 2<sup>23</sup>-1 or -2<sup>23</sup>) operations, or exceeds the capacity of 46 bits for double length register operations (i.e., 2<sup>46</sup>-1 or -2<sup>46</sup>). During GEN 3 commands, when bit 5 of the command word is a "one", overflow occurs when bit 23 (sign bit) of the A Register is changed during a left shift.

The status of the Overflow flip-flop is monitored and cleared by the JNO (GEN 2) command to establish program location control. The status of the Overflow flip-flop is stored by the SPB command; restoring of the Overflow flip-flop status may then be accomplished by the LPR command.

## Test Flip-Flop

The Test flip-flop (F1ETST) serves as a memory element to remember the status of a previous test and to retain this status until changed by the program. The

program may then use the status of the Test flip-flop to initiate a branch.

The Test flip-flop control logic provides the following functions:

- During the execution of DMT, GEN 1, and TXH commands, the Test flip-flop is used to remember the status of the condition tested.
- BTR and BTS commands will cause a branch in the program depending on the status of the Test flip-flop.
- The SPB command stores the status of the Test flip-flop; the status may be restored using the LPR command.
- The SET (GEN 1) command unconditionally sets the Test flip-flop; the RST (GEN 1) command unconditionally clears the Test flip-flop.

### Operators Console

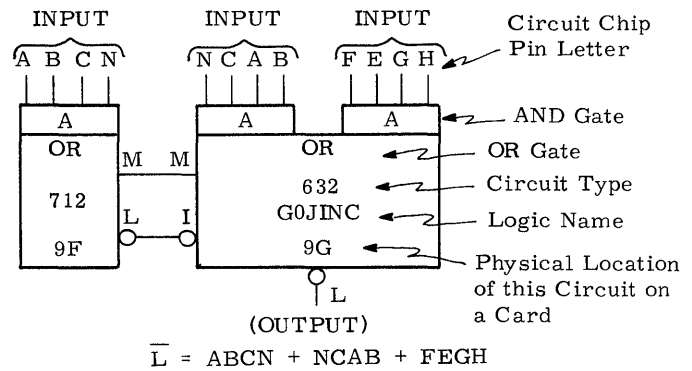
The Operators Console provides one method by which the operator may communicate with the computer in machine language. It allows the operator to load and execute programs and monitor the running program. A detailed description of the Operators Console is contained under the mnemonic CON in this section.

## LOGIC SYMBOLS AND NOMENCLATURE

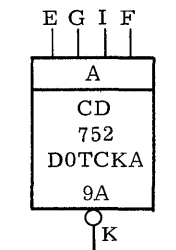
The following paragraphs provide a very brief discussion of the logic symbols and nomenclature used in the GE-PAC 4020 logic drawings.

Fig. DESC. 2 illustrates the most commonly used logic symbols and the logic equation associated with each symbol. Each symbol designates the integrated circuit chip pin numbers, the AND (A) and OR inputs, an identification number of the circuit chip type, the logic name of the logic element, and the physical location of the logic element on the circuit card. Each of these designations will be discussed in the following paragraphs.

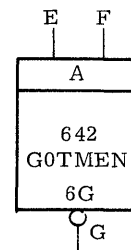
Since there are many different types of integrated circuit chips, each logic symbol identifies the type. This circuit type number is derived from the drawing number assigned to the individual circuit type. For instance, the drawing number for a 632 circuit is 68A8363P2; for a 752 circuit, 68A8375P2; for a 642, 68A8364P2; etc. Since the 68A83\_\_P\_\_ part of the drawing number is the same for all circuits, it is deleted from the circuit type number contained in the logic symbol.



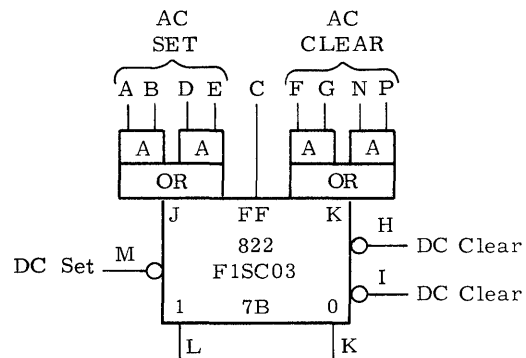
NAND Logic Symbol



$\bar{K} = E \cdot G \cdot I \cdot F$   
Clock Driver



$\bar{G} = E \cdot F$   
Two-Input NAND/NOR Gate



$$\bar{L} = [(A \cdot B + D \cdot E) \cdot C] + \bar{M}$$

$$\bar{L} = [(F \cdot G + N \cdot P) \cdot C] + \bar{H} + \bar{I}$$

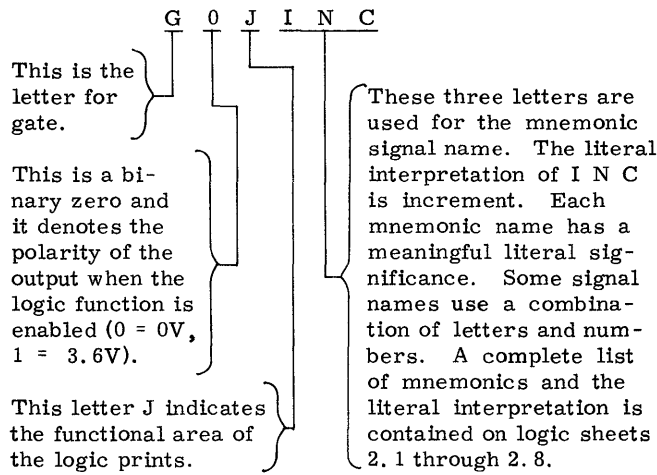
Flip-Flop (AND Inputs)

1 = +3.6V = True  
 0 = 0V = False

Fig. DESC. 2. Logic Symbols

The logic name consists of six digits to define four parts: Device Code, Polarity of output signal when the function is enabled, Functional Area of the logic prints, and the

mnemonic signal name. The logic name for the NAND symbol of Fig. DESC. 2 is described for clarity.



The physical location of this circuit on the card is designated by numbered and lettered co-ordinates of the circuit position of the card. Fig. DESC. 3 illustrates a circuit card and the co-ordinate designations.

The circuit chip pin letters specify the connections of the chip. Fig. DESC. 3 illustrates a chip and the pin designation locations.

Fig. DESC. 4 illustrates the wiring nomenclature and circuit card locations provided in the logic drawings.

The circuit card pin number defines the card connector and the pin number of that connector. As shown in Fig. DESC. 3, the connector may be defined by the lettered co-ordinate. Since each connector contains 14 pins, the remaining digits specify which pin of the connector.

The card location and size is also defined on the logic drawings as shown in Fig. DESC. 4. The first letter specifies the panel on which the card is located, the

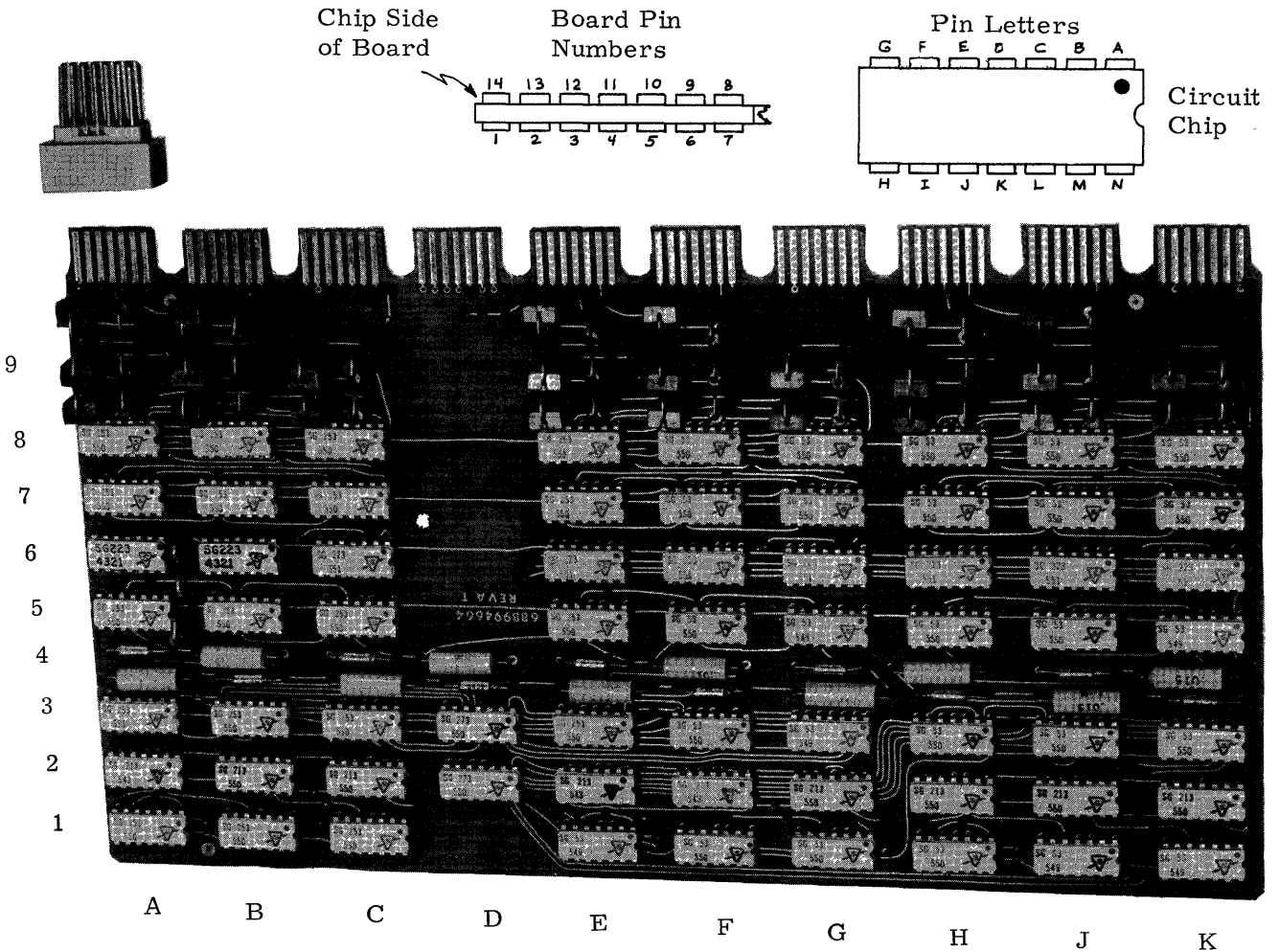


Fig. DESC. 3. Circuit Card



next two numbers specify the card slot of the panel that the card is located, and the last two letters define the card size.

A typical page will contain up to four panels, as shown in Fig. DESC. 5. The first letter then designates in which panel the card is located. The next two numbered digits specify the card slot within the panel, since 32 card slots may be provided in a panel. The last two lettered digits indicate the size of the card by indicating the connectors provided. Therefore, AK is a 10-inch card utilizing connectors A through K. The letters AE would indicate a 5-inch card using connectors A through E, and the letters FK would indicate a 5-inch card using connectors F through K.

Since wire connections must be made between the individual circuit cards, a wire list is used to specify the required connections. Fig. DESC. 6 illustrates a sample logic drawing and the wire list associated with the logic drawing. The wire list gives the wire number, the name of the signal connected by the wire, where the wire originates, the route it must take, and the destination of the wire for each connection.

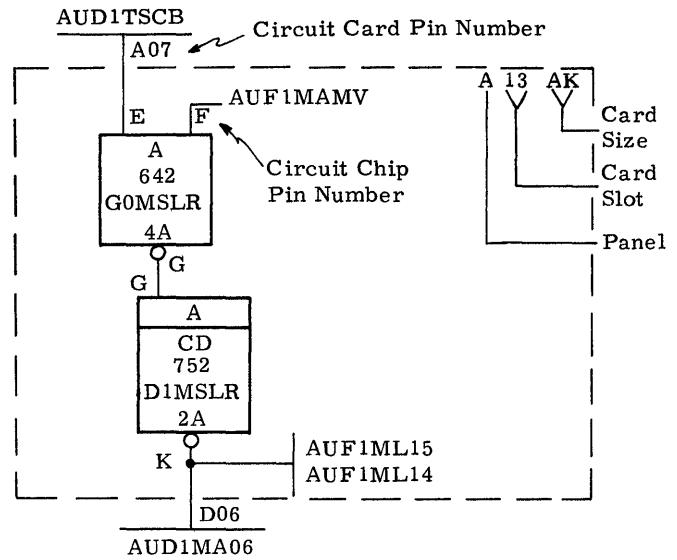


Fig. DESC. 4. Logic Wiring Nomenclature

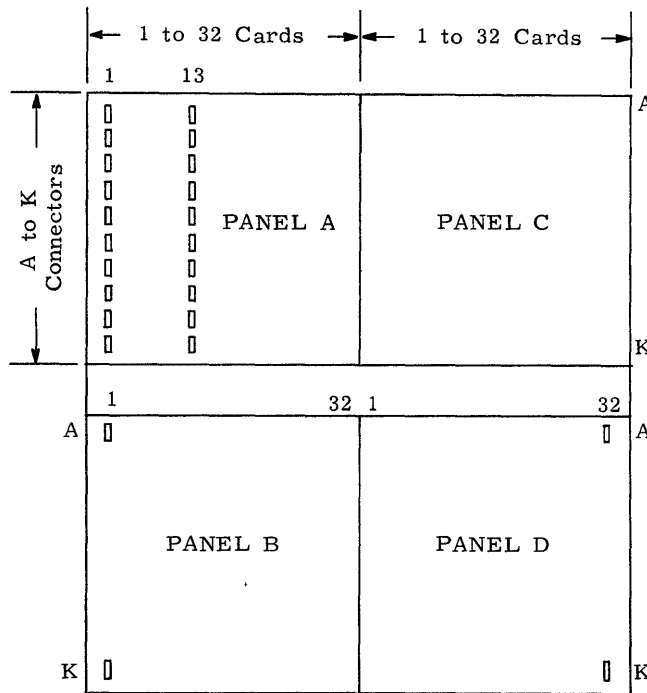


Fig. DESC. 5. Typical View of a Page

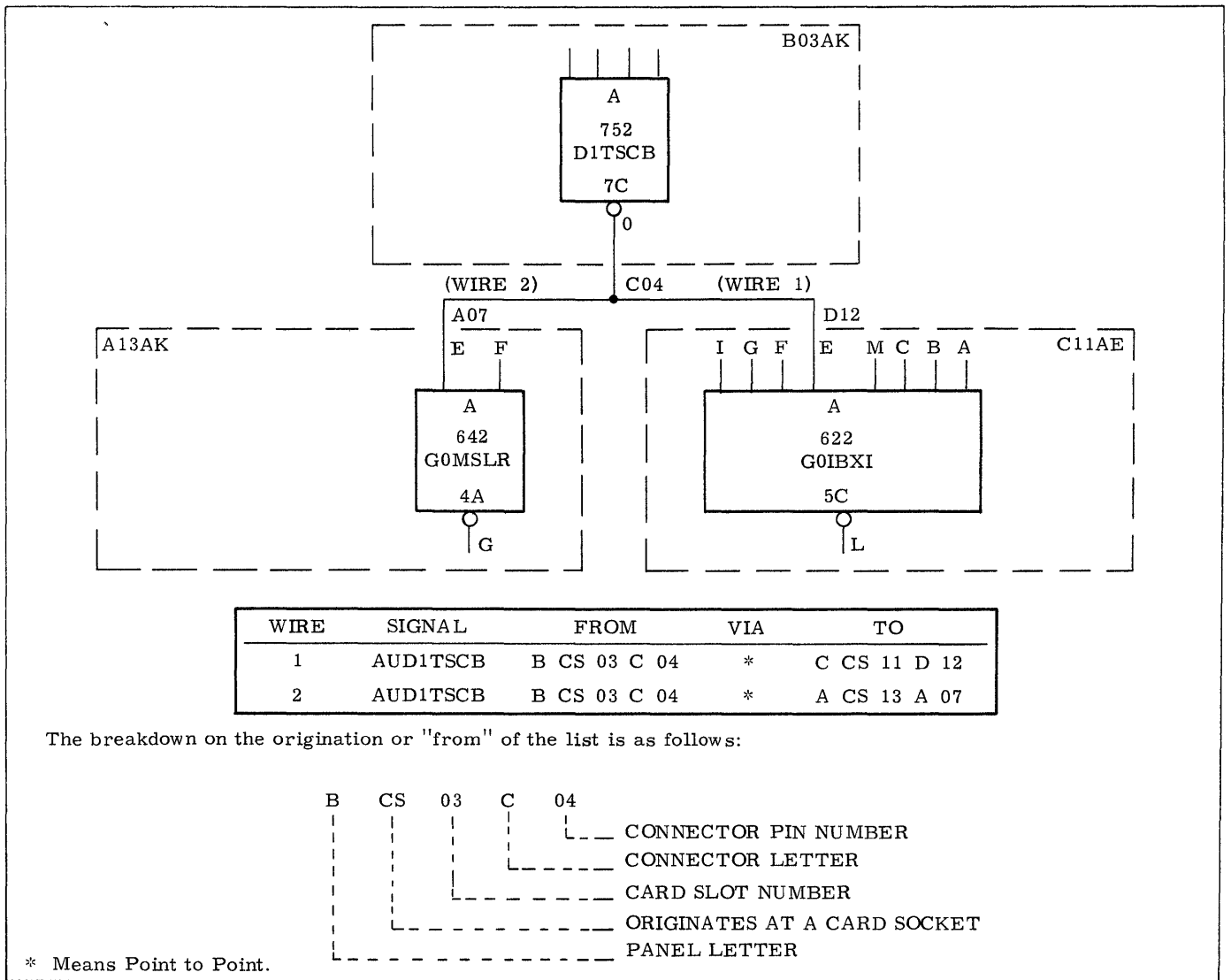


Fig. DESC.6. Wire List

# CONSOLE

## INTRODUCTION

The GE-PAC 4020 Console (Fig. CON. 1) is the basic man/machine communication link and control element of the Central Processor. It provides all the necessary indications and switches to apply power, initialize, load programs and monitor the operation of the machine.

As shown in Fig. CON. 1, additional controls and indicators are provided on the console as an option (4DP4022A0013). All of these controls and indicators are described in the following text. Logic for the Console is contained in Drawing 68C972367, sheets 125 thru 132. Table CON. 1 lists the operation of the controls and indicators and the functions performed by each. The purpose of this discussion is to acquaint the reader with the operation of the Console and the function of the switches and indicators contained thereon. Basic procedures to enter data in a register, read in punched cards or paper tape, store words in memory locations, display memory locations, and to store a constant throughout all core memory locations are also provided.

## Console to A, B

Using the Register Entry Pushbuttons, data may be entered in the A or B Register as determined by the Register Select Switch. Fig. CON. 3 contains a block diagram of the logic utilized to implement this function. With the Console Enabled, in the MANual mode, and the A or B Register selected, pressing the Register Entry pushbuttons sets the corresponding bits in the A or B Register. The status of the pushbuttons are transferred to the A or B Register via the Parallel Adder Unit. As illustrated in the Register Select discussion above, the contents of the selected register are also gated to the Parallel Adder Unit. Therefore, the A or B Register must first be cleared prior to pressing the Register Entry pushbuttons. Operation to clear the A or B Register is described below.

## CLEAR Register

The CLEAR Reg. switch, when pressed, clears the B or A Register depending upon which is selected by the Register Select switch. In the MANual mode with the

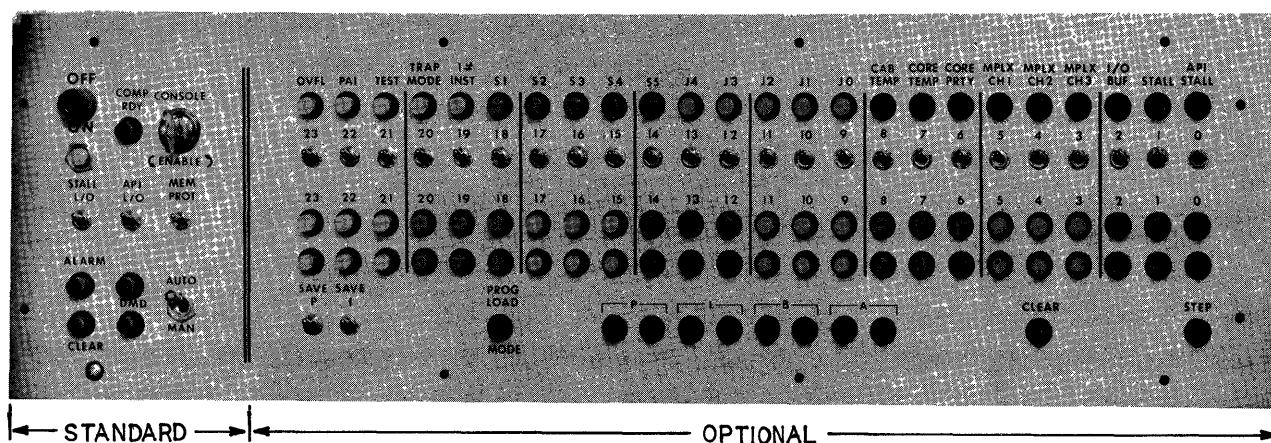


Fig. CON. 1. Operators Console

## DESCRIPTION

### Register Select-P,I,B,A:

In the MANual mode, with the Console Enabled, the contents of the register selected by the Register Select switches are displayed by the Register Display indicators. In this manner, the operator may observe the contents of the P, I, B, or A Registers.

Fig. CON. 2 contains a block diagram of the logic circuits involved. The contents of the particular register selected is gated to the Parallel Adder Unit during  $\overline{TSCA}$  when in the MAN mode, with the console enabled, and the CLEAR switch not pressed. The contents of the Parallel Adder Unit are then displayed by the indicators. Since sequencing is held in  $\overline{TSCA}$  between pressing the STEP switch (see description below), a display of the selected register is provided.

Console Enabled, and the A or B Register Selected, the status of the Register Entry switches is gated to the selected Register via the Parallel Adder Unit. Since none of the Register Entry switches are pressed, this will clear the selected register. As shown in Fig. CON. 1, pressing the CLEAR Reg. switch, disables gating the contents of the Selected Register to the Parallel Adder Unit. In this manner, since no data is applied to the Parallel Adder Unit, the A or B Register will be cleared.

### MANual STEP Operation

Pressing the STEP switch with the Console Enabled and in the MANual mode, will cause the instruction contained in the B Register to be executed. Instruction sequencing is halted after executing this instruction with the instruction just executed in the I Register and the next instruction in the B Register.

Fig. CON. 4 illustrates the basic timing, with logic

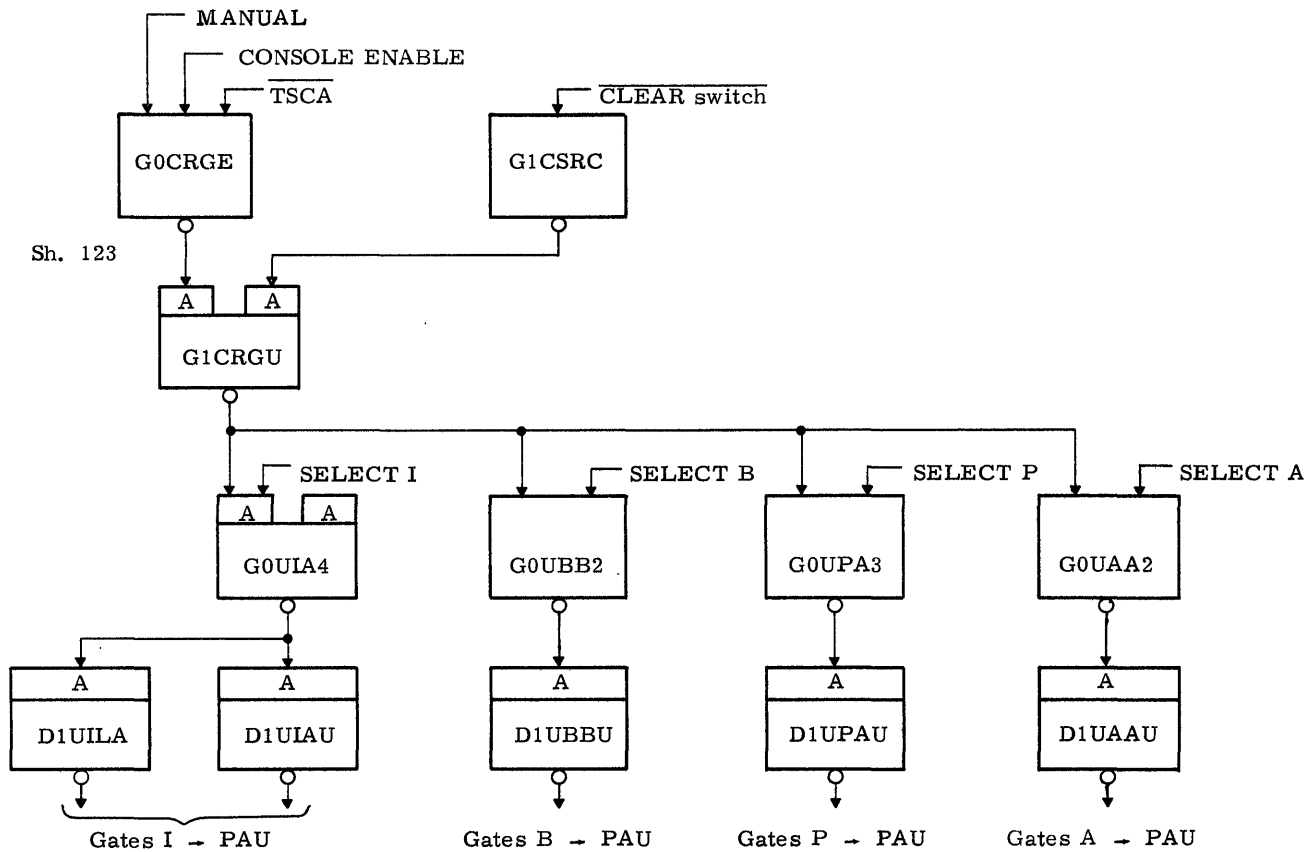


Fig. CON. 2. Register Select Block Diagram

equations, of the MANUAL STEP operation. When in the MANUAL mode, sequencing is halted in State 1, time 2 envelope. Pressing and releasing the STEP switch will allow the instruction contained in the B Register to be executed. Sequencing will then be halted during time 2 envelope of the following State 1. In the MANUAL Mode, sequencing is held in time 2 envelope by enabling G1TMEN and disabling G1CTAE during State 1. Pressing and releasing the STEP switch will enable G1CTAE and disable G1TMEN and allow sequencing to continue until the next State 1.

## CONSOLE PROCEDURES

The following procedures are provided to assist the operator of the Programming and Maintenance Console. To use these procedures, the Console must be enabled by both the key switch and CONSOLE OFF toggle switch (CSWOCE).

### Enter Data in A or B Register

1. Place MAN - AUTO toggle switch to MAN.
2. Select desired Register by pressing A or B Register Select switch.
3. Press the CLEAR Reg. switch.
4. Press the Data Entry pushbutton switches for desired bit pattern.

### Store a Word in Core Memory Location Y

1. Place MAN - AUTO toggle switch to MAN.
2. Press A Register Select switch.
3. Press CLEAR Reg. switch.
4. Press the Data Entry pushbuttons for desired bit pattern.
5. Press B Register Select switch.
6. Press CLEAR Reg. switch.
7. Using Data Entry pushbuttons, set a STA (320Y) in the B Register.
8. Press STEP switch.

### Display the Contents of Memory Location Y

1. Place MAN - AUTO toggle switch to MAN.
2. Press B Register Select Switch.
3. Press CLEAR Reg. Switch.
4. Press the Register Entry pushbuttons to set a LDA (000Y) into the B Register.
5. Press STEP switch.
6. Press A Register Select Switch. The contents of the addressed core cell will be displayed by the indicators.

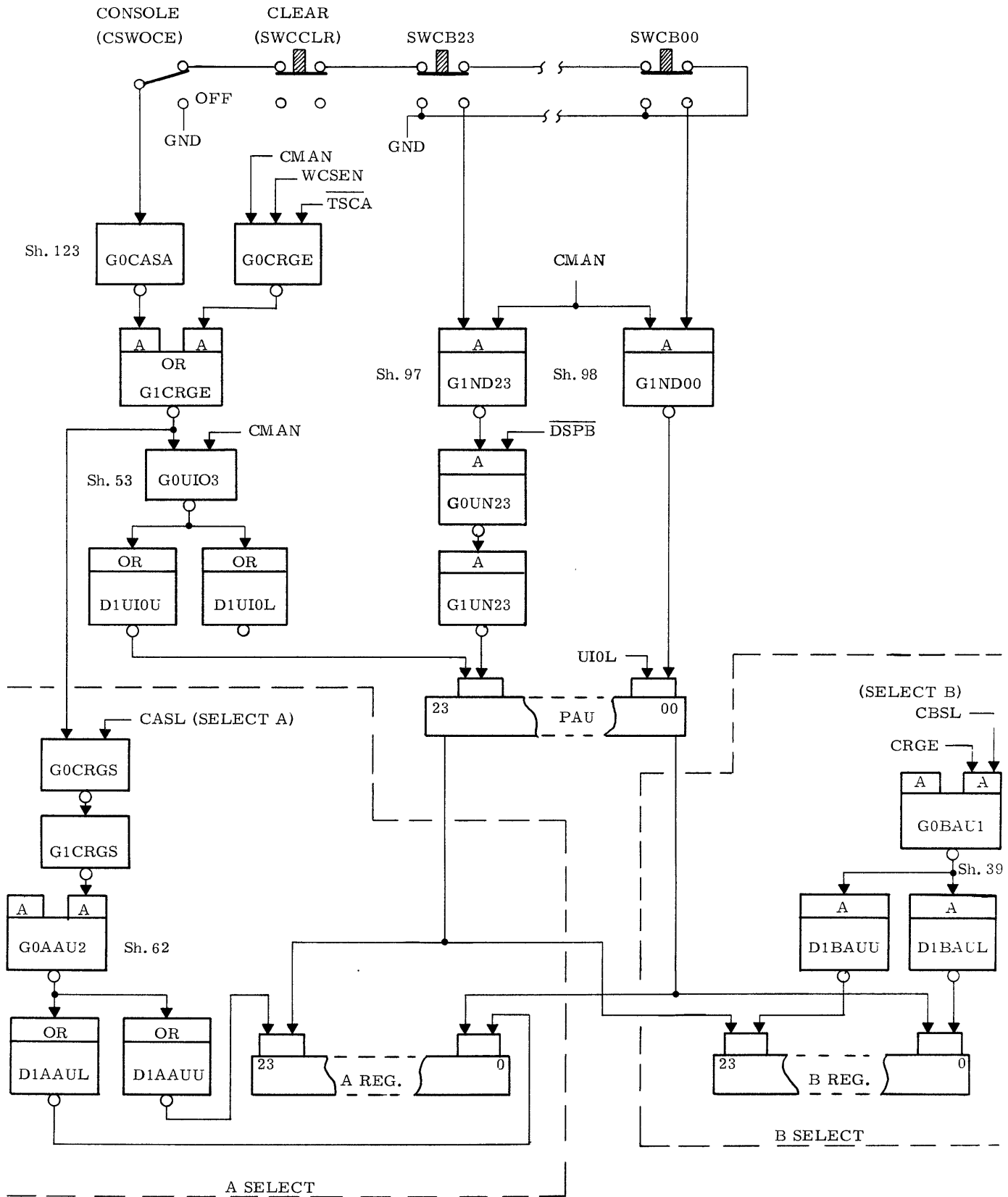
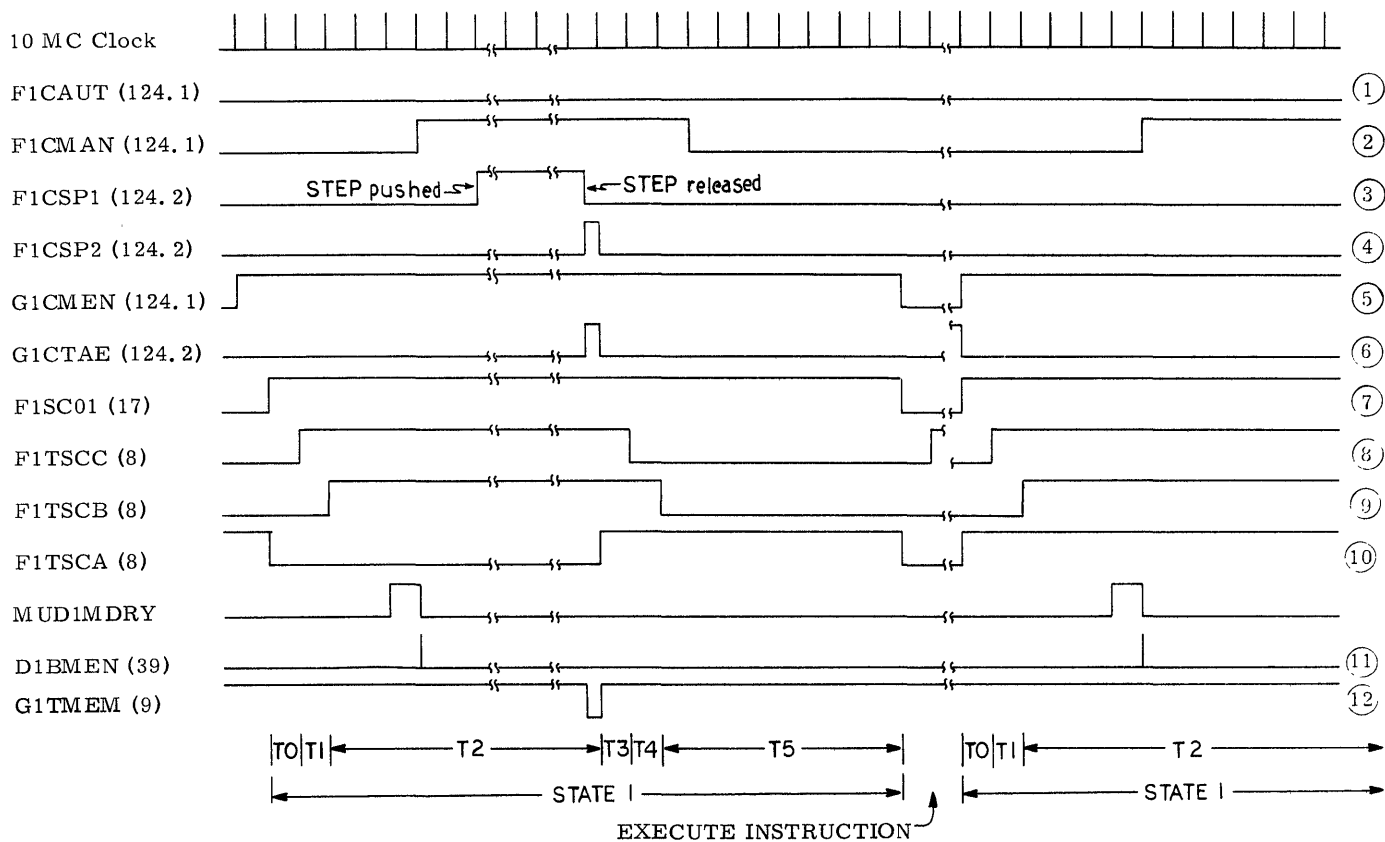


Fig. CON. 3. Console To A, B Block Diagram



- ①  $F1CAUT = \overline{CAUM} \cdot CCLK \cdot (\overline{MUFOPPER} + \overline{STOP\ ON\ PARITY\ ERROR})$   
 $SWCAUM = MANUAL \cdot CONSOLE\ ENABLE$
- ②  $F1CMAN = \overline{CAUT} \cdot MDRY \cdot CMEN \cdot CCLK$   
 $G1CMEN = SC01 \cdot \overline{MVDA}$   
 $\overline{F1CMAN} = TSCB \cdot TSCA \cdot CCLK$
- ③  $F1CSP1 = \overline{SSTPN0} = \text{STEP Switch pushed}$   
 $\overline{F1CSP1} = SSTPNC = \text{STEP Switch released}$
- ④  $F1CSP2 = \overline{CSP2} \cdot \text{Clocked by CSP1}$   
 $\overline{F1CSP2} = CSCA$
- ⑤  $G1CMEN = SC01 \cdot \overline{MVDA}$
- ⑥  $G1CTAE = TSCB (CSP2 + \overline{CMEN})$
- ⑦  $F1SC01 = SSS1 \cdot TLPE \cdot SCLK$
- ⑧  $F1TSCC = TSC2 \cdot \overline{TSCB} \cdot TCK2$   
 $G1TSC2 = \overline{TSCA}$   
 $\overline{F1TSCC} = TSCB \cdot TSCA \cdot TCK2$
- ⑨  $F1TSCB = \overline{TSCA} \cdot TSCC \cdot TCK2$   
 $\overline{F1TSCB} = \overline{TSCC} \cdot TCK2$
- ⑩  $F1TSCA = \overline{TMEN} \cdot TSCB \cdot TCK2$   
 $\overline{F1TSCA} = TLPE \cdot TCK2$
- ⑪  $D1BMEN = MDRY \cdot BMRQ \cdot \overline{STOR} \cdot BCLK$
- ⑫  $G1TMEN = CMAN \cdot CTAE$

Fig. CON. 4. Manual Mode Step Sequencing

#### To Coat Core Memory With A Constant

1. Place MAN-AUTO toggle switch to MAN.
2. Press B Register Select switch.
3. Press CLEAR Reg. switch.
4. Press Register Entry switches for 07100001 bit pattern (LXK 1, 1).
5. Press STEP switch.
6. Press CLEAR Reg. switch.
7. Press Register Entry switches for 32100001 bit pattern (STA 1, 1)
8. Press A Register Select switch.
9. Press CLEAR Reg. switch.
10. Press Register Entry switches for the desired constant.
11. Press STEP switch.
12. Place SAVE I switch in save position.
13. Place MAN-AUTO toggle switch to AUTO.
14. The constant will now be stored in each memory location.
15. Place MAN-AUTO switch in MAN.
16. Lower SAVE I switch.

#### To Handload Instructions into Sequential Core locations Y, Y + 1, Y + 2, ... (Y ≥ 2).

1. Place MAN-AUTO toggle switch to MAN.
2. Press B Register Select switch.
3. Press CLEAR Reg. switch.
4. Press Register Entry switches for 07100001 (LXK 1, 1) bit pattern.
5. Press STEP switch.
6. Press CLEAR Reg. switch.
7. Press Register Entry switches for 321XXXXX (STA Y-1, 1) where XXXXX = Y - 1.
8. Press A Register Select switch.
9. Press CLEAR Reg. switch.
10. Press Register Entry switches to bit configuration of first instruction to be stored.
11. Press STEP switch.
12. Place SAVE I switch in save position.
13. For each instruction to be stored:
  - a. Press CLEAR Reg. switch.
  - b. Enter instruction in Register Entry switches.
  - c. Press STEP switch.

The I Register contains the address of the last location stored.

#### Manually Display The Contents of Sequential Memory locations Y, Y + 1, Y + 2, ... (Y ≥ 2).

1. Place MAN-AUTO toggle switch to MAN.
2. Press B Register Select switch.
3. Press CLEAR Reg. switch.
4. Press Register Entry switches for 07100001 (LXK 1, 1) bit pattern.
5. Press STEP switch.
6. Press CLEAR Reg. switch.
7. Press Register Entry switches for 001XXXXX (LDA Y-1, 1) where XXXXX = Y - 1.
8. Press STEP switch.
9. Place SAVE I switch in save position.
10. Press A Register select switch. Contents of selected memory location is displayed by Register Display indicators.
11. Press STEP switch for each sequential location display. The I Register contains the address of the location being displayed.

#### Load a Bootstrap Routine

The following procedure may be used to enter or load into memory a "Bootstrap" load routine contained by paper tape or punched tape. This routine assumes that the Bootstrap routine is assembled so that the first instruction is to be stored in memory location one.

1. Place MAN-AUTO toggle switch to MAN.
2. Ready the tape or card reader by inserting the proper tape or card and placing the device on-line.
3. Switch the API Lockout switch to the Lockout position.
4. Initialize computer by depressing the ON push-button.
5. Switch the Save-P switch to the Save-P position.
6. Hand load a "LDA" instruction into location zero.
  - a. Press B Register Select switch.
  - b. Press CLEAR switch.
  - c. Press Register Entry switches for 32000000g (STA) bit pattern.
  - d. Press A Register Select switch.
  - e. Press CLEAR switch.
  - f. Press STEP switch.
7. Hand load a "LDA" instruction into noninhibitible clock and DMT API response addresses.
  - a. Press B Register Select switch.
  - b. Press CLEAR switch.

- c. Press Register Entry switches for STA to noninhibit clock (and DMT) API location.
  - d. Press A Register Select switch.
  - e. Press CLEAR switch.
  - f. Press STEP switch.
8. Hand load the TIM Control Word into the corresponding API response address.
- a. Press Register Entry switches for "00000000" bit configuration for paper tape or to "00500000" bit configuration for cards.
  - b. Press B Register Select switch.
  - c. Press CLEAR switch.
  - d. Press Register Entry switches for "32000202" (paper tape) or "3200020" (cards).
  - e. Press STEP switch.
9. Hand load OPR command into B Register.
- a. Press Register Entry switches for 2502710D (D = Device Address)
10. Hand load DC1 code into A Register.
- a. Press A Register Select switch.
  - b. Press CLEAR switch.
  - c. Press Register Entry switches for "00000021" bit configuration.
11. Permit API's by placing API switch in non-lockout position.
12. Place AUTO/MAN switch to AUTO.
13. After card or read operation, place SAVE P switch in non-save position.
- Bootstrap load program reads into memory starting at location one.

**NOTE**

Step 7 may be omitted by clearing core memory prior to step 1.



CONTROL OR INDICATOR	ACTIVE IN MODES MANUAL, AUTO OR CONSOLE OFF	PUSHBUTTON OR TOGGLE SWITCH	INDICATOR	LATCHING OR MOMENTARY	DESCRIPTION
API STALL (DRCD00)	ALL MODES		X		Comes on when API Watchdog timer times out.
API L/O (SWCAPI)	AUTO & MAN	T		L	Inhibits all API's and disables API Watchdog in the lockout mode. CONSOLE OFF enables API and Watchdog. Refer to the API description in this book set.
AUTO/MAN (SWCAUM)	AUTO & MAN	T		L	In AUTO mode, computer sequencing enabled. Computer runs automatically. In MAN mode computer sequencing occurs each time STEP switch is depressed. . . For one instruction. CONSOLE OFF forces AUTO mode. Refer to Manual Mode Description.
ALARM (DRCERR)	ALL MODES			X	Comes on when alarm or error condition exists. Turned off by CLEAR ALARM Switch or Initialize.
CAB. TEMP. (DRCD08)	ALL MODES			X	Comes on when the heat sink in the Bulk Power Supply or Memory Power Supply reaches $260^{\circ} \pm 5^{\circ}$ F. When the temperatures of the heat sink reaches $287^{\circ} \pm 5^{\circ}$ F, the power supplies will stop operating, thus the system is shut down.
CLEAR ALARM (SWCERR)	ALL MODES	PB		M	Clears all error/alarm conditions that are Ored to the ALARM indicator.
CLEAR REG. (SWCCLR)	MAN	PB		M	Will clear the A or B Register as selected on the Reg. Select switches. Refer to Clear Register description.
COMP RDY(DRCRDY)	ALL MODES			X	Comes on when power is applied. Turns off when power is removed. Displays 5V logic only. Refer to the Power Distribution section of this book set for details.
CORE PRTY (DRCD06)	ALL MODES			X	Comes on when a memory parity error is detected, CLEAR ALARM resets Parity Error FF and turns off light. JNP and initialize also clears the flip-flop and turns off the light.
CORE TEMP (DRCD07)	ALL MODES			X	Comes on when core stack temperature (4015B) is not within allowable range. Turns off when temperature is within allowable range. If parity error occurs during out of limits time memory sequencing is inhibited.
CONSOLE ENABLE (SWCSEN)	AUTO & MAN	KEY		L	When in the disable position only the DMD, RCS Switches, indicators, ALARM CLEAR and OFF are active. Contacts 10, 11, and 12 may be used to provide optional external display or control.
DMD (LCDMD)	ALL MODES		X	M	Sets Demand Flip-Flop when depressed and released. Demand Flip-Flop is cleared and light turned off by JND or initialize. Refer to JND command and description.

Continue On Next Page

Table CON. 1. Controls and Indicators

Table CON. 1. (Cont.)

CONTROL OR INDICATOR	ACTIVE IN MODES MANUAL, AUTO OR CONSOLE OFF	PUSHBUTTON OR TOGGLE SWITCH	INDICATOR	LATCHING OR MOMENTARY	DESCRIPTION
I $\neq$ INST (DRCD19)	ALL MODES		X		Comes on at completion of Multiply/Divide or prior to, and upon completion of TIM/TOM to indicate I Register does not contain last instruction. Cleared when next instruction is executed.
I/O BUF (DRCD02)	ALL MODES		X		Comes on when a parity or timing error occurs in the I/O Buffer. Cleared by JNE.
J <sub>4</sub> - J <sub>0</sub> (DRCD13-09)	ALL MODES		X		Displays the contents of the J Counter. The J Counter is affected by GEN 1, GEN 3, MPY, DVD, and TIM/TOM operations.
MEM PROT (SWCMEN)	ALL MODES			L	Disables Memory Protect when in Lockout position, when in the non-lockout position Memory Protect is active, if TMFF is set.
MPLX CH. 1, CH. 2, CH. 3 (DRCD03, 4, 5)	ALL MODES		X		Comes on when associated device produces a parity error. Turned off by initialize or OUT command.
Overflow (OVFL) (DRCD23)	ALL MODES		X		Comes on when Arithmetic Overflow occurs. Turned off by Initialize, JNO or LPR commands. (See LPR & JNO commands.)
OFF (SWCOFF)	ALL MODES	T		M	Removes DC power from CPU. Refer to Power Distribution section of this book set.
ON/Initialize (SWCSON)	MAN	T		M	Applies DC power to CPU and Initializes CPU. Initialize clears B, P, I Reg., S <sub>2-5</sub> , Demand, PAI, All Alarms, and priority interrupt flip-flops. Refer to the Power Distribution description of the book set.
PAI (DRCD22)	ALL MODES		X		Comes on when PAI command is executed to indicate interrupts are permitted. Turned off by IAI, LDP, LPR, SPB commands and Initialize. See LPR, LDP, SPB command description.
PROG LOAD (SWCPGL)	MAN & AUTO	PB		M	Not implemented at this printing. Refer to Bootstrap load routine contained in this section.
Reg. Select A, B, I, P (SWCSLA, B, I, P)	MAN		X	L	Each switch selects its associated register. Used in conjunction with the register entry and display switchlights. All Registers may be displayed but only A & B may receive data from the Reg. Entry switches. Activating any one of the Reg. Select Switches disables the others. Refer to Register Select discussions.
Reg. Entry/Display (SWCB00-23)	MAN	PB	X	M	The Switches are used in conjunction with the Reg. Select Switch to enter data into A or B Register. The lights will display the contents of the register selected (I, P, B, or A) by the Register Select Switch. Refer to Register Select Discussions.

Continue on Next Page

Table CON. 1. Controls and Indicators

Table CON. 1. (Cont.)

CONTROL OR INDICATOR	ACTIVE IN MODES MANUAL, AUTO OR CONSOLE OFF	PUSHBUTTON OR TOGGLE SWITCH	INDICATOR	LATCHING OR MOMENTARY	DESCRIPTION
Read Console Sw's. (SWCC00-23)	ALL MODES	T		L	Used in conjunction with the RCS command to enter the contents of these 24 switches into the A Register. The up position represents a 0, while the down position equals a 1. Refer to the RCS command descriptions.
SAVE P (SWCSSP)	AUTO & MAN	T		L	When in the up position the P Register cannot be altered. When in the down position, P operates normally. CONSOLE OFF disables the SAVE P function.
SAVE I (SWCSSI)	AUTO & MAN	T		L	When in the up position the I Register cannot be altered during the instruction fetch cycle. In the down position I operates normally. CONSOLE OFF disables the SAVE I function.
STEP (SWCSTP)	MAN	PB		M	In the MAN mode, each depression and release causes the computer to execute one instruction. Refer to the Manual Mode description.
S <sub>1</sub> - S <sub>5</sub> (DRCD18-14)	ALL MODES		X		These indicators display the 5 sequence states of the CPU.
STALL (DRCD01)	ALL MODES		X		Comes on when Stall Alarm timer times out. May be cleared by Initialize, Stall L/O or the SSA instruction.
STALL L/O (SWCSTL)	AUTO & MAN	T		L	When in the Lockout mode the Stall Alarm is disabled. CONSOLE OFF enables STALL ALARM. Refer to the Stall Alarm description.
TRAP MODE (DRCD20)	ALL MODES		X		Comes on when trap or watchdog error occurs. Turned off by SPB executed from API response address 20 <sub>8</sub> or 24 <sub>8</sub> or Initialize. Refer to Memory Protect descriptions.
TEST (DRCD21)	ALL MODES		X		Comes on when Test flip-flop is set. The Test flip-flop can be set by: DM T, TXH, some GEN 1 commands and LPR.  The Test flip-flop may be reset by DM T, TXH, some GEN 1 instructions, LPR, and Initialize. Refer to these command descriptions.

Table CON. 1. Controls and Indicators

# COMMANDS

The following discussion is provided to assist the reader in a more thorough understanding of the individual command descriptions that follow.

## COMMAND FORMATS

The commands, used by the GE-PAC 4022A Arithmetic Unit fall into five categories: Full Operand, GEN 1, GEN 2, GEN 3, and Quasi. The microcoded format of each of these command types is shown in Fig. CMD. 1.

### Full Operand

The Full Operand commands are the most widely used commands of the Arithmetic Unit. These commands are used to perform arithmetic operations, logical operations, index control operation, and data transfers to and from memory. These commands designate the core address for information in memory as an operand address. The operand address is a full 14 bits (13-0), which may be augmented by relative addressing (14) and/or indexing (17-15).

A detailed discussion of each of these commands is provided in alphabetical order. Each command description provides the mnemonic associated with the command, the command type, format, and a description of the command usage. In addition, a specific hardware oriented discussion is provided including a block diagram of the registers used, and a timing diagram including logic equations, of the more important control signals enabled within the Arithmetic Unit.

### GEN 1

GEN 1 commands are differentiated from other commands by the OP Code (bits 23 through 18) 05g. GEN 1 commands are further subdivided into commands by the microcoding of the operation portion (bits 14 through 0) of the command word. GEN 1 commands are used primarily for bit manipulation of the A Register, but may be microcoded to affect the J Counter and Test flip-flop.

Thirty-seven different GEN 1 commands are described under the mnemonic GEN 1. Each description contains the microcoded format of the command, a description of

the command usage, and a description of the hardware operation when executing the command.

### GEN 2

GEN 2 commands are differentiated from other commands by the OP Code (bits 23 through 18) 25g. GEN 2 commands are also subdivided into commands by the microcoding of the operation portion (bits 14 through 0) of the command word. These commands are employed by the GE-PAC system to: (1) select modules and devices in the input/output equipment, (2) transfer data to or from these devices, and (3) provide for program control transfers (jump) as determined by various internal and external conditions.

Twenty-one different GEN 2 commands are described under the mnemonic GEN 2. Each description contains the microcoded format, discussion of the command usage, and a detailed discussion of the hardware operation when executing the command.

### GEN 3

GEN 3 commands are differentiated from other commands by the OP Code 45g. GEN 3 commands are also subdivided into commands by the microcoding of the operation portion of the command word. These commands are used to manipulate the contents of the A and Q Registers, affect the J Counter, and optionally affect the Overflow flip-flop.

Eight individual GEN 3 commands are described in detail under the mnemonic GEN 3.

### Quasi

Quasi commands provide operations not included in the hardware. These commands supply the programmer with a mnemonic which allows the running program to be linked with a subroutine. Quasi commands are identified by OP Codes 40g - 77g if the command is not an MPY, DVD, LDQ, STQ, or GEN 3.

The Quasi command word is divided into four areas: bits 23 through 18, the next command address; bits 17

	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FULL OPERAND	OP					CODE					INDEXING ADDRESS	*	OPERAND											
GEN 1	0					5					INDEXING ADDRESS	C	A	B	T	S	K							
GEN 2	2					5					INDEXING ADDRESS	S			K3			K2			K1			K0
GEN 3	4					5					INDEXING ADDRESS	SEE GEN 3 DESCRIPTION												K
QUASI	40g - 77g EXCEPT MPY, DVD, LDQ, STQ, OR GEN 3						INDEXING ADDRESS	*	OPERAND															

\*Relative Address Bit

Fig. CMD. 1. Command Formats

through 15, index address modification; bit 14, relative addressing; and, 13 through 0, the operand.

When a Quasi command is "fetched" from memory, indexing and/or relative address modification of the operand portion of the command is performed if specified. After modification, if required, the operand portion of the Quasi command is stored in index cell 2. The next instruction is then addressed from the OP Code portion of the Quasi command. Therefore, program control is transferred to a memory location between 40g and 77g. This location will normally contain an SPB command to a subroutine to accomplish the operation. This subroutine may, of course, use the contents of index cell (operand portion of the Quasi command) as an operand address or data value.

A detailed description of the Quasi command is contained under the mnemonic QSI.

## BASIC TIMING OF FULL OPERAND COMMANDS

The following paragraphs describe the basic Arithmetic Unit timing for full operand commands. The basic timing for GEN 1, GEN 2, GEN 3, TIM/TOM, and the execution states of MPY and DVD are described with the command description of these commands.

Because the basic timing is identical for all Sequence States of most commands, the command descriptions that follow this section do not include this timing except where differences exist.

The basic clock signal used throughout the 4020 system is generated from a 10 megacycle crystal controlled oscillator (sheet 7). The output of the oscillator is applied through a variable  $25 (\pm 10\%)$  nanosecond single shot circuit. Therefore, out of the single shot,  $25 \pm 10\%$  nanosecond pulses occurring at a 10 megacycle rate are applied through clock drivers to the system.

Sequential timing within the Arithmetic Unit is controlled by a 3-state gray code counter (F1TSCA, F1TSCB, and F1TSCC). The condition of these flip-flops (set or clear) are used to specify specific time intervals to perform various functions. Six different time intervals are specified by this counter during the execution of full operand commands. For ease of description, the time intervals are called Time 0 Envelope through Time 5 Envelope. The count sequence and corresponding time envelope nomenclature are:

<u>F1TSCA</u>	<u>F1TSCB</u>	<u>F1TSCC</u>	
0	0	0	Time 0 Envelope
0	0	1	Time 1 Envelope
0	1	1	Time 2 Envelope
1	1	1	Time 3 Envelope
1	1	0	Time 4 Envelope

1	0	0	Time 5 Envelope
1	0	1	Time 6 Envelope - This timing envelope is entered only during execution of MPY, DVD, TIM/TOM, and GEN commands. Refer to these command descriptions for details.

Fig. CMD.2 contains a timing diagram and logic equations of the Sequence Time Counter. This timing diagram applies to all commands except the execution states of GEN 1, GEN 2, GEN 3, MPY, DVD, and TIM/TOM operations.

Last Pulse (D1TLPE) of any Sequence State or the Initialize signal (DOTINT) clears the gray code counter forming Time 0 Envelope. Clearing F1TSCA enables F1TSC2. With TSC2 enabled, the first clock pulse sets F1TSCC, forming Time 1 Envelope. The next clock pulse sets F1TSCB, forming Time 2 Envelope. Time 2 Envelope is enabled until Data Ready (MUD1MDRY) is received from the core memory module. This normally provides a duration of 6 clock pulses. The clock pulse during Data Ready sets F1TSCA, forming Time 3 Envelope. The next clock pulse clears F1TSCC, forming Time 4 Envelope. The next clock pulse clears F1TSCB, forming Time 5 Envelope. Time 5 Envelope is enabled until Last Pulse is generated. Last Pulse is generated from the Memory Release signal (MUD1MRLS), applied by the Core Memory module at the end of the memory timing cycle. This provides Time 5 Envelope with a duration of approximately 6 clock pulses. Last Pulse Envelope clears the Sequence Time Counter for the next timing cycle. The duration of Time 0 through Time 5 is approximately 1.6 microseconds.

As described above, the duration of Time 2 Envelope and the duration of Time 5 Envelope is dependent upon the memory cycle. The memory cycle begins with Memory Request (G1SMRQ) when access to memory is granted by the memory priority scheme. Memory Request is normally enabled by the Memory Release (MUD1MRLS) signal generated in the memory. If, however, the previous or next cycle does not require memory access (i. e., State 4 of GEN 1 or GEN 2 command, or when State 4 is extended for the execution of GEN 3, MPY, DVD, or TIM/TOM), the Memory Request signal is inhibited until Time 0 Envelope. In either condition, a time duration of 1.7 microseconds is required by the memory to determine access priority and to complete the memory cycle. Therefore, if Memory Request is inhibited until Time 0, the duration of Time 2 Envelope will be extended by one clock pulse while awaiting Data Ready. This is, of course, a comparatively rare occurrence.

## Instruction Sequencing

All instructions performed by the 4022A Arithmetic Unit follow a definite set pattern or sequence for "fetching"

the instruction from memory, performing index address modification if required, and executing the instruction. Sequence Control State flip-flops F1SC01, F1SC02, F1SC03, F1SC04, and F1SC05 are provided to control this pattern or sequence.

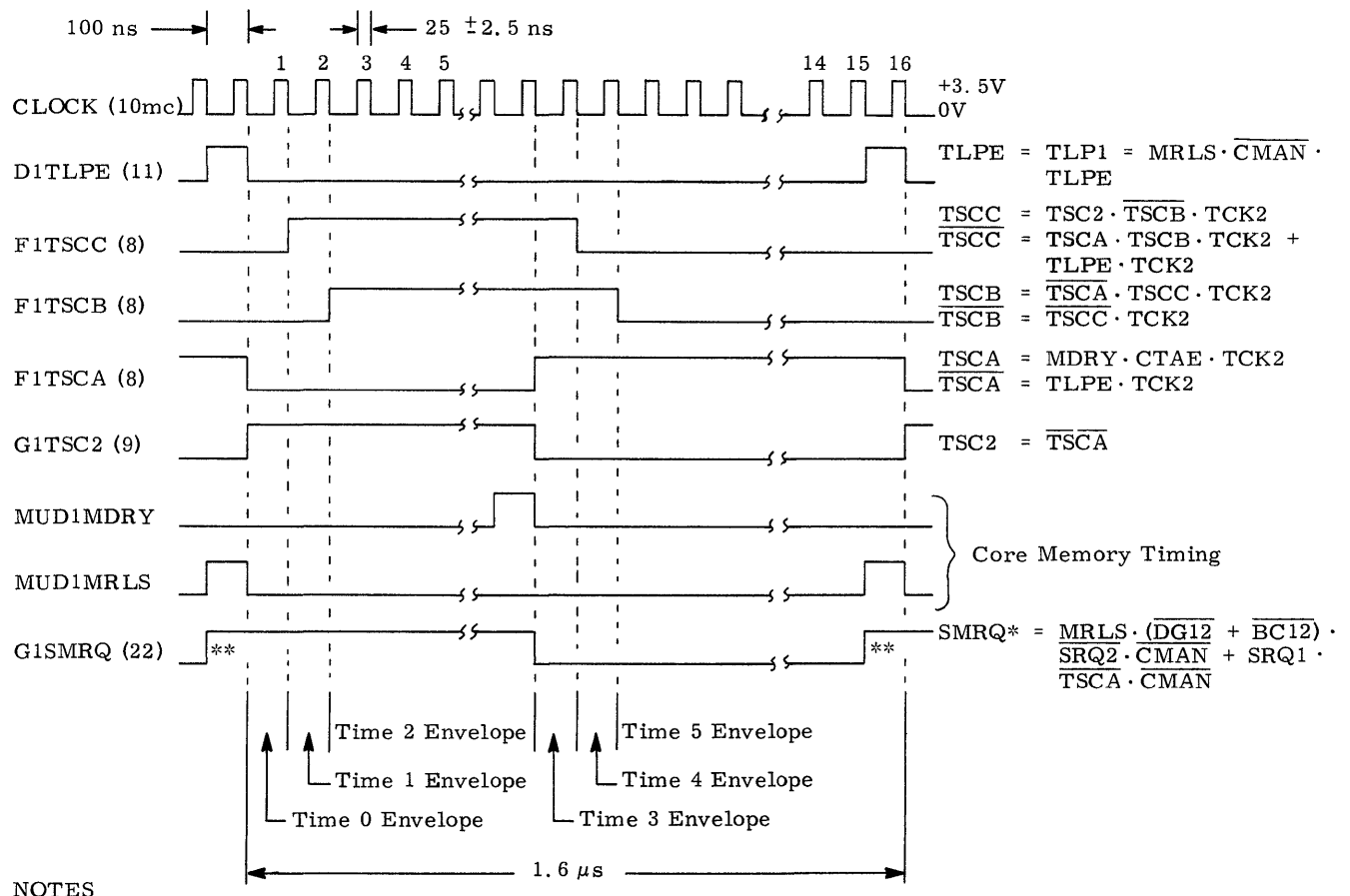
Fig. CMD.3 graphically illustrates the Sequence States required to fetch, index, and execute any instruction. Briefly, the function performed during each Sequence Control State is described below.

- Sequence Control State 1 - F1SC01: During this Sequence State, all commands are "fetched" from memory. Non-indexed XEC, BRU, BTR, and BTS commands are also executed during Sequence Control State 1.
- Sequence Control State 2 - F1SC02: During Sequence State 2 index address modification occurs. Also, STX, TXH, and DMT commands use State 2 for a portion of the execution.

- Sequence Control State 3 - F1SC03: State 3 is used during the execution of MPY, DVD, STQ, STX, and TIM/TOM operations.
- Sequence Control State 4 - F1SC04: State 4 is the execution state for most commands.
- Sequence Control State 5 - F1SC05: State 5 is used during MPY, DVD, GEN 3, STQ, LDQ, and TIM/TOM instructions to complete their execution cycle.

### Sequence Control State 1

Sequence Control State 1 defines the "fetch" cycle for all instructions. Because State 1 is nearly the same for all commands, it is not described in the command descriptions later in this section unless it performs unique functions (e.g., XEC, BRU, TXH, BTR, and BTS). Therefore, the following discussion describes the detailed operation of the Arithmetic Unit during State 1. This discussion applies to all commands.



\* Memory Request is not enabled during State 4 of GEN 1 and GEN 2 commands.

\*\* Memory Request is not enabled until Time 0 when following a Sequence State that does not use memory or a Sequence State that is extended (i.e., following State 4 of GEN. 1, 2, 3, MPY, DVD, or TIM/TOM). When Memory Request is not enabled until Time 0, Time 2 is extended by 1 clock pulse awaiting data ready.

Fig. CMD. 2. Full Operand Basic Timing Diagram

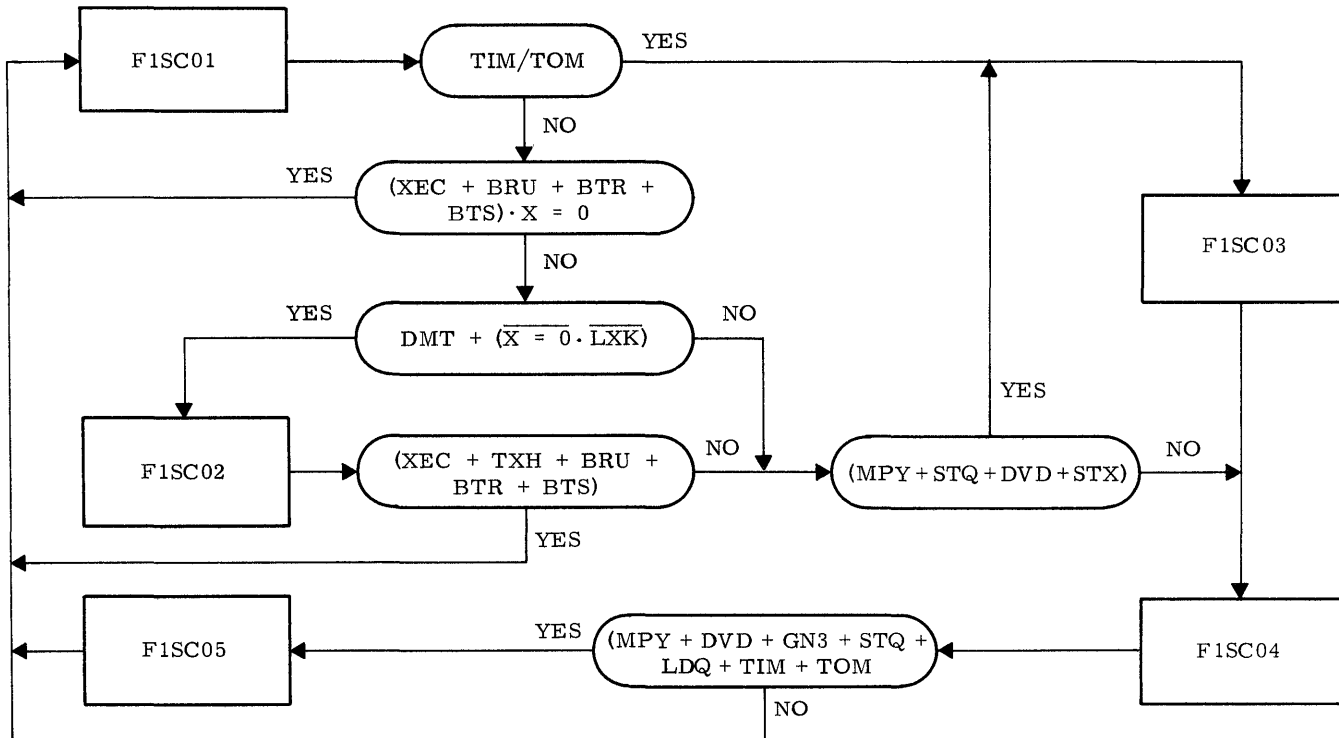


Fig. CMD. 3. Instruction Sequencing

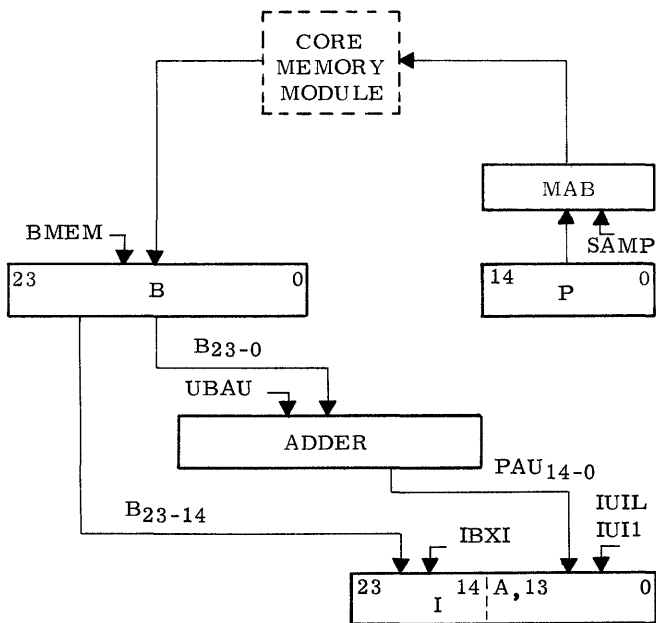


Fig. CMD. 4. State 1 Block Diagram

Fig. CMD. 4 contains a block diagram of the Arithmetic Unit operation during a normal Sequence State 1. A timing diagram and logic equations for State 1 are contained in Fig. CMD. 5. Refer to these aids during the following discussion.

During Sequence State 1, memory is always requested (G1SMRQ) to "fetch" the next command from memory. Memory is addressed from the P Register during State 1 except when following a branch instruction (SPB, BRU, BTS, BTR, LDP, or LPR), an XEC command, an Automatic Program Interrupt, a Memory Protect Error, or when a new Protect Status Word is required for the optional Memory Protect logic. Upon receipt of the Data Ready signal from the core memory module, the command located in the addressed memory location is gated to the B Register by D1BMEN. Bits 23 through 14 of the command are gated from B to the I Register (IBXI). This places the OP Code of the instruction fetched in the I Register where it is decoded and the operation to be performed is determined. The operand address portion of the command (bits 13-0) is gated to the I Register via the Adder Unit. If the command is relative addressed (bit 14 is a "one"), relative address modification of the operand address occurs in the Adder Unit prior to being transferred to the I Register. Relative Addressing is described separately under the mnemonic RLT in the command description portion of the description.

At memory release, Last Pulse Envelope is enabled to end State 1. Sequencing will then continue for execution of the command as described under the mnemonic of the command.

## INDEXING

Indexing of a command involves the changing of the command word operand by augmenting it with the contents of

a designated X core cell (core cells 1 through 7). The GE-PAC command format allocates 3 bits (15, 16, and 17) to specify Index Address Modification. Special considerations for indexing the GEN 1, GEN 2, GEN 3, and certain full operand commands are described below.

- GEN 1 and GEN 3 commands involving a variable K constant (bits 0 through 5) may be indexed. That is, if the K constant is to vary, it may be changed by indexing. All other GEN 1 and GEN 3 commands have fixed command information in bits 13 through 0 and should not be indexed.
- GEN 2 commands that select a device in the input/output equipment may be indexed. Those GEN 2 commands not selecting a device should not be indexed so as to retain the intelligence coded in the command word.
- LDX, LXC, LXX, INX, STX, and TXH commands use the index bits for instruction control and do not change the command operand.

A detailed description of Index Address Modification is contained under the mnemonic INDEX.

### RELATIVE ADDRESSING

Relative addressing modifies the operand portion of the command word (bits 13 through 0) by adding the core cell address of the command to the command operand. All Quasi and Full Operand commands, except NOP and TXH may be relative addressed. Relative addressing is performed when bit 14 of the Full Operand or Quasi command word is a "one".

Relative addressing allows selection of core memory locations up to  $\pm 8K$  from the location of the relative addressed command.

A detailed discussion of relative addressing is contained under the mnemonic RLT.

### MEMORY WRAP AROUND

In a system that uses less than the maximum memory, instructions may be executed with an address greater than the implemented memory size.

In such cases the memory will wrap around or fetch zeros as shown in the chart below. The point at which the implemented memory size and actual 4K block addressed, cross, indicates the 4K block that will be affected.

		Implemented Memory							
		4K	8K	12K	16K	20K	24K	28K	32K
4K Block Addressed	0- 4K	0-4	0-4	0- 4	0- 4	0- 4	0- 4	0- 4	0- 4
	4- 8K	0-4	4-8	4- 8	4- 8	4- 8	4- 8	4- 8	4- 8
	8-12K	0-4	0-4	8-12	8-12	8-12	8-12	8-12	8-12
	12-16K	4-8	4-8	*	12-16	12-16	12-16	12-16	12-16
	16-20K	0-4	0-4	0- 4	0- 4	16-20	16-20	16-20	16-20
	20-24K	0-4	4-8	4- 8	4- 8	*	20-24	20-24	20-24
	24-28K	0-4	0-4	8-12	8-12	*	*	24-28	24-28
	28-32K	0-4	4-8	*	12-16	*	*	*	28-32

\* A zero word is read with no parity error. A write operation would go undetected unless Memory Protect were used.

Memory wrap around is accomplished by wiring to the memory or multiplexer only those address bits required to address the maximum implemented memory location. That is, if the implemented memory is 16K words, then

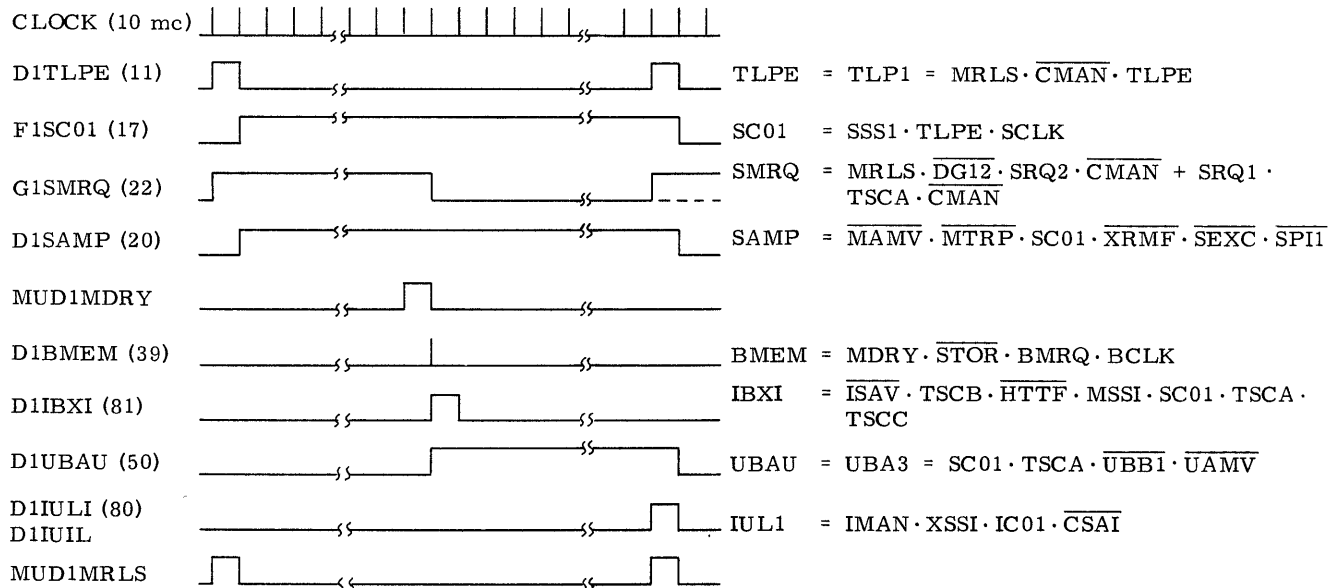


Fig. CMD. 5. Sequence Control State 1 Timing Diagram



address bits 0 through 13 are wired to the memory, etc.

## COMMAND DESCRIPTION CONVENTIONS

Each command is described individually and includes the following information:

- a. Mnemonic.
- b. Operation Format.
- c. Effective Operand Address.
- d. Command Type (e. g., Full Operand, Quasi, GEN 1).
- e. A chart indicating the registers, memory cells, and basic flip-flops affected by the command.
- f. Description of the instruction operation.
- g. Block diagram of the registers effected.
- h. Timing diagram of the primary control signals enabled.
- i. Logic equations for enabling the primary control signals.
- j. Description of the hardware operation.

Within the command descriptions, the following terms and symbols are used:

- X - Specifies the index word used in the execution of a command.
- Y - Represents the operand address of the command and implies the use of an operand from storage.

Z - Represents the effective operand address. It is developed as a function of X, \*, and Y.

\* - Indicates relative address modification (relative to the instruction's own location).

K - A constant in the address portion of certain commands.

f - Refers to the function of that which follows (e. g.,  $Z = F(X, *, Y)$  should be read as Z is equal to the function of indexing, relative addressing, and the operand address). Subscripts refer to bit positions of the designated register (e. g., B<sub>13-0</sub> delineates bit positions 13 through 0 of the B Register).

C ( ) - Refers to the contents of whatever is enclosed within the parenthesis (e. g., C (A<sub>23-0</sub>) is read as the contents of A Register bits 23 through 0.

All command descriptions are presented in alphabetical order except GEN 1, GEN 2, GEN 3, and Quasi commands which are contained under their group mnemonic in alphabetical order (i. e., GEN 1, GEN 2, etc.).

In addition to the discussion of individual commands, a discussion of Index Modification is contained under the mnemonic INDEX, and a discussion of Relative Addressing is contained under the mnemonic RLT.

Table CMD.1 contains an alphabetical listing of all commands operable with the 4022A Arithmetic Unit. Table CMD.2 lists these same commands according to their OP Code. Using these tables, the command format may be readily obtained for any command mnemonic, and the command mnemonic may be obtained for any command OP Code.

Command	Description	Type	Operation Format	Command	Description	Type	Operation Format
ABL	Append Beginning of List	Quasi	57X*Y	LDZ	Load Zeros Into A	GEN 1	05000000
ABT	Abort Device D's Operation	GEN 2	25X3D	LMO	Load Minus One	GEN 1	05060000
ACT	Activate Device D	GEN 2	25X1D	LMR	Load Mask Register	GEN 2	25000302
ADD	Add	F. O.	11X*Y	LMR <sub>2</sub>	Load Mask Register <sub>2</sub>	GEN 2	25000300
ADO	Add One To Bit K	GEN 1	0500700K	LPR	Load Place And	F. O.	35X*Y
AEL	Append End Of List	Quasi	47X*Y	LXC	Load X With Count	F. O.	17X00000
AKA	Add K To A	Quasi	60X0K	LXK	Load X With K	F. O.	07X*K
ANA	Logical And To A	F. O.	20X*Y	MAQ	Move A To Q	GEN 3	45004330
BRU	Branch Unconditionally	F. O.	14X*Y	MPY	Multiply	F. O.	55X*Y
BTR	Branch If Test FF Reset	F. O.	30X*Y	NEG	Negate	GEN 1	05013000
BTS	Branch If Test FF Set	F. O.	34X*Y	NOP	No Operation	F. O.	26200000
CBK	Change Bit K	GEN 1	05X4700K	OOM	Operate On Memory	Quasi	62X*Y
CLO	Count Least Significant Ones	GEN 1	05004137	OPR	Operate Device D	GEN 2	25X2D
CLZ	Count Least Significant Zeros	GEN 1	05070137	ORA	Logical OR To A	F. O.	21X*Y
CMO	Count Most Significant Ones	GEN 1	05004237	OUT	Out To Device D	GEN 2	25X4D
CMZ	Count Most Significant Zeros	GEN 1	05070237	PAI	Permit Automatic Interrupt	GEN 2	25020000
CPL	Complement A	GEN 1	05010000	RBK	Reset Bit K	GEN 1	0504500K
DAD	Double Length Add	Quasi	51X*Y	RBL	Remove Beginning Item From List	Quasi	56X*Y
DLA	Double Left Arithmetic	GEN 3	45X0644K	RCS	Read Console Switches	GEN 2	25050000
DLD	Double Length Load	Quasi	41X*Y	REL	Remove Ending Item From List	Quasi	46X*Y
DLL	Double Left Logical	GEN 3	45X0720K	REV	Reset Test FF Bit K Even	GEN 1	05X7040K
DMT	Decrement Memory And Test	F. O.	060*Y	RNZ	Reset Test FF A Non-Zero	GEN 1	05004470
DRA	Double Right Arithmetic	GEN 3	45X0440K	ROD	Reset Test FF Bit K Odd	GEN 1	05X0440K
DRC	Double Right Circular	GEN 3	45X0530K	RST	Reset Test FF	GEN 1	05004737
DRL	Double Right Logical	GEN 3	45X0430K	SBK	Set Bit K	GEN 1	05X4600K
DST	Double Length Store	F. O.	63X*Y	SEL	Select Device D	GEN 2	25X0D
DSU	Double Length Subtract	Quasi	61X*Y	SET	Set Test FF	GEN 1	05004637
DVD	Divide	F. O.	65X*Y	SEV	Set Test FF Bit K Even	GEN 1	05X7050K
ERA	Exclusive OR To A	F. O.	10X*Y	SKA	Subtract K From A	Quasi	50X0Y
FAD	Floating Point Add	Quasi	70X*Y	SLA	Shift Left Arithmetic	GEN 3	45X0204K
FDV	Floating Point Divide	Quasi	73X*Y	SLL	Shift Left Logical	GEN 3	45X0200K
FIX	Fix-Floating Number	Quasi	74X0K	SNZ	Set Test FF A Non-Zero	GEN 1	05004570
FLO	Float-Fixed Number	Quasi	74X2K	SOD	Set Test FF Bit K Odd	GEN 1	05X0450K
FMP	Floating Multiply	Quasi	72X*Y	SPB	Save Place And Branch	F. O.	33X*Y
FMS	Floating Mode Shift	Quasi	74X1K	SRA	Shift Right Arithmetic	GEN 1	05X1404K
FSU	Floating Subtract	Quasi	71X*Y	SRC	Shift Right Circular	GEN 1	05X0404K
IAI	Inhibit Automatic Interrupt	GEN 2	25030000	SRL	Shift Right Logical	GEN 1	05X0004K
IAI <sub>2</sub>	Inhibit Automatic Interrupt <sub>2</sub>	GEN 2	25000304	SSA	Set Stall Alarm	GEN 2	25010000
IBK	Isolate Bit K	GEN 1	05X0100K	STA	Store A	F. O.	32X*Y
IN	In From Device D	GEN 2	25X5D	STI	Store A Indirect	Quasi	53X*Y
INX	Increment X	F. O.	26X*K	STM	Select Trapping Mode	GEN 2	25000001
JCB	Jump If Channel Busy	GEN 2	25X6D2D	STQ	Store Q	F. O.	44X*Y
JDR	Jump If Data Ready	GEN 2	25X6D4D	STX	Store X	F. O.	06X*Y
JND	Jump If No Demand	GEN 2	25040000	SUB	Subtract	F. O.	31X*Y
JNE	Jump If No Error	GEN 2	25X7D	TER	Test Even Reset Bit K	GEN 1	05X4560K
JNO	Jump If No Overflow	GEN 2	25060000	TES	Test Even Set Bit K	GEN 1	05X4660K
JNP	Jump If No Parity	GEN 2	25070000	TEV	Test Bit K Even	GEN 1	05X7070K
JNR	Jump If Not Ready	GEN 2	25X6D	TNM	Test Not Minus One	GEN 1	05070770
LBM	Load Bit Mask	GEN 1	05X6300K	TNZ	Test A Non-Zero	GEN 1	05004770
LDA	Load A	F. O.	00X*Y	TOD	Test Bit K Odd	GEN 1	05X0470K
LDI	Load A Indirect	Quasi	52X*Y	TOR	Test Odd Reset Bit K	GEN 1	05X4570K
LDK	Load A With K	Quasi	40X*K	TOS	Test Odd Set Bit K	GEN 1	05X4670K
LDO	Load One Into Bit K	GEN 1	05X0300K	TSC	Test and Shift Circular	GEN 1	05X0464K
LDP	Load Place	F. O.	15X*Y	TXH	Test X High	F. O.	24X0-K
LDQ	Load Q	F. O.	42X*Y	TZC	Test Zero And Complement	GEN 1	05064670
LDX	Load X	F. O.	16X*Y	TZE	Test A Zero	GEN 1	05004670
				XEC	Execute	F. O.	04X*Y

Table CMD. 1. GE-PAC 4020 Command Listing

OPERATION FORMAT	COMMAND	OPERATION FORMAT	COMMAND	OPERATION FORMAT	COMMAND
00X*Y	LDA	06X*Y	STX	34X*Y	BTS
04X*Y	XEC	060*Y	DMT	35X*Y	LPR
05000000	LDZ	07X*K	LXK	40X*K	LDK
05004137	CLO	10X*Y	ERA	41X*Y	DLD
05004237	CMO	11X*Y	ADD	42X*Y	LDQ
05004470	RNZ	14X*Y	BRU	44X*Y	STQ
05004570	SNZ	15X*Y	LDP	45004330	MAQ
05004637	SET	16X*Y	LDX	45X0200K	SLL
05004670	TZE	17X00000	LXC	45X0204K	SLA
05004737	RST	20X*Y	ANA	45X0430K	DRL
05004770	TNZ	21X*Y	ORA	45X0440K	DRA
05010000	CPL	24X0-K	TXH	45X0530K	DRC
05013000	NEG	25X0D	SEL	45X0644K	DLA
05060000	LMO	25X1D	ACT	45X0720K	DLL
05064670	TZC	25X2D	OPR	46X*Y	REL
05070137	CLZ	25X3D	ABT	47X*Y	AEL
05070237	CMZ	25X4D	OUT	50X0Y	SKA
05070770	TNM	25X5D	IN	51X*Y	DAD
05X0004K	SRL	25X6D	JNR	52X*Y	LDI
05X0100K	IBK	25X6D2D	JCB	53X*Y	STI
05X0300K	LDO	25X6D4D	JDR	55X*Y	MPY
05X0404K	SRC	25X7D	JNE	56X*Y	RBL
05X0440K	ROD	25000001	STM	57X*Y	ABL
05X0450K	SOD	25000300	LMR <sub>2</sub>	60X0K	AKA
05X0464K	TSC	25000302	LMR	61X*Y	DSU
05X0470K	TOD	25000304	IAI <sub>2</sub>	62X*Y	OOM
05X0700K	ADO	25010000	SSA	63X*Y	DST
05X1404K	SRA	25020000	PAI	65X*Y	DVD
05X4500K	RBK	25030000	IAI	70X*Y	FAD
05X4560K	TER	25040000	JND	71X*Y	FSU
05X4570K	TOR	25050000	RCS	72X*Y	FMP
05X4600K	SBK	25060000	JNO	73X*Y	FDV
05X4660K	TES	25070000	JNP	74X0K	FIX
05X4670K	TOS	26X*K	INX	74X1K	FMS
05X4700K	CBK	26200000	NOP	74X2K	FLO
05X6300K	LBM	30X*Y	BTR	75X*Y	Optional
05X7040K	REV	31X*Y	SUB	76X*Y	Optional
05X7050K	SEV	32X*Y	STA	77X*Y	Optional
05X7070K	TEV	33X*Y	SPB		

Table CMD. 2. GE-PAC 4020 Command Operation Format Listing

# ADD - ADD Z TO A

FULL OPERAND	23	11	18	17	15	14	13	Y	0	$Z = f(X, *, Y)$
				X	*					

ADD sums the contents of core cell Z with the contents of the A Register and stores the result in A. If the result is too large to be stored in the 23 data bits of the A Register (i.e., more negative than  $-2^{23}$  or more positive than  $2^{23}-1$ ), the Overflow flip-flop (F1UOFL) is set. Overflow may occur only when the signs of the original numbers in A and Z are alike. Overflow is then detected if a sign change occurs in the sum.

A non-indexed ADD command is executed during Sequence Control State 4 (SC04). Memory location Z is addressed from  $I_A, 13-0$  (DISAMI) during State 4. The contents of memory location Z are gated to the B Register by D1BMEM during the Clock pulse of Memory Data Ready (MUD1MDRY). From B, the contents of memory location Z are gated (D1UBAU) to the Adder Unit at the same time the contents of the A Register are gated (D1UAAU) to the Adder Unit. The summation occurs in the Adder Unit and the sum is gated (D1AAUL,U) to the A Register at the Clock of Memory Release (MUD1MRLS).

If arithmetic overflow - either positive or negative - occurs during the summation, the Overflow flip-flop (F1UOFL) is set. The following examples illustrate (a) a positive overflow, and (b) a negative overflow. For simplicity, five bit (4 data and a sign) registers are illustrated. Consider the most significant bit as bit 23 and the next most significant bit as bit 22.

Consider:  $(A = +5) + (Z = +12) = 17$ . Seventeen is too large to be contained in 4 bits plus a sign bit; thus, the Overflow flip-flop is set.

Contents of Z	0 1100	(12)
Contents of A	0 0101	( 5)
Sum	1 0001	Overflow Set = $23S \cdot 23C \cdot 22C$

(a) POSITIVE OVERFLOW.

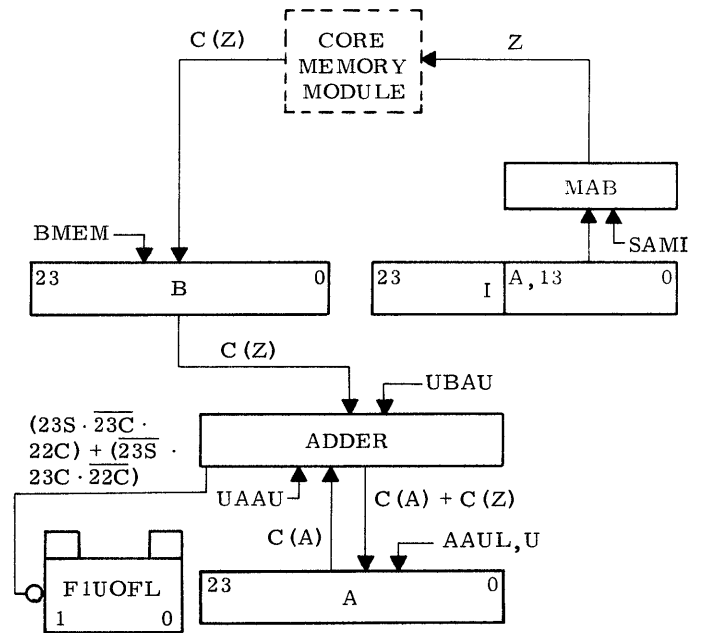
Consider:  $(A = -8) + (Z = -13) = -21$ . Minus 21 is too large to be contained in 4 bits plus a sign bit; thus, the Overflow flip-flop is set.

Contents of Z	1 0010	(-13)
Contents of A	1 1000	( -8)
Sum	0 1010	Overflow Set = $23S \cdot 23C \cdot 22C$

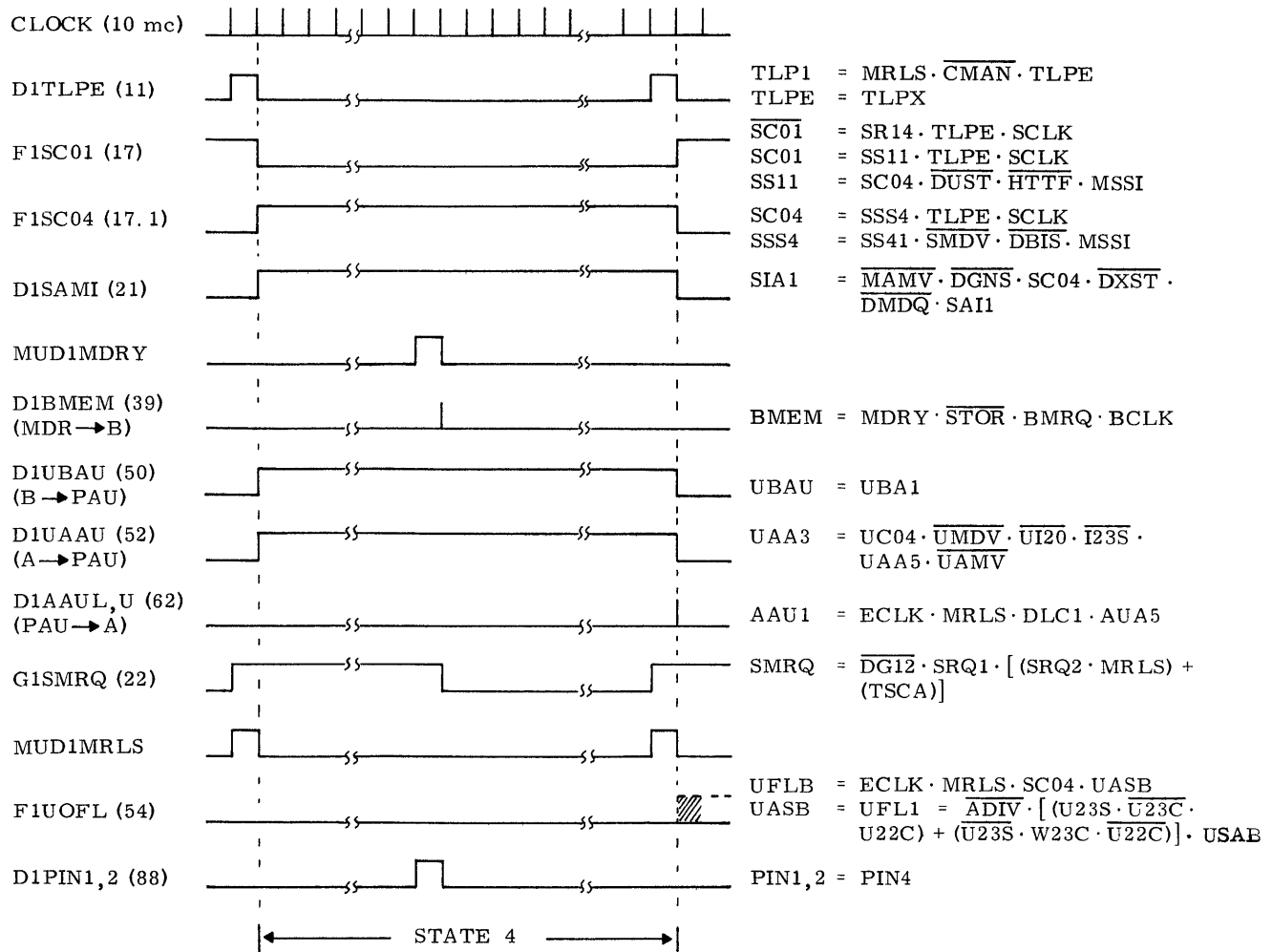
(b) NEGATIVE OVERFLOW.

Non-Indexed Word Times.	2 (S1, S4)
Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$C(A) + C(Z)$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	Set if overflow occurs
F1ETST	
$J_{4-0}$	
Memory Z	

COMMAND CHARACTERISTICS



ADD BLOCK DIAGRAM



ADD TIMING DIAGRAM

# ANA - LOGICAL AND TO A

FULL OPERAND	23	20	18	17	15	14	13	Y	0	$Z = f(X, *, Y)$
				X	*					

ANA performs the logic AND of the contents of core cell Z with the contents of A. The corresponding bits of Z and A are compared. If the corresponding bits are both "one", a "one" is placed in that position of A. If either or both of the compared positions are "zero", a "zero" is placed in that position of A.

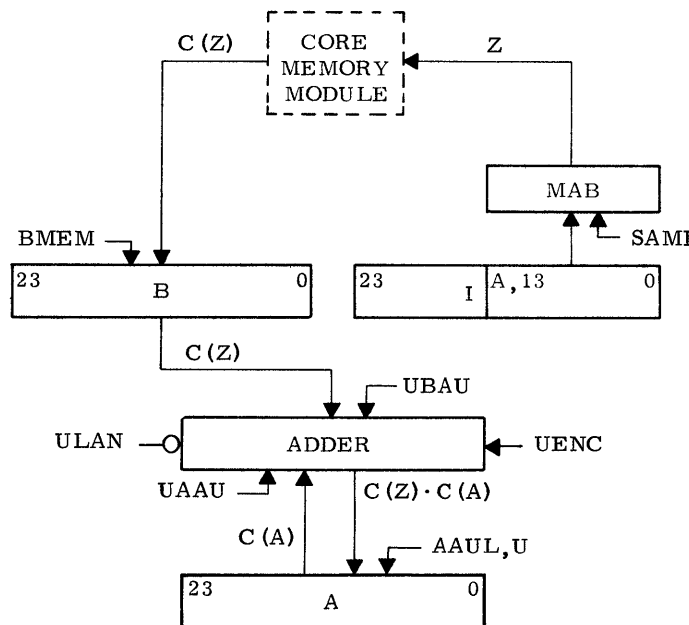
A non-indexed ANA command is executed during Sequence Control State 4 (SC04). Memory location Z is addressed from  $I_{A,13-0}$  (D1SAMI) during State 4. The contents of memory cell Z are then gated to the B Register by D1BMEM during the clock pulse of Memory Data Ready (MUD1MDRY). From B, the contents of memory location Z are gated to the Adder Unit (D1UBAU). At the same time, the contents of the A Register are gated (D1UAAU) to the Adder Unit along with a control signal, D0ULAN, and the Enable Carry signal, G1UENC. D0ULAN, in conjunction with G1UENC, enables the logical AND function of the Adder Unit. The result of the logical AND is then gated back to the A Register to complete the ANA execution cycle.

To exemplify the ANA comparison, consider the following; four bits are used for simplicity.

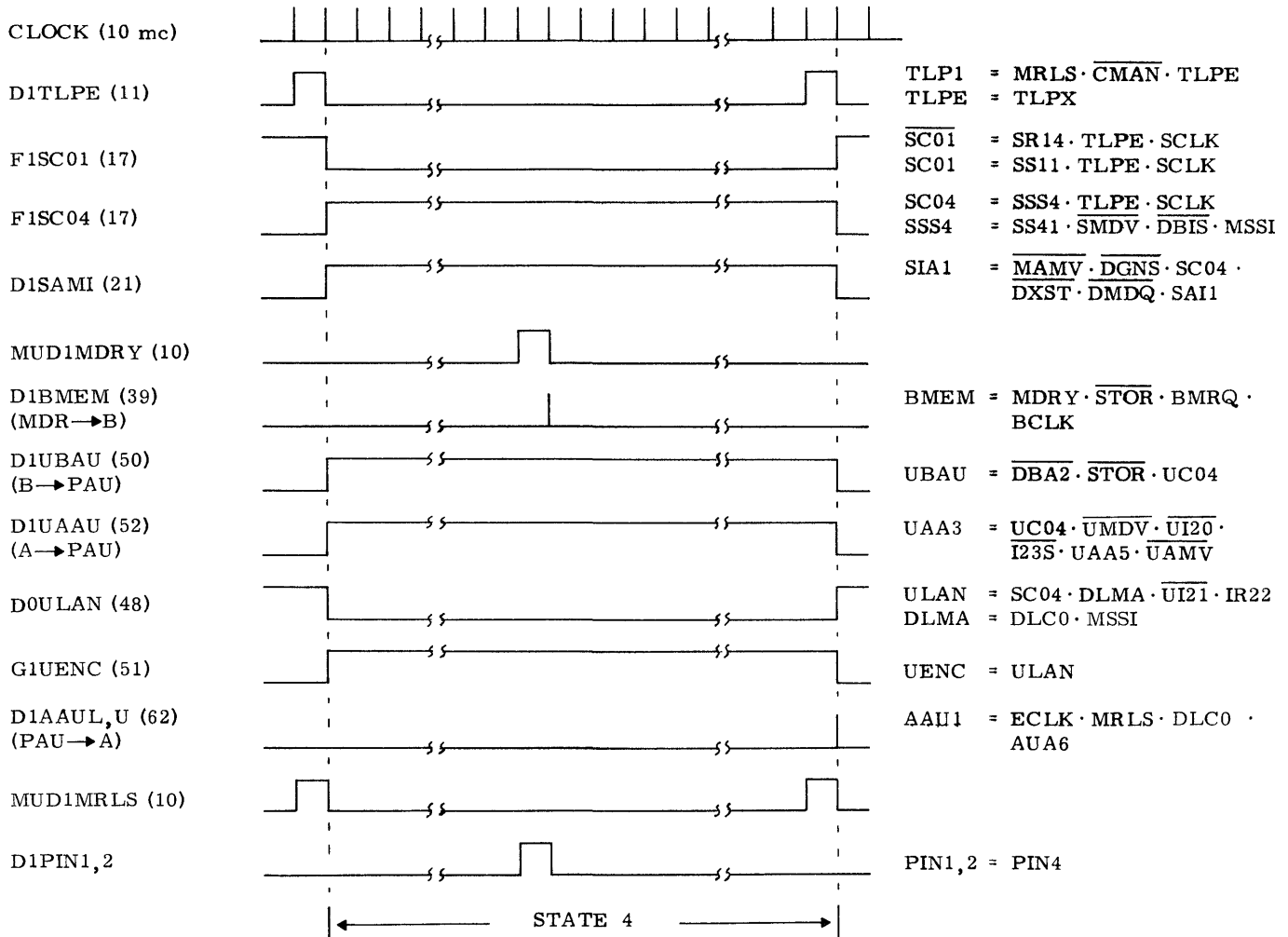
Contents of the A Register = 0011  
 Contents of cell Z = 0101  
 Result Placed in A Register = 0001.

Non-Indexed Word Times.	2 (S1, S4)
Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$C(Z_{23-0})$ AND with $C(A_{23-0})$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	
Memory Z	

COMMAND CHARACTERISTICS

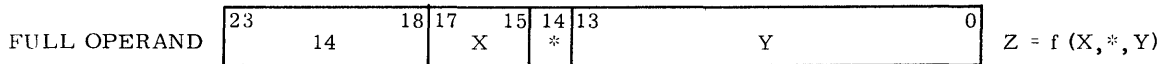


ANA BLOCK DIAGRAM



ANA TIMING DIAGRAM

# BRU - BRANCH UNCONDITIONALLY

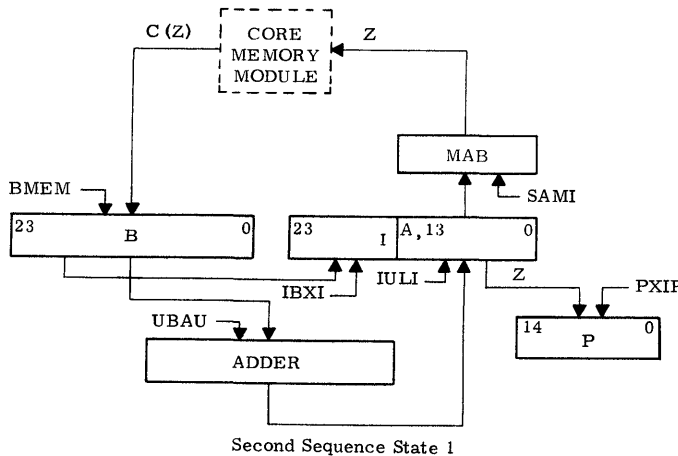
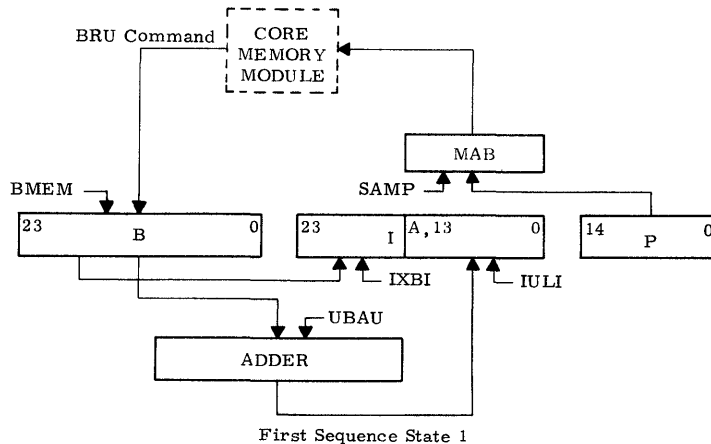


BRU unconditionally transfers program control to the designated core address Z. This is accomplished by calculating the address Z, if it is relative addressed and/or indexed, and then transferring this address to the Program (P) Register.

The BRU command is executed during Sequence Control State 1. For ease of understanding, two successive Sequence State 1's (i.e., 2 word times) are shown in the Block and Timing Diagrams. During the first word time of Sequence State 1 (F1SC01), the BRU command is "fetched" from memory in the normal manner except that the clock of Last Pulse Envelope (D1TLPE) sets the Remember flip-flop (F1XRMF). At the end of the first word time, State 1 is not cleared; instead, State 1 is held for another "fetch" cycle to obtain the command located in memory location Z. During this second word time, memory is addressed from IA<sub>13-0</sub> and the command in location Z is "fetched" from memory in the normal manner. Also, during this second word time, the contents of IA<sub>13-0</sub> are transferred to the Program Register (PXIP) to complete transfer of program control. At the end of this second word time, the Remember flip-flop is cleared.

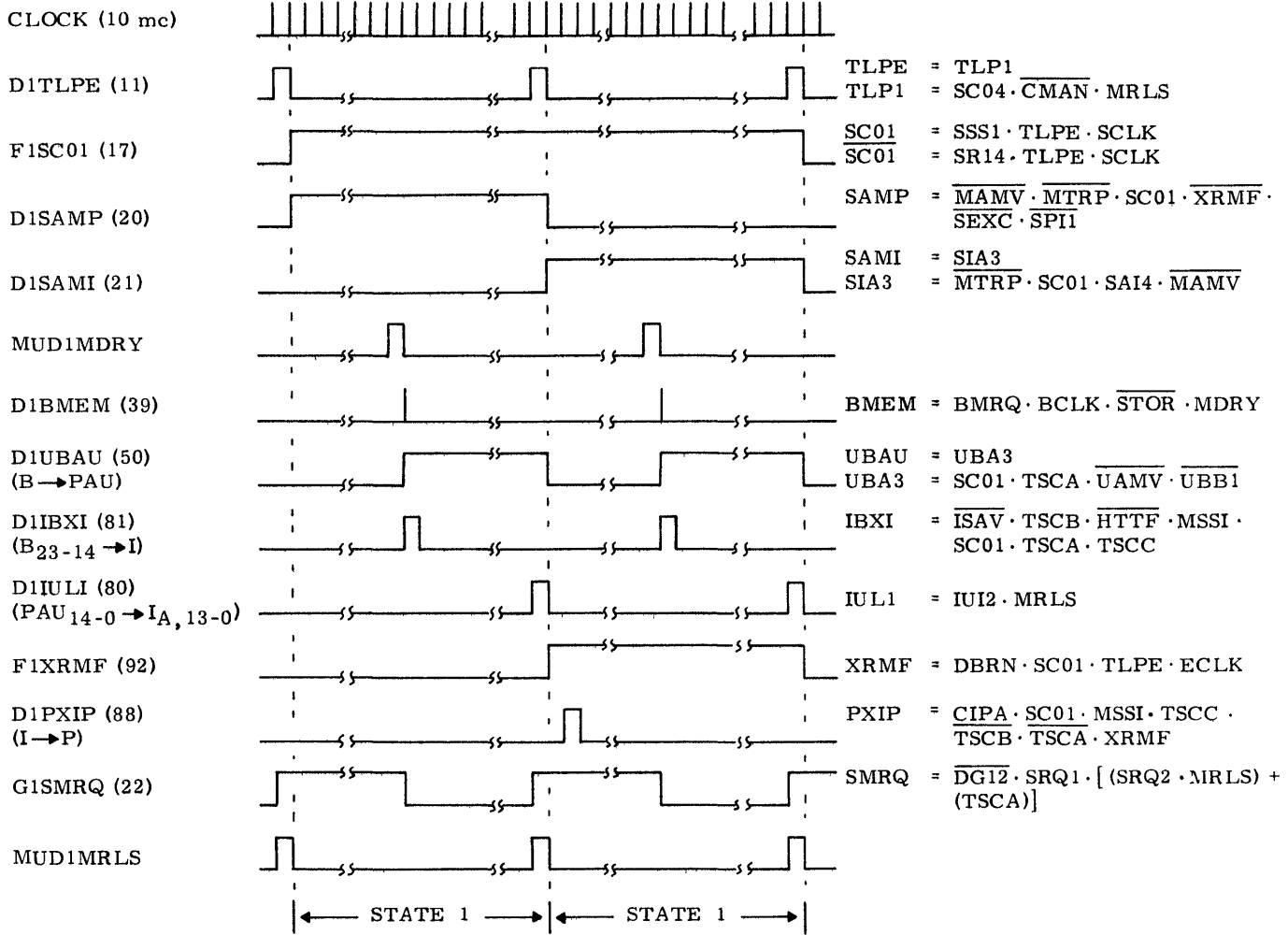
Non-Indexed Word Times.	1 (S1)
Interruptable Following Execution?	No
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	
Q <sub>23-0</sub>	
P <sub>14-0</sub>	Z
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	
Memory Z	

COMMAND CHARACTERISTICS



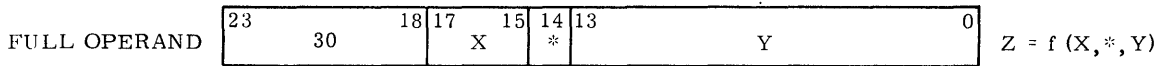
BRU BLOCK DIAGRAM





BRU TIMING DIAGRAM

# BTR - BRANCH IF TEST FLIP-FLOP RESET



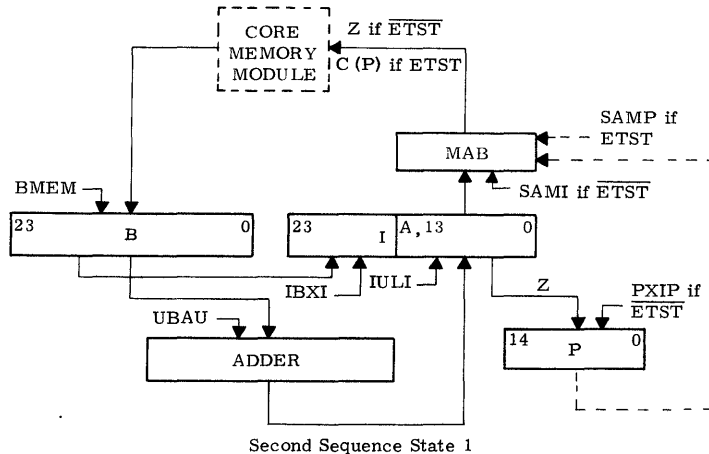
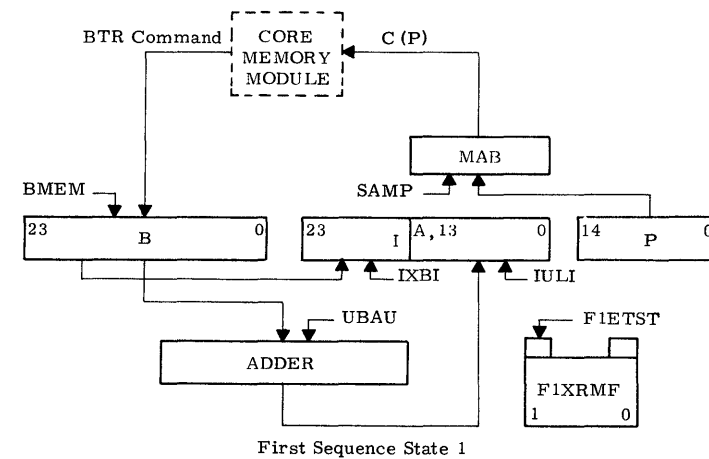
BTR transfers program control to memory location Z if the Test flip-flop (F1ETST) is reset. If the Test flip-flop is set, program control continues in sequence (i. e. , C (P) +1). The status of the Test flip-flop is not changed by the BTR command.

To illustrate the operation of the BTR command, two successive word times are required (i. e. , two Sequence Control State 1's). However, only one word time is used solely by the BTR command, since the second word time is used to "fetch" the next command. The Block and Timing Diagrams illustrate both word times of State 1. During the first word time, the BTR command is "fetched" from memory in the normal manner. At the clock of Last Pulse Envelope, the Remember flip-flop is set if the Test flip-flop (F1ETST) is reset. If the Test flip-flop is set, the Remember flip-flop remains reset.

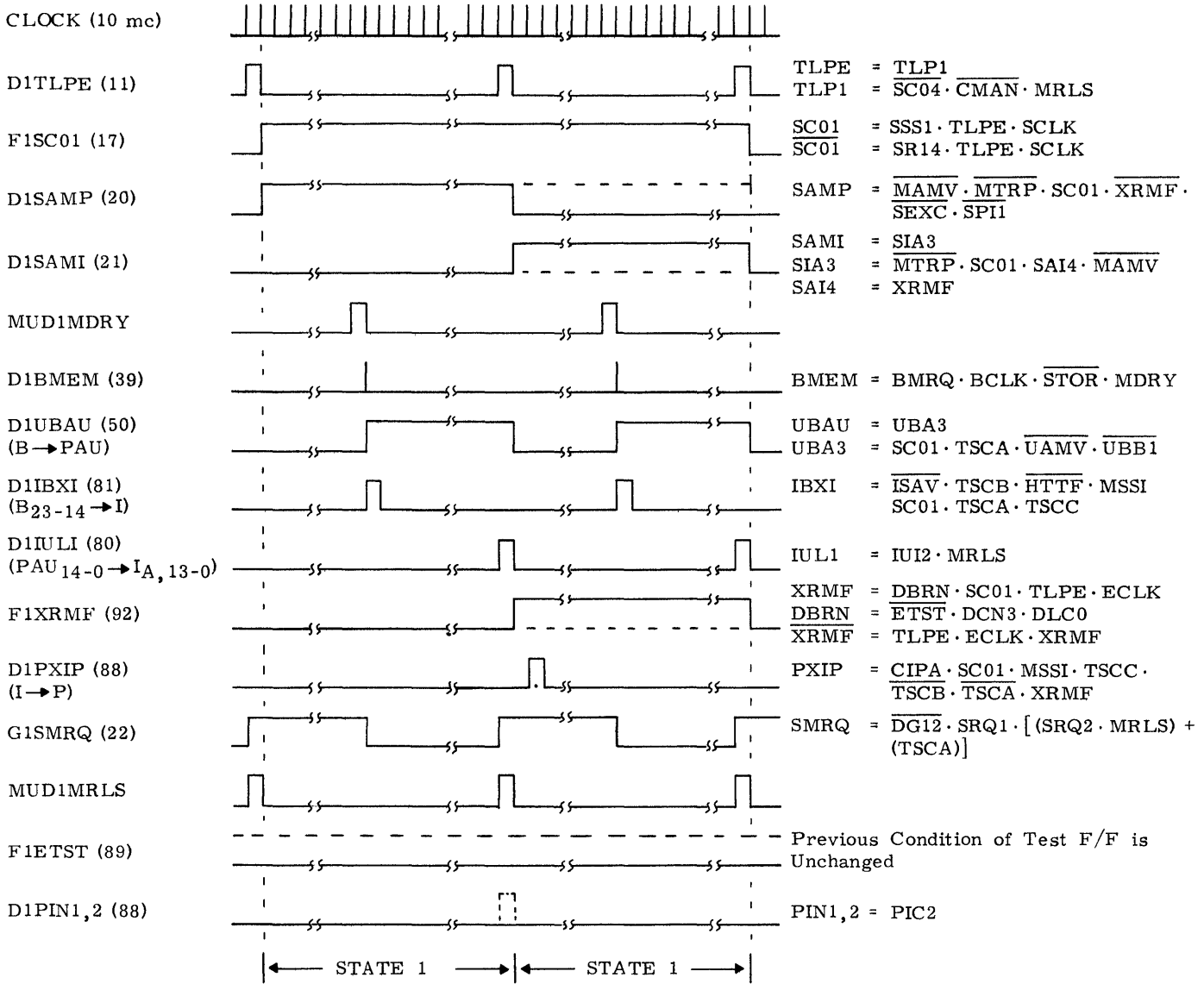
During the second word time, memory is addressed from  $I_{A,13-0}$  (D1SAMI) if the Remember flip-flop is set (i. e. , if the Test flip-flop is reset). In this manner, memory location Z is addressed. Also, if the Remember flip-flop is set,  $I_{A,13-0}$  is transferred to the P Register (PXIP) to complete the transfer of program control. If, however, the Remember flip-flop is reset, memory is addressed from the P Register and program control continues in the normal sequence.

Non-Indexed Word Times.	1 (S1)
Interruptable Following Execution?	No
<b>CHANGES FOLLOWING EXECUTION</b>	
A <sub>23-0</sub>	
Q <sub>23-0</sub>	
P <sub>14-0</sub>	Z if $\overline{F1ETST}$ C(P) +1 if F1ETST
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	
Memory Z	

COMMAND CHARACTERISTICS

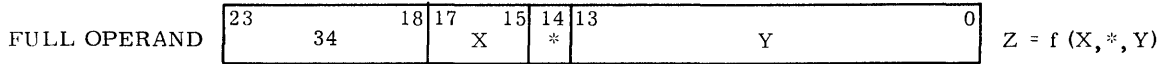


BTR BLOCK DIAGRAM



BTR TIMING DIAGRAM

# BTS - BRANCH IF TEST FLIP-FLOP SET



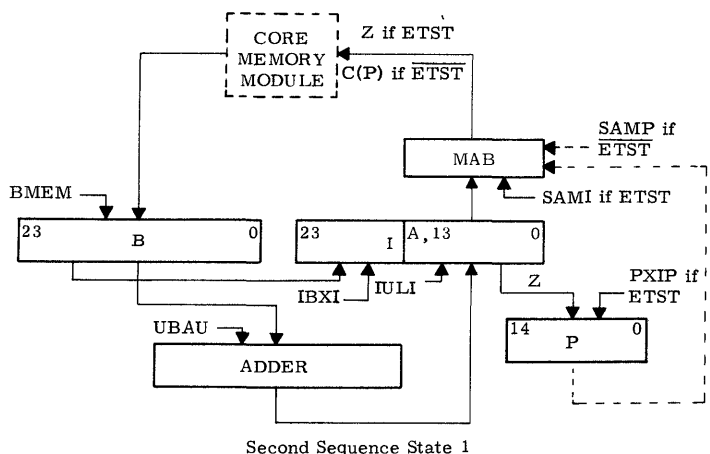
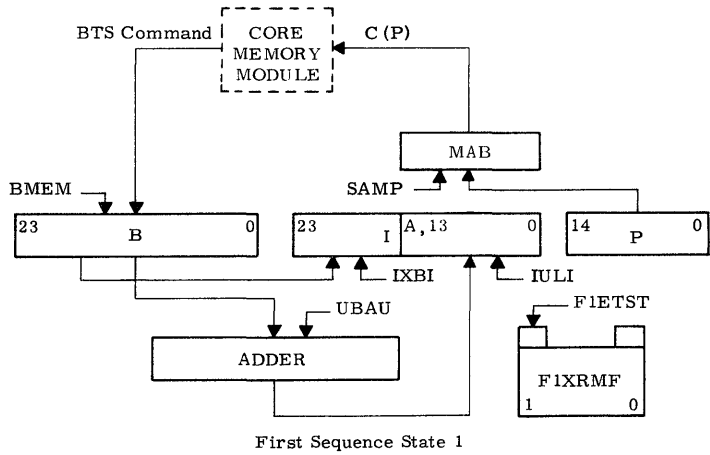
BTS transfers program control to memory location Z if the Test flip-flop (F1ETST) is set. If the Test flip-flop is reset, program control continues in sequence (i. e., C(P) + 1). The status of the Test flip-flop is not changed by the BTS command.

To illustrate the operation of the BTS command, two successive word times are required (i. e., two Sequence Control State 1's). However, only one word time is used solely by the BTS command, since the second word time is used to "fetch" the next command. The Block and Timing Diagrams illustrate both word times of State 1. During the first word time, the BTS command is "fetched" from memory in the normal manner. At the clock of Last Pulse Envelope, the Remember flip-flop is set if the Test flip-flop (F1ETST) is set. If the Test flip-flop is reset, the Remember flip-flop remains reset.

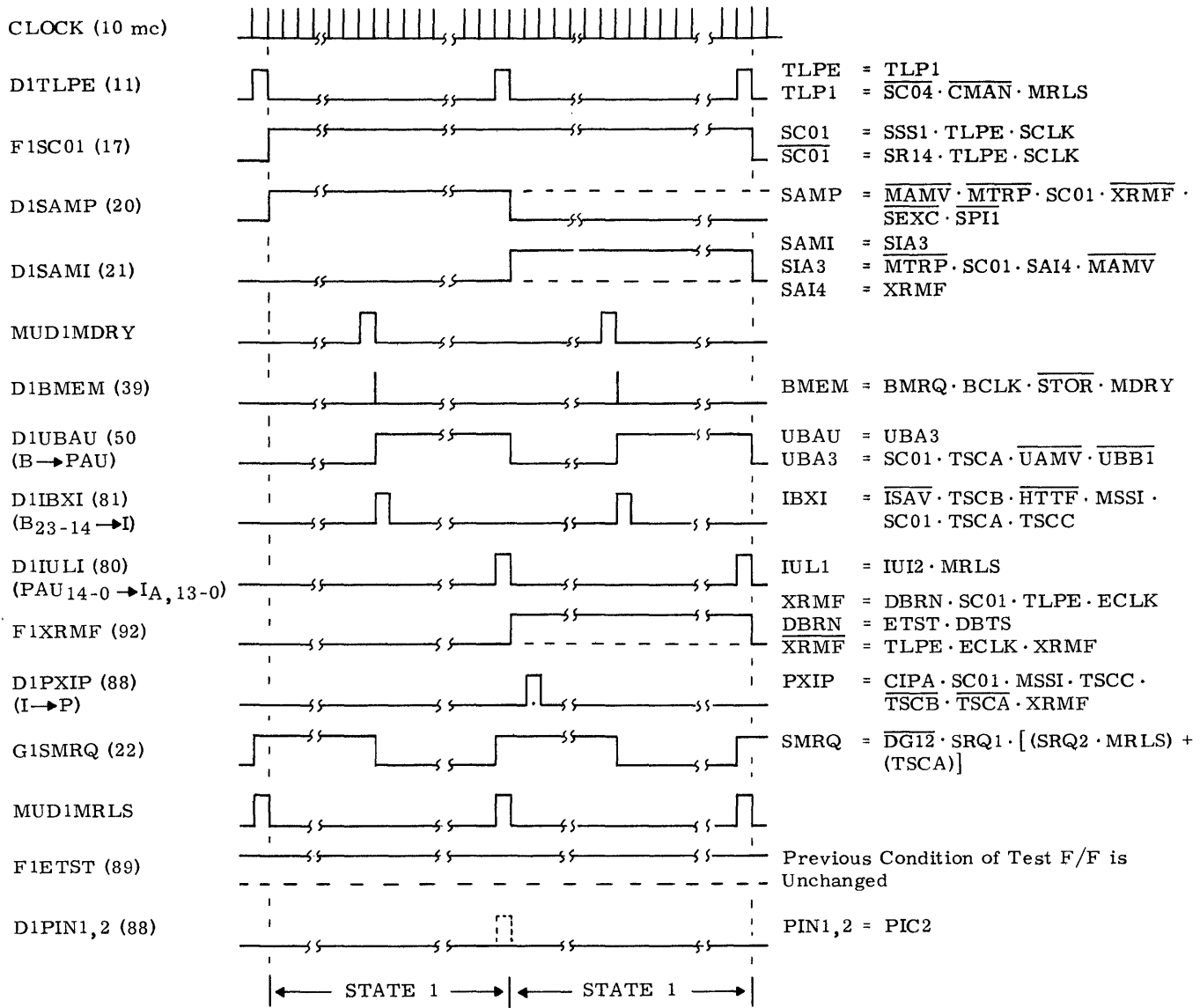
During the second word time, memory is addressed from  $I_A, 13-0$  (D1SAMI) if the Remember flip-flop is set (i. e., if the Test flip-flop is set). In this manner, memory location Z is addressed. Also, if the Remember flip-flop is set,  $I_A, 13-0$  is transferred to the P Register (PXIP) to complete the transfer of program control. If, however, the Remember flip-flop is reset, memory is addressed from the P Register and program control continues in the normal sequence.

Non-Indexed Word Times.	1 (S1)
Interruptable Following Execution?	No
<b>CHANGES FOLLOWING EXECUTION</b>	
A <sub>23-0</sub>	
Q <sub>23-0</sub>	
P <sub>14-0</sub>	Z if F1ETST C(P) + 1 if $\overline{F1ETST}$
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	
Memory Z	

COMMAND CHARACTERISTICS

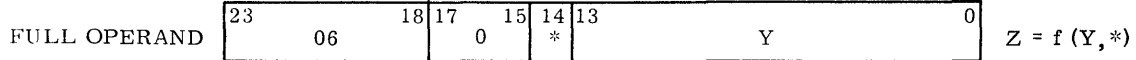


BTS BLOCK DIAGRAM



BTS TIMING DIAGRAM

# DMT - DECREMENT MEMORY AND TEST



DMT subtracts 1 from the contents of memory cell Z each time it is executed. If the DMT is not the result of an API and the original contents of Z were not equal to 0, the Test flip-flop (F1ETST) is set. If the original contents of Z were equal to 0 and the DMT command is not the result of an API, the Test flip-flop is cleared. If the DMT command is the result of an API and the original contents of Z were equal to 0, a signal is applied to API for "Echo" generation. The Test flip-flop is not affected by a DMT resulting from an API. When an index address is specified (bits 17 through 15 not equal to 0) the DMT command is executed as an STX command.

The DMT command is executed during Sequence Control States 2 and 4. During State 2, memory is addressed from IA<sub>13-0</sub> (D1SAMI) and the contents of memory location Z are gated to the B Register. From B, the contents of memory location Z are gated to the B input of the Adder Unit (D1UBBU). At the same time, both the true and complemented B Register contents are gated to the A input of the Adder Unit (D1UBAU, D1UBNA). Gating both the true and complemented B Register contents to the A input applies a "one" to the A input of each Adder bit position. This is the same as applying a -1 to the A inputs of the Adder. The Test flip-flop is cleared at Time 4 if the DMT command is not being executed from an interrupt response address (AIF0SPI2). If a carry occurs from bit 23 of the Adder, due to the summation (contents of B plus -1), the Test flip-flop is set at the clock of Memory Release. This indicates that the original contents of Z were not equal to zero. If a carry from bit 23 of the Adder does not occur, the Test flip-flop remains in the reset state, indicating that the original contents of Z were equal to zero. If a carry from bit 23 of the Adder does not occur, when the DMT command is executed from an interrupt response address, G1WEKO is enabled allowing the API module to cause an interrupt at a different address. At the clock of Last Pulse Envelope, the Adder outputs (contents of Z -1) are gated back to the B Register.

During State 4, memory is again addressed from IA<sub>13-0</sub> and the contents of B are gated back to memory location Z. In this manner, the original contents of Z -1 are stored back in memory location Z.

The following examples are used to illustrate that a carry occurs from PAU<sub>23</sub> whenever the value to be decremented is not equal to 0. Only when the value being decremented goes from 0 to -1 will the absence of a carry occur from PAU<sub>23</sub>. For simplicity, word lengths of 5 bits are used. Consider the most significant bit of the example to be PAU<sub>23</sub>.

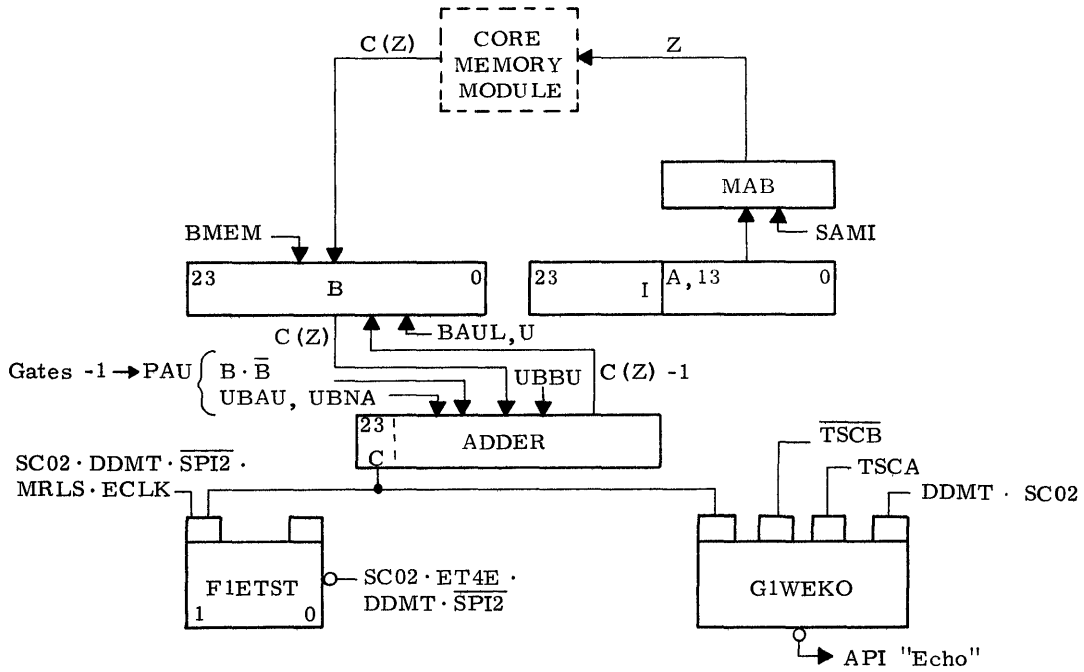
Non-Indexed Word Times.	3 (S1, S2, S4)
Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C (P) +1
F1WPMT	
F1UOFL	
F1ETST	Set if C(Z) ≠ 0 Reset if C(Z) = 0*
J <sub>4-0</sub>	
Memory Z	

### COMMAND CHARACTERISTICS

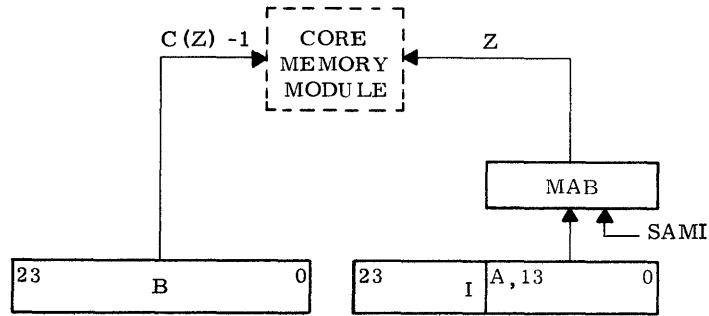
\* The Test flip-flop is not affected if the DMT command is executed as the result of an Automatic Program Interrupt and is located in the interrupt response address. In this case, an "Echo" signal is applied to the API module when the original contents of Z were equal to zero. This will normally cause a different program interrupt.

(Z = 2)	00010	11111	
( -1)	11111	00001	Plus Carry = Test F/F Set*
(Z = 1)	00001	11111	
	11111	00000	Plus Carry = Test F/F Set*
(Z = 0)	00000	11111	
	11111	11111	No Carry = Test F/F Reset* or "Echo" to API
(Z = -1)	11111	11111	
	11111	11110	Plus Carry = Test F/F Set*.

\* If DMT is not executed from API response address.

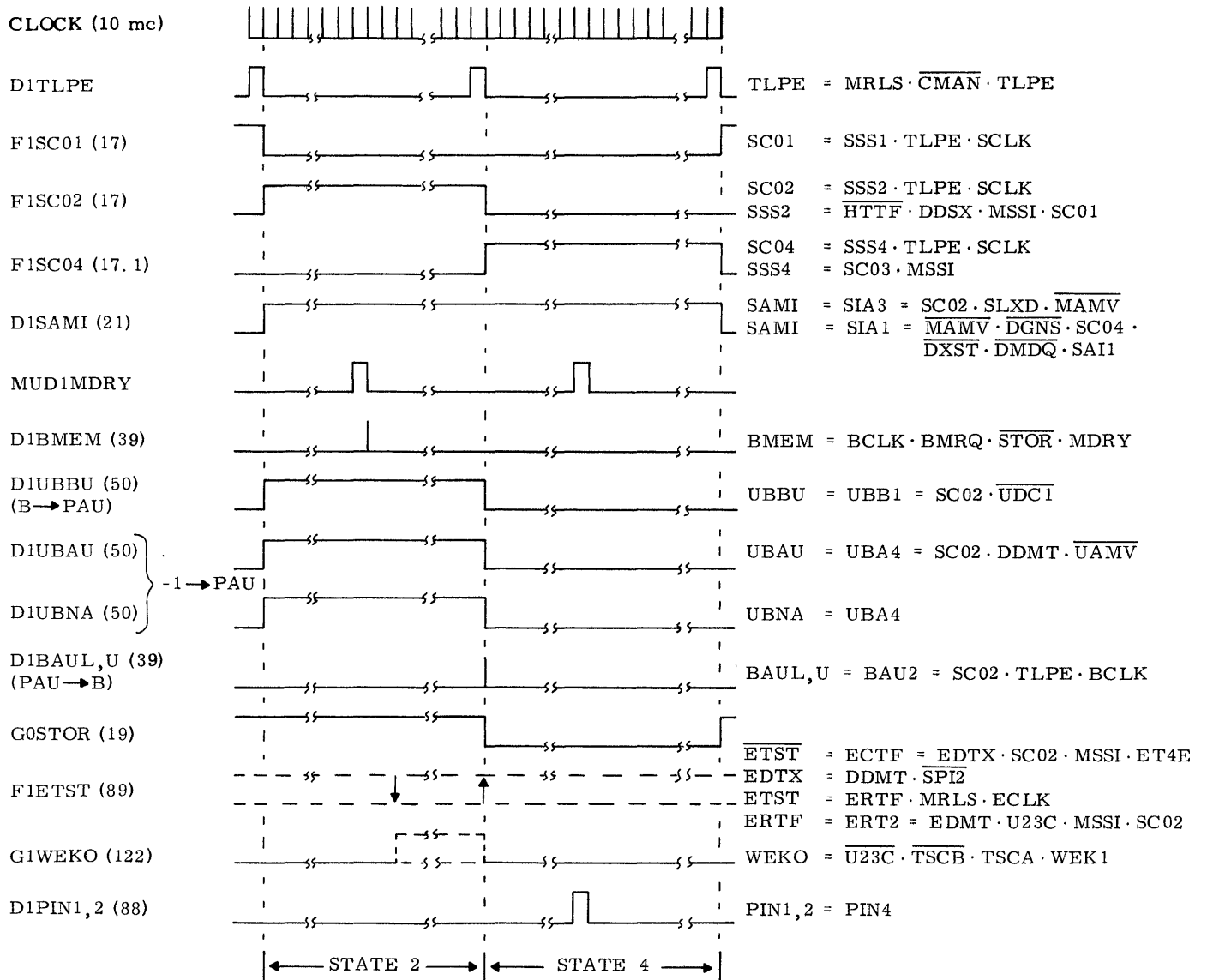


Sequence State 2



Sequence State 4

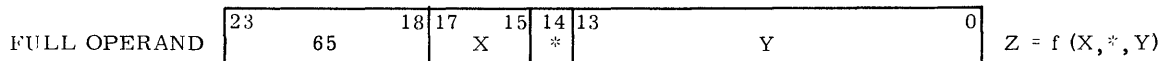
DMT BLOCK DIAGRAM



DMT TIMING DIAGRAM



# DVD - DIVIDE



DVD divides the 47-bit dividend, 24 bits of the A Register coupled with bits 22 through 0 of the Q Register, by the divisor contained in location Z. The quotient is placed in the Q Register and the remainder is placed in the A Register. The magnitude of the divisor must be greater than the magnitude of the A Register. If the divisor is not larger than A, the Overflow flip-flop (F1UOFL) is set and the DVD result will be incorrect. The sign of A ( $A_{23}$ ) applies to the remainder, and the sign of Q ( $Q_{23}$ ) applies to the quotient.

**NOTE**

Either positive or negative numbers may be divided, however, the remainder always has the sign of the divisor, which may yield an unexpected result. The difference between the divisor and the remainder obtained yields the expected divisor (e.g.,  $(+4) \div (-2) = -3$ , remainder -2).

In actual mathematical operations within the computer, however, this is of little consequence since the quotient formed by any division is only an approximation (i.e., the remainder must be considered). Only when small integer values are divided is the quotient obtained of any significant difference from the expected quotient.

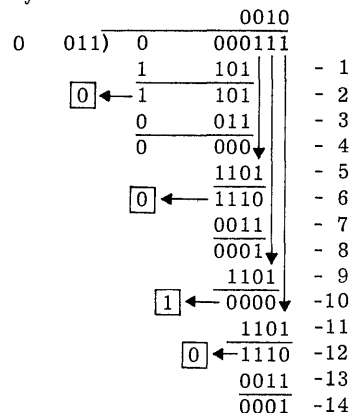
Non-Indexed Word Times.	S1, S3, S4, S5 13.7 $\mu$ s
Interruptable Following Execution?	Yes
<b>CHANGES FOLLOWING EXECUTION</b>	
A <sub>23-0</sub>	Remainder
Q <sub>23-0</sub>	Quotient
P <sub>14-0</sub>	C(P) +1
F1WPMT	
F1UOFL	Set if Overflow
F1ETST	
J <sub>4-0</sub>	37 <sub>8</sub>
Memory Z	

## COMMAND CHARACTERISTICS

Binary Equivalent:

$$\begin{array}{r} 0 \ 000111 \\ 0 \ 011 \end{array} = 0 \ 010, \text{ remainder } 0 \ 001.$$

Binary Method:



1. Subtract divisor.
2. Negative result . . 0  $\rightarrow$  quotient (Sign).
3. Add divisor to restore dividend.
4. Restored dividend.
5. Shift left dividend and subtract divisor.
6. Negative result . . 0  $\rightarrow$  quotient.
7. Add divisor to restore dividend.
8. Restored dividend.
9. Shift left dividend and subtract divisor.
10. Positive result . . 1  $\rightarrow$  quotient.
11. Shift left dividend and subtract divisor.
12. Negative result . . 0  $\rightarrow$  quotient.
13. Add divisor for remainder "fix up".
14. Remainder.

Example 1

## BINARY DIVISION

Division of positive numbers is the process of counting how many times one number (the divisor) can be subtracted from another number (the dividend) while still leaving a positive result. The number of times the divisor can be subtracted from the dividend is the result, or quotient. The value remaining after the repeated subtractions becomes the remainder.

The simplest form of binary division is accomplished by first subtracting the divisor from the most significant portion of the dividend. If the subtraction is valid (i.e., a positive result is obtained), a "one" is placed in that position of the quotient. The dividend is then shifted one place to the left and the divisor again subtracted from the dividend. If the subtraction is not valid (i.e., a negative result is obtained), a "zero" is placed in that position of the quotient and the divisor is added to the negative result to restore the dividend. The dividend is then shifted one place to the left and the divisor again subtracted from the dividend. The remainder, after the division, is that portion of the dividend remaining after the last valid subtraction. This method of forming the quotient is illustrated in the following example. The most significant bit of the divisor and the dividend are sign bits.

Decimal Problem:

$$7 \div 3 = 2, \text{ remainder } 1.$$

The first subtraction in Example 1 is described by:

$$D = 2^n (d) \rightarrow D$$

where: D = dividend

d = divisor

n = register length minus 1.

The multiplication of the divisor (d) by  $2^n$  merely aligns the divisor with the correct portion of the dividend and is accomplished by shifting left the divisor n places. For each successive subtraction, the divisor is moved one place to the right:

1st Sub:	$-2^n (d)$	=	1 101000
2nd Sub:	$-2^{n-1} (d)$	=	1 10100
3rd Sub:	$-2^{n-2} (d)$	=	1 1010
Final Sub:	$-2^{n-n} (d)$	=	1 101.

Notice in the previous example that, if the subtraction is invalid (i. e., quotient bit set to "zero"), it is necessary to add back the divisor to restore the dividend to its correct form. That is, the number  $2^n (d)$  is first subtracted from the dividend, then added back. In the 4022A Arithmetic Unit, this function is combined with the next subtraction, which takes  $2^{n-1} (d)$  from the dividend:

$$\begin{aligned} +2^n (d) - 2^{n-1} (d) &= 2^{n-1} (2d - d) \\ &= 2^{n-1} (d). \end{aligned}$$

This number is generated by adding the divisor to the dividend at the  $2^{n-1}$  position.

Example 2 more closely resembles the mechanics of the division process within the Arithmetic Unit. For comparison, the same values ( $7 \div 3$ ) used in Example 1 are used in Example 2.

By comparing Example 2 with Example 1, notice that 4 less steps are required to perform the division in Example 2. Steps 3, 4, and 5 of Example 1 are performed in step 3 below, providing the same result. That is, when a negative result from a subtraction was obtained in Example 1, the divisor was added back to restore the dividend, then the dividend was shifted and the divisor subtracted. In Example 2, when a negative result from a subtraction was obtained, this negative result was shifted and the divisor added to it. The final result of each operation was the same (i. e., step 6 of Example 1 has the same result as step 4 in Example 2). In this manner, a much faster divide cycle is obtained.

In both Examples 1 and 2, notice that the first subtraction is used to determine the sign bit of the quotient. The divide cycle time in the 4022A Arithmetic Unit is further decreased by eliminating this step. Within the Arithmetic Unit, the sign of the quotient is determined by comparing the sign of the divisor with the sign of the dividend. If the signs are alike, the sign bit is "zero" (positive); if the signs are unlike, the sign bit is "one" (negative). Therefore, the first subtraction is not required to determine the sign of the quotient.

			0010	
0	011)0		000111	- 1
		1	101	- 2
		0	101	- 3
			0011	- 4
		0	1110	- 5
			0011	- 6
		1	0000	- 7
			1101	- 8
		0	1110	- 9
			0011	- 10
			0001	

1. Subtract divisor.
2. Negative result ∴ 0 → quotient (Sign).
3. Shift left dividend and add divisor.
4. Negative result ∴ 0 → quotient.
5. Shift left dividend and add divisor.
6. Positive result ∴ 1 → quotient.
7. Shift left dividend and subtract divisor.
8. Negative result ∴ 0 → quotient.
9. Add divisor for remainder "fix-up".
10. Remainder.

#### Example 2

			0010	
0	011)0		000111	- 1
		1	1101	- 2
		0	1110	- 3
			0011	- 4
		1	0000	- 5
			1101	- 6
		0	1110	- 7
			0011	- 8
			0001	- 9

1. Sign of divisor and dividend alike ∴ 0 → quotient.
2. Shift left dividend and subtract divisor.
3. Negative result ∴ 0 → quotient.
4. Shift left dividend and add divisor.
5. Positive result ∴ 1 → quotient.
6. Shift left dividend and subtract divisor.
7. Negative result ∴ 0 → quotient.
8. Add divisor for remainder "fix-up".
9. Remainder.

#### Example 3

Example 3 illustrates the mechanics of division by comparing the signs of the divisor and dividend to determine the sign of the quotient. The same values ( $7 \div 3$ ) used in Examples 1 and 2 are used for comparison.

## COMMAND DESCRIPTION

Fig. DVD. 1 contains a basic flow chart of the Sequence Control States required to execute the DVD command and lists the basic functions performed within each Sequence State. Block diagrams of States 3, 4, and 5 are contained in Fig. DVD. 2.

### Sequence State 1

The DVD command is "fetched" from memory during a normal Sequence State 1. At Last Pulse of State 1, the

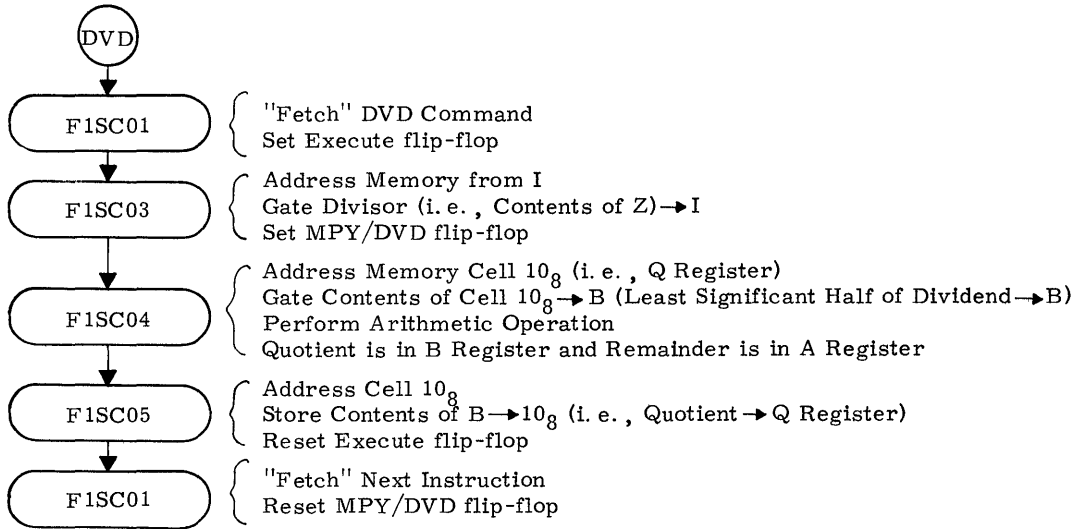


Fig. DVD. 1. DVD Basic Flow Chart

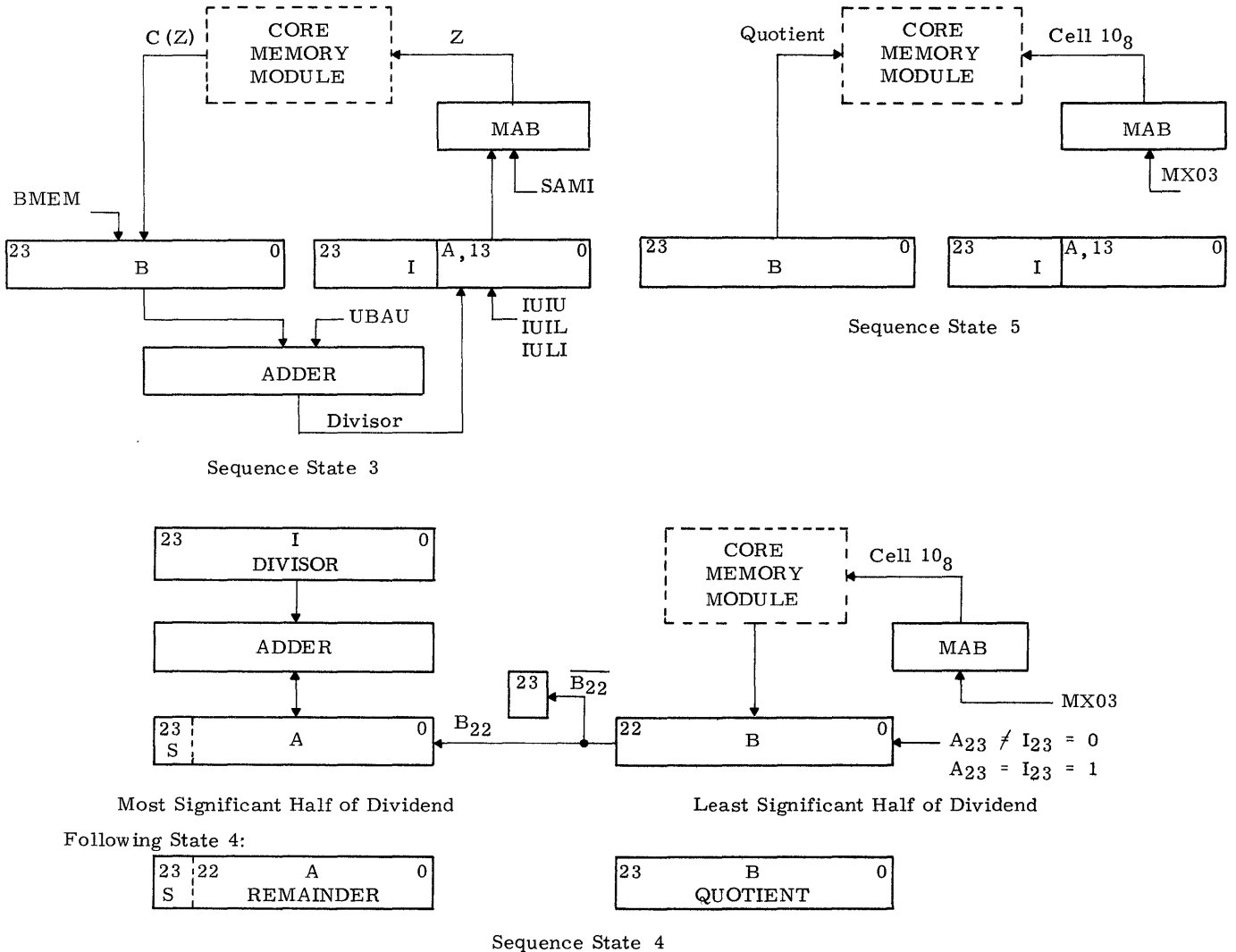


Fig. DVD. 2. DVD Block Diagram

Execute flip-flop (F1XEXC) is set. Therefore, at the end of State 1, the DVD command is contained by the I Register and the Execute flip-flop is set. Following State 1, a non-indexed DVD command enters Sequence State 3.

### Sequence State 3

A timing diagram, including logic equations, for State 3 is contained in Fig. DVD. 3. During State 3, memory is addressed from  $I_A_{13-0}$  (D1SAMI) and the contents of memory location Z (divisor) are gated to the B Register. From the B Register, the divisor is gated to the Adder (D1UBAU) and from the Adder to the I Register (D1IUIU, D1IUIL, D1IULI). At Time 3, the Execute MPY/DVD flip-flop (F1XMDV) is set to indicate that the I Register does not contain the instruction and to provide control for the DVD operation. The DVD command is differentiated from the MPY command by having both F1XMDV and F1XEXC (State 1) set. Following State 3, State 4 is entered.

### Sequence State 4

Sequence State 4 is used to bring the least significant half of the dividend ( $Q_{22-0}$ ) to the B Register and to perform the actual arithmetic operation. To provide these functions, the duration of State 4 is extended by entering Time 6 Envelope.

For ease of understanding, a detailed flow chart of State 4 is contained in Fig. DVD. 4. The logic elements and

associated logic equation used to perform the individual steps of the flow chart are contained in Table DVD. 1.

An example of the arithmetic operation performed in State 4 is provided in Table DVD. 2. For simplicity, 9-bit registers are illustrated. The timing diagram of State 4, contained in Fig. DVD. 5, illustrates the timing associated with this example. Using these aids, little difficulty should be encountered in determining the operation of the Arithmetic Unit for any dividend and divisor values.

During the first portion of State 4, memory cell  $10g$  is addressed ( $G0MX03$ ) and the contents of the Q Register are gated to the B Register. Bits 22-0 of the Q Register contain the least significant half of the dividend.

The J Counter is initially cleared and then preset to 7 ( $G1JP07$ ). The J Counter is then incremented at each shift of the A and B Registers. The count value of the J Counter is then used to determine when the divide cycle is completed ( $J = 37g$ ). The Delay Time Counter is initially preset to  $30g$  and provides timing control for determining when to add (or subtract) and when to shift for each quotient bit generated.

During Time 3 Envelope, the signs of the divisor ( $I_{23}$ ) and dividend ( $A_{23}$ ) are compared to determine the sign bit of the quotient and to provide control for the Overflow check. If the signs are alike, the Strings flip-flop (F1XSTG) is reset; if the signs are unlike, the Strings

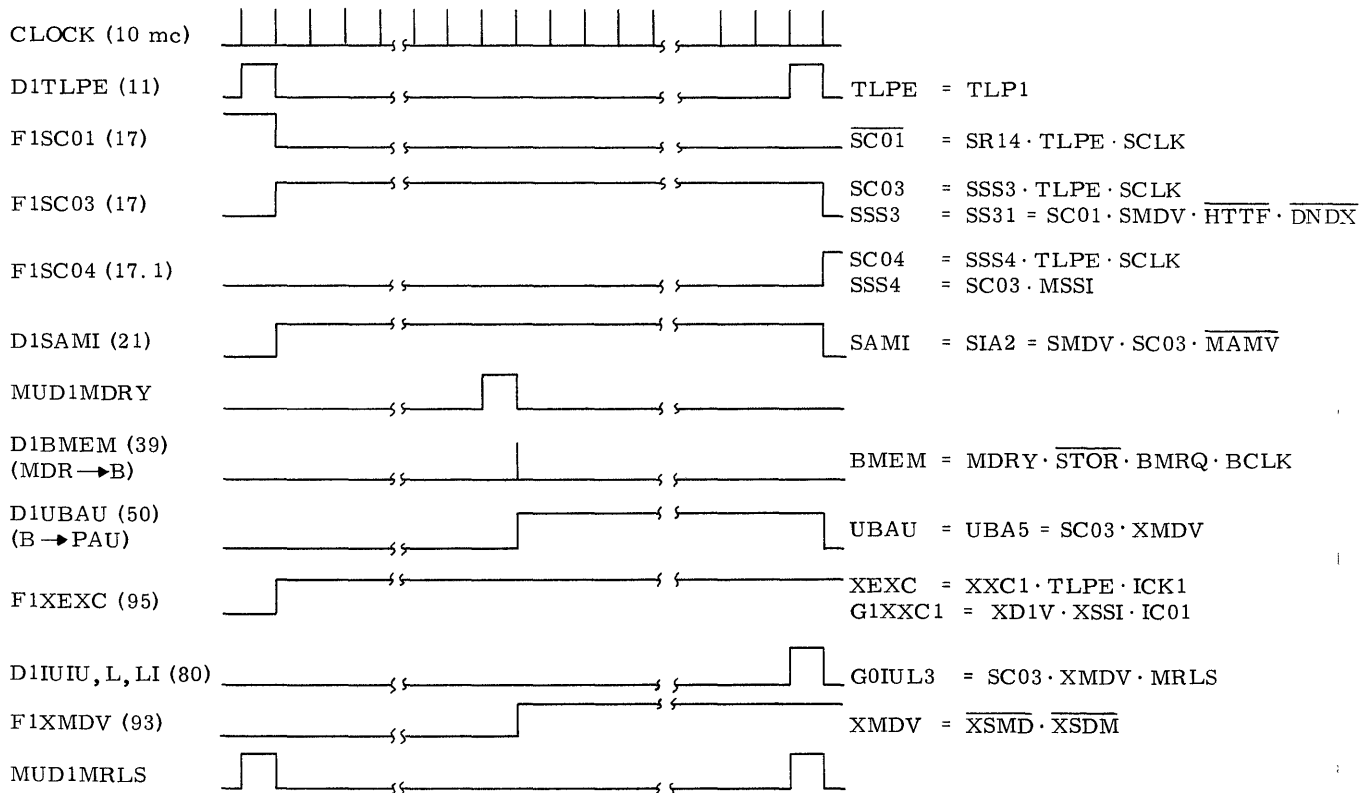


Fig. DVD. 3. DVD Sequence State 3 Timing

flip-flop is set. Also, if the signs are alike, B<sub>0</sub> is armed to set at the first shift. This "one" will represent a positive quotient, since the complement of this bit will be shifted to B<sub>23</sub> at the end of the DVD operation. If the signs are unlike, B<sub>0</sub> will be armed to reset at the first shift, specifying a negative quotient.

The Overflow check is used to determine if the value of the quotient will be too large to be contained in a 24-bit register. If the absolute magnitude of the A Register is equal to or greater than the contents of the I Register, the quotient will exceed the capacity of 24 bits and the Overflow flip-flop (F1UOFL) is set. The Overflow test is made by performing either an addition or subtraction of the A and I Registers. If the signs of A and I are alike (Strings flip-flop reset), I is subtracted from A and if the sign of the result in the Adder does not change, then A is greater than or equal to I and the Overflow flip-flop is set. If I and A have unlike signs (Strings flip-flop set), then I and A are added. If the sign of the result in the Adder is equal to the sign of A, A is greater than or equal to I and the Overflow flip-flop is set. If, however, following the addition or subtraction the sign of the result is not the same as the sign of A, the quotient may be contained in 24 bits and a valid division may take place.

After the test for Overflow, the A and B Registers are shifted left with the complement of the sign bit, as determined by like or unlike signs, gated to B<sub>0</sub>. At the shift, the complement of B<sub>22</sub> is gated to B<sub>23</sub> and the true output of B<sub>22</sub> is gated to A<sub>0</sub>. The J Counter is incremented to indicate that one of the 24 quotient bits has been determined.

Time 6 Envelope is then entered. The Delay Time Counter is preset to 268 and the first addition or subtraction is performed to determine the first data quotient bit and to generate the partial remainder. If the most significant bit of the A Register and the sign of the I Register were alike prior to the last shift (Strings flip-flop reset), a subtraction is performed. If prior to the last shift, the most significant bit of A and the sign of I were unlike (Strings flip-flop set), an addition is performed. The result of this addition or subtraction is gated to the A Register and the most significant bits of A and I are again compared. If they are alike, a valid effective subtraction was performed and the quotient bit (B<sub>0</sub>) is armed to set at the shift. If the most significant bits are unlike, an invalid effective subtraction was performed and the quotient bit (B<sub>0</sub>) is armed to reset. The Strings flip-flop is also armed to set (if most significant bits are unlike) or reset (if most significant bits are alike) at the next shift to provide control for the next effective subtraction.

The A and B Registers are then shifted left with the quotient bit determined above gated to B<sub>0</sub>. The complement of B<sub>22</sub> is gated to B<sub>23</sub> and the true output of B<sub>22</sub> is gated to A<sub>0</sub>. At the shift, the J Counter is incremented to indicate that a quotient bit has been determined.

Each data bit of the quotient is determined in this manner until the last data bit has been determined (J = 368). When the last data bit has been determined, only the B

Register is shifted left. This gates the final data bit to B<sub>0</sub>, the sign bit to B<sub>23</sub>, and preserves the partial remainder in the A Register. At this shift, the J Counter is incremented to 378, indicating that the required number of shifts (24) have occurred.

Following the 24 shifts, the quotient is contained in the B Register, and the "fix-up" of the remainder, if required, is performed. If, prior to the last shift, the sign of the divisor (I<sub>23</sub>) and the sign of the remainder (A<sub>23</sub>) were unlike, the Strings flip-flop was set at the shift of B. If the Strings flip-flop is set, then one too many subtractions were performed and I is added to A to provide the correct remainder. If the Strings flip-flop is reset, the contents of A are gated to the Adder and back to A unchanged.

Following State 4, State 5 is entered to store the quotient contained in the B Register, in cell 108 (Q Register).

### Sequence State 5

A timing diagram and logic equations for State 5 are contained in Fig. DVD. 6.

Memory cell 108 is addressed (G0MX03) and the quotient is gated from the B Register to the Q Register. At Last Pulse of State 5, the Execute flip-flop (F1XEXC) is reset.

During State 1 of the command following DVD, the Execute MPY/DVD flip-flop (F1XMDV) is reset.

#### NOTE

The previous description applies to both positive and negative values. However, when the divisor is negative, the quotient and remainder must be modified to obtain the expected result. If the divisor is negative, a positive one must be added to the quotient and the divisor must be subtracted from the remainder to obtain the expected result. The following example (using decimal numbers) illustrates the quotient and remainder obtained, proves that it is correct, and how it may be modified to obtain the expected result.

Decimal Problem:  $\frac{+4}{-2}$

Result Obtained:  
Quotient = -3, Remainder = -2

Prove Result is Correct:  
Divisor x Quotient + Remainder = Dividend  
 $(-2) \times (-3) + (-2) = +4$

Result Modification:  
Add +1 to Quotient;  $-3 + (+1) = -2$   
Subtract Divisor from Remainder:  
 $(-2) - (-2) = 0.$

As previously described, this is of little consequence in actual mathematical operations within the computer and normally the quotient is used without any "fix-up".

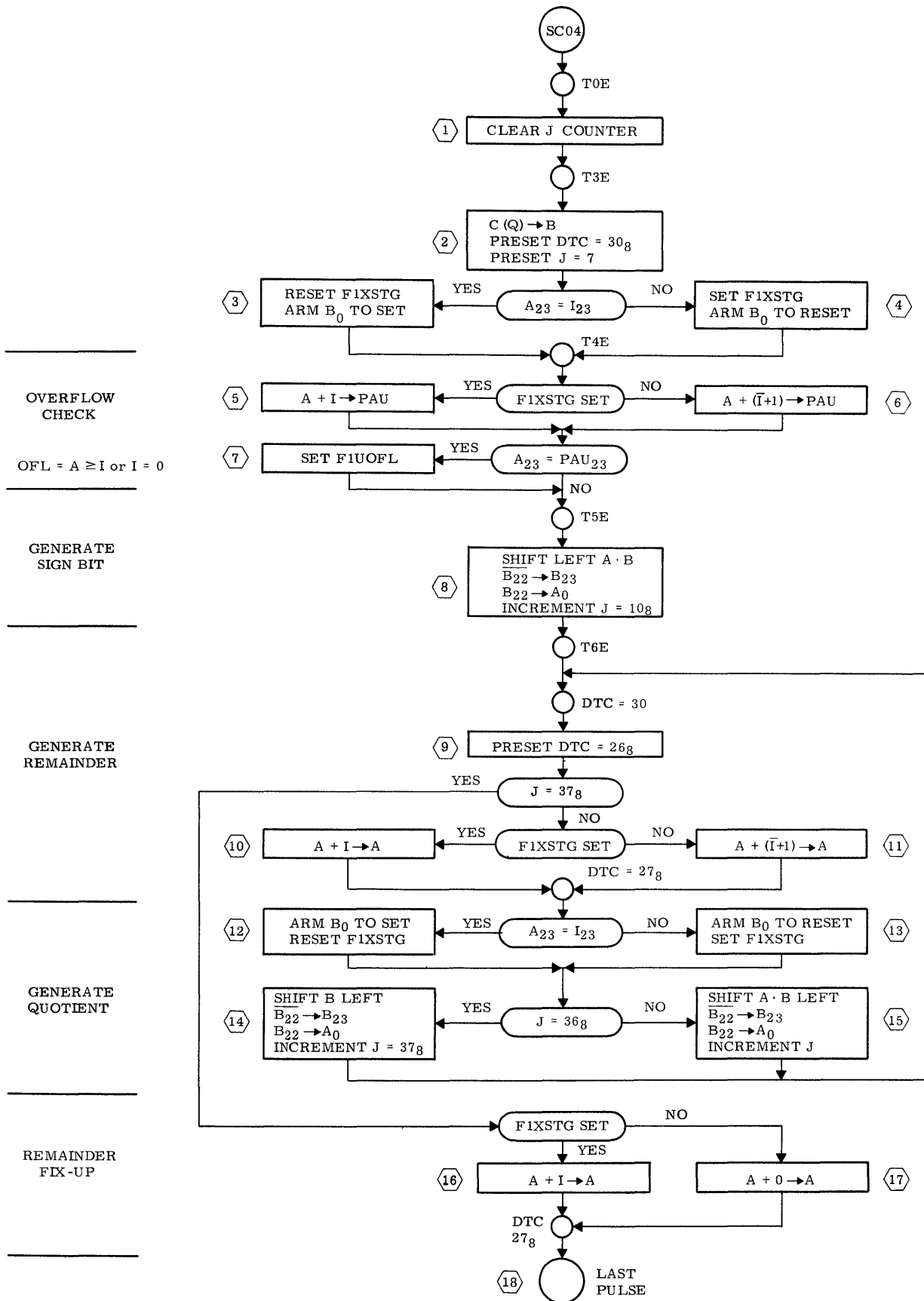


Fig. DVD. 4. State 4 Flow Chart

<p>① Clear J Counter:  <math>D0JJE0 (70) = JCK1 \cdot \overline{TSCA} \cdot \overline{TSCB} \cdot JS4G</math>  <math>G1JS4G (70) = SC04 \cdot DMD1</math></p> <p>② Preset Delay Time Counter to <math>30_8</math>:  <math>G1TPAF (12) = TP30 + TD30</math>  <math>G0TP30 (12) = TSCB \cdot TSCA \cdot TSCC \cdot DMGT</math>  Preset J Counter to 7:  <math>G1JP07 (70) = TP30 \cdot XMDV</math></p> <p>③ Reset Strings flip-flop:  <math>\overline{F1XSTG} (94) = \overline{G0XCA1} \cdot \overline{XDVT} \cdot \overline{ICK1}</math>  <math>G0XCA1 (94) = (\overline{XA23} \cdot \overline{XI23}) + (\overline{AR23} \cdot \overline{IR23})</math>  <math>G1XDVT (94) = UDVI \cdot XT3E</math>  Arm <math>B_0</math> to set at first shift ( <math>\textcircled{8}</math> ):  <math>G1BSL2 (38) = BDIV \cdot [(BI23 \cdot BA23) + (\overline{AR23} \cdot \overline{IR23})]</math></p> <p>④ Set Strings flip-flop:  <math>F1XSTG (94) = XCA1 \cdot XDVT \cdot ICK1</math>  <math>G1XCA1 (94) = (XA23 \cdot \overline{IR23}) + (XI23 \cdot \overline{AR23})</math>  Arm <math>B_0</math> to reset at first shift ( <math>\textcircled{8}</math> ):  <math>G1BSL1 (38) = BDIV \cdot [(BI23 \cdot \overline{AR23}) + (BA23 \cdot \overline{IR23})]</math></p> <p>⑤ Add A and I:  <math>D1UAAU (52) = UAA4 = XMDV \cdot UC04</math>  <math>D1UILA, AU (51) = UIA4 = UC04 \cdot UDIV \cdot XSTG</math></p> <p>⑥ Subtract I from A:  <math>D1UAAU (52) = UAA4 = XMDV \cdot UC04</math>  <math>D1UINA (51) = UIN1 = UC04 \cdot UDIV \cdot \overline{XSTG} \cdot \overline{JE37}</math>  <math>D1UENC (51) = UIN1</math></p> <p>⑦ Set Overflow flip-flop at shift ( <math>\textcircled{8}</math> ):  <math>F1UOFL (54) = UFL5 \cdot UFL6 \cdot ECLK</math>  <math>G1UFL5 (54) = (UA23 \cdot U23S) + (\overline{AR23} \cdot \overline{U23S})</math>  <math>G1UFL6 (54) = ABSL = DDIV \cdot JC04 \cdot TT5E \cdot MRLS</math></p> <p>⑧ Shift left A and B Registers one place:  <math>D1BSLL, U, 1 (40) = ABSL</math>  <math>G0ABSL (63) = DDIV \cdot JC04 \cdot TT5E \cdot MRLS</math>  <math>D1ASLL, U (63. 1) = ABSL</math>  Gate <math>\overline{B_{22}}</math> to <math>B_{23}</math> at shift:  <math>G1B3LD (38) = \overline{BR22} \cdot BDIV</math>  <math>G1B2LD (38) = \overline{BR22} \cdot BDIV</math>  Gate <math>B_{22}</math> to <math>A_0</math> at shift:  <math>G1AMLN (64) = AML1 \cdot \overline{BR22}</math>  <math>G1AMLD (64) = AML1 \cdot AB22</math>  Increment J Counter:  <math>N1JINC (71) = G0JINC = XMDV \cdot TT5E \cdot MRLS \cdot JC04</math></p> <p>⑨ Preset the Delay Time Counter to <math>26_8</math>:  <math>G1TP26 (12) = TD30 \cdot \overline{JE07} \cdot XMDV \cdot DDIV</math></p>	<p>⑩ Add I to A and gate result to A:  <math>D1UAAU (52) = XMDV \cdot UC04</math>  <math>D1UILA, AU (51) = UC04 \cdot UDIV \cdot XSTG</math>  <math>D1AAU, L (62) = AAU2 = ECLK \cdot TE26 \cdot AMDV</math></p> <p>⑪ Subtract I from A and gate result to A:  <math>D1UAAU (52) = UAA4 = XMDV \cdot UC04</math>  <math>D1UINA (51) = UIN1 = UC04 \cdot UDIV \cdot \overline{XSTG} \cdot \overline{JE37}</math>  <math>G1UENC (51) = UIN1</math></p> <p>⑫ Arm <math>B_0</math> to set at shift:  <math>G1BSL2 (38) = BDIV \cdot [(BI23 \cdot BA23) + (\overline{AR23} \cdot \overline{IR23})]</math>  Reset Strings flip-flop at shift:  <math>\overline{F1XSTG} (94) = \overline{XDVT} \cdot \overline{XCA1} \cdot \overline{ICK1}</math>  <math>G1XDVT (94) = UDVT \cdot ASLB</math>  <math>G0XCA1 (94) = (XA23 \cdot \overline{IR23}) + (\overline{AR23} \cdot \overline{XI23})</math></p> <p>⑬ Arm <math>B_0</math> to reset at shift:  <math>G1BSL1 (38) = BDIV \cdot [(BI23 \cdot \overline{AR23}) + (BA23 \cdot \overline{IR23})]</math>  Set Strings flip-flop at shift:  <math>F1XSTG (94) = XCA1 \cdot XDVT \cdot ICK1</math></p> <p>⑭ Shift B left one place:  <math>D1BSLL, U, 1 (40) = ASLB</math>  <math>G0ASLB (63) = DDIV \cdot TE27 \cdot TT6E \cdot \overline{JE37}</math>  Gate the complement of <math>B_{22}</math> to <math>B_{23}</math>:  <math>G1B3LD (64) = \overline{BR22} \cdot BDIV</math>  <math>G1B2LD (64) = \overline{BR22} \cdot BDIV</math>  Increment the J Counter to <math>37_8</math>:  <math>N1JINC = G0JINC = \overline{JE37} \cdot TT6E \cdot XMDV \cdot TEFF</math></p> <p>⑮ Shift left A and B one place:  <math>D1BSLL, U, 1 (40) = ASLB</math>  <math>G0ASLB (63) = DDIV \cdot TE27 \cdot TT6E \cdot \overline{JE37}</math>  <math>D1ASLL, U (63. 1) = ASL1</math>  <math>G0ASL1 (63) = DDIV \cdot \overline{J367} \cdot TT6E \cdot TE27</math>  Gate the complement of <math>B_{22}</math> to <math>B_{23}</math>:  <math>G1B3LD (64) = \overline{BR22} \cdot BDIV</math>  <math>G1B2LD (64) = \overline{BR22} \cdot BDIV</math>  Gate <math>B_{22}</math> to <math>A_0</math>:  <math>G1AMLN (64) = AML1 \cdot \overline{BR22}</math>  <math>G1AMLD (64) = AML1 \cdot BR22</math>  Increment the J Counter:  <math>N1JINC = G0JINC = \overline{JE37} \cdot TT6E \cdot XMDV \cdot TEFF</math></p> <p>⑯ Add A to I and gate result to A:  <math>D1UAAU (52) = XMDV \cdot UC04</math>  <math>D1UILA, AU (51) = UC04 \cdot UDIV \cdot XSTG</math>  <math>D1AAU, L (62) = ECLK \cdot TE26 \cdot AMDV</math></p> <p>⑰ Gate A to Adder and back to A:  <math>D1UAAU (52) = XMDV \cdot UC04</math>  <math>D1AAU, L (62) = ECLK \cdot TE26 \cdot AMDV</math></p> <p>⑱ Generate Last Pulse:  <math>D1TLPE (11) = \overline{JE37} \cdot AMDV \cdot TE27</math></p>
--	---

Table DVD. 1. State 4 Flow Chart Equations

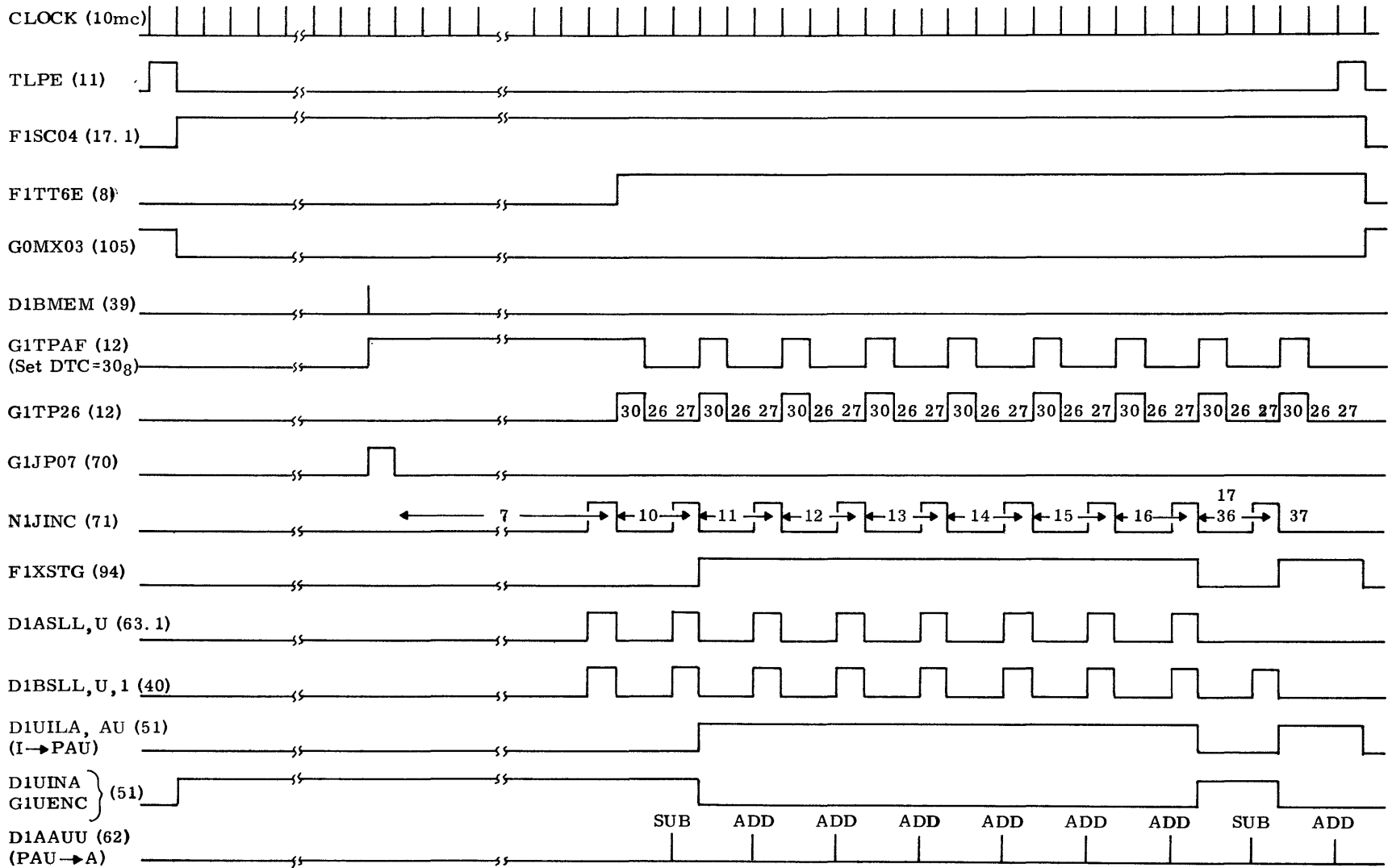


Fig. DVD. 5. DVD Sequence State 4 Example (+27 ÷ +13)



$$\begin{array}{r}
 \text{S} \\
 \text{I} = 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1 = +13 \\
 \text{A} = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 = 0 \\
 \text{B} = 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1 = +27 \\
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} +13 \overline{) +27} \\
 \underline{\phantom{0000}26} \\
 \phantom{0000}1
 \end{array}$$

J COUNT	DTC	XSTG	A REGISTER								B REGISTER								B <sub>0</sub> INPUT	ACTION							
			S	7	6	5	4	3	2	1	0	S	7	6	5	4	3	2			1	0					
7	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	T3E - A <sub>23</sub> = I <sub>23</sub>			
7	30		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	T4E - Overflow Test	
10	30		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1	T5E - Shift - $\overline{\text{Sign}}$
10	26		1	1	1	1	1	0	0	1	1														0	$\overline{\text{XSTG}}$ = Subtract	
11	27	1	1	1	1	1	0	0	1	1	0														0	Shift	
	30																										
	26		1	1	1	1	1	0	0	1	1														0	XSTG = Add	
12	27	1	1	1	1	1	0	0	1	1	0														0	Shift	
	30																										
	26		1	1	1	1	1	0	0	1	1														0	XSTG = Add	
13	27	1	1	1	1	1	0	0	1	1	1														0	Shift	
	30																										
	26		1	1	1	1	1	0	1	0	0														0	XSTG = Add	
14	27	1	1	1	1	1	0	1	0	0	1														0	Shift	
	30																										
	26		1	1	1	1	1	0	1	1	0														0	XSTG = Add	
15	27	1	1	1	1	1	0	1	1	0	0														0	Shift	
	30																										
	26		1	1	1	1	1	1	0	0	1														0	XSTG = Add	
16	27	1	1	1	1	1	1	0	0	1	1														0	Shift	
	30																										
	26		0	0	0	0	0	0	0	0	0														1	XSTG = Add	
17=36	27	0	0	0	0	0	0	0	0	0	1														1	Shift	
	30																										
	26		1	1	1	1	1	0	1	0	0														0	$\overline{\text{XSTG}}$ = Subtract	
37	27	1																							0	Shift B Only	
	30																										
	26		0	0	0	0	0	0	0	0	0	1														XSTG = Add = Remainder Fix-Up	
	27																									Last Pulse	

↑ Remainder = 1      ↑ Quotient = 2

Table DVD.2. Divide Example (27 ÷ 13)

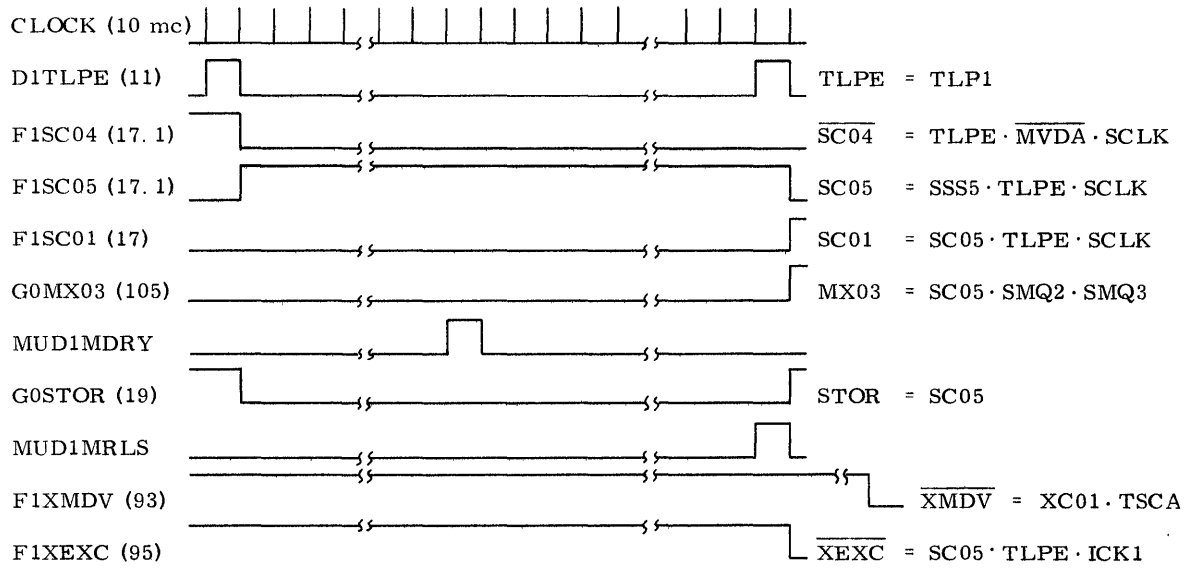
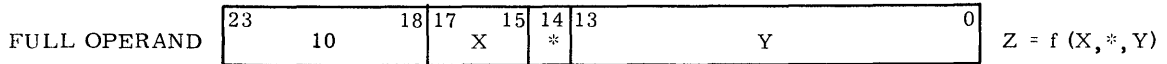


Fig. DVD. 6. Sequence State 5 Timing

# ERA - EXCLUSIVE OR TO A



ERA compares the corresponding bits of A with those of core cell Z. If the corresponding bits of both A and Z are alike, a "zero" is placed in that position of A. If the corresponding bits of A and Z are not alike, a "one" is placed in that position of A.

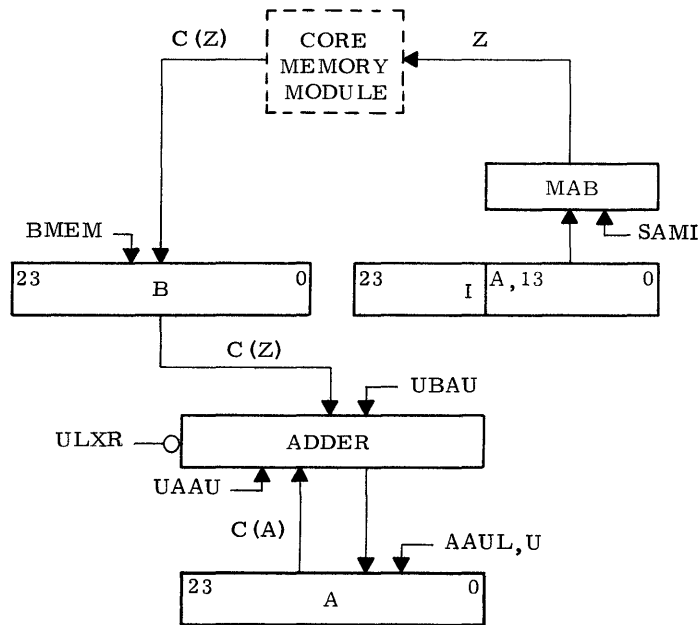
A non-indexed ERA command is executed during Sequence Control State 4 (SC04). Memory location Z is addressed from  $I_A, 13-0$  (D1SAMI) during State 4. The contents of memory location Z are then gated to the B Register by D1BMEN during the clock pulse of Memory Data Ready (MUD1MDRY). From B, the contents of memory location Z are gated to the Adder Unit (D1UBAU). At the same time, the contents of the A Register are gated to the Adder Unit (D1UAAU). The Logical OR control signal, D0ULXR, is also applied to the Adder Unit to enable the Logical OR function. Effectively, the contents of memory cell Z and the A Register are summed in the Adder with carry generation inhibited by D0ULXR to provide the Logical OR result. The result is gated back to the A Register (D1AAUL,U) to complete the ERA execution cycle.

To exemplify the ERA comparison, consider the following; 4 bits are used for simplicity.

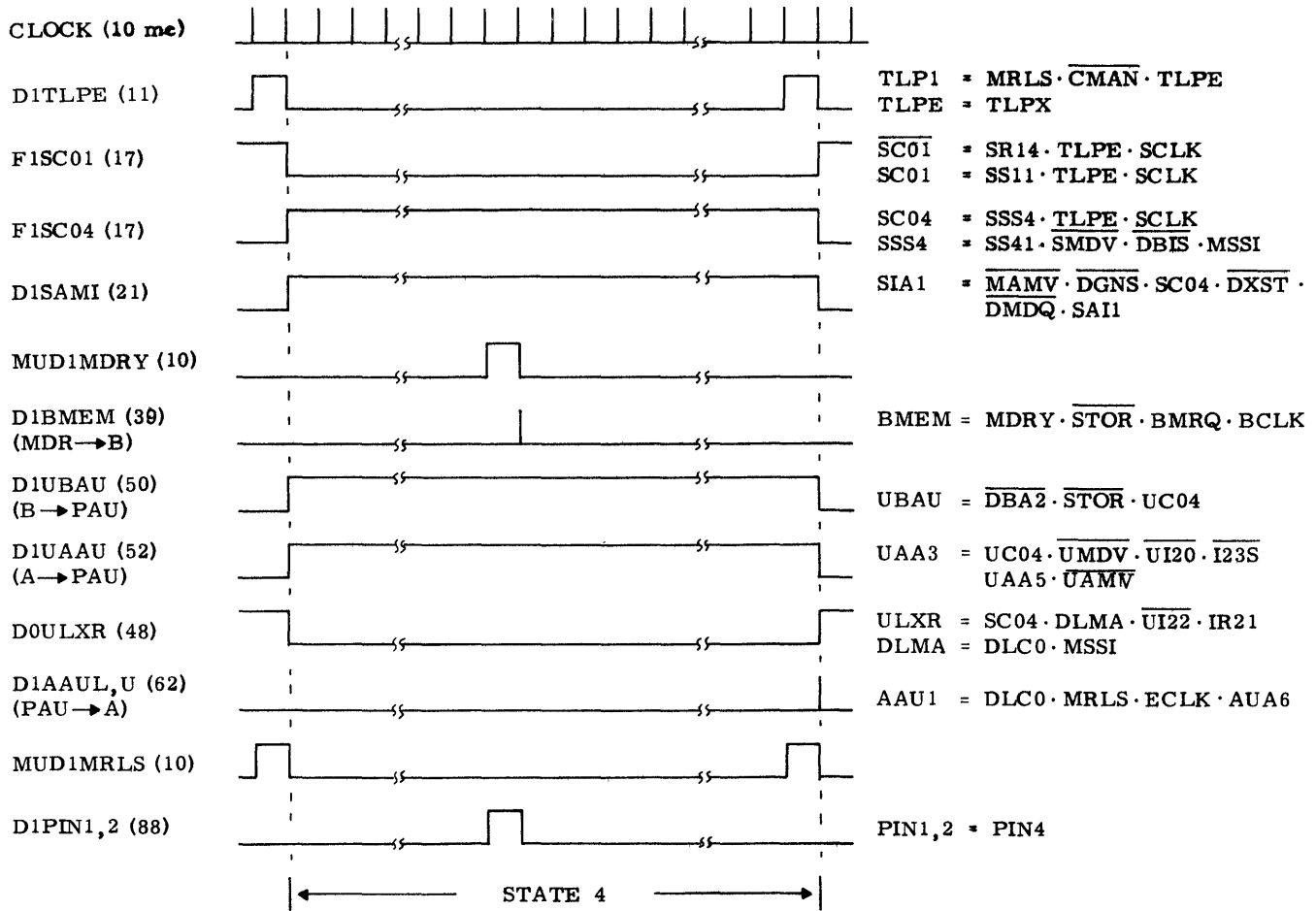
- Contents of A Register = 0011
- Contents of cell Z = 0101
- Result Placed in A Register = 0110.

Non-Indexed Word Times.	2 (S1,S4)
Interruptable Following Execution?	Yes
<b>CHANGES FOLLOWING EXECUTION</b>	
$A_{23-0}$	$C(Z_{23-0}) \oplus C(A_{23-0})$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	
Memory Z	

COMMAND CHARACTERISTICS



ERA BLOCK DIAGRAM



ERA TIMING DIAGRAM

# GEN 1 COMMANDS

GEN 1 commands are used for bit manipulation of the A Register. By controlling the operation of the serial Full Adder (G1AFNS), individual bits of the A Register may be shifted in position, masked by ones or zeros, tested for polarity, or counted for the number of ones or zeros contained therein, etc. Microcoding of the instruction may be manipulated to affect the J counter and Test flip-flop as well as manipulate the A Register.

GEN 1 commands are identified by the operation code 05 (bits 23-18). They may be indexed, if desired, but cannot be relative addressed. Indexing of GEN 1 must be handled with care since it may change the microcoded action bits, thus changing the command functions. The microcoded action bits of the command format (bits 14-0) provide the logic signals necessary to perform the individual commands. Fig. GN1.1 illustrates the use of the microcoded bits and the control each provides. The instruction format is divided into microcoded categories as defined in the following:

05 = GEN 1 Instruction Octal Code.

B = B Control - Controls the B input to the Full Adder.

X = Index Word Indicator.

T = Test Control - 1 indicates Test flip-flop (F1ETST) operation.  
- 0 indicates no Test flip-flop operation.

C = Carry Control - Controls the Full Adder Carry flip-flop (F1AFNP) and, therefore, the Carry input of the Full Adder.

S = Shift Control.

A = A Control - Controls the A input to the Full Adder.

K = Length of shift or bit position of A Register designator.

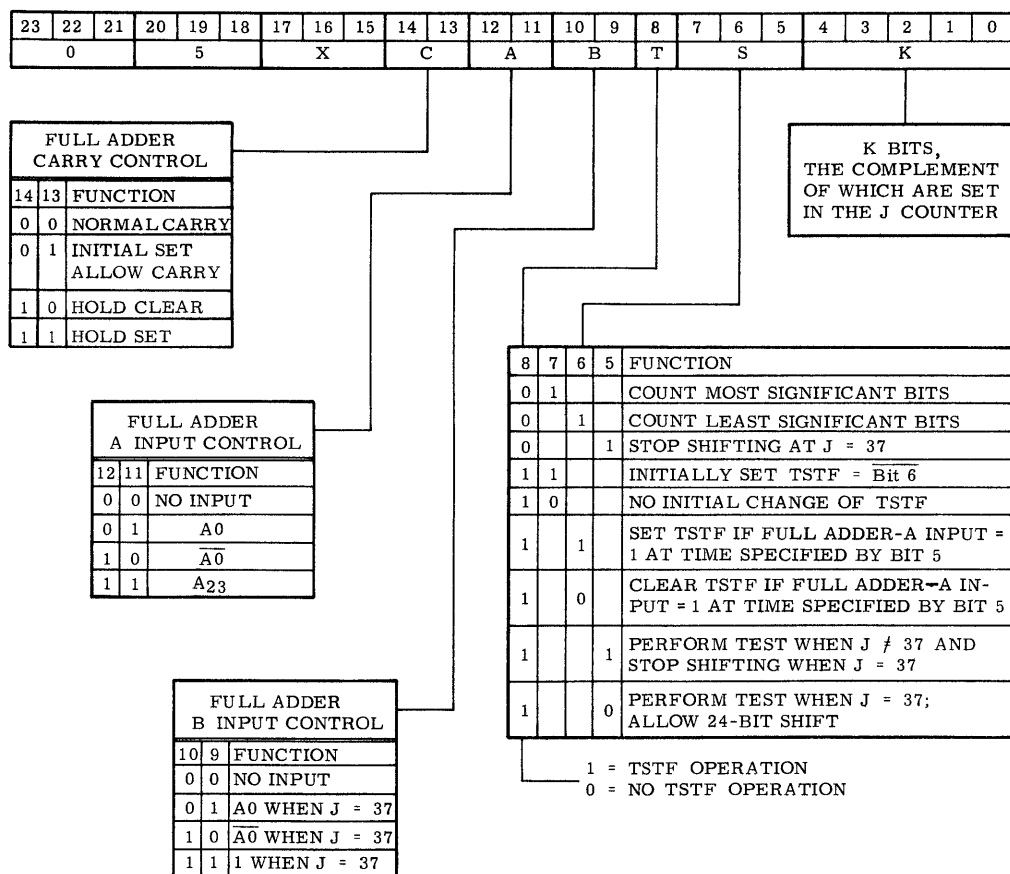


Fig. GN1.1. Action Control Bits of GEN 1 Commands

There are 1024 unique GEN 1 instruction octal words. Thirty-seven of these have been determined to be useful enough to justify assigning them mnemonics and are listed in Table GN1.1. There are three categories into which these commands fall: bit manipulation, Test flip-flop operation, and shift A right. By associating the control bit configuration of Table GN1.1 to the control bit action of Fig. GN1.1, it may be seen how the various commands are implemented.

Basic timing for all GEN 1 commands is described below. Timing and Block Diagrams of the GEN 1 commands in table GN1.1 denoted by shading, are provided to illustrate the operation of GEN 1 commands. These commands were selected as being representative of all GEN 1 commands. A general description of all GEN 1 commands is presented. Because of the similarity of commands within each group, little difficulty should be encountered in determining how any GEN 1 command is implemented.

## BASIC TIMING

All GEN 1 commands are "fetched" during a normal Sequence Control State 1. Following State 1, all non-indexed GEN 1 commands are executed during Sequence Control State 4. Since the execution of GEN 1 commands do not require the use of memory, Memory Request (G1SMRQ) is inhibited during State 4. The basic timing of State 4 of all GEN 1 commands is the same. These basic timing signals are shown in Fig. GN1.2.

The Grey Code Sequence Time Counter (F1TSCA, B, C) is incremented by the first six clock pulses of State 4. That is, since memory is not requested, the Sequence Time Counter cannot and does not await Data Ready and Memory Release. Because GEN 1 commands require serial shifting of the A Register, State 4 is extended by using the Delay Time Counter (F1TAFF - F1TEFF) to allow sufficient time to shift the A Register and to determine when all 24 shifts have occurred.

The Delay Time Counter is a straight binary counter. It is incremented until it is equal to  $30_8$ , by each clock pulse after Time 6 Envelope is entered. Time 6 Envelope is entered after 6 clock pulses of State 4 as determined by the Sequence Time Counter. Allowing the Delay Time Counter to increment to  $30_8$ , defines  $24_{10}$  clock pulses that may be used to shift the A Register. When the Delay Time Counter is equal to  $30_8$ , Last Pulse (D1TLPE) is enabled to end State 4 and consequently completes the execution of the GEN 1 command.

The timing diagram also illustrates that the J Counter is cleared (D0JJE0) and the complement of  $I_{4-0}$  (K Bits) are always transferred to the J Counter.

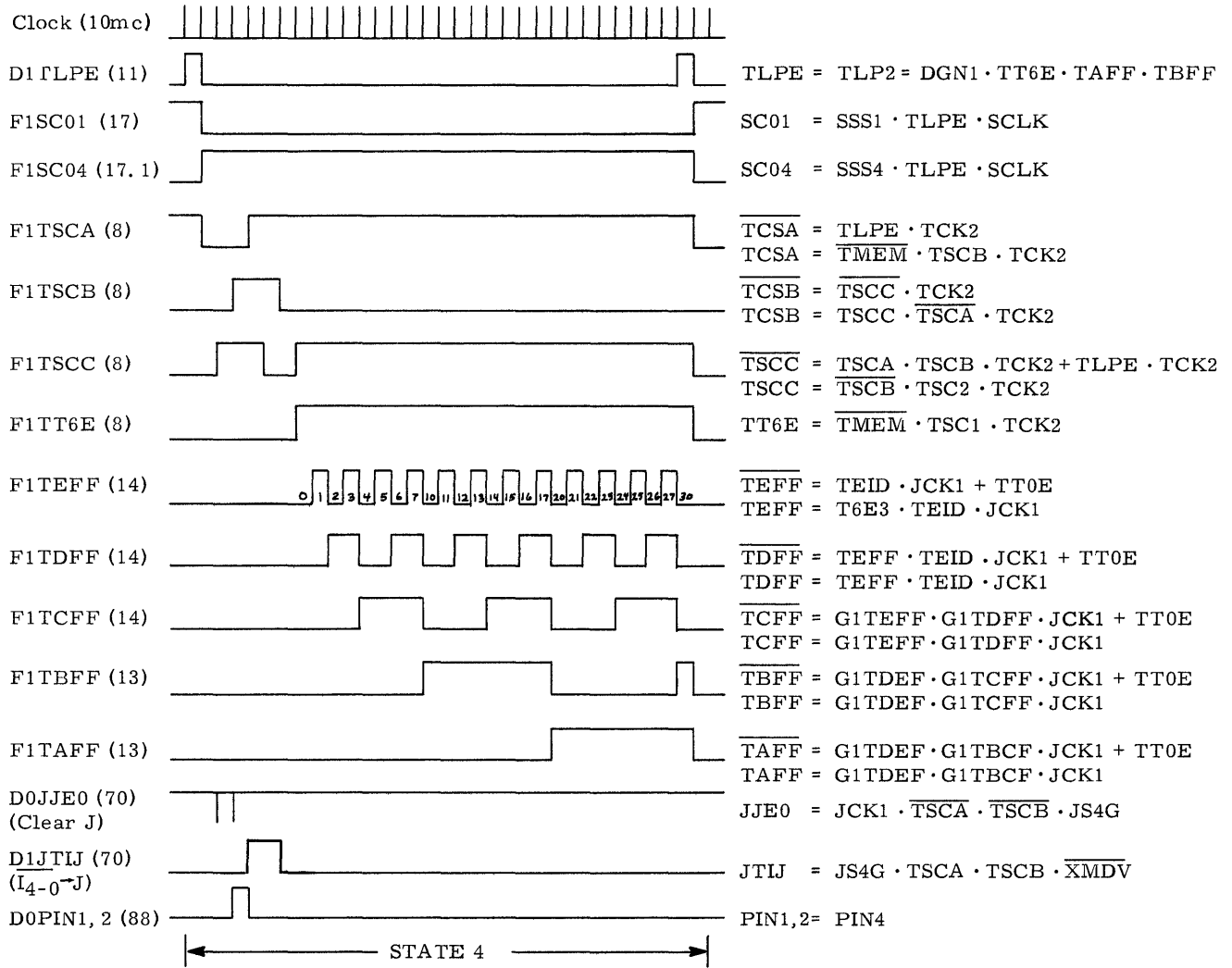


Fig. GN1.2. GEN 1 Basic Timing Diagram

MNEMONIC	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIT MANIPULATION															
ADO	0	0	0	1	1	1	0	0	0	0	← K →				
CBK	1	0	0	1	1	1	0	0	0	0	← K →				
CLO	0	0	0	1	0	0	0	0	1	0	1	1	1	1	1
CLZ	1	1	1	0	0	0	0	0	1	0	1	1	1	1	1
CMO	0	0	0	1	0	0	0	1	0	0	1	1	1	1	1
CMZ	1	1	1	0	0	0	0	1	0	0	1	1	1	1	1
CPL	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
IBK	0	0	0	0	0	1	0	0	0	0	← K →				
LBM	1	1	0	0	1	1	0	0	0	0	← K →				
LDO	0	0	0	0	1	1	0	0	0	0	← K →				
LDZ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LMO	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
NEG	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
RBK	1	0	0	1	0	1	0	0	0	0	← K →				
SBK	1	0	0	1	1	0	0	0	0	0	← K →				
TEST FLIP-FLOP OPERATION															
REV	1	1	1	0	0	0	1	0	0	0	← K →				
RNZ	0	0	0	1	0	0	1	0	0	1	1	1	0	0	0
ROD	0	0	0	1	0	0	1	0	0	0	← K →				
RST	0	0	0	1	0	0	1	1	1	0	1	1	1	1	1
SET	0	0	0	1	0	0	1	1	0	0	1	1	1	1	1
SEV	1	1	1	0	0	0	1	0	1	0	← K →				
SNZ	0	0	0	1	0	0	1	0	1	1	1	1	0	0	0
SOD	0	0	0	1	0	0	1	0	1	0	← K →				
TER	1	0	0	1	0	1	1	1	0	0	← K →				
TES	1	0	0	1	1	0	1	1	0	0	← K →				
TEV	1	1	1	0	0	0	1	1	1	0	← K →				
TNM	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0
TNZ	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0
TOD	0	0	0	1	0	0	1	1	1	0	← K →				
TOR	1	0	0	1	0	1	1	1	1	0	← K →				
TOS	1	0	0	1	1	0	1	1	1	0	← K →				
TSC	0	0	0	1	0	0	1	1	0	1	← K →				
TZC	1	1	0	1	0	0	1	1	0	1	1	1	0	0	0
TZE	0	0	0	1	0	0	1	1	0	1	1	1	0	0	0
SHIFT A RIGHT															
SRA	0	0	1	1	0	0	0	0	0	1	← K →				
SRC	0	0	0	1	0	0	0	0	0	1	← K →				
SRL	0	0	0	0	0	0	0	0	0	1	← K →				


 Timing and Block diagrams of the Gen 1 commands denoted by shading are provided in the following text and are representative of the different types of GEN 1 commands.

Table GN1.1. Microcoded Bit Configuration of GEN 1 Commands



## ADO-ADD ONE TO BIT K

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		070		0	K		

ADO adds plus one to bit position Z in the A Register. Carries out of A<sub>23</sub>, resulting from the summation, are lost but the Overflow flip-flop (F1UOFL) is not affected.

During State 4, the complement of I<sub>4-0</sub> (Z) is gated to the J Counter. The A Register is then shifted right circular thru the serial adder, with the J Counter incremented at each shift. When the J Counter is incremented to 37<sub>8</sub>, bit position Z will be shifted from A<sub>0</sub> and a "one" is added to it in the serial adder. The result is shifted to A<sub>23</sub>. The Carry flip-flop (F1AFNP) provides normal carry propagation if a carry resulted from the summation. The A Register continues to shift until 24 shifts have occurred, as determined by the Delay Time Counter.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	$C(A) + 2^Z$
Q <sub>23-0</sub>	
P <sub>15-0</sub>	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	$27_8 - Z$
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
GOAFNB	67	$JE37 \cdot AGN1 \cdot (IR10 \cdot AR00 + IR09 \cdot AR00)$	Enabled when J = 37 <sub>8</sub>
G1AFNB	67	$AI10 \cdot AI09 \cdot JE37 \cdot DGN1$	Enabled when J = 37 <sub>8</sub>
GOAFNA	65	$AR00 \cdot ANA2$	Gates A <sub>0</sub> to Adder
F1AFNP	66	$AFNP = AFL3 \cdot AFNC \cdot BCLK$ $\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot BCLK$	Normal Carry propagation
DIASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. 24 times
N1J1NC	71	$JIN2 \cdot JIN4$	Increment J 24 times

## CBK-CHANGE BIT K

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		470		0	K		

CBK complements bit Z of the A Register. All other bit positions of A remain unchanged. Z may be specified from 0 to 30 (decimal), however, if Z exceeds 23, A will be unchanged.

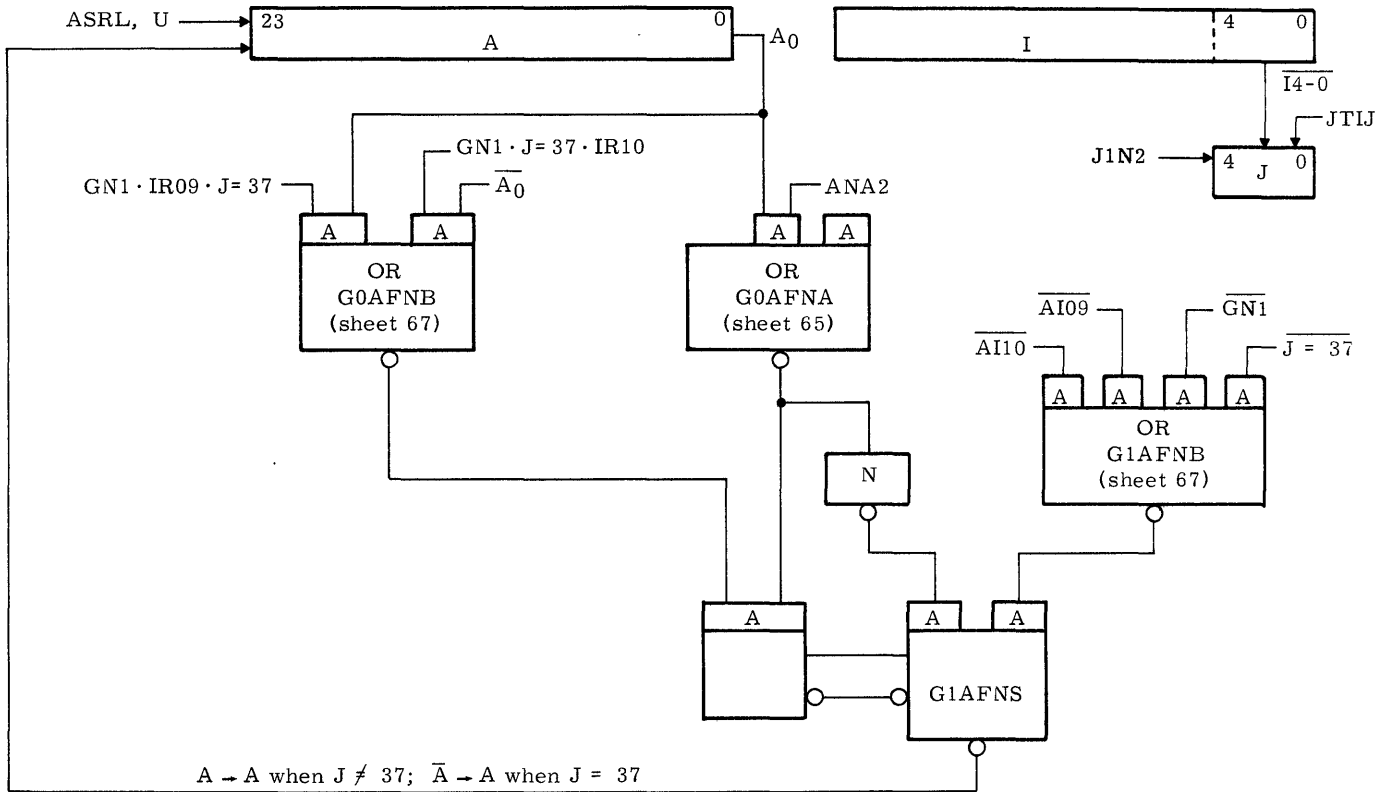
Timing and block diagrams of the CBK command are contained in Fig. GN1.3, 4.

During State 4, the complement of I<sub>4-0</sub> (Z) is gated to the J Counter. The A Register is then shifted right circular thru the serial adder, with the J Counter incremented at each shift. When the J Counter is incremented to 37<sub>8</sub>, bit Z will be shifted from A<sub>0</sub> and a "one" is added to it in the serial adder. The result is shifted to A<sub>23</sub>. Any carry resulting from the summation is lost. In this manner, only bit Z is changed. The A Register continues to be shifted until 24 shifts have occurred, as determined by the Delay Time Counter.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	$a_{23} \dots \overline{a_Z} \dots a_0$
Q <sub>23-0</sub>	
P <sub>14-0</sub>	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	$27_8 - Z$
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$JE37 \cdot AGN1 \cdot (IR10 \cdot AR00 + IR09 \cdot \overline{AR00})$	Enabled when $J = 37_8$
G1AFNB	67	$AI10 \cdot AI09 \cdot JE37 \cdot DGN1$	Enabled when $J = 37_8$
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to Adder
F1AFNP	66	$\overline{AFNP} = SC01 \cdot BCLK$	Inhibit carry propagation
DIASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. 24 places
N1J1NC	71	$JIN2 \cdot JIN4$	Increment J 24 times



Condition	G0AFNB Output	G1AFNB Output	G0AFNA Output	F1AFNP Output	G1AFNS Output
$A_0 \cdot J \neq 37$	1	0	0	0	1
$\overline{A}_0 \cdot J \neq 37$	1	0	1	0	0
$A_0 \cdot J = 37$	0	1	0	0	0
$\overline{A}_0 \cdot J = 37$	0	1	1	0	1

Fig. GN1.3. CBK Block Diagram

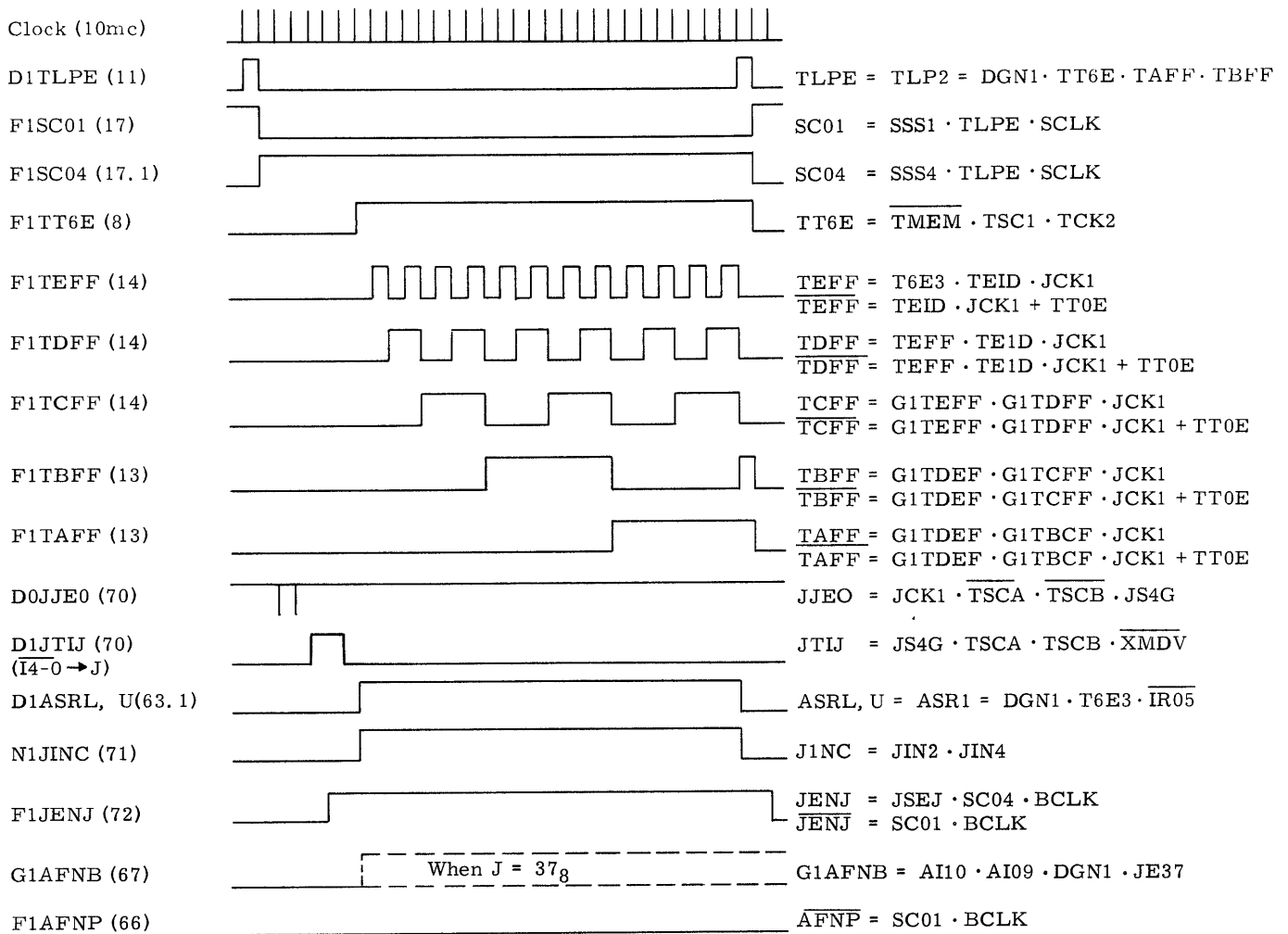
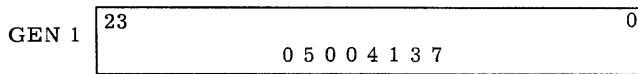


Fig. GN1.4. CBK Timing Diagram

### CLO-COUNT LEAST SIGNIFICANT ONES



CLO counts the number of "one" bits to the right of the rightmost "zero" bit in the A Register. The count value is placed in the J Counter. If A equals 77777778, the count in J is 24<sub>10</sub>.

NOTE

An LXC command must follow a CLO command before another GEN 1 command is executed. This is required to save the J Counter value.

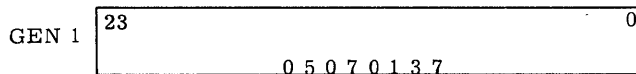
During State 4, the complement of I<sub>4-0</sub> is gated to the J Counter. Since I<sub>4-0</sub> are all "ones", this clears the J Counter. The A Register is then shifted right circular thru the serial adder. The B inputs (G0AFNB, G1AFNB) are disabled during the CLO command. As the A Register is shifted right, each "one" bit increments the J Counter until the first "zero" bit is shifted from A<sub>0</sub>. The first "zero" bit clears the Enable J flip-flop (F1JENJ) which inhibits further incrementation of the J Counter. Therefore, at the end of 24 shifts, the J Counter contains the number of least significant "one" bits in the A Register.

Interruptable Following Execution?		No
CHANGES FOLLOWING EXECUTION		
A <sub>23-0</sub>		
Q <sub>23-0</sub>		
P <sub>14-0</sub>	C(P) + 1	
F1WPMT		
F1UOFL		
F1ETST		
J <sub>4-0</sub>	Number of rightmost "ones" in the A Register	
Memory Z		

COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$G0AFNB = (\overline{IR10} \cdot \overline{IR09}) + JE37$	Disabled
G1AFNB	67	$G1AFNB = (\overline{IR10} \cdot \overline{AR00}) + (\overline{IR09} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to Adder
F1AFNP	66	$\overline{AFNP} = AFL3 \cdot \overline{AFNC} \cdot \overline{BCLK}$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JINC = JIN2 \cdot \overline{AFNA} \cdot \overline{IR05}$ $JIN2 = F1JENJ$ $JENJ = JDEJ \cdot ASR1 \cdot \overline{AFNA}$	Counts least significant "ones" from $A_0$

### CLZ-COUNT LEAST SIGNIFICANT ZEROS



CLZ counts the number of "zero" bits to the right of the rightmost "one" bit in the A Register. The count value is placed in the J Counter. If A equals 00000000<sub>8</sub> the count in J is 24<sub>10</sub>.

#### NOTE

An LXC command must follow a CLZ command before another GEN 1 command is executed. This is required to save the J Counter value.

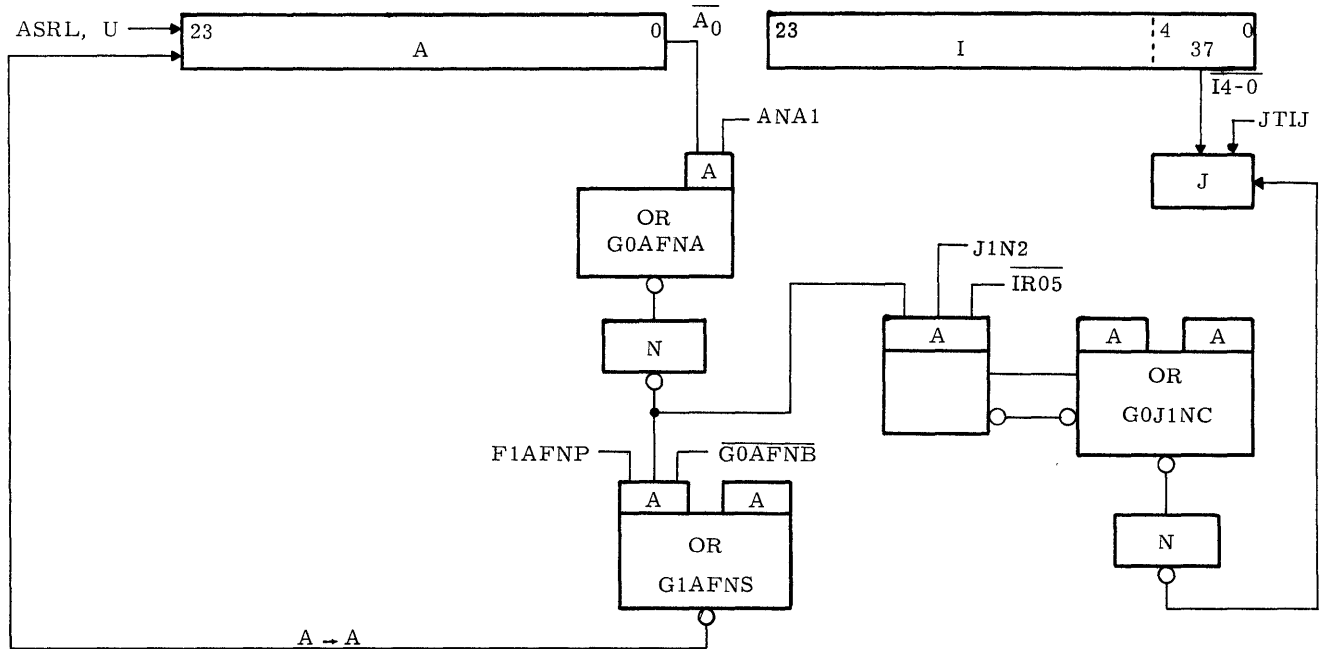
Timing and block diagrams of the CLZ command are contained in Fig. GN1.5, 6.

During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. Since  $I_{4-0}$  are all "ones", this clears the J Counter. The A Register is then shifted right circular with the complement of  $A_0$  applied to the Adder. The Carry flip-flop (F1AFNP) is held set applying a continuous "one" to the Adder. In this manner, the Adder output is equal to the true output of  $A_0$ . The Adder output is shifted to  $A_{23}$  leaving the A Register unchanged after 24 shifts. As the A Register is shifted, the J Counter is incremented by each "zero" from the A Register until the first "one" is detected. The first "one" from  $A_0$  clears the Enable J flip-flop (F1JENJ) which inhibits further incrementation of the J Counter. Therefore, at the end of 24 shifts, the J Counter will contain the number of least significant "zero" bits in the A Register.

Interruptable Following Execution?	No
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	Number of rightmost "zeros" in the A Register
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = JE37 + (AI10 \cdot \overline{AR00}) + (AI09 \cdot \overline{AR00})$	Disabled
G1AFNB	67	$\overline{AFNB} = JE37 + (\overline{IR10} \cdot \overline{IR09})$	Disabled
G0AFNA	65	$\overline{AR00} \cdot ANA1$	Gates $\overline{A_0}$ to Adder
F1AFNP	66	$AFL2$	Held set, applies "1" to Adder
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shift A Reg. 24 places
N1J1NC	71	$JINC = JIN2 \cdot \overline{AFNA} \cdot \overline{IR05}$ $JIN2 = F1JENJ$ $JENJ = JDEJ \cdot ASR1 \cdot \overline{AFNA}$	Count least significant "zeros" from $A_0$



Condition	G0AFNB Output	G1AFNB Output	G0AFNA Output	F1AFNP Output	G1AFNS Output	F1JENJ Output
$A_0$	1	0	1	1	1	-
$\overline{A_0}$	1	0	0	1	0	0

Fig. GN1. 5. CLZ Block Diagram

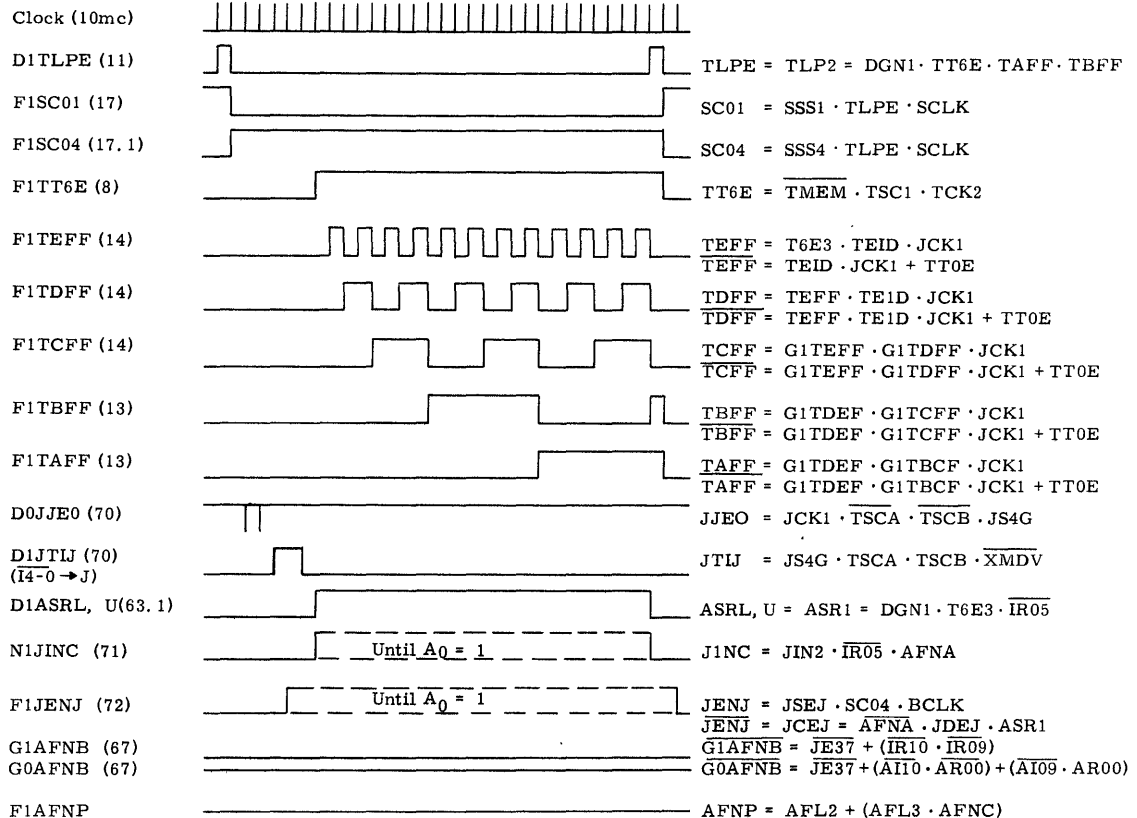
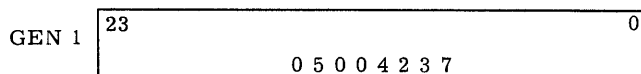


Fig. GN1. 6. CLZ Timing Diagram

## CMO-COUNT MOST SIGNIFICANT ONES



CMO counts the number of "one" bits to the left of the leftmost "zero" bit in the A Register. The count value is placed in the J Counter. If A equals 77777777<sub>8</sub>, the count in J is 24<sub>10</sub>.

### NOTE

An LXC command must follow a CMO command before another GEN 1 command is executed. This is required to save the J Counter value.

Timing and block diagrams of the CMO command are contained in Fig. GN1.7, 8.

During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. Since  $I_{4-0}$  are all "ones", this clears the J Counter. The A Register is then shifted right circular with the true output of  $A_0$  shifted thru the Adder. Each "zero" bit shifted from  $A_0$  causes the J Counter to be reset. In this manner, after the 24 shifts of the A Register, the count in the J Counter corresponds to the number of most significant "one" bits.

Interruptable Following Execution?	No
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C(P) + 1
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	Number of left most "ones" in the A Register
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{G0AFNB} = (\overline{IR10} \cdot \overline{IR09}) + \overline{JE37}$	Disabled
G1AFNB	67	$\overline{G1AFNB} = (\overline{IR10} \cdot \overline{AR00}) + (\overline{IR09} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to Adder
F1AFNP	66	$AFNP = AFL3 \cdot \overline{AFNC} \cdot BCLK$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN2 \cdot AFNA \cdot \overline{IR05}$	Each "one" increments the J Counter
G1JP00	70	$ASR1 \cdot JP0A$	Each "zero" clears the J Counter

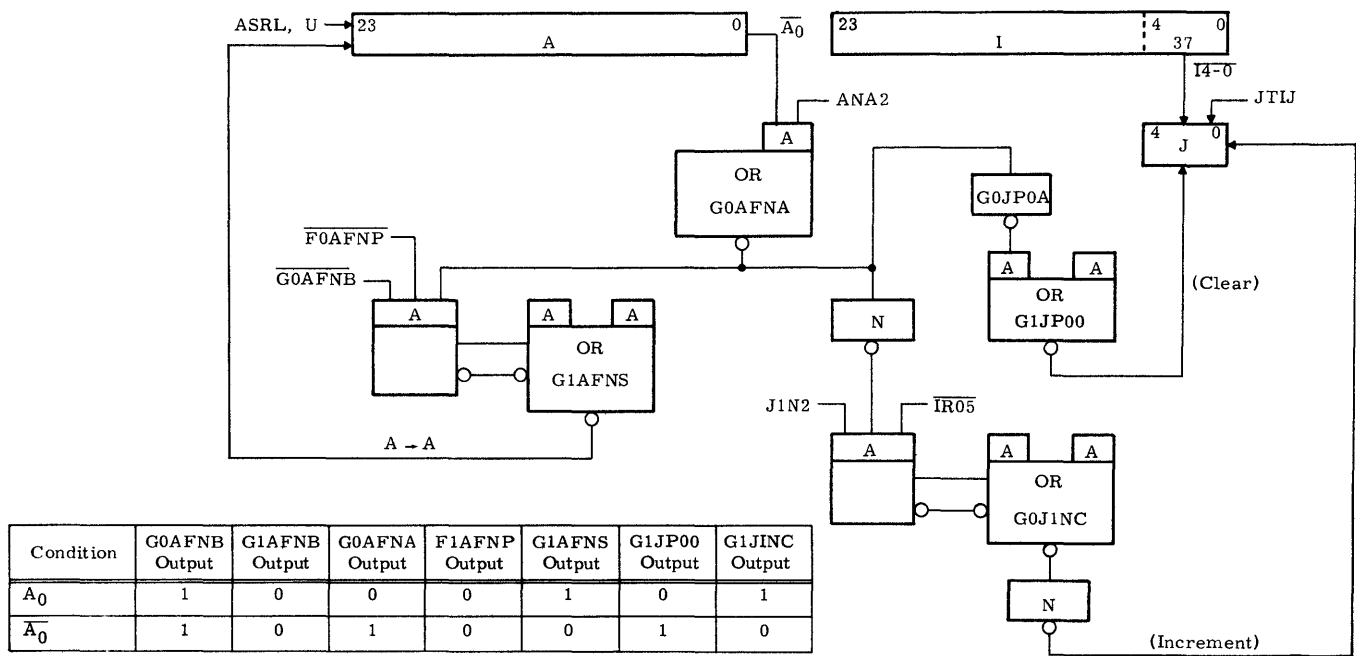


Fig. GN1.7. CMO Block Diagram

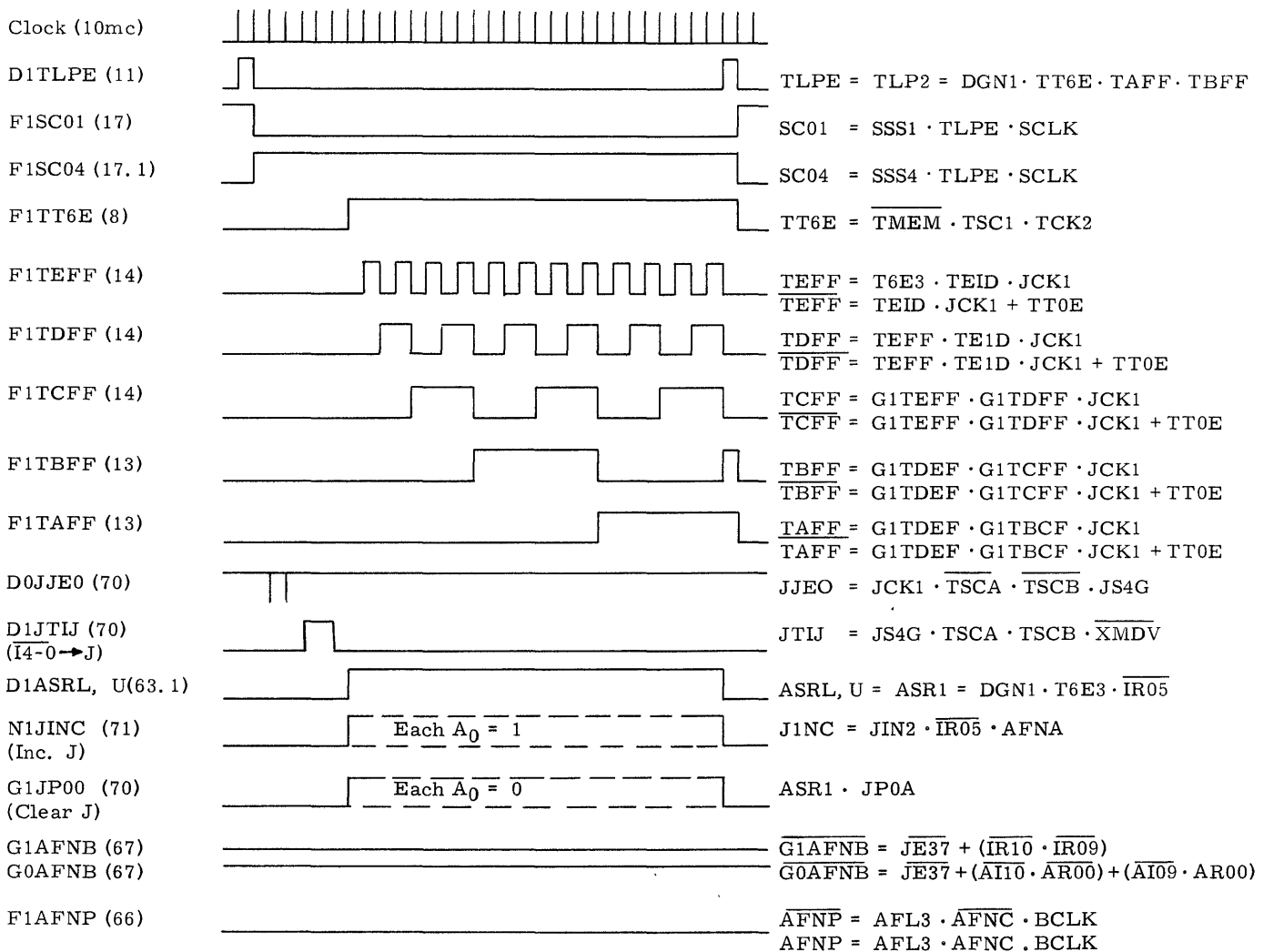


Fig. GN1.8. CMO Timing Diagram

## CMZ-COUNT MOST SIGNIFICANT ZEROS

GEN 1 23  
0 5 0 7 0 2 3 7  
0

CMZ counts the number of "zero" bits to the left of the leftmost "one" bit in the A Register. The count value is placed in the J Counter. If A equals 00000000<sub>8</sub>, the count in J is 24<sub>10</sub>.

### NOTE

An LXC command must follow a CMZ command before another GEN 1 command is executed. This is required to save the J Counter value.

During State 4, the complement of I<sub>4-0</sub> is gated to the J Counter. Since I<sub>4-0</sub> are all "ones", this clears the J Counter. The A Register is then shifted right circular with the complement of A<sub>0</sub> applied to the Adder. The Carry flip-flop (F1AFNP) is held set applying a continuous "one" to the Adder. In this manner, the Adder output is equal to the true output of A<sub>0</sub>. This Adder output is shifted to A<sub>23</sub> leaving the A Register unchanged after 24 shifts. As the A Register is shifted, each "zero" bit increments the J Counter. Each "one" bit clears the J Counter. In this manner, after 24 shifts, the J Counter will contain the number of most significant "zero" bits contained by the A Register.

Interruptable Following Execution?	No
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C(P) + 1
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	Number of left most "zeros" contained in the A Register
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = JE37 + (AI10 \cdot AR00) + (\overline{AI09} \cdot AR00)$	Disabled
G1AFNB	67	$\overline{AFNB} = JE37 + (\overline{IR10} \cdot \overline{IR09})$	Disabled
G0AFNA	65	$\overline{AR00} \cdot ANA1$	Gates $\overline{A_0}$ to Adder
F1AFNP	66	$AFL2 + (AFL3 \cdot AFNC \cdot BCLK)$	Held set applies a "1" to Adder
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Register right 24 places
N1J1NC	71	$JIN2 \cdot AFNA \cdot \overline{IR05}$	Each "zero" increments the J Counter
G1JP00	70	$ASR1 \cdot JPOA$	Each "one" clears the J Counter

## CPL-COMPLEMENT A

GEN 1 23  
0 5 0 1 0 0 0 0  
0

CPL inverts each bit in the A Register; that is, each "one" is replaced by a "zero" and each "zero" is replaced by a "one".

During State 4, the A Register is shifted right 24 places with the complement of A<sub>0</sub> shifted through the Serial Full Adder and back to A<sub>23</sub>. The B input of the Adder is inhibited. Since no summation is performed, no carries result. Therefore, at the end of 24 shifts, the ones complement of the original contents of A is contained in the A Register.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	C ( $\overline{A_{23-0}}$ )
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C(P+1)
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	27 <sub>8</sub>
Memory Z	

### COMMAND CHARACTERISTICS



Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{JE37} + (AI10 \cdot \overline{AR00}) + (\overline{AI09} \cdot \overline{AR00})$	Disabled
G1AFNB	67	$\overline{AFNB} = \overline{JE37} + (\overline{IR10} \cdot \overline{IR09})$	Disabled
G0AFNA	65	$\overline{AR00} \cdot \overline{ANA1}$	Gates $\overline{A_0}$ to Adder
F1AFNP	66	$\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$ $\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$	Remains cleared because no summation is performed
DIASRL, U	63.1	$\overline{ARS1} = \overline{DGN1} \cdot \overline{T6E3} \cdot \overline{IR05}$	Shifts A Reg. right 24 places
NIJ1NC	71	$\overline{JIN4} \cdot \overline{JIN2}$	Allows incrementation of J Counter

### IBK-ISOLATE BIT K

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		010		0	K		

IBK leaves bit Z of the A Register unchanged and clears all other bits in A.

During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. The A Register is then shifted right circular. At each shift of the A Register, the J Counter is incremented. Until the J Counter is incremented to 37<sub>8</sub>, the serial adder inputs are disabled, and "zeros" are shifted to A<sub>23</sub>. When J is equal to 37<sub>8</sub>, the bit specified by Z is shifted from A<sub>0</sub> and applied to the B inputs of the serial adder. The output of the Serial Adder is then equal to A<sub>0</sub> and is shifted to A<sub>23</sub>. Therefore, this bit is unchanged. The inputs of the Serial Adder are enabled only when the J Counter is equal to 37<sub>8</sub>. The J Counter will continue to be incremented at each shift of A until all 24 shifts have occurred. Therefore, following State 4, the A Register will contain all "zeros" except the bit specified by Z which will be unchanged.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	$2^3_0 \dots C(A_Z) \dots 0^0$
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C(P) + 1
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	27 <sub>8</sub> - Z
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AGN1} \cdot \overline{IR09} \cdot \overline{JE37} \cdot \overline{AR00}$	Enables A <sub>0</sub> to Adder when J = 37 <sub>8</sub>
G1AFNB	67	$\overline{AI09} \cdot \overline{AR00} \cdot \overline{DGN1} \cdot \overline{JE37}$	
G0AFNA	65	$\overline{AFNA} = \overline{ANA3} \cdot \overline{DGN3} \cdot \overline{ANA1} \cdot \overline{ANA2}$	Disabled
F1AFNP	66	$\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$ $\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$	Remains cleared because no summation is performed
DIASRL, U	63.1	$\overline{ASR1} = \overline{DGN1} \cdot \overline{T6E3} \cdot \overline{IR05}$	Shifts A Reg. right 24 places
NIJ1NC	71	$\overline{JIN2} \cdot \overline{JIN4}$	Enables incrementation of J Counter

## LBM-LOAD BIT MASK

GEN 1	23	18	17	15	14	6	5	4	0
	05		X		630		0		K

LBM places "zero" in bit Z of the A Register and sets all other bits in A to "one".

During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The A Register is then shifted right circular with the J Counter incremented at each shift. The Carry flip-flop (F1AFNP) is held set throughout the shift of A. All other Adder inputs are disabled until  $J = 37_8$ . Therefore, the Carry flip-flop forces a "one" to the Adder and the "one" result is shifted to  $A_{23}$ . When J is equal to  $37_8$ , a "one" is also applied to the B input. This provides a "zero" Adder output which is shifted to  $A_{23}$ . Therefore, when J is equal to  $37_8$ , a "zero" is set in that position of the A Register. This corresponds to bit position Z. Since the J Counter is incremented at each shift of the A Register a "zero" adder output will only occur when bit position Z is shifted from  $A_0$ . At all other shifts, a "one" Adder output will be generated from the forced carry.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$2^3_1 \dots Z_0 \dots 1^0$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	$27_8 - Z$
Memory Z	

COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$AGN1 \cdot JE37 (IR09 \cdot AR00 + IR10 \cdot \overline{AR00})$	Applies a "1" to Adder when $J = 37_8$
G1AFNB	67	$AI10 \cdot AI09 \cdot DGN1 \cdot JE37$	
G0AFNA	65	$\overline{ARNA} = \overline{ANA3} \cdot \overline{DGN3} \cdot \overline{ANA1} \cdot \overline{ANA2}$	Disabled
F1AFNP	66	$AFNP = AFL2$	Held set, applies "1" to Adder
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN4 \cdot JIN2$	Enables incrementation of J Counter

## LDO-LOAD ONE INTO BIT K

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		030		0		K	

LDO places a "one" in bit Z of the A Register. All other bits of A are cleared (zero).

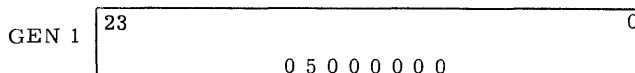
During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The A Register is then shifted right circular with the J Counter being incremented at each shift. All inputs to the Serial Adder are disabled, providing a "zero" to  $A_{23}$  at each shift except when J is equal to  $37_8$ . When J is equal to  $37_8$ , a "one" is applied to the B input of the Serial Adder providing a "one" to  $A_{23}$ . The J Counter is equal to  $37_8$  when the A Register bit specified by Z is shifted from  $A_0$ . Since the J Counter is incremented at each shift of A, only bit Z will be a "one". All other bits of A will be "zero".

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$2^3_0 \dots Z_1 \dots 0^0$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	$27_8 - Z$
Memory Z	

COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$AGN1 \cdot JE37 \cdot (\overline{IR09} \cdot \overline{AR00} + \overline{IR10} \cdot \overline{AR00})$	Applies a "1" to Adder when $J = 37_8$
G1AFNB	67	$AI10 \cdot AI09 \cdot DGN1 \cdot JE37$	
G0AFNA	65	$\overline{AFNA} = \overline{ANA3} \cdot \overline{DGN3} \cdot \overline{ANA1} \cdot \overline{ANA2}$	Disabled
F1AFNP	66	$AFNP = AFL3 \cdot AFNC \cdot BCLK$ $\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot BCLK$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN4 \cdot JIN2$	Enables incrementation of J Counter

### LDZ-LOAD ZEROS INTO A



LDZ replaces the contents of the A Register with "zeros".

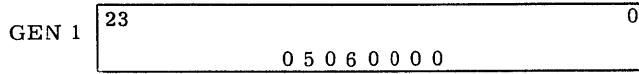
During State 4, the A Register is shifted right 24 places, however, the Serial Adder inputs are disabled providing a "zero" output to  $A_{23}$ . Therefore, after 24 shifts of A have occurred, the entire A Register is cleared.

Interruptable Following Execution?		Yes
CHANGES FOLLOWING EXECUTION		
$A_{23-0}$	23	"Zeros" 0
$Q_{23-0}$		
$P_{14-0}$		$C(P) + 1$
F1WPMT		
F1UOFL		
F1ETST		
$J_{4-0}$		27 <sub>8</sub>
Memory Z		

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR09} \cdot \overline{IR10}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI09} \cdot \overline{AR00}) + (\overline{AI10} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$\overline{AFNA} = \overline{ANA3} \cdot \overline{DGN3} \cdot \overline{ANA1} \cdot \overline{ANA2}$	Disabled
F1AFNP	66	$AFNP = AFL3 \cdot AFNC \cdot BCLK$ $\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot BCLK$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN4 \cdot JIN2$	Enables incrementation of J Counter

## LMO-LOAD MINUS ONE INTO A



LMO places a "one" in each bit of the A Register (equals minus 1).

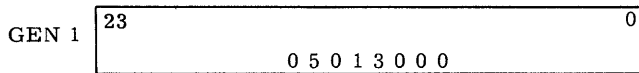
During State 4, the A Register is shifted 24 places to the right. During the shifts, the Carry flip-flop (F1AFNP) is held set. All other Serial Adder inputs are inhibited. In this manner, a "one" is applied to A<sub>23</sub> at each shift. Therefore, at the end of 24 shifts, the A Register will contain all "ones" which is equal to a minus one in two's complement integer representation.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	22 "Ones" 0
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C(P) + 1
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	27 <sub>8</sub>
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR09} \cdot \overline{IR10}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI09} \cdot \overline{AR00}) + (\overline{AI10} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$\overline{AFNA} = \overline{ANA3} \cdot \overline{DGN3} \cdot \overline{ANA1} \cdot \overline{ANA2}$	Disabled
F1AFNP	66	AFL2	Held set. Applies a "1" to Adder
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	JIN4 · JIN2	Enables incrementation of J Counter

## NEG-NEGATE



NEG replaces the contents of the A Register with its 2's complement value.

A timing and block diagram of the NEG command is contained in Fig. GN1.9, 10.

During State 4, the complement of I<sub>4-0</sub> is gated to the J Counter. Since I<sub>4-0</sub> are all "ones", this sets the J Counter equal to 37<sub>8</sub>. The A Register is then shifted right circular with the complement of A<sub>0</sub> applied to the A input of the Serial Full Adder. Because the J Counter is equal to 37<sub>8</sub> at the first shift, a "one" is also applied to the B input of the Adder. The result of the summation of the complement of A<sub>0</sub> and one is gated to A<sub>23</sub>. Any carry resulting from the summation is propagated by the Carry flip-flop (F1AFNP). After the first shift of A, the J Counter is no longer equal to 37<sub>8</sub> and the B input of the Adder is inhibited. Therefore, after 24 shifts of the A Register, the A Register contains the 2's complement of the original contents of A.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	- C(A)
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C(P) + 1
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	27 <sub>8</sub>
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$JE37 \cdot AGN1 \cdot (AR00 \cdot IR09 + \overline{AR00} \cdot IR10)$	Applies "1" to Adder (B input) at first shift because $J = 37_8$
G1AFNB	67	$AI10 \cdot AI09 \cdot DGN1 \cdot JE37$	
G0AFNA	65	$\overline{AR00} \cdot ANA1$	Gates $\overline{A_0}$ to Adder (A input)
F1AFNP	66	$\overline{AFNP} = AFL3 \cdot \overline{AFNC} \cdot BCLK$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Allows normal carry
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN4 \cdot JIN2$	Enables incrementation of J Counter

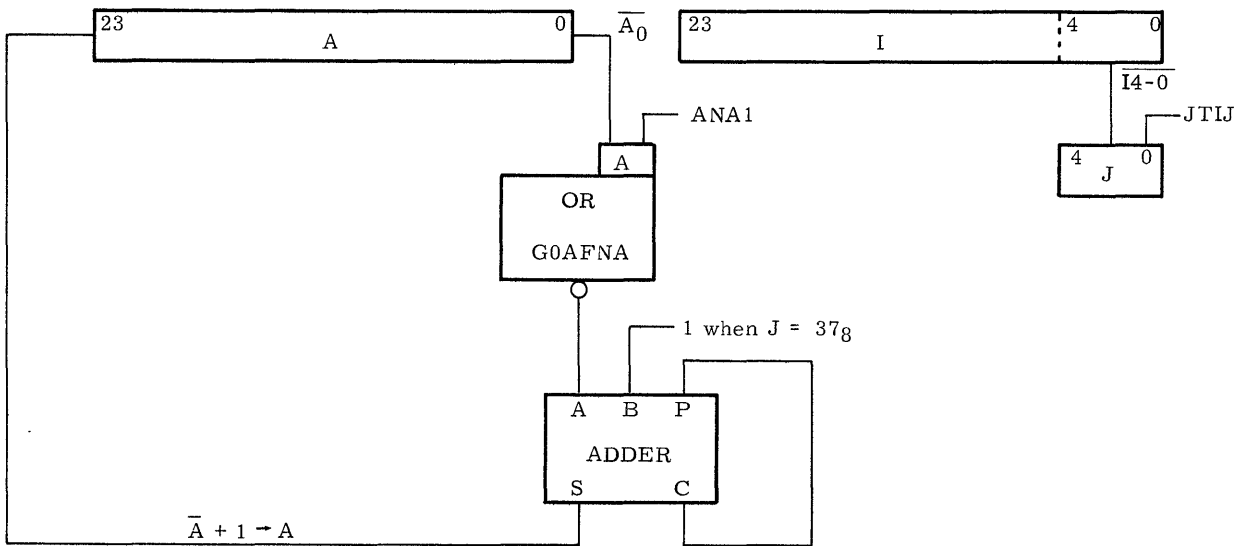


Fig. GN1.9. NEG Block Diagram

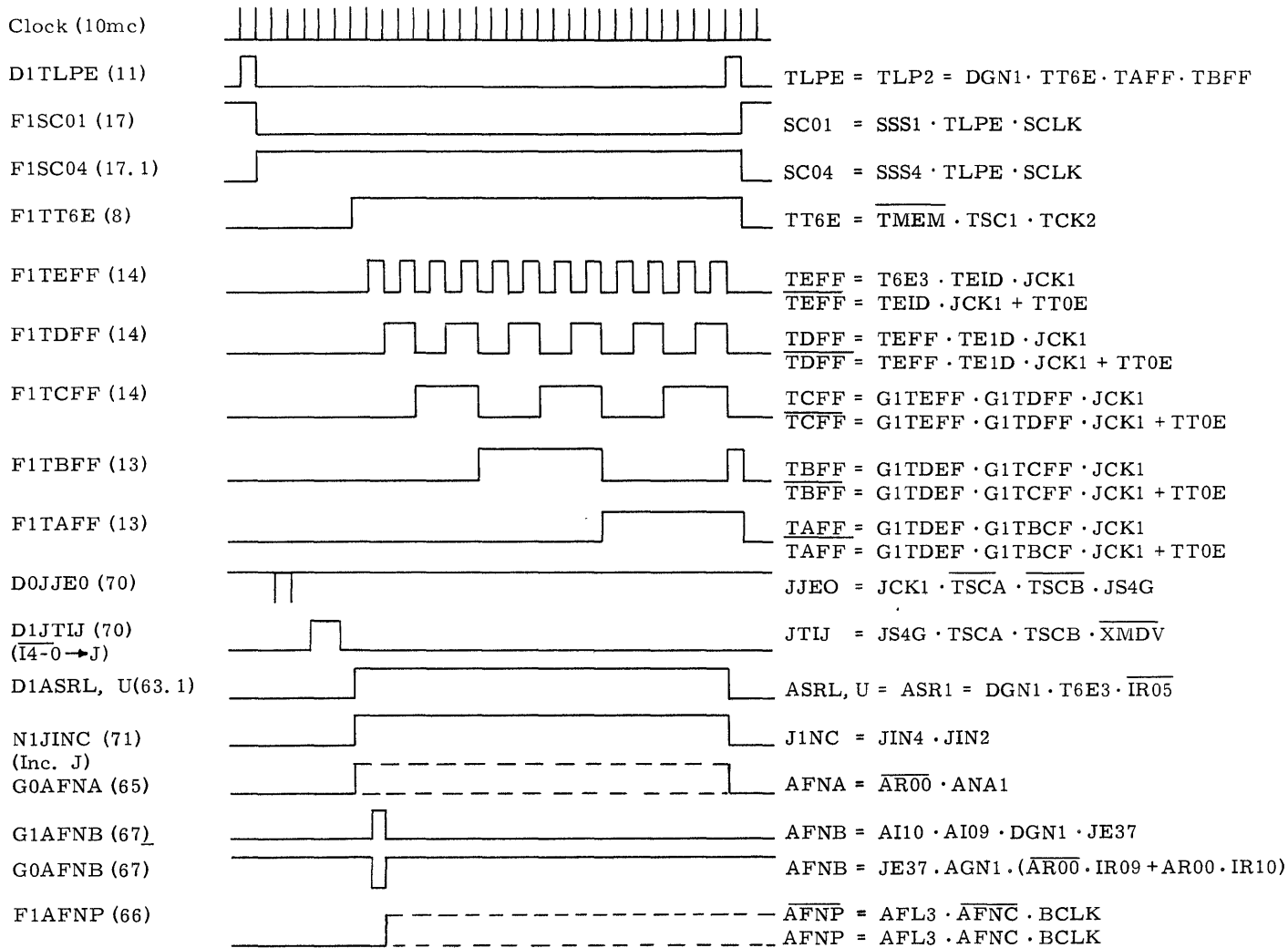


Fig. GN1.10. NEG Timing Diagram

### RBK-RESET BIT K

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		450	0		K		

RBK places a "zero" into bit Z of the A Register. All other bits in A remain unchanged.

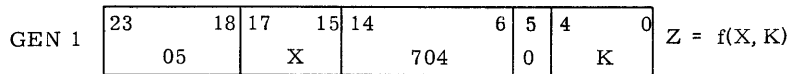
During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. The A Register is then shifted right circular with  $A_0$  applied to the A input of the Serial Full Adder. At each shift of A, the J Counter is incremented. When J is equal to  $37_8$ , the bit specified by Z is shifted from  $A_0$  and  $A_0$  is also applied to the B input of the Adder. When  $A_0$  is applied to both the A and B inputs of the Adder, the sum output will always be "zero". This "zero" is gated to  $A_{23}$  at the shift. Since J is equal to  $37_8$  only when bit Z is shifted from  $A_0$ , this bit will always be a "zero". The remaining bits of the A Register will be unchanged because  $A_0$  is only applied to the Adder A input at the remaining shifts.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$23 \quad Z_0 \quad 0$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	$27_8 - Z$
Memory Z	

COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$AGN1 \cdot IR09 \cdot JE37 \cdot AR00$	Enables $A_0$ to B input of Adder when $J = 37_8$
G1AFNB	67	$AR00 \cdot AI09 \cdot DGN1 \cdot JE37$	
G0AFNA	65	$AR00 \cdot ANA2$	Enables $A_0 \rightarrow A$ input of Adder
F1AFNP	66	$\overline{AFNP} = \overline{AFL3} + \overline{AFL2}$	Held clear
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN4 \cdot JIN2$	Enables incrementation of J Counter

### REV-RESET TEST FLIP-FLOP IF BIT K IS EVEN



REV clears the Test flip-flop (F1ETST) if bit Z in the A Register is a "zero". If bit Z is a "one", the status of the Test flip-flop is unchanged.

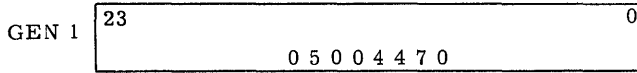
During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The A Register is then shifted right circular with the complement of  $A_0$  applied to the A input of the Adder. The Carry flip-flop (F1AFNP) is held set throughout the shift of A applying a "one" to the B input of the Adder. The summation of 1 plus the complement of  $A_0$  provides a Sum output of the Adder equal to the true  $A_0$ . This output is gated to  $A_{23}$ , leaving the A Register unchanged. The J Counter is incremented at each shift of A. When the J Counter is equal to  $37_8$ , bit Z is shifted from the A Register. If this bit is a "zero", the Test flip-flop (F1ETST) is cleared. If bit Z is a "one", the status of the Test flip-flop is unchanged.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Reset if $A_Z = 0$
$J_{4-0}$	$27_8 - Z$
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$\overline{AR00} \cdot ANA1$	$\overline{A_0}$ to Adder A input
F1AFNP	66	$IR13 + IR14$	Held set
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$\overline{ETST} = \overline{IR06} \cdot \overline{ESTF} \cdot \overline{ECLK}$	Reset Test flip-flop if $A_Z = 0$

## RNZ-RESET TEST FLIP-FLOP IF A IS NON-ZERO



RNZ clears the Test flip-flop (F1ETST) if any bit in the A Register is a "one". If all bits in A are "zero", the status of the Test flip-flop is unchanged. The original contents of A are unchanged by the RNZ command.

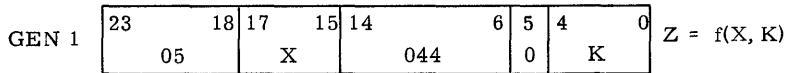
During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. Since  $I_{2-0}$  are "zeros" from the RNZ command, this presets the J Counter equal to 7. The A Register is then shifted right circular with  $A_0$  applied to the A input of the Serial Adder. All other Adder inputs are disabled. Therefore, the Sum output of the Adder is equal to  $A_0$ . This Sum output is applied to  $A_{23}$  leaving the A Register unchanged. Any "one" shifted from  $A_0$  will cause the Test flip-flop (F1ETST) to be cleared. The J Counter is incremented at each shift of the A Register. Since the J Counter was preset to  $7_8$ , after 24 shifts of A the J Counter will equal  $37_8$ . When J equals  $37_8$  further shifting of A is inhibited. Therefore, the A Register is shifted 24 places and only 24 places.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Reset if $C(A) \neq 0$
$J_{4-0}$	$37_8$
Memory Z	

COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$AR00 \cdot \overline{ANA2}$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$ $AFNP = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$	Remains clear because no summation is performed
D1ASRL, U	63.1	$ASR1 = \overline{DGN1} \cdot \overline{T6E3} \cdot \overline{JE37}$	Shifts A Reg. right until $J = 37_8$
N1J1NC	71	$JIN2 \cdot JIN1 \cdot \overline{JE37}$	Enables incrementation of J Counter
F1ETST	89	$\overline{ETST} = \overline{ESTF} \cdot \overline{IR06} \cdot \overline{ECLK}$ $ESTF = \overline{IR05} \cdot \overline{EG1T} \cdot \overline{AFNA} \cdot \overline{ASR1}$	Any "1" from $A_0$ resets the Test flip-flop

## ROD-RESET TEST FLIP-FLOP IF K IS ODD



ROD clears the Test flip-flop (F1ETST) if bit Z in the A Register is a "one". If bit Z is a "zero", the status of the Test flip-flop is unchanged. The original contents of A are unchanged by the ROD command.

During State 4, the complement of  $I_{4-0}$  ( $Z$ ) is gated to the J Counter. The A Register is then shifted right circular with  $A_0$  applied to the A input of the Serial Full Adder. All other inputs of the Adder are disabled. Therefore, the Sum output of the Adder is equal to  $A_0$ . This Sum output is applied to  $A_{23}$ , leaving the A Register unchanged. The J Counter is incremented at each shift of the A Register. When J is equal to  $37_8$ , bit Z is shifted from  $A_0$ . If bit Z is a "one", the Test flip-flop (F1ETST) is cleared. If bit Z is a "zero", the status of the Test flip-flop is unchanged.

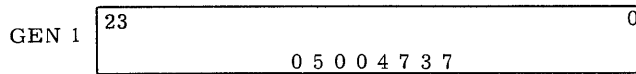
Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Reset if $A_Z = 1$
$J_{4-0}$	$27_8 - Z$
Memory Z	

COMMAND CHARACTERISTICS



Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot AR00)$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = AFL3 \cdot \overline{AFNC} \cdot BCLK$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts the A Reg. right 24 places
N1J1NC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$\overline{ETST} = \overline{IR06} \cdot ESTF \cdot ECLK$ $ESTF = JE37 \cdot EG1T \cdot AFNA \cdot ASR1$	Resets Test flip-flop if $A_Z = "1"$

### RST-RESET THE TEST FLIP-FLOP



RST unconditionally clears the Test flip-flop (F1ETST). The contents of A are unchanged by RST.

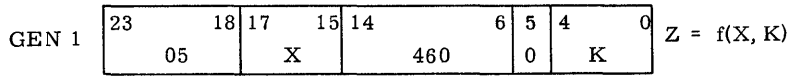
During State 4, the Test flip-flop (F1ETST) is reset. The A Register is circular shifted right with  $A_0$  applied to the A input of the Serial Full Adder. All other inputs to the Adder are disabled. Therefore, the Sum output of the Adder is equal to  $A_0$ . This Sum output is applied to  $A_{23}$ , leaving the A Register unchanged after 24 shifts.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Reset
$J_{4-0}$	$30_8$
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot AR00)$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = AFL3 \cdot \overline{AFNC} \cdot BCLK$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts the A Reg. right 24 places
N1J1NC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$\overline{ETST} = ECTF = IR07 \cdot EI06 \cdot EG1T \cdot \overline{ET4E}$	Reset unconditionally

## SBK-SET BIT K



SBK sets bit Z in the A Register to "one". All other bits in the A Register are unchanged.

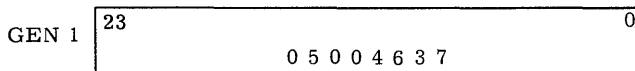
During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. The A Register is then shifted right circular with  $A_0$  applied through the Serial Full Adder to  $A_{23}$ . At each shift of A, the J Counter is incremented. When J is equal to  $37_8$ , bit Z is shifted from  $A_0$ . At this time, the complement of  $A_0$  is applied to the B input of the Adder. The summation of  $\overline{A_0}$  plus  $A_0$  will always provide a "one" output to  $A_{23}$ . Therefore bit Z of the A Register is set to a "one". Since the J Counter is equal to  $37_8$  only when bit Z is shifted from  $A_0$ , all other bits will be unchanged.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$Z_1$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	$27_8 - Z$
Memory Z	

COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$AGN1 \cdot JE37 \cdot IR10 \cdot \overline{AR00}$	Gates $\overline{A_0}$ to Adder B input when $J = 37_8$
G1AFNB	67	$AI10 \cdot \overline{AR00} \cdot JE37$	
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to Adder A input
F1AFNP	66	$\overline{AFNP} = \overline{IR13} + IR0A$	Held clear
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN4 \cdot JIN2$	Enables incrementation of J Counter

## SET-SET TEST FLIP-FLOP



SET unconditionally sets the Test flip-flop (F1ETST). The original contents of the A Register are unchanged.

During State 4, the Test flip-flop (F1ETST) is set. The A Register is circular shifted right with  $A_0$  applied to the A input of the Serial Full Adder. All other inputs to the Adder are disabled. Therefore, the Sum output of the Adder is equal to  $A_0$ . This Sum output is applied to  $A_{23}$ , leaving the A Register unchanged after 24 shifts.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set
$J_{4-0}$	$30_8$
Memory Z	

COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot AR00)$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = AFL3 \cdot \overline{AFNC} \cdot BCLK$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$GOESTF = \overline{IR06} \cdot IR07 \cdot EG1T \cdot ET4E$	Set unconditionally

### SEV-SET TEST FLIP-FLOP IF BIT K IS EVEN

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
		05		X		705	0	K		

SEV sets the Test flip-flop (F1ETST) if bit Z in the A Register is a "zero". If bit Z in A is a "one", the status of the Test flip-flop is unchanged.

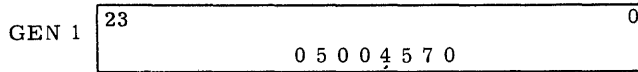
During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The A Register is then shifted right circular with the complement of  $A_0$  applied to the A input of the Adder. The Carry flip-flop (F1AFNP) is held set applying a "one" to the B input of the Adder. The summation of 1 plus the complement of  $A_0$  provides a Sum output of the Adder equal to the true  $A_0$  output. This Sum output is applied to  $A_{23}$ , leaving the A Register unchanged after 24 shifts. The J Counter is incremented at each shift of A. When J is equal to 378, bit Z of the A Register is shifted from  $A_0$ . If this bit is a "zero", the Test flip-flop (F1ETST) is set. If this bit is a "one", the Test flip-flop status is unchanged.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if $A_Z = 0$
$J_{4-0}$	$278 - Z$
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot AR00)$	Disabled
G0AFNA	65	$\overline{AR00} \cdot ANA1$	$\overline{A_0}$ to A input of the Adder
F1AFNP	66	$IR13 \cdot IR0A$	Held set
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$ETST = ESTF \cdot EI06 \cdot ECLK$ $ESTF = JE37 \cdot EG1T \cdot AFNA \cdot ASR1$	Set Test flip-flop if $A_Z = 0$

## SNZ-SET TEST FLIP-FLOP IF A IS NON-ZERO



SNZ sets the Test flip-flop (F1ETST) if any bit in the A Register is a "one". If all bits in A are "zero", the status of the Test flip-flop is unchanged.

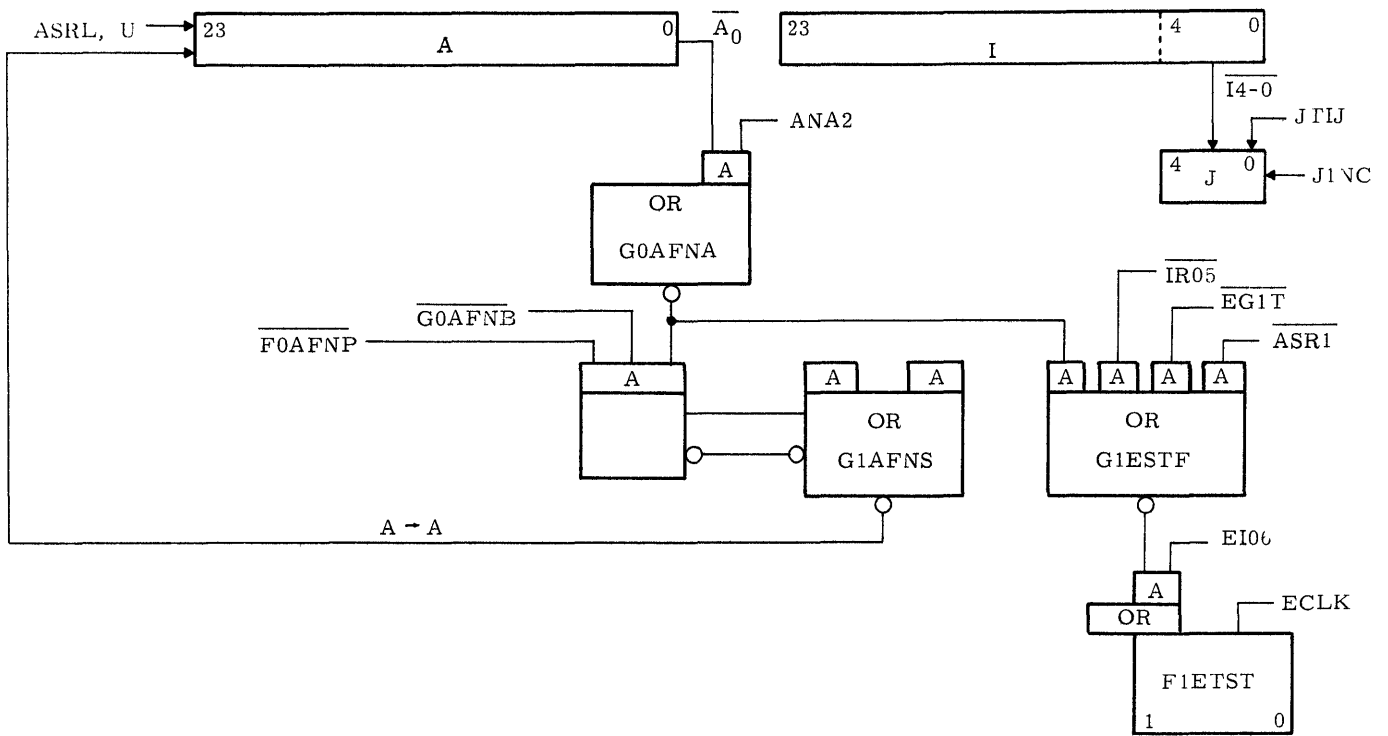
A timing and block diagram of the SNZ command is contained in Fig. GN1. 11, 12.

During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. Since bits 2 through 0 of the SNZ command are "zeros", this pre-sets the J Counter equal to 7. The A Register is then shifted right with  $A_0$  applied to the A input of the Serial Full Adder. All other Adder inputs are disabled. Therefore, the Sum output of the Adder is equal to  $A_0$ . This Sum output is applied to  $A_{23}$ , leaving the A Register unchanged after 24 shifts. If at any shift of A,  $A_0$  is a "one", the Test flip-flop (F1ETST) is set. If all bits of A are "zero", the status of the Test flip-flop is unchanged. The J Counter is incremented at each shift of the A Register and when equal to  $37_8$  inhibits further shifting of A. Since the J Counter was preset to 7, when J equals  $37_8$ , 24 shifts of A have occurred.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if $C(A) \neq 0$
$J_{4-0}$	$37_8$
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$ $\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{JE37}$	Shifts A Reg. right until $J = 37_8$
N1J1NC	71	$JIN2 \cdot JIN1 \cdot \overline{JE37}$	Enables incrementation of J Counter until $J = 37_8$
F1ETST	89	$\overline{ETST} = G1ESTF \cdot \overline{EI06} \cdot \overline{ECLK}$ $\overline{ETST} = \overline{IR05} \cdot \overline{EG1T} \cdot \overline{AFNA} \cdot \overline{ASR1}$	Any "1" from $A_0$ sets the Test flip-flop



Condition	G0AFNB Output	G1AFNB Output	G0AFNA Output	F1AFNP Output	G1AFNS Output	F1ETST
$A_0$	1	0	0	0	1	1
$\overline{A_0}$	1	0	1	0	0	-

Fig. GN1. 11. SNZ Block Diagram

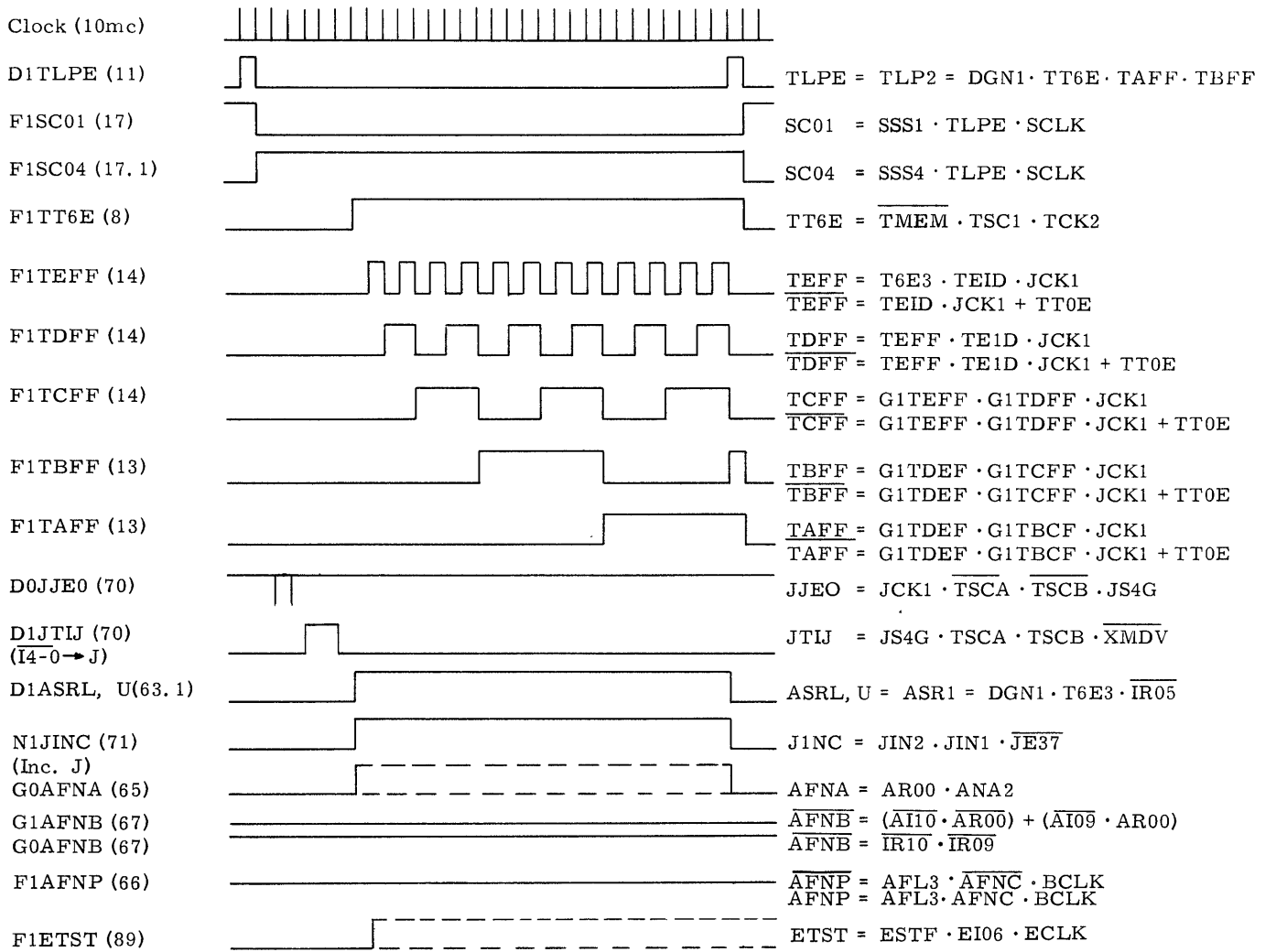


Fig. GN1.12. SNZ Timing Diagram

### SOD-SET TEST FLIP-FLOP IF BIT K IS ODD

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		045		0		K	

SOD sets the Test flip-flop (F1ETST) if bit Z in the A Register is a "one". If bit Z in A is a "zero", the status of the Test flip-flop is unchanged.

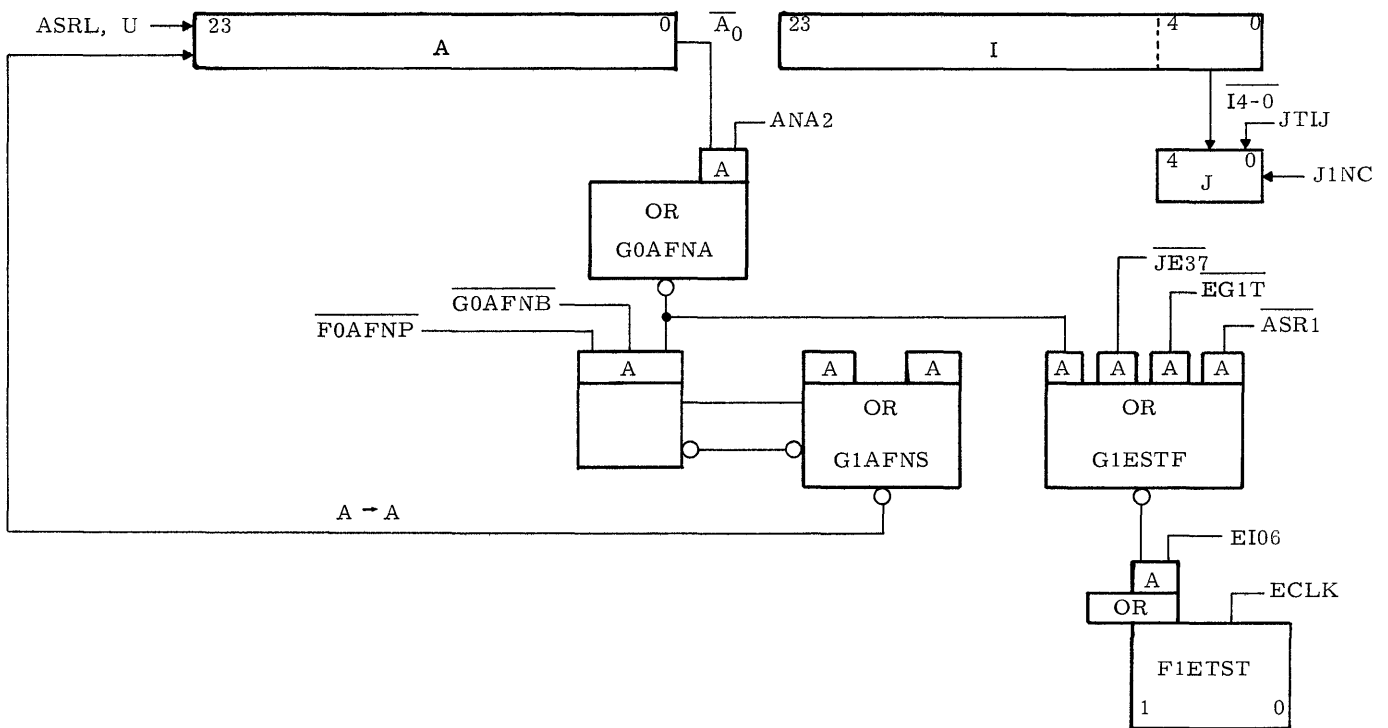
A timing and block diagram of the SOD command are contained in Fig. GN1.13, 14.

During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The A Register is then shifted right circular with  $A_0$  applied to the A input of the Serial Full Adder. All other inputs of the Adder are disabled. Therefore, the Sum output of the Adder is equal to  $A_0$ . This Sum output is applied to  $A_{23}$ , leaving the A Register unchanged after 24 shifts. The J Counter is incremented at each shift of A. When J is equal to  $37_8$ , bit Z is shifted from  $A_0$  and if a "one", the Test flip-flop is set.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if $C(A_Z) = 1$
$J_{4-0}$	$27_8 - Z$
Memory Z	

COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$ $\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = \overline{DGN1} \cdot \overline{T6E3} \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1JINC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$\overline{ETST} = \overline{G1ESTF} \cdot \overline{EI06} \cdot \overline{ECLK}$ $\overline{G1ESTF} = \overline{JE37} \cdot \overline{EG1T} \cdot \overline{AFNA} \cdot \overline{ASR1}$	Set Test flip-flop if $A_Z = 1$



Condition	G0AFNB Output	G1AFNB Output	G0AFNA Output	F1AFNP Output	G1AFNS Output
$A_0$	1	0	0	0	1
$\overline{A_0}$	1	0	1	0	0

Fig. GN1.13. SOD Block Diagram

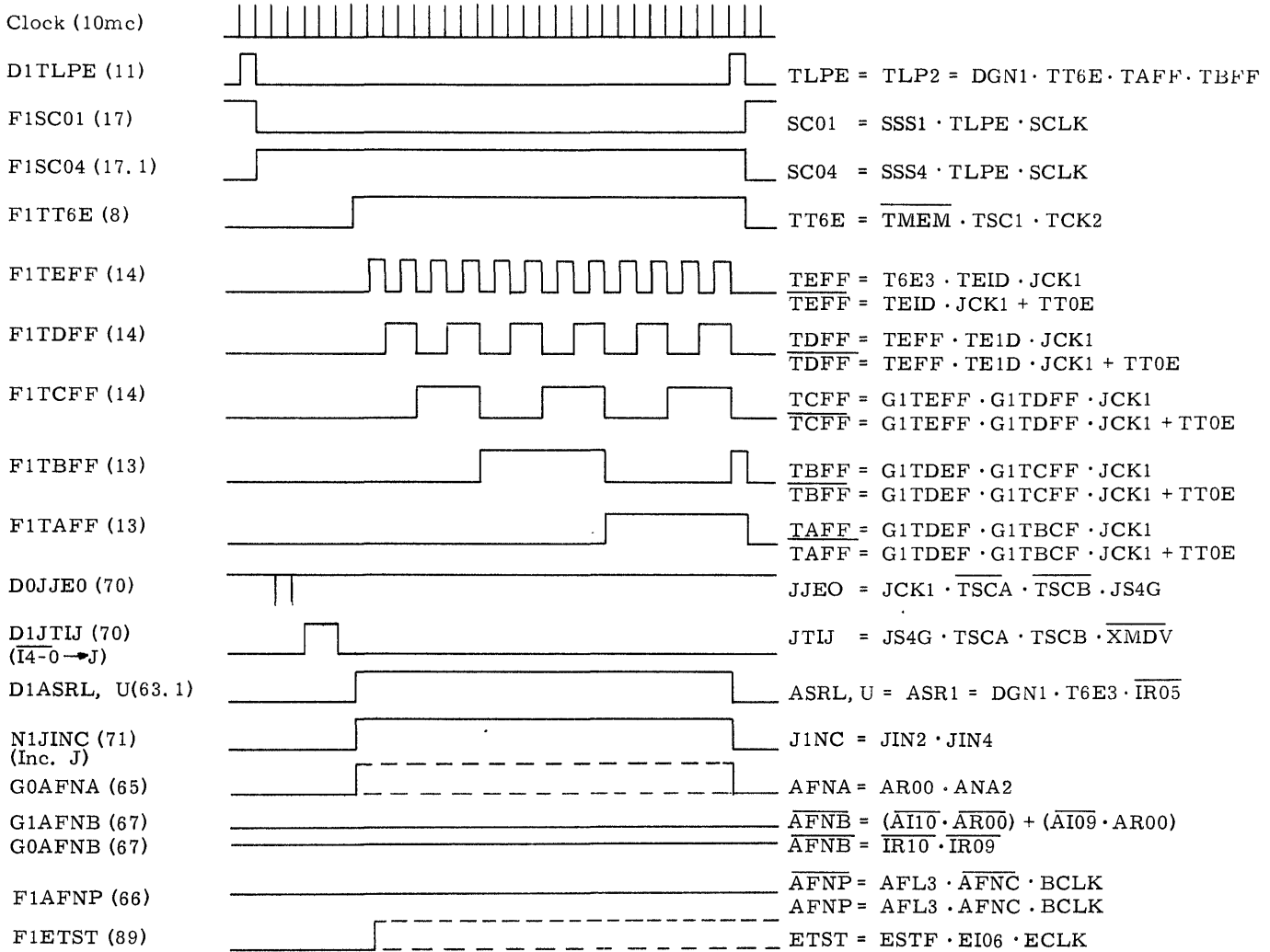


Fig. GN1.14. SOD Timing Diagram

### SRA-SHIFT A RIGHT ARITHMETIC

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		140		1		K	

SRA shifts the contents of the A Register Z places to the right. The sign bit (23) of A is unchanged. Bits shifted out of A<sub>0</sub> are lost. Bits shifted into A<sub>22</sub> are the same as the sign bit. The maximum number of shifts is 24 places.

A timing and block diagram of the SRA command is contained in Fig. GN1.15, 16.

During State 4, the complement of I<sub>4-0</sub> is gated to the J Counter. The A Register is then shifted right with A<sub>23</sub> shifted to A<sub>22</sub> and with A<sub>23</sub> also shifted through the Serial Full Adder back to A<sub>23</sub>, leaving A<sub>23</sub> unchanged. At each shift of the A Register, the J Counter is incremented. When J is equal to 37<sub>8</sub>, the number of shifts specified by Z have occurred and further shifting is inhibited. Therefore, the A Register is shifted right the number of places specified by Z.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	See Text
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C(P) + 1
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	37 <sub>8</sub>
Memory Z	

COMMAND CHARACTERISTICS



Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$ANA3 \cdot AR23$	Gates A <sub>23</sub> to A input of Adder
F1AFNP	66	$\overline{AFNP} = AFL3 \cdot \overline{AFNC} \cdot BCLK$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{JE37}$	Shifts A Reg. right until J = 37 <sub>8</sub>
N1J1NC	71	$\overline{JE37} \cdot JIN1 \cdot JIN2$	Increments J Counter until J = 37 <sub>8</sub>

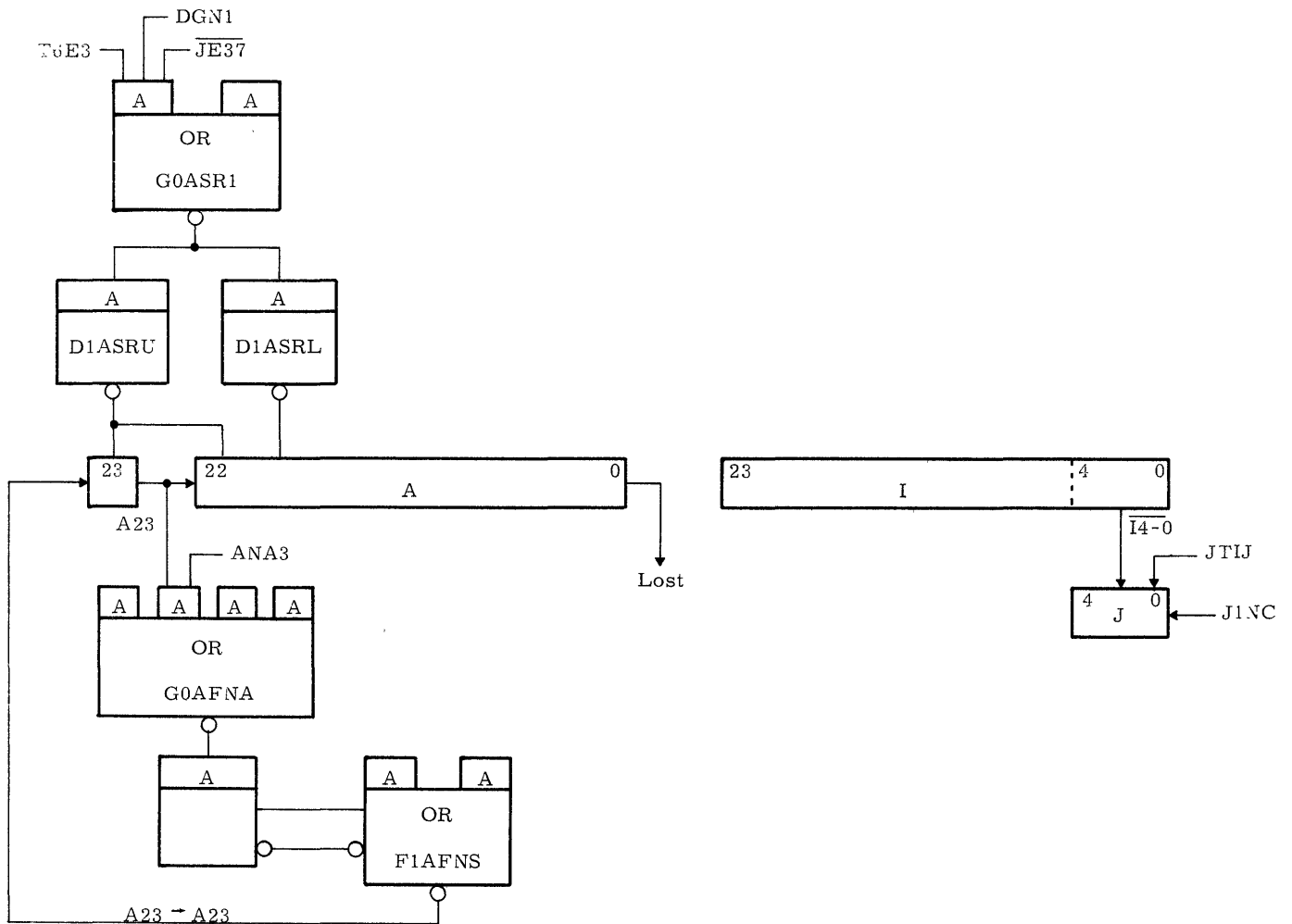


Fig. GN1.15. SRA Block Diagram

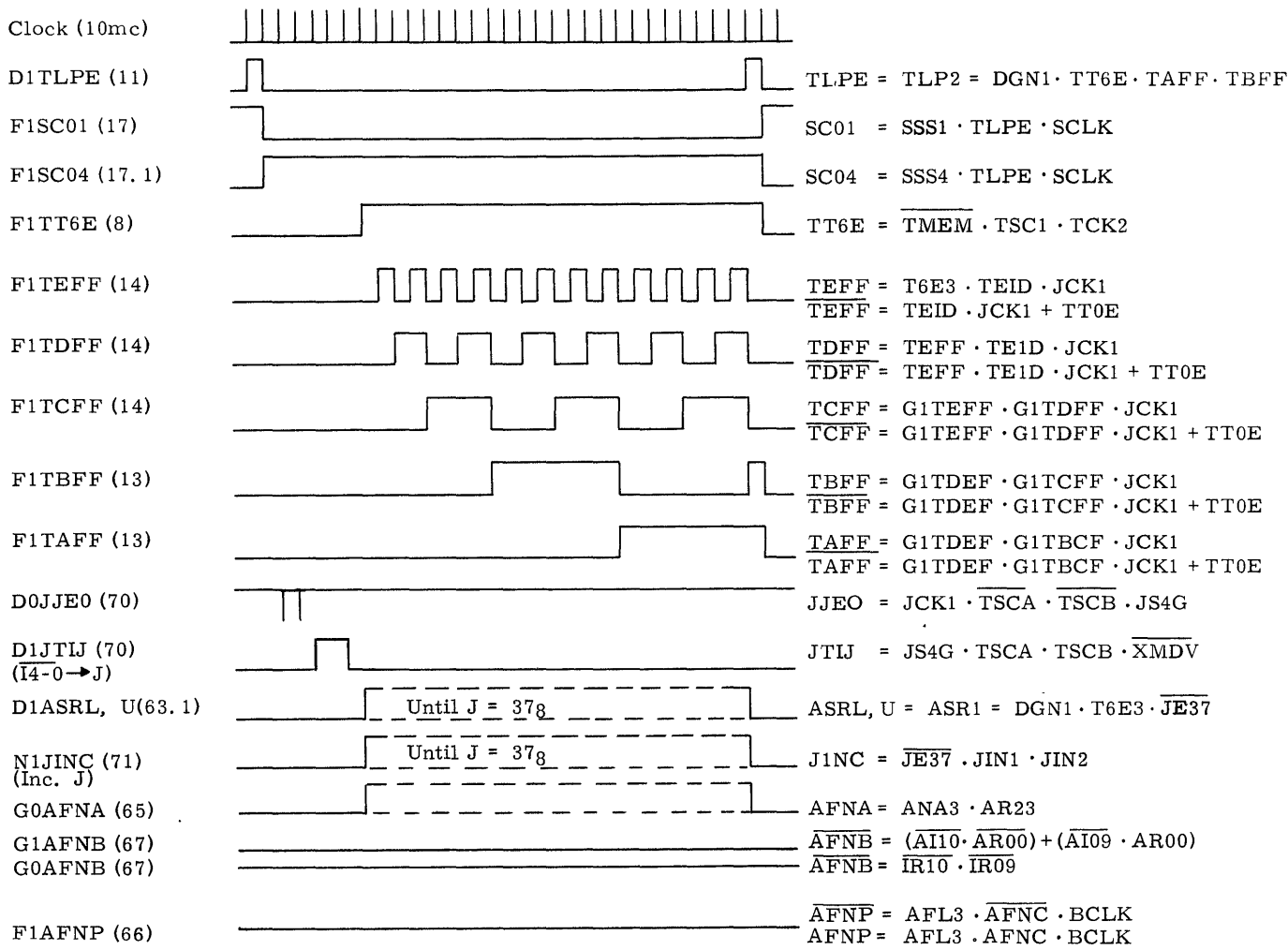
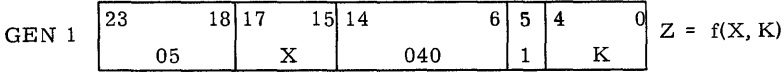
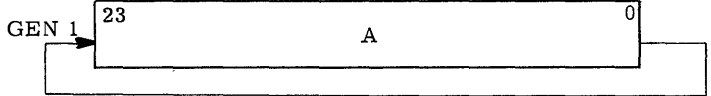


Fig. GN1.16. SRA Timing Diagram

**SRC-SHIFT RIGHT CIRCULAR**



SRC shifts the contents of the A Register Z places to the right in a circular fashion; that is, the bit shifted from A<sub>0</sub> is inserted in A<sub>23</sub>, replacing the bit shifted out of A<sub>23</sub>. The maximum number of shifts is 24.



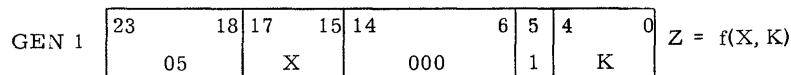
During State 4, the complement of I<sub>4-0</sub> (Z) is gated to the J Counter. The A Register is then shifted right circular with A<sub>0</sub> applied through the A input of the Serial Adder back to A<sub>23</sub>. At each shift of the A Register, the J Counter is incremented. When J is equal to 37<sub>8</sub>, the number of shifts specified by Z have occurred and further shifting of A is inhibited.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	See Text
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C(P) + 1
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	37 <sub>8</sub>
Memory Z	

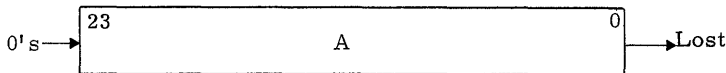
COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot AR00)$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{JE37}$	Shifts A Reg. right until $J = 37_8$
N1J1NC	71	$\overline{JE37} \cdot JIN1 \cdot JIN2$	Increments J Counter until $J = 37_8$

### SRL-SHIFT RIGHT LOGICAL



SRL shifts all twenty-four bits of the A Register to the right Z places. Zeros are shifted in through  $A_{23}$ . Bits shifted out of  $A_0$  are lost. The maximum number of shifts is 24 places.



During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The A Register is then shifted right with the J Counter being incremented at each shift. When J is equal to  $37_8$ , the number of shifts specified by Z have occurred and further shifting of A is inhibited. Bits shifted from  $A_0$  are lost. "Zeros" are shifted into  $A_{23}$  at each shift.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	See Text
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	$37_8$
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot AR00)$	Disabled
G0AFNA	65	$\overline{AFNA} = \overline{ANA1} \cdot \overline{ANA2} \cdot \overline{ANA3}$	Disabled
F1AFNP	66	$\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Remains cleared because no inputs to Adder
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{JE37}$	Shifts A Reg. right until $J = 37_8$
N1J1NC	71	$\overline{JE37} \cdot JIN1 \cdot JIN2$	Enabled J incrementation until $J = 37_8$

## TER-TEST EVEN AND RESET BIT K

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		456		0		K	

TER sets the Test flip-flop (F1ETST) if Bit Z in the A Register is "zero", and places a "zero" back in bit Z. If bit Z is "one" the Test flip-flop is cleared and a "zero" is placed in bit Z.

### NOTE

The Test flip-flop will be set if Z is greater than  $23_{10}$ .

During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The Test flip-flop (F1ETST) is unconditionally set. The A Register is shifted right circular with  $A_0$  applied to the A input of the Serial Full Adder. All other inputs to the Adder are disabled until the J Counter is equal to  $37_8$ . In this manner, the Sum output of the Adder is equal to  $A_0$  until J is equal to  $37_8$  and this Sum output is applied to  $A_{23}$  leaving these bits of A unchanged. The J Counter is incremented at each shift of A. When J is equal to  $37_8$ ,  $A_0$  is also applied to the B input of the Adder. This always results in a Sum output of "zero". This "zero" is gated to  $A_{23}$ , placing a "zero" in position Z of the A Register. Also, when J is equal to  $37_8$ , if the bit shifted from A ( $A_Z$ ) is a "one", the Test flip-flop is cleared. If this bit is a "zero", the Test flip-flop remains in the set state.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$23 \quad A_Z \quad 0$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if $A_Z = 0$ ; Reset if $A_Z = 1$
$J_{4-0}$	$27_8 - Z$
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$AGN1 \cdot IR09 \cdot JE37 \cdot AR00$	Gates $A_0$ to B input of Adder when $J = 37_8$
G1AFNB	67	$AR00 \cdot AR09 \cdot DGN1 \cdot JE37$	
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = (SC01 \cdot BCLK) + \overline{IR13} + IR0A$	Held Clear
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1JINC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$ETST = GOESTF = \overline{IR06} \cdot IR07 \cdot ET4E$ , $\overline{ETST} = \overline{EG1T} \cdot \overline{IR06} \cdot ECLK$	Set Test F/F; Clear Test if $A_Z = 1$

## TES-TEST EVEN AND SET BIT K

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		466		0		K	

TES sets the Test flip-flop (F1ETST) if bit Z in the A Register is a "zero", and replaces bit Z with a "one". If bit Z is a "one", the Test flip-flop is cleared and a "one" is placed back in bit Z.

### NOTE

The Test flip-flop will be set if Z is greater than  $23_{10}$ .

During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The Test flip-flop (F1ETST) is unconditionally set. The A Register is shifted right circular with  $A_0$  applied to the A input of the Serial Full Adder. All other inputs to the Adder are disabled until the J Counter is equal to  $37_8$ . In this manner, the Sum output of the Adder is equal to  $A_0$ , unless J is equal to  $37_8$ , and this Sum output is gated to  $A_{23}$  leaving these bits of A unchanged. The J

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$23 \quad Z_1 \quad 0$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if $A_Z = 0$ ; Reset if $A_Z = 1$
$J_{4-0}$	$27_8 - Z$
Memory Z	

### COMMAND CHARACTERISTICS

Counter is incremented at each shift of A. When J is equal to  $37_8$ , the complement of  $A_0$  is applied to the B input of the Adder. The summation of  $A_0$  and  $\overline{A_0}$  always provides a Sum output of "one". This "one" is applied to  $A_{23}$ , thereby setting bit Z to a "one". Also, when J is equal to  $37_8$ , if the bit shifted from  $A_0$  ( $A_Z$ ) is a "one", the Test flip-flop is cleared. If this bit is a "zero", the Test flip-flop remains in the set state.

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$AGN1 \cdot JE37 \cdot IR10 \cdot \overline{AR00}$	Gates $\overline{A_0}$ to A input of Adder when $J = 37_8$
G1AFNB	67	$AI10 \cdot \overline{AR00} \cdot DGN1 \cdot JE37$	
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = (SC01 \cdot BCLK) + \overline{IR13} + IR0A$	Held clear
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$\overline{ETST} = G0ESTF = IR06 \cdot IR07 \cdot ET4E \cdot EG1T$ $\overline{ETST} = G1ESTF \cdot \overline{IR06} \cdot ECLK$	Set Test F/F; Clear Test F/F if $A_Z = 1$

### TEV-TEST BIT K EVEN

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		707		0	K		

TEV sets the Test flip-flop (F1ETST) if bit Z in the A Register is "zero". If bit Z in the A is "one", the Test flip-flop is cleared.

NOTE

The Test flip-flop is cleared if Z is greater than  $23_{10}$ .

During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The Test flip-flop is unconditionally cleared. The Carry flip-flop (F1AFNP) is held set applying a "one" to the B input of the Serial Adder. The A Register is then shifted right with the complement of  $A_0$  applied to the A input of the Serial Adder. The summation of  $A_0$  and the forced "one" from F1AFNP provides a Sum output of the Adder equal to the true  $A_0$ . This sum output is applied to  $A_{23}$ , leaving the A Register unchanged. At each shift of A, the J Counter is incremented. When J is equal to  $37_8$ , bit Z is shifted from  $A_0$ . If  $A_0$  is a "zero", the Test flip-flop is set. If  $A_0$  is a "one" when J equals  $37_8$ , the Test flip-flop remains cleared.

Interruptable Following Execution?		Yes
CHANGES FOLLOWING EXECUTION		
$A_{23-0}$		
$Q_{23-0}$		
$P_{14-0}$		$C(P) + 1$
F1WPMT		
F1UOFL		
F1ETST		Set if $A_Z = 0$ ; Reset if $A_Z = 1$
$J_{4-0}$		$27_8 - Z$
Memory Z		

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot AR00)$	Disabled
G0AFNA	65	$\overline{AR00} \cdot ANA2$	Gates $\overline{A_0}$ to A input of Adder
F1AFNP	66	$AFL2 + \overline{AFL3}$	Held set
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$\overline{ETST} = EI06 \cdot G1ESTF \cdot ECLK$ $ETST = ECTF = IR07 \cdot EI06 \cdot EG1T \cdot \overline{ET4E}$	Clear Test F/F; Set Test F/F if $A_Z = 0$

### TNM-TEST NOT MINUS ONE

GEN 1 

23	0
0 5 0 7 0 7 7 0	

TNM sets the Test flip-flop (F1ETST) if any bit in the A Register is a "zero". When all bits of the A Register are "ones" (minus 1) the Test flip-flop is cleared.

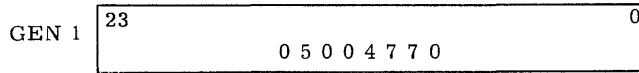
During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. This presets the J Counter equal to 7, since bits 2 through 0 of the TNM command are "zeros". The Test flip-flop (F1ETST) is reset. The Carry flip-flop (F1AFNP) is held set applying a "one" to the B input of the Serial Full Adder. The A Register is then shifted to the right with the complement of  $A_0$  applied to the Serial Full Adder. The Sum output of the Adder is equal to the true  $A_0$ . This Sum output is applied to  $A_{23}$ , leaving A unchanged after 24 shifts. If at any shift of A,  $A_0$  is a "zero", the Test flip-flop is set. If all bits of A are "ones", the Test flip-flop remains in the reset state. At each shift of the A Register, the J Counter is incremented. When the J Counter is equal to  $37_8$ , 24 shifts of the A Register have occurred and further shifting is inhibited.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if any bit in A = 0; Reset if all bits in A = 1
$J_{4-0}$	$37_8$
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot AR00)$	Disabled
G0AFNA	65	$\overline{AR00} \cdot ANA2$	Gates $\overline{A_0}$ to A input of Adder
F1AFNP	66	$AFL2 + \overline{AFL3}$	Held set
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{JE37}$	Shifts A Reg. right until J = 37
N1J1NC	71	$\overline{JE37} \cdot JIN1 \cdot JIN2$	Enables incrementation of J Counter until J = $37_8$
F1ETST	89	$\overline{ETST} = ECTF = EI07 \cdot EI06 \cdot EG1T \cdot \overline{ET4E}$ $ETST = G1ESTF \cdot EI06 \cdot ECLK$	Clear Test F/F; Set Test F/F if any bit of A = 0

## TNZ-TEST A NON-ZERO



TNZ sets the Test flip-flop (F1ETST) if any bit in the A Register is a "one". If all bits in the A Register are "zero", the Test flip-flop is cleared.

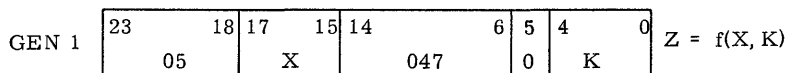
During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. This presets J equal to 7, since bits 2 through 0 of the TNZ command are "zeros". The Test flip-flop (F1ETST) is cleared. The A Register is then shifted right circular with  $A_0$  applied to the A input of the Serial Full Adder. All other inputs to the Adder are disabled providing a Sum output equal to  $A_0$ . This Sum output is applied to  $A_{23}$ , leaving A unchanged after 24 shifts. If at any shift of A,  $A_0$  is equal to a "one", the Test flip-flop is set. If all bits of A are "zero", the Test flip-flop remains in the reset state. The J Counter is incremented at each shift of A. When the J Counter is equal to  $37_8$ , 24 shifts of A have occurred and further shifting of A is inhibited.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if $C(A) \neq 0$ ; Reset if $C(A) = 0$
$J_{4-0}$	$37_8$
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (AI09 \cdot AR00)$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot BCLK$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{JE37}$	Shifts A Reg. right until $J = 37_8$
N1J1NC	71	$\overline{JE37} \cdot JIN1 \cdot JIN2$	Enables J incrementation until $J = 37_8$
F1ETST	89	$\overline{ETST} = ECTF = IR07 \cdot EI06 \cdot EG1T \cdot ET4E$ $ETST = EI06 \cdot G1ESTF \cdot ECLK$	Clear Test F/F; Set Test F/F if any bit of A = "1"

## TOD-TEST BIT K ODD



TOD sets the Test flip-flop (F1ETST) if bit Z in the A Register is a "one". If bit Z in A is "zero", the Test flip-flop is cleared.

NOTE

The Test flip-flop is cleared if Z is greater than  $23_{10}$ .

During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. The Test flip-flop (F1ETST) is reset. The A Register is shifted right circular with  $A_0$  applied to the A input of the Serial Full Adder. All other inputs to the Adder are inhibited. Therefore, the Sum output of the Adder is equal to  $A_0$ . This Sum output is applied to  $A_{23}$ , leaving the A Register unchanged after 24 shifts. At each shift of A, the J Counter is incremented. When J is equal to  $37_8$ , bit Z is shifted from A and if bit Z is a "one", the Test flip-flop is set. If bit Z is a "zero", the Test flip-flop remains in the reset state.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if $A_Z = 1$ ; Reset if $A_Z = 0$
$J_{4-0}$	$27_8 - Z$
Memory Z	

### COMMAND CHARACTERISTICS

If bit Z is a "zero", the Test flip-flop remains in the reset state.

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$ $AFNP = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$	Remains cleared because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$\overline{ETST} = \overline{ECTF} = \overline{IR07} \cdot \overline{EI06} \cdot \overline{EG1T} \cdot \overline{ET4E}$ $ETST = EI06 \cdot G1ESTF \cdot ECLK$	Clear Test F/F; Set Test F/F if $A_Z = 1$

### TOR-TEST ODD AND RESET BIT K

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		457		0		K	

TOR sets the Test flip-flop (F1ETST) if bit Z in the A Register is a "one" and replaces bit Z with a "zero". If bit Z in A is a "zero", the Test flip-flop is cleared and bit Z remains "zero".

#### NOTE

The Test flip-flop is cleared if Z is greater than  $23_{10}$ .

During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The Test flip-flop (F1ETST) is reset. The A Register is shifted right with  $A_0$  applied to the A input of the Serial Full Adder. All other inputs to the Adder are disabled, except when J is equal to  $37_8$ . When J is not equal to  $37_8$ , the Sum output of the Adder is equal to  $A_0$ . This Sum output is applied to  $A_{23}$ , leaving these bits of A unchanged. The J Counter is incremented at each shift of A and when equal to  $37_8$ , bit Z is shifted from  $A_0$ . If bit Z is a "one", the Test flip-flop is set. Also, when J is equal to  $37_8$ ,  $A_0$  is applied to the B input of the Adder. The Sum of  $A_0$  and  $A_0$  will always result in a "zero" Sum output of the Adder. The Sum output is applied to  $A_{23}$ , providing a "zero" in position Z of the A Register after 24 shifts have occurred.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$Z_0$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if $A_Z = 1$ Reset if $A_Z = 0$
$J_{4-0}$	$27_8 - Z$
Memory Z	

#### COMMAND CHARACTERISTICS



Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$AGN1 \cdot IR09 \cdot JE37 \cdot AR00$	Gates $A_0$ to B input of Adder When $J = 37_8$
G1AFNB	67	$AR00 \cdot AI09 \cdot DGN1 \cdot JE37$	
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFL2} + \overline{AFL3} + (AFL1 \cdot ECLK)$	Held reset
DIASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$\overline{ETST} = ECTF = IR07 \cdot EI06 \cdot EG1T \cdot ET4E$ $ETST = G1ESTF \cdot EI06 \cdot ECLK$	Clear Test F/F; Set Test F/F if $A_Z = 1$

### TOS-TEST ODD AND SET BIT K

GEN 1	23	18	17	15	14	6	5	4	0	$Z = f(X, K)$
	05		X		467		0	K		

TOS sets the Test flip-flop (F1ETST) if bit Z in the A Register is a "one" and bit Z remains a "one". If bit Z in A is a "zero", the Test flip-flop is cleared and bit Z is replaced with a "one".

#### NOTE

The Test flip-flop is cleared if Z is greater than  $23_{10}$ .

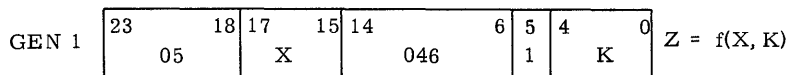
During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The Test flip-flop (F1ETST) is reset. The A Register is shifted right with  $A_0$  applied to the A input of the Serial Full Adder. All other inputs to the Adder are disabled, except when the J Counter is equal to  $37_8$ . Therefore, the Sum output of the Adder is equal to  $A_0$ , unless J is equal to  $37_8$ . This Sum output is applied to  $A_{23}$ , leaving these bits of A unchanged. The J Counter is incremented at each shift of A. When J is equal to  $37_8$ , bit Z is shifted from the A Register. The complement of bit Z is applied to the B input of the Adder. The result of the summation of  $A_0$  and  $\overline{A_0}$  results in a "one" Sum output of the Adder. This "one" is applied to  $A_{23}$ , making bit Z a "one". Also, when J is equal to 37, if  $A_0$  is a "one", the Test flip-flop is set.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$23 \quad Z_1 \quad 0$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if $A_Z = 1$ ; Reset if $A_Z = 0$
$J_{4-0}$	$27_8 - Z$
Memory Z	

#### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$AGN1 \cdot JE37 \cdot IR10 \cdot \overline{AR00}$	Gates $\overline{A_0}$ to B input of Adder when $J = 37_8$
G1AFNB	67	$AI10 \cdot \overline{AR00} \cdot DGN1 \cdot JE37$	
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFL2} + \overline{AFL3} + (AFL1 \cdot ECLK)$	Held reset
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{IR05}$	Shifts A Reg. right 24 places
N1J1NC	71	$JIN2 \cdot JIN4$	Enables incrementation of J Counter
F1ETST	89	$\overline{ETST} = ECTF = IR07 \cdot EI06 \cdot EG1T \cdot ET4E$ $ETST = G1ESTF \cdot EI06 \cdot ECLK$	Clear Test F/F; Set Test F/F if $A_Z = 1$

### TSC-TEST AND SHIFT CIRCULAR



TSC shifts the contents of the A Register right circular Z places. If all bits shifted out of  $A_0$  are "zero", the Test flip-flop (F1ETST) is set, otherwise it is cleared. The maximum number of shifts is 2410.

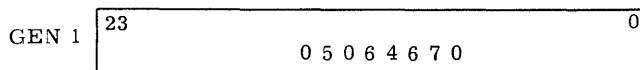
During State 4, the complement of  $I_{4-0}$  (Z) is gated to the J Counter. The Test flip-flop (F1ETST) is set. The A Register is then shifted right circular with  $A_0$  applied to the A input of the Serial Full Adder. All other Adder inputs are disabled providing a Sum output of the Adder equal to  $A_0$ . This Sum output is applied to  $A_{23}$ . If any bit shifted from  $A_0$  is a "one", the Test flip-flop is cleared. The J Counter is incremented at each shift of A. When J is equal to  $37_8$ , the number of shifts specified by Z have occurred and further shifting is inhibited.

Interruptable Following Execution?		Yes
CHANGES FOLLOWING EXECUTION		
$A_{23-0}$	C(A)shifted right circular Z places	
$Q_{23-0}$		
$P_{14-0}$	$C(P) + 1$	
F1WPMT		
F1UOFL		
F1ETST	Set } Reset }	See Text
$J_{4-0}$	$37_8$	
Memory Z		

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot IR09$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot AR00)$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = AFL3 \cdot \overline{AFNC} \cdot BCLK$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Remains reset because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{JE37}$	Shifts A Reg. right until $J = 37_8$
N1J1NC	71	$\overline{JE37} \cdot JIN1 \cdot JIN2$	Enables incrementation of J Counter until $J = 37_8$
F1ETST	89	$\overline{ETST} = G0ESTF = \overline{IR06} \cdot IR07 \cdot ET4E \cdot EG1T$ $ETST = G1ESTF \cdot \overline{IR06} \cdot ECLK$	Set Test Flip-flop; Clear if any bit shifted from $A_0 = 1$

## TZC-TEST ZERO AND COMPLEMENT



TZC sets the Test flip-flop (F1ETST) if all bits in the Register are "zero". If any bit in the A Register is a "one", the Test flip-flop is cleared. TZC also replaces the contents of the A Register with its 1's complement.

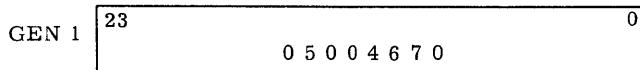
During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. This presets the J Counter to 7 because bits 2 through 0 of the TZC command are "zeros". The Test flip-flop (F1ETST) is set. The A Register is then shifted right with  $A_0$  applied to the A input of the Serial Full Adder. The Carry flip-flop (F1AFNP) is held set throughout the shift of A applying a "one" to the P input of the Adder. The Sum output of the Adder is equal to the 1's complement  $A_0$  (i. e.  $A_0$  plus 1 = complement of  $A_0$ ). This sum output is applied to  $A_{23}$  placing the 1's complement of A back in A. If at any shift of A,  $A_0$  is a "one", the Test flip-flop is cleared. The J Counter is incremented at each shift of A. When J is equal to  $37_8$ , 24 shifts of A have occurred and further shifting is inhibited.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$C(\overline{A_{23-0}})$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if $C(A) = 0$ ; Reset if $C(A) \neq 0$
$J_{4-0}$	$37_8$
Memory Z	

COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$AFL2 + \overline{AFL3}$	Held set
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{JE37}$	Shifts A Reg. right until $J = 37_8$
N1J1NC	71	$\overline{JE37} \cdot JIN1 \cdot JIN2$	Enables incrementation of J Counter until $J = 37_8$
F1ETST	89	$\overline{ETST} = G0ESTF \cdot \overline{IR06} \cdot \overline{IR07} \cdot ET4E \cdot EG1T$ $\overline{ETST} = G1ESTF \cdot \overline{IR06} \cdot \overline{ECLK}$	Set Test flip-flop Clear if $C(A) \neq 0$

## TZE-TEST A ZERO



TZE sets the Test flip-flop (F1ETST) if all bits in the A Register are "zero". If any bit in the A Register is a "one", the Test flip-flop is cleared. The original contents of A are unchanged.

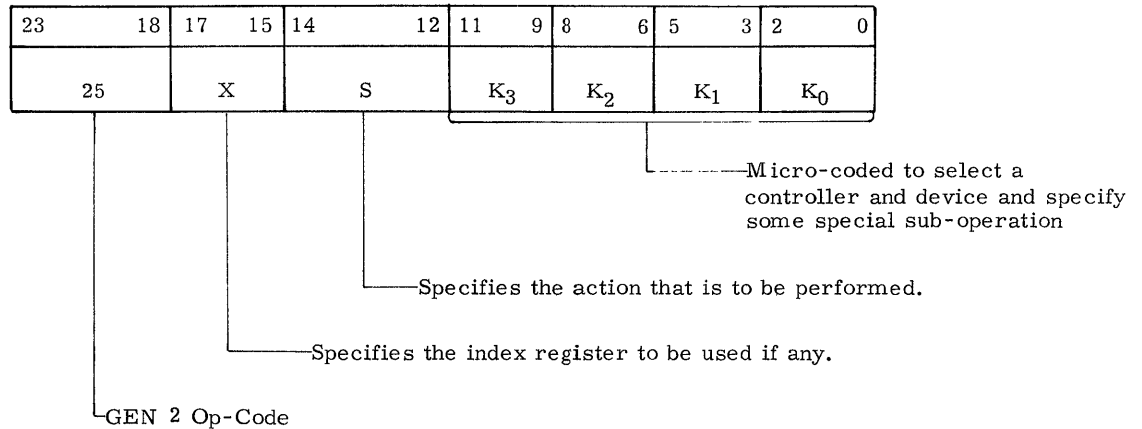
During State 4, the complement of  $I_{4-0}$  is gated to the J Counter. This presets the J Counter equal to 7 because bits 2 through 0 of the TZE command are "zeros". The Test flip-flop is set. The A Register is then shifted right circular with  $A_0$  applied to the A input of the Serial Full Adder. All other inputs of the Adder are inhibited providing a Sum output of the Adder equal to  $A_0$ . This Sum output is applied to  $A_{23}$ , leaving the A Register unchanged after 24 shifts. If at any shift of A,  $A_0$  is a "one", the Test flip-flop is cleared. The J Counter is incremented at each shift of A. When J is equal to  $37_8$ , 24 shifts of A have occurred and further shifting is inhibited.

Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	Set if $C(A) = 0$ ; Reset if $C(A) \neq 0$
$J_{4-0}$	$37_8$
Memory Z	

### COMMAND CHARACTERISTICS

Logic Element	Logic Sheet	Logic Equation	Function
G0AFNB	67	$\overline{AFNB} = \overline{IR10} \cdot \overline{IR09}$	Disabled
G1AFNB	67	$\overline{AFNB} = (\overline{AI10} \cdot \overline{AR00}) + (\overline{AI09} \cdot \overline{AR00})$	Disabled
G0AFNA	65	$AR00 \cdot ANA2$	Gates $A_0$ to A input of Adder
F1AFNP	66	$\overline{AFNP} = \overline{AFL3} \cdot \overline{AFNC} \cdot \overline{BCLK}$ $AFNP = AFL3 \cdot AFNC \cdot BCLK$	Remains reset because no summation is performed
D1ASRL, U	63.1	$ASR1 = DGN1 \cdot T6E3 \cdot \overline{JE37}$	Shifts A Reg. right until $J = 37_8$
N1J1NC	71	$\overline{JE37} \cdot JIN1 \cdot JIN2$	Enables incrementation of J Counter until $J = 37_8$
F1ETST	89	$\overline{ETST} = G0ESTF = \overline{IR06} \cdot \overline{IR07} \cdot \overline{ET4E} \cdot \overline{EG1T}$ $\overline{ETST} = G1ESTF = \overline{IR06} \cdot \overline{ECLK}$	Set Test flip-flop; Clear if any bit shifted from $A_0 = 1$

# GEN 2 INPUT/OUTPUT COMMANDS



GEN 2 commands are micro-coded and used primarily to control input/output sub-systems. GEN 2 commands, however, are also used to perform some operations internal to the Central Processor. To further illustrate the microcoding of the GEN 2 format, Fig. GN2. 1, 2, and 3 are provided. These figures show the specific microcoded functions of GEN 2 commands when operating various groups of controllers. Fig. GN2. 1 illustrates the command format when operating input/output sub-systems that utilize the memory multiplexer. Fig. GN2. 2 illustrates the GEN 2 format when operating I/O devices through the I/O Buffer. Fig. GN2. 3 illustrates the GEN 2 format when operating other I/O controllers.

The GEN 2 command is identified by the operation code 25g. GEN 2 commands are divided into two categories, internal and external, according to the K3 bits (11 thru 9). If the K3 bits are equal to 0, the command is internal. If any of the K3 bits is a "one", the command is external. External commands are Input/Output commands that select modules and devices for inputting and outputting data. A single GEN 2 command will select and address a specific device and specify the action that is to be performed. The selection of a device and the transfer of data is accomplished with a single GEN 2 command.

Internal GEN 2 commands are used to control the API module, test for Parity, Overflow, or Demand indications, read the console switches, reset the Stall Alarm, turn off or on alarm contacts and set the Trapping Mode.

GEN 2 commands cannot be relative addressed. They may be interrupted upon completion except for IAI<sub>1</sub>, IAI<sub>2</sub>, JNR, JDR, JCB and JNO. Since indexing may alter bit positions 14-0 of the GEN 2 command, caution must be exercised when specifying an index register.

The following table lists the GEN 2 commands as defined by the S (14-12) and K3 (11-9) bits.

S Bit	Internal (K3=0)	External (K3≠0)
0	(optional) { STM, LMR, LMR <sub>2</sub> , IAI <sub>2</sub> , ALARM	SEL
1	SSA	ACT
2	PAI	OPR
3	IAI <sub>1</sub>	ABT
4	JND	OUT
5	RCS	IN
6	JNO	JNR, JDR, JCB
7	JNP	JNE

## BASIC TIMING

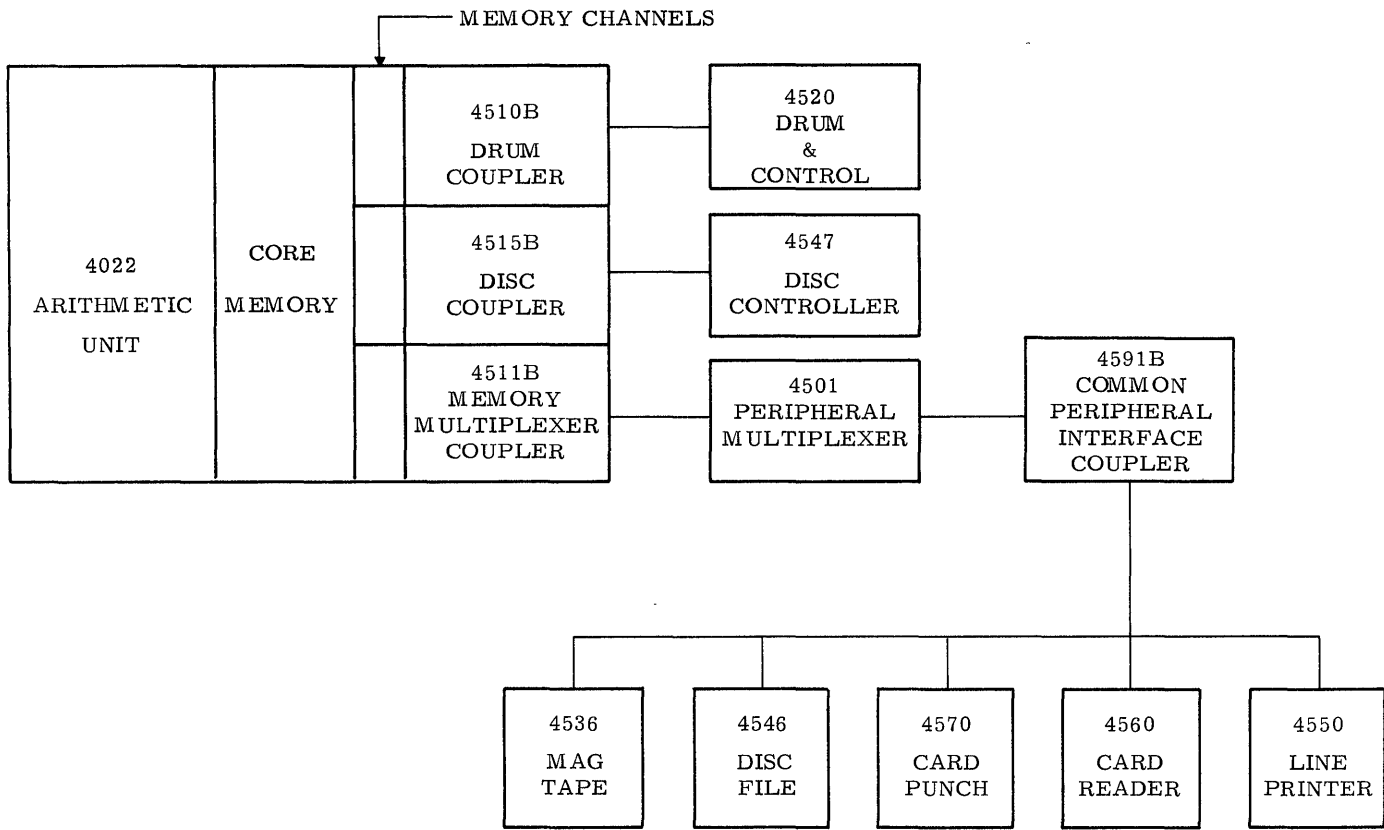
The basic timing of GEN 2 commands is divided into three groups; internal (K3=0), high-speed external (K3=7), and normal external (K3≠0 or 7). The basic timing of each of these groups is described in the following paragraphs. Following this discussion of basic timing, each GEN 2 command is described individually.

### Internal GEN 2 Commands

GEN 2 commands with K3=0 perform various functions within the Central Processor and are executed at very high speeds (i. e. 2. 2 μs.).

Fig. GN2. 4 contains a timing diagram and logic equations for GEN 2 commands with K3=0. These commands are executed during Sequence State 4 which has a duration of 0. 6 microseconds.

Since memory is not required to execute GEN 2 commands, memory is not requested and the Sequence Time Counter (FITSCA-C) is not held in Time 2 (SCA·SCB·SCC) awaiting Data Ready. Instead, it is allowed to increment to Time 5 (SCA·SCB·SCC) which generates Last Pulse Envelope (D1TLPE), ending the execution cycle.



23	18	17	15	14	12	11	9	8	6	5	3	2	0
	25	X	S			K <sub>3</sub>		K <sub>2</sub>		K <sub>1</sub>		K <sub>0</sub>	

4591 only. Specifies Pointer Word address.  
 (00 - 07 Specifies Pointer  
 Word address 30-37 respectively)

Specifies Memory Channel

- 10 = Channel one
- 20 = Channel two
- 40 = Channel three

Specifies action to be performed

- 0 = Not Used
- 1 = Activate Interrupt
- 2 = Disc (Return to zero)  
Drum (Not Used)  
4591 (Not Used)
- 3 = Abort operation
- 4 = Initiate operation
- 5 = Disc (Read Status)  
Drum (Not Used)  
4591 (Program Load - Manual Only)
- 6 = Controller Busy Test
- 7 = Error Test

Specifies Index Register

GEN 2 Op-Code

Fig. GN2.1. Memory Multiplexer Channels

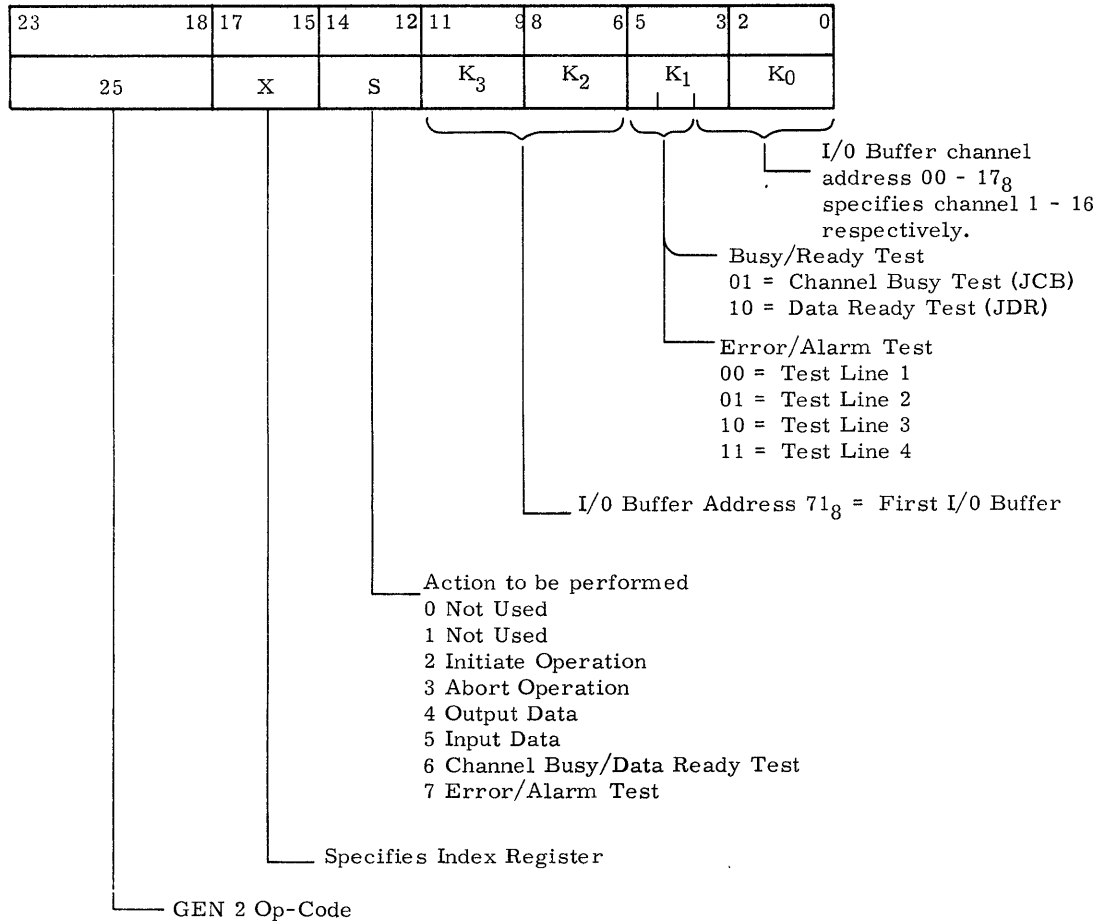
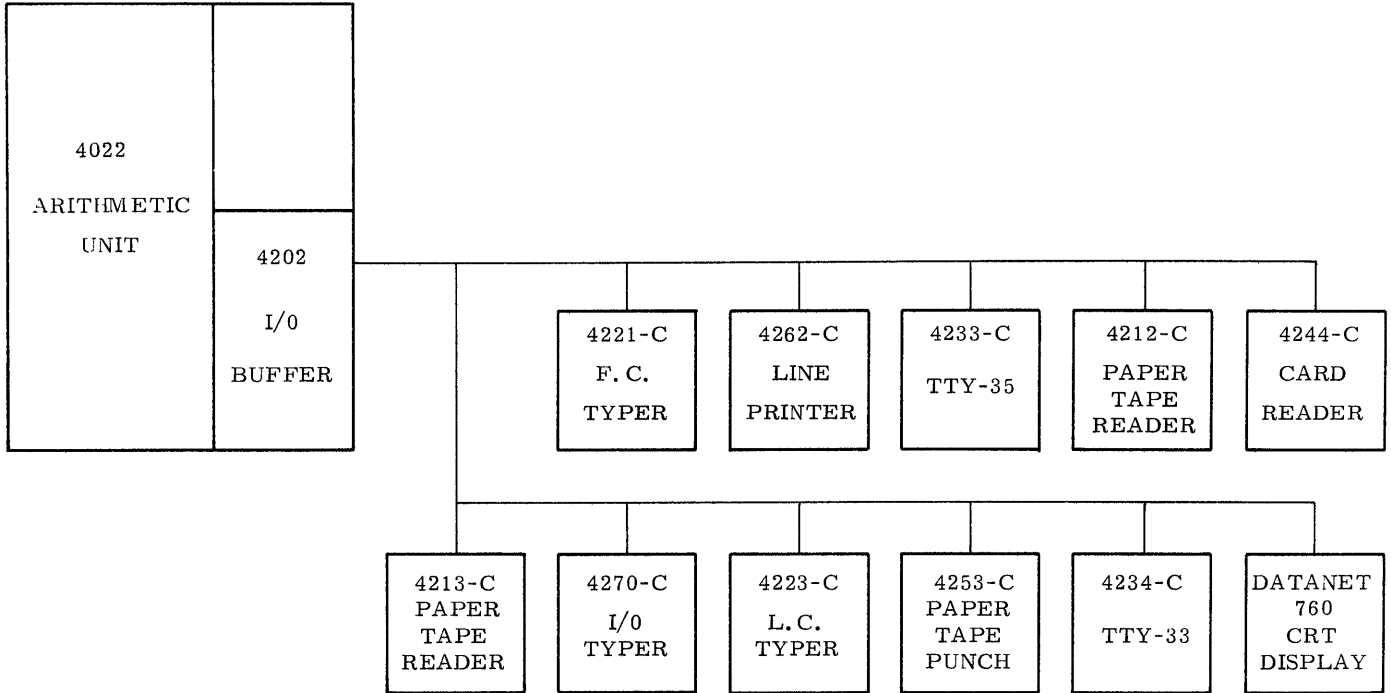


Fig. GN2. 2. I/O Buffer Application

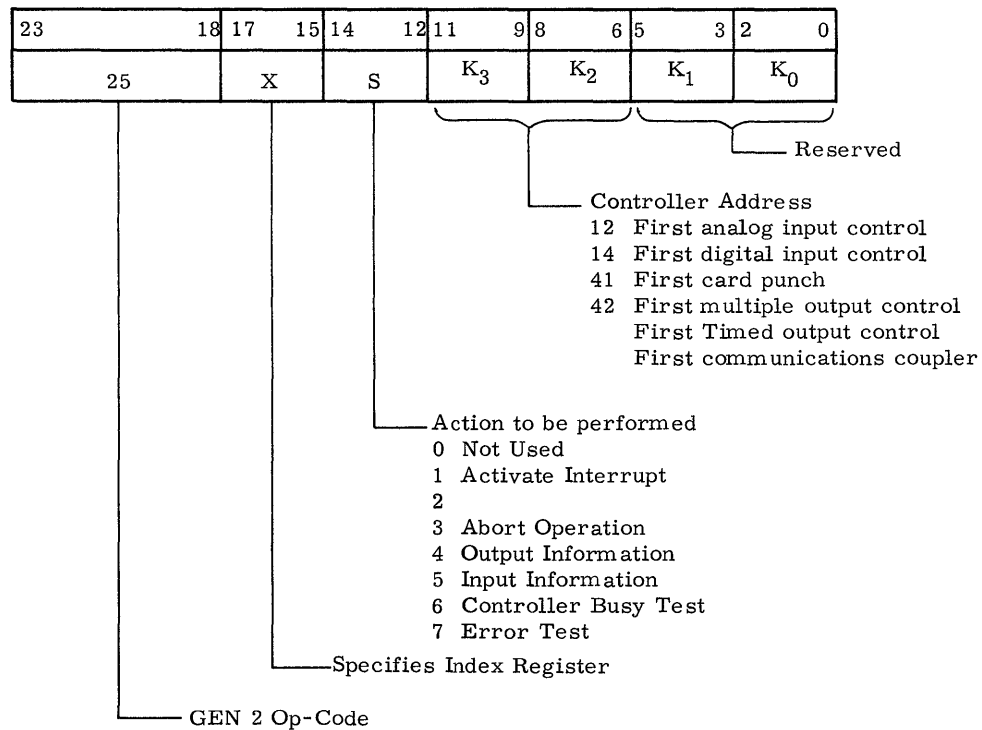
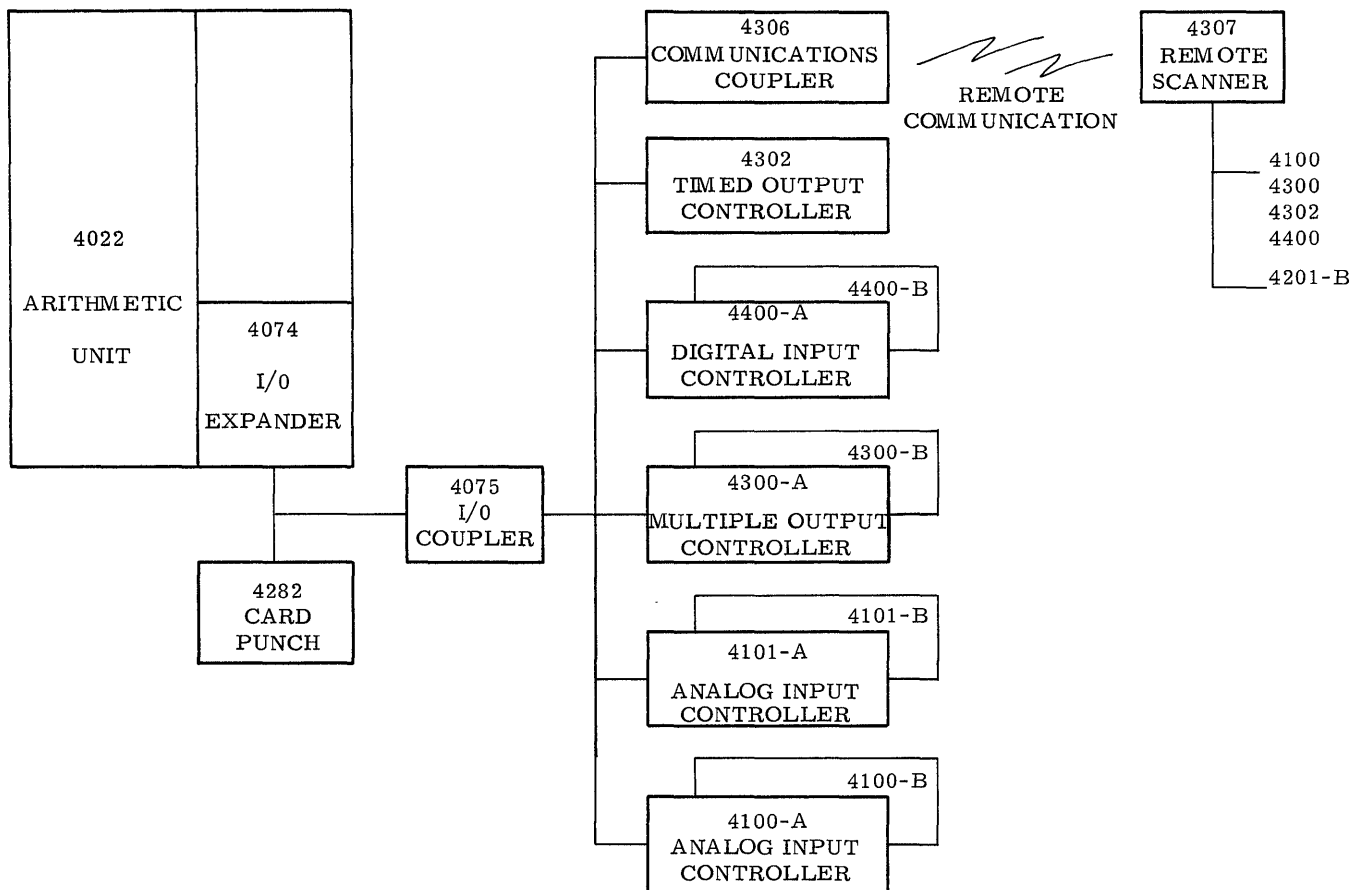


Fig. GN2.3. Controller Applications



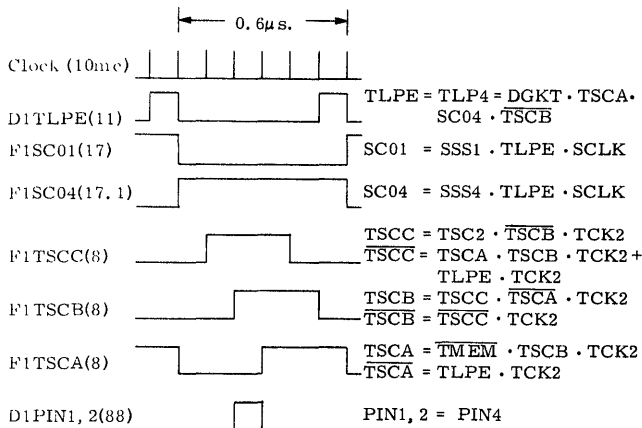


Fig. GN2.4. Internal GEN 2 Basic Timing

### High-Speed External GEN 2 Commands

High-speed GEN 2 commands are those GEN 2 commands having the K3 bits equal to 7. These commands are used with I/O Devices that operate through the I/O Buffer at high-speeds. High-speed GEN 2 commands are executed in 8.5 microseconds.

Fig. GN2.5 illustrates the basic timing signals and the associated logic equations for the execution state (State 4) of GEN 2 commands with K3 equal to 7. Sequence Control State 4 has a duration of 6.9 microseconds for these commands. The duration of State 4 is controlled primarily by Time 6 Envelope (F1TT6E), the Delay Time (binary) Counter (F1TAFF-F1TEFF), and the Extended Time Counter flip-flops (F1TEC1, 2).

The Sequence Time Counter (F1TSCA, B, C) is incremented by the first six Clock pulses of State 4 to Time 6 Envelope (F1TT6E). At Time 3 (SCA · SCB · SCC) of the Sequence Time Counter, the Delay Time Counter is preset to a count of  $30_8$  by G1TPAF, and the Extended Time Counter flip-flops (F1TEC1, 2) are set. At Time 4 (SCA · SCB · SCC) of the Sequence Time Counter, the Delay Time Counter is preset to a count of 4 by D0TP04. During Time 6 Envelope, the Delay Time Counter is incremented by each Clock pulse until it is equal to  $30_8$ . When the Delay Time Counter is equal to  $30_8$ , F1TEC2 is reset and the Delay Time Counter is again preset to 4 by D0TP04. The Delay Time Counter is again incremented until it is equal to  $30_8$ . At this time, F1TEC1 is reset and the Delay Time Counter is again preset to 4 by D0TP04. F1TEC2 is set if the command is not a JNE, JNR, JDR or JCB command with the jump condition true (i. e. if G1TP1X). The Delay Time Counter is again incremented by each Clock pulse until it is equal to  $30_8$ . At this time, with the Delay Time Counter equal to  $30_8$  and the Extended Time Counter flip-flops (F1TEC1, 2) reset, Last Pulse Envelope (D1TLPE) is enabled to end the execution cycle.

Phase A and Phase B timing signals are applied to the I/O Subsystem by D0NPHA and D0NPHB. Notice that D0NPHB is not enabled for external jump commands (JNE, JNR, JCB, JDR) when the jump condition is true.

### External GEN 2 Commands (K3 ≠ 7)

External GEN 2 commands that do not have the K3 bits equal to 7 are executed in 26.5 microseconds. Most external GEN 2 commands are in this category.

Fig. GN2.6 illustrates the basic timing signals and the associated logic equations for the execution state (State 4) of these external GEN 2 commands. Sequence Control State 4 has a time duration of 24.9 microseconds. This time duration is controlled in much the same manner as the High-Speed GEN 2 commands described above, except that the Delay Time Counter is incremented by every fourth Clock pulse instead of at every Clock pulse. The Frequency Divider (F1TED1, 2) is used to provide the Delay Time Counter triggers at every 4th Clock pulse of these external GEN 2 commands.

The Sequence Time Counter (F1TSCA, B, C) is incremented by the first six Clock pulse of State 4 to Time 6 Envelope (F1TT6E). At Time 3 (SCA · SCB · SCC) of the Sequence Time Counter, the Delay Time Counter (F1TAFF-F1TEFF) is preset to  $30_8$  by G1TPAF and the Extended Time Counter flip-flops (F1TEC1, 2) are set. At Time 4 (SCA · SCB · SCC) of the Sequence Time Counter, the Delay Time Counter is preset to 4 by D0TP04. Each Clock pulse of Time 6 Envelope increments the Frequency Divider flip-flops (F1TFD1, 2), when the Delay Time Counter is not equal to  $30_8$ . F1TFD1 and F1TFD2 act as a binary counter and are, therefore, both set at every fourth Clock pulse. When both F1TFD1 and F1TFD2 are set, G0TDFD is enabled. When enabled, G0TDFD enables G1TEID and allows incrementation of the binary Delay Time Counter (F1TAFF-F1TEFF). Therefore, every fourth Clock pulse increments the Delay Time Counter until it is equal to  $30_8$ . When the Delay Time Counter is equal to  $30_8$ , the Extended Time Counter flip-flop, F1TEC2, is cleared and the Delay Time Counter is preset to 4 by D0TP04. The Delay Time Counter is again incremented at every fourth Clock pulse until it is again incremented to  $30_8$ . At this time the Extended Time Counter flip-flop, F1TEC1, is reset and the Delay Time Counter is again preset to 4 by D0TP04. The Delay Time Counter is again incremented by every fourth Clock pulse until equal to  $30_8$ . At this time, with F1TEC1 and F1TEC2 reset and the Delay Time Counter equal to  $30_8$ , Last Pulse Envelope (D1TLPE) is enabled ending the execution cycle.

Phase A and Phase B timing signals are applied to the I/O Subsystem by D0NPHA and D0NPHB. Notice that D0NPHB is inhibited during external jump commands (JNE, JNR, JCB, JDR) when the jump condition is true.

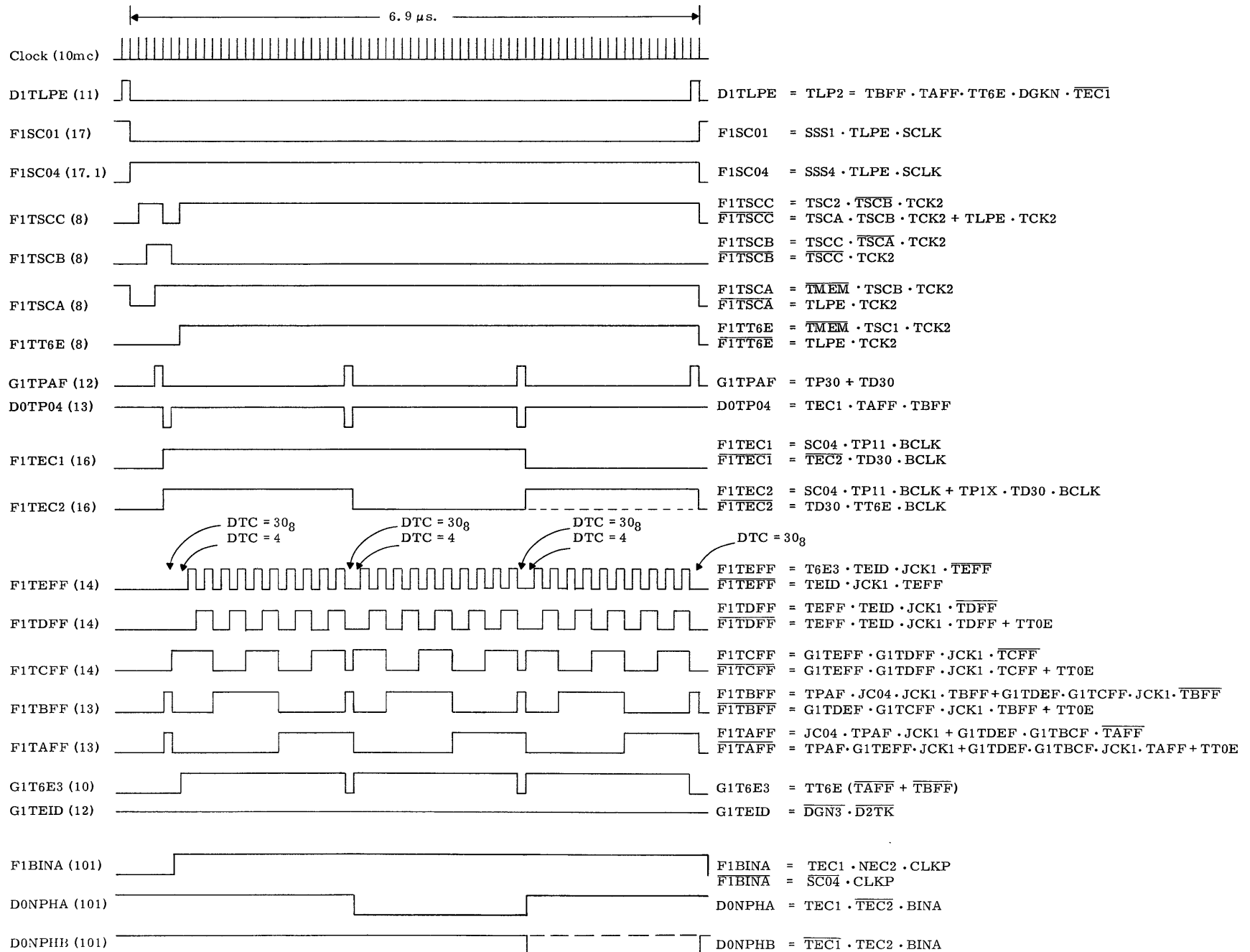


Fig. GN2. 5. High-Speed External GEN 2 Basic Timing

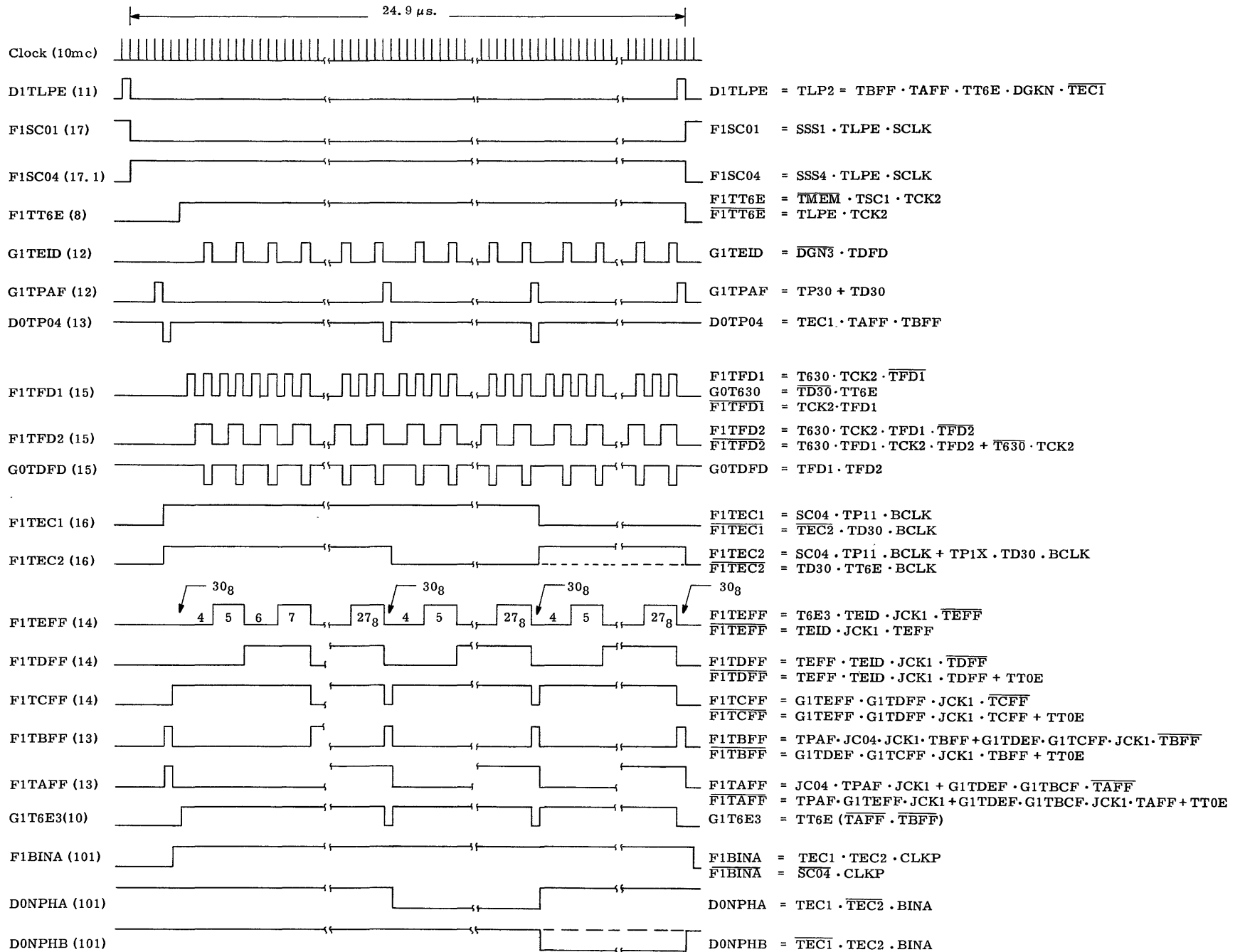
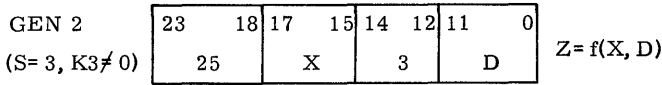


Fig. GN2. 6. External GEN 2 (K3/7) Basic Timing

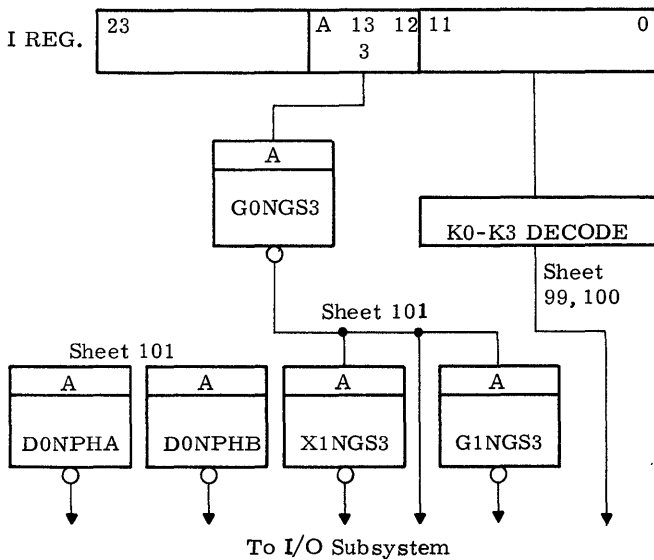
## ABT-ABORT DEVICE D's OPERATION



ABT terminates the operation of the addressed device and initializes the addressed channel. The operation may or may not have been completed, thus, ABT provides an interrupt to indicate that the ABT command has been executed. Some device operations cannot be terminated instantly, (Teletype, Magnetic Disc seek) therefore, the program must account for this delay to complete termination of the operation. For example, the Teletype requires 100 ms to transmit one character. If ABT is executed before the full character has been transmitted, the channel will be initialized but the Teletype unit must complete its cycle. Therefore, if another command is issued to the Teletype before it completes that cycle, a synchronization problem occurs.

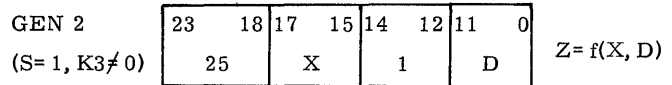
Since ABT is an external GEN 2 command (K3≠0), the duration of Sequence Control State 4 will be 6.9 microseconds if K3 is equal to 7, or 24.9 microseconds if K3 is not equal to 7. The timing diagram contained in Fig. GN2.5 or GN2.6 applies to the ABT command.

Because the S bits (14-12) of the ABT command are equal to 3, the NGS3 gates are enabled by the corresponding bits of the I Register. Bits 11 through 0 of the ABT command are decoded from the I Register, enabling the corresponding K decode gates. These signals, along with the Phase A and Phase B timing signals, are applied to the I/O Subsystem. Refer to the I/O Subsystem description (e.g. Disc, Drum, etc.) for a detailed discussion of the operation of the ABT command within the Subsystem.



ABT Block Diagram

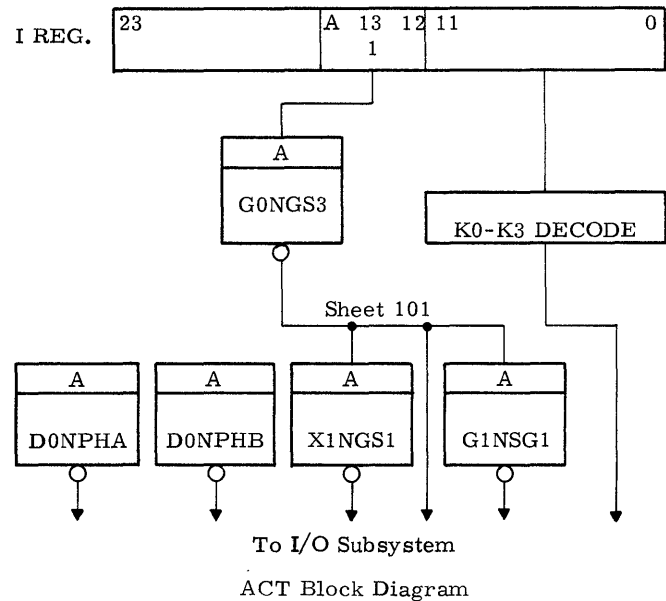
## ACT-ACTIVATE DEVICE D's INTERRUPT



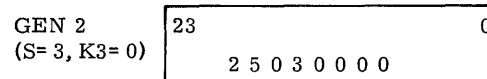
ACT indirectly initiates programmed operation of device Z by initiating an Automatic Program Interrupt from the device. If device Z is not in operation, ACT simulates the completion of an operation to provide the Automatic Program Interrupt (i.e. the ready signal is cycled to not ready and back to ready). If device Z is in operation, the ACT command is ignored. The ACT command may be used to initiate subroutines for devices connected to modules such as the Analog Input Controller, Multiple Output Controller, Drum, Disc, etc.

Since the ACT command is an external command (K3≠0), the duration of Sequence Control State 4 will be 6.9 microseconds if K3 is equal to 7 or 24.9 microseconds if K3 is not equal to 7. The timing diagram shown in Fig. GN2.5 or GN2.6 applies directly to the ACT command.

The S (bits 14-12) and K (bits 11-0) bits of the ACT command are decoded from the I Register. Since S is equal to 1, the NGS1 gates are enabled. This signal along with the K decode gates enabled and the Phase A and Phase B timing signals are applied to the addressed I/O Subsystem. Refer to the description of the particular Subsystem addressed for details of the ACT operation within the Subsystem.



## IAI<sub>1</sub>-INHIBIT AUTOMATIC INTERRUPT

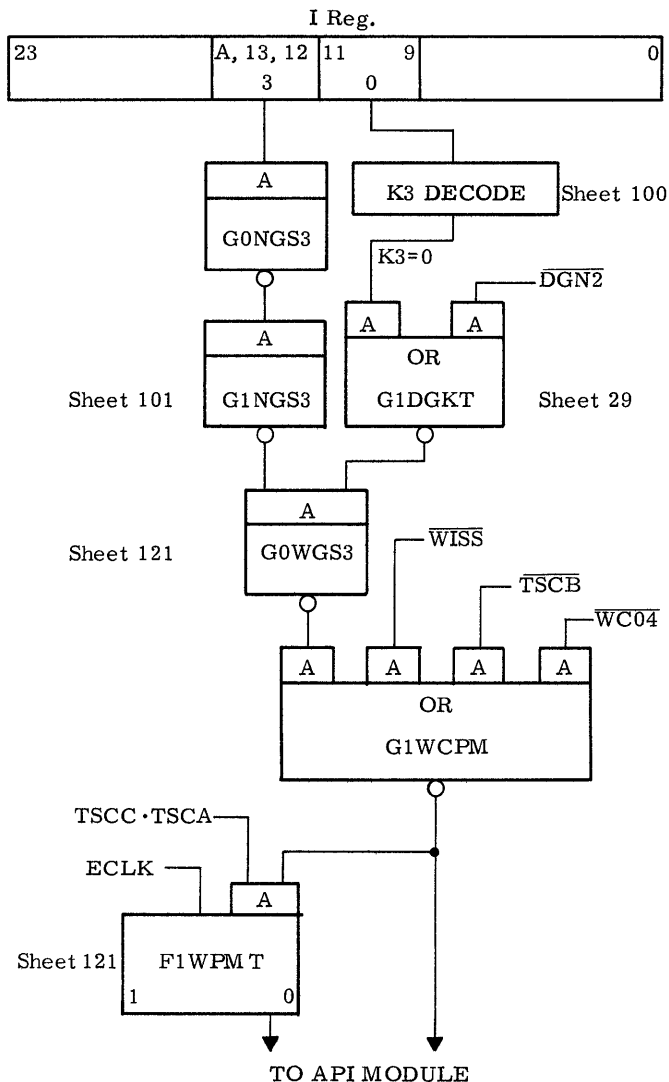


IAI<sub>1</sub> clears the Permit Automatic Interrupt flip-flop

(F1WPMT) to inhibit inhibitable program interrupts. Non-inhibitable interrupts are not affected by the IAI<sub>1</sub> command or Permit Automatic Interrupt flip-flop.

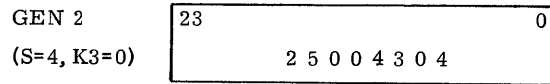
Since the IAI<sub>1</sub> command is an internal GEN 2 command, Sequence Control State 4 has a duration of 0.6 microseconds. The timing diagram and logic equations shown in Fig. GN2.4 apply to the IAI<sub>1</sub> command.

The S bits of the IAI<sub>1</sub> command are decoded from I<sub>A, 13, 12</sub> to enable the NGS3 gates. The K3 bits are decoded from I<sub>11-9</sub> and enable the NK30 gates. At time 3 (TSCA · TSCB · TSCC) the Permit Automatic Interrupt flip-flop (F1WPMT) is reset. F1WPMT when reset, applies a signal to the Automatic Program Interrupt module inhibiting inhibitable type interrupts.



IAI<sub>1</sub> Block Diagram

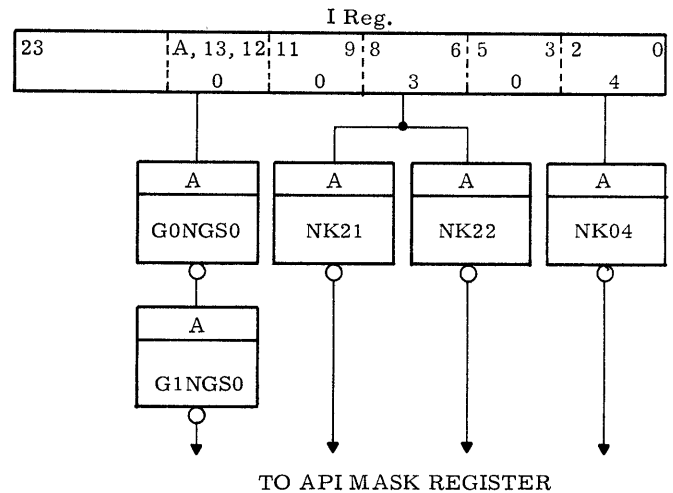
## IAI<sub>2</sub>-INHIBIT AUTOMATIC INTERRUPT(Optional)



IAI<sub>2</sub> is used in conjunction with the optional Automatic Interrupt Mask Register. IAI<sub>2</sub> inhibits all interrupts, both inhibitable and non-inhibitable provided the Permit Automatic Interrupt flip-flop has been previously cleared.

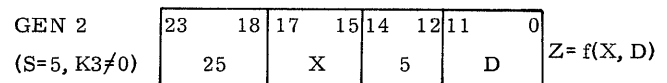
Since the IAI<sub>2</sub> command is an internal GEN 2 command, Sequence Control State 4 has a duration of 0.6 microseconds. The timing diagram and logic equations shown in Fig. GN2.4 apply to the IAI<sub>2</sub> command.

The S and K bits of the IAI<sub>2</sub> command are decoded from the I Register. Decoding these bits apply the S=0 and corresponding K decode signals to the optional API Mask Register logic. From these signals, the Mask Register logic inhibits all interrupts by clearing the Time Counter, clearing the Priority Interrupt flip-flops and disabling Echo generation.

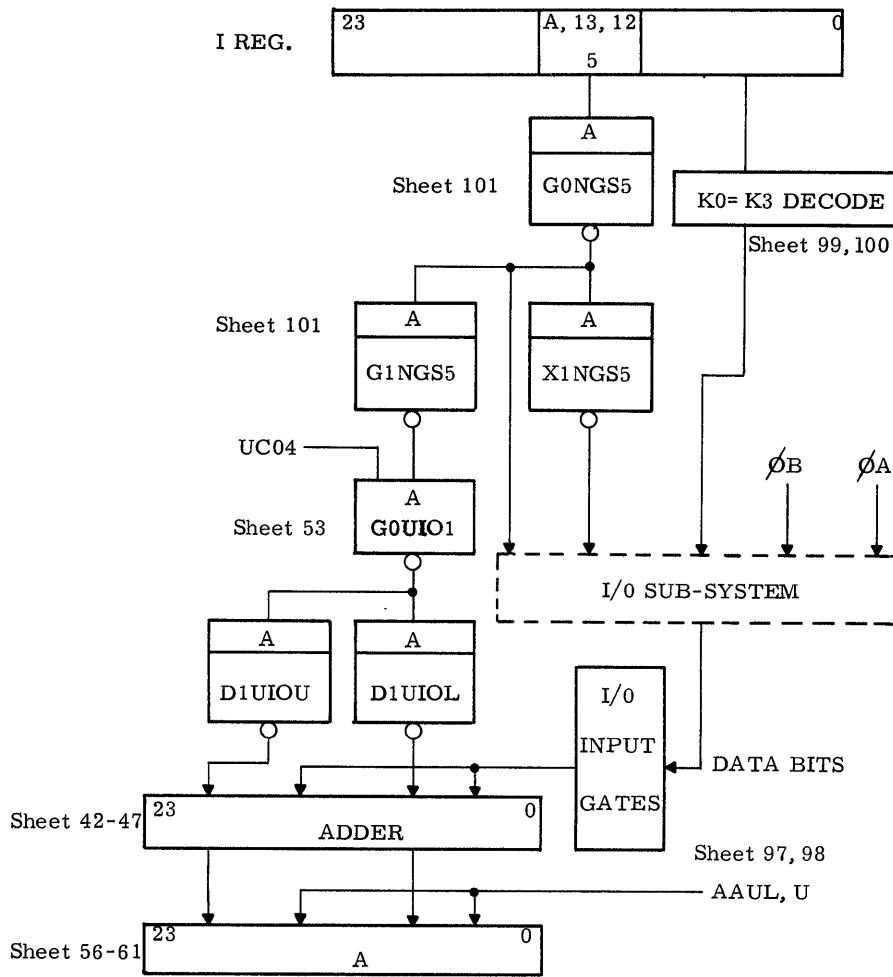


IAI<sub>2</sub> Block Diagram

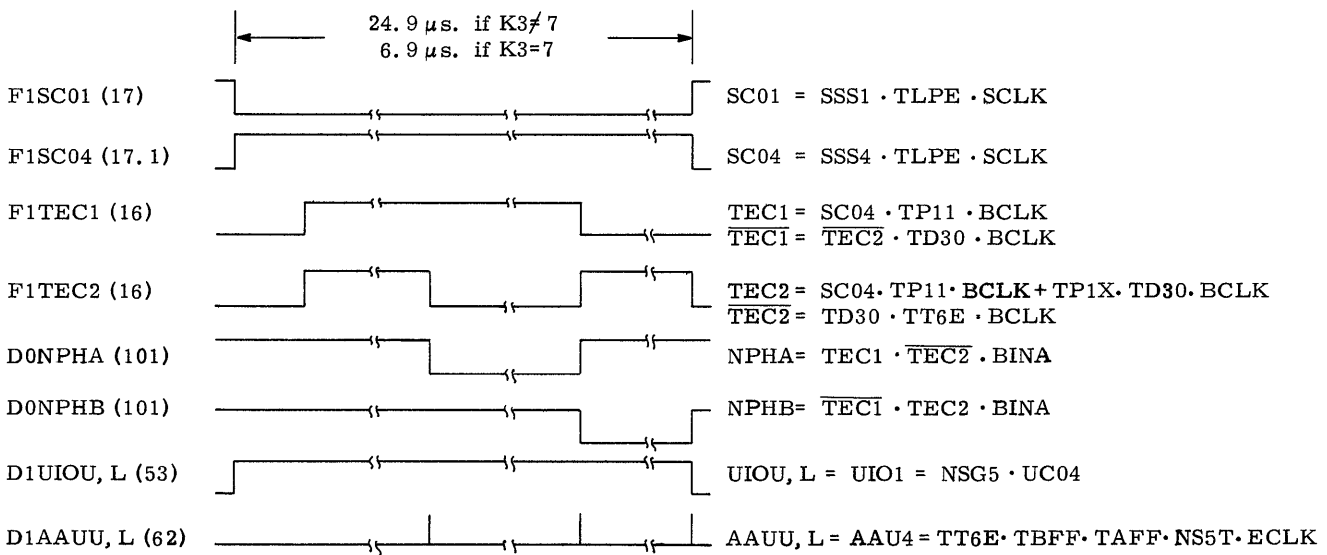
## IN-INPUT FROM DEVICE D



IN transfers data from the addressed device to the A Register. The IN command is not required on subsystems connected to the Memory Multiplexer channels with the exception of the Model 4591 Common Peripheral Interface Coupler. A manually executed IN command (via the Console) addressed to the Model 4591 initiates a program load operation. Refer to the Model 4591 for further details.



IN Block Diagram



IN Timing Diagram

The duration of State 4 is 24.9 microseconds if the K3 bits are not equal to 7, or the duration is 6.9 microseconds if the K3 bits are equal to 7. The basic timing diagram in Fig. GN2.5 or GN2.6 applies to the IN command.

Since the S bits of the IN command are equal to 5, X1NGS5 is enabled and the resultant signal is applied to the I/O Sub-system along with the Phase A and Phase B timing signals and the decoded address signals (K0 - K3). Data from the addressed subsystem is applied through the I/O Input Gates (logic sheets 97 and 98) and gated to the Adder Unit by D1UIOU, L. From the Adder Unit, the data bits are gated to the A Register by D1AAUU, L.

### JCB-JUMP IF CHANNEL BUSY

GEN 2	23	18	17	15	14	12	11	6	5	4	3	0
(S = 6, K3 ≠ 0)	25	X		6	D	0	1				D	

JCB transfers program control to the second sequential instruction (P + 2) if the addressed channel is busy. If the addressed channel is ready, program control is transferred to the first sequential instruction (P + 1).

Hardware operation of the JCB command within the Arithmetic Unit is identical to that of the JDR and JNR commands. Refer to the description of the JNR command for further details.

### JDR-JUMP IF DATA READY

GEN 2	23	18	17	15	14	12	11	6	5	4	3	0
(S = 6, K3 ≠ 0)	25	X		6	D	1	0				D	

JDR transfers program control to the second sequential instruction (P + 2) if the addressed input channel is busy and its data ready indicator is set. Control is also transferred to the second sequential instruction (P + 2) when the addressed output channel is busy and its buffer ready indicator is set. If either indicator of the addressed channel is reset, program control is transferred to the first sequential location (P + 1). When the addressed channel is not busy, program control is transferred to the first sequential location (P + 1).

Hardware operation of the JDR command within the Arithmetic Unit is identical to that of the JCB and JNR commands. Refer to the description of the JNR command for further details.

### JND-JUMP IF NO DEMAND

GEN 2	23											0
(S=4, K3=0)		2	5	0	4	0	0	0	0	0	0	

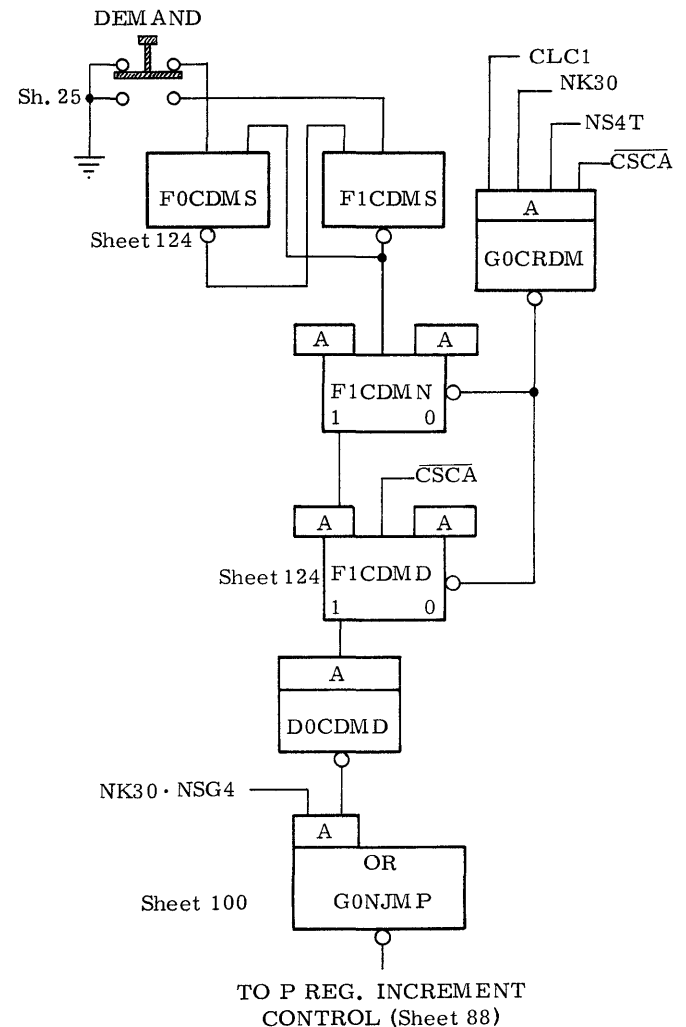
JND transfers program control to the second sequential

location (P + 2) if the Demand flip-flop (F1CDMD) is reset. If the Demand flip-flop is set, JND clears it and advances program control to the first sequential location (P + 1). The Demand flip-flop is set by pressing and then releasing the DEMAND switch on the computer console.

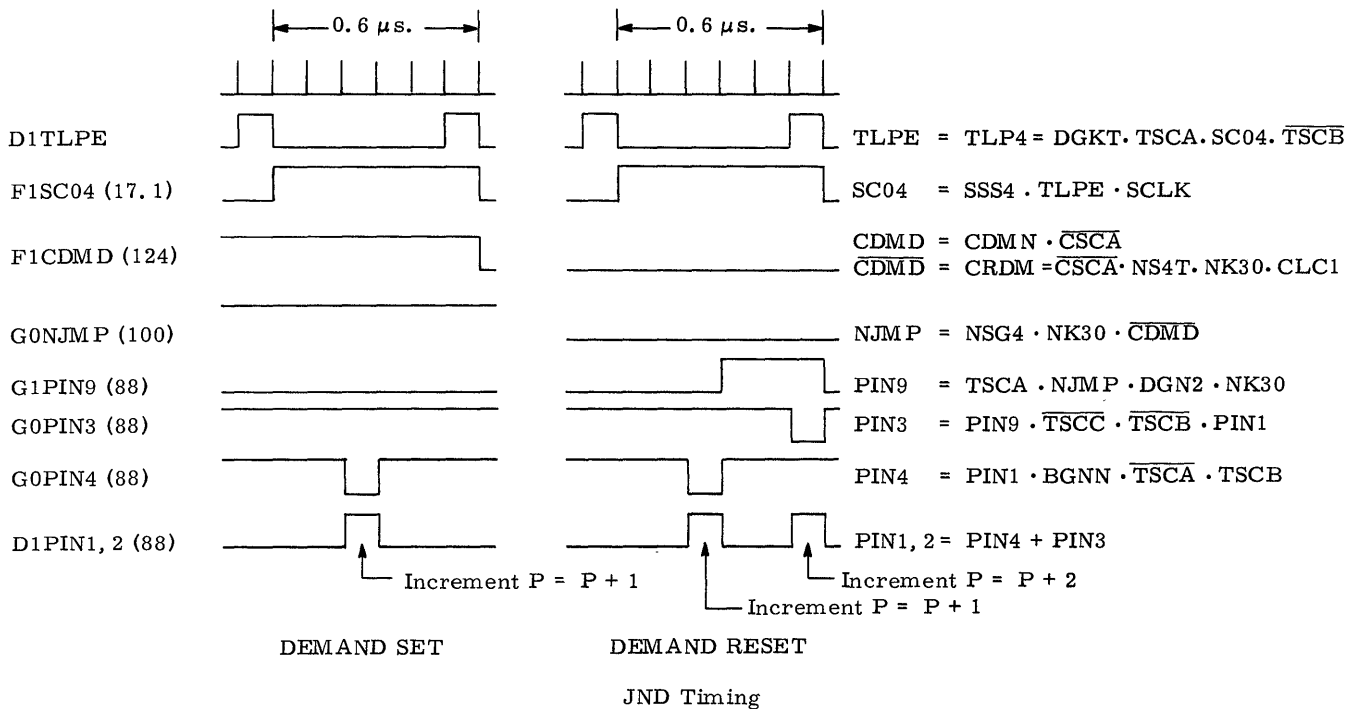
Since the JND command is an internal GEN 2 command, Sequence Control State 4 has a duration of 0.6 microseconds. The timing diagram and logic equations shown in Fig. GN2.4 apply to the JND command.

The P Register is always incremented during Time 2 ( $\overline{TSCA} \cdot TSCB$ ) of the JND command. If the Demand flip-flop (F1CDMD) is in the reset state, the P Register is again incremented at Time 5 ( $TSCA \cdot TSCB \cdot TSCC$ ). If the Demand flip-flop is set, the JND command resets it during time 0 of the following State 1.

The Demand flip-flop is set by pressing and then releasing the DEMAND switch on the computer console. Pressing the DEMAND switch applies a ground to F1CDMS providing a "one" output to arm F1CDMD. Releasing the



JND Block Diagram



DEMAND switch removes the ground from F1CDMS and applies it to F0CDMS. This provides a "one" output from F0CDMS which is applied to F1CDMS causing its output to fall to "zero" thereby setting F1CDMD. F1CDMD is cleared by the decoded JND command (NS4T · NK30) at the beginning of the following State 1.

Setting F1TEC2 enables generation of the Phase B signal to the addressed module and inhibits incrementing of the P Register for the second time. If, however no error or alarm exists, G0NJMP is enabled. When enabled, setting of F1TEC2 is inhibited when the Delay Time Counter is equal to 30g. Inhibiting F1TEC2, inhibits generation of Phase B and enables G1PIN8. G1PIN8 enables G0PIN3 causing the P Register to be incremented a second time (P + 2).

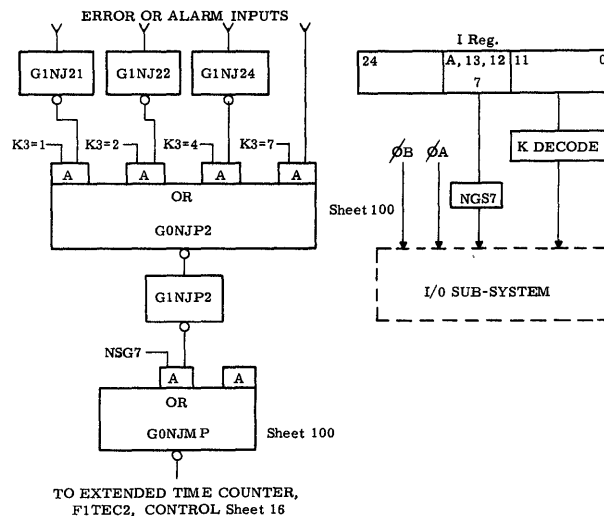
### JNE-JUMP IF DEVICE D NOT IN ERROR

GEN 2	23	18	17	15	14	12	11	0	Z = f(X, D)
(S=7, K3≠0)	25		X		7		D		

JNE transfers program control to the second sequential location (P + 2) if no error or alarm exists in the addressed module. If an error or alarm exists, program control is transferred to the first sequential instruction (P + 1). Depending on the module addressed, the JNE command may also clear the error or alarm indicator in the module.

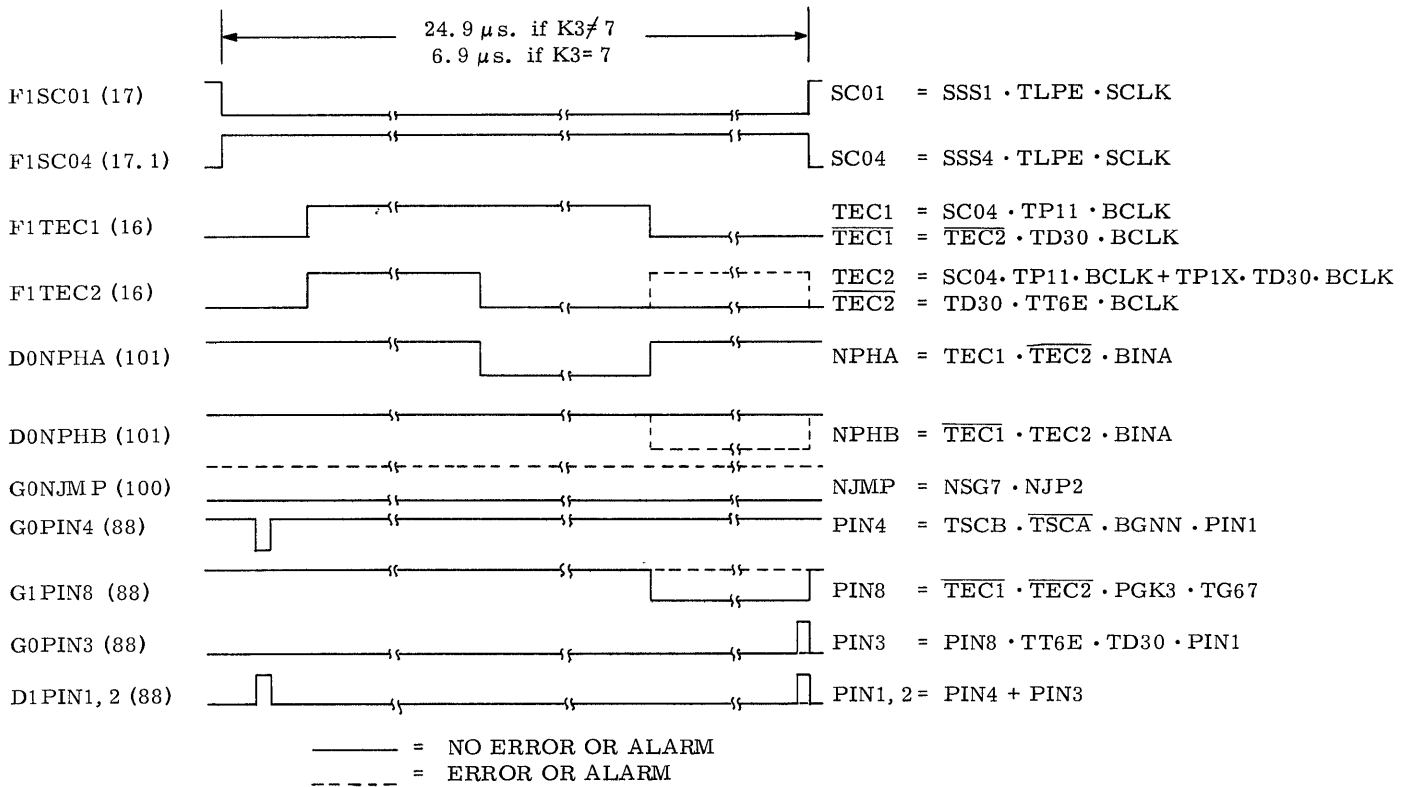
The duration of Sequence Control State 4 is 24.9 microseconds if K3 (bits 11 - 9 of the JNE command) is not equal to 7 or the duration is 6.9 microseconds if K3 of the JNE command is equal to 7. Refer to the basic timing diagram contained in Fig. GN2.5 or GN2.6.

The S bits of the JNE command are decoded from IA, 13, 12 to enable NGS7. This signal, along with the decoded K3-0 signals and Phase A and Phase B, are applied to the I/O Subsystem. The P Register is incremented (G0PIN4) at Time 2 of State 4. The error or alarm signal from the addressed module is applied through G0NJP2. If an error or alarm exists, G0NJMP is disabled. When disabled, G0NJMP enables setting of F1TEC2 when the Delay Time Counter is equal to 30g.



JNE Block Diagram

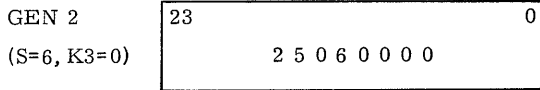




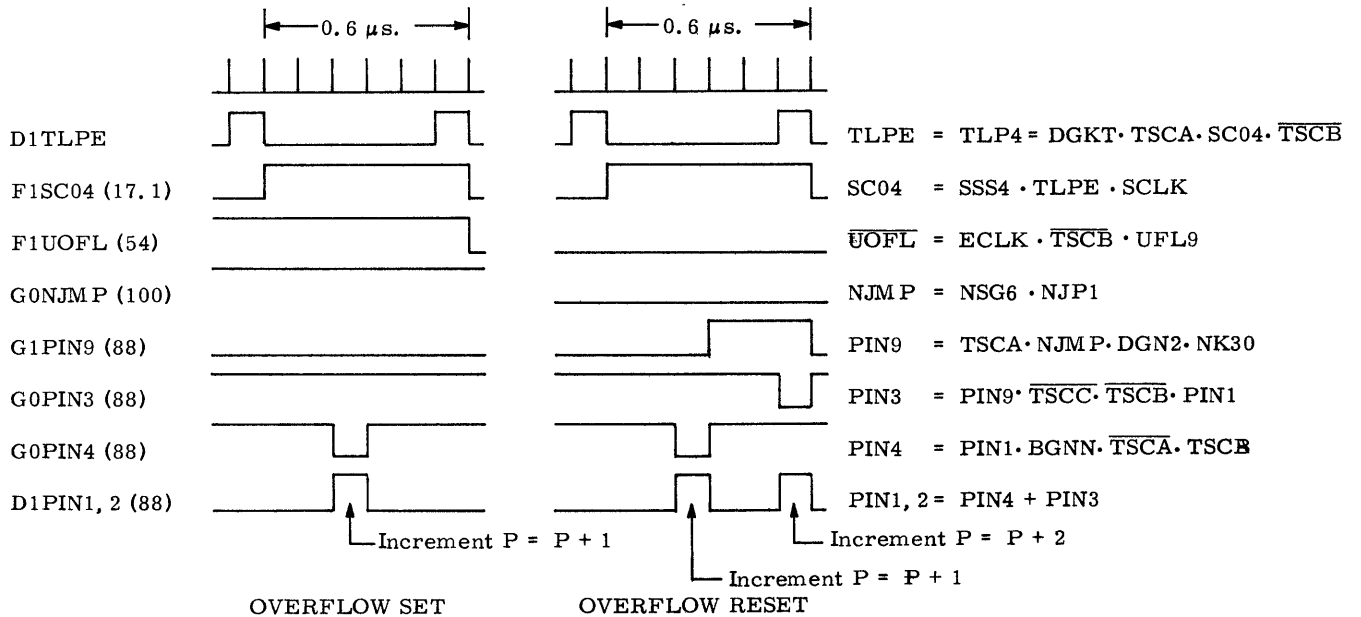
JNE Timing Diagram

**JNO-JUMP IF NO OVERFLOW**

JNO transfers program control to the second sequential location (P + 2) if the Overflow flip-flop (F1UOFL) is cleared. If the Overflow flip-flop is set, JNO clears it and transfers program control to the first sequential location (P + 1).



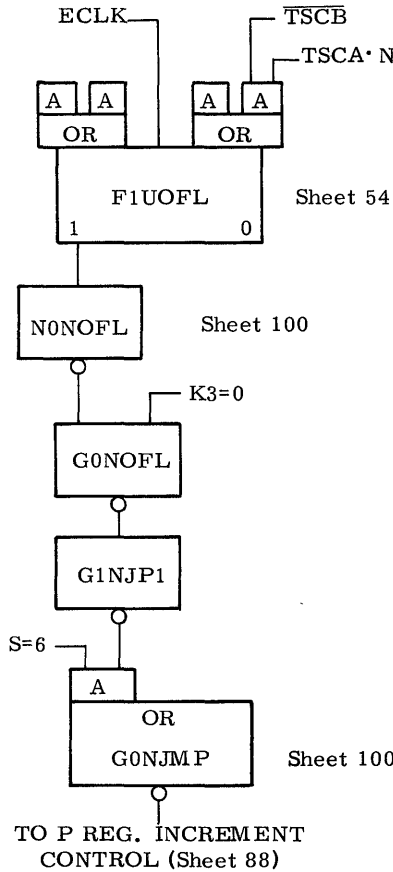
Since the JNO command is an internal GEN 2 command,



JNO Timing

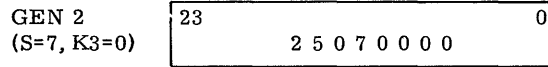
Sequence Control State 4 has a duration of 0.6 micro-seconds. The timing diagram and logic equations shown in Fig. GN2. 4 apply to the JNO command.

The P Register is always incremented during Time 2 ( $\overline{TSCA} \cdot TSCB$ ) of the JNO command. If the Overflow flip-flop (F1UOFL) is in the reset state, the P Register is again incremented at Time 5 ( $TSCA \cdot \overline{TSCB} \cdot \overline{TSCC}$ ). If the Overflow flip-flop is set, the JNO command resets it during Time 5 (G1UFL9).



JNO Block Diagram

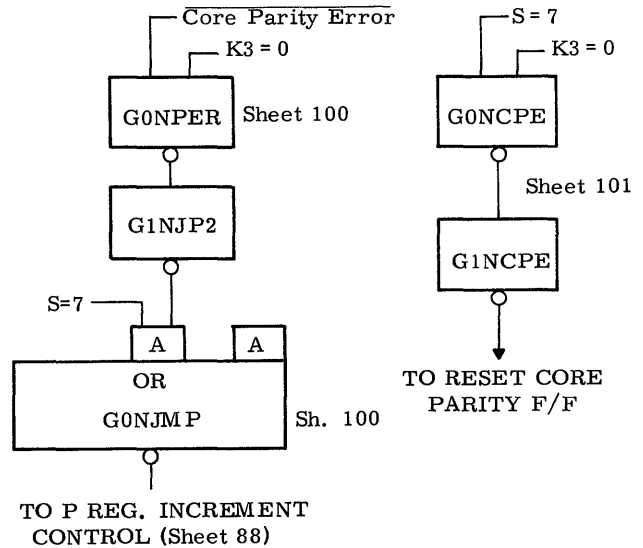
### JNP-JUMP IF NO PARITY ERROR



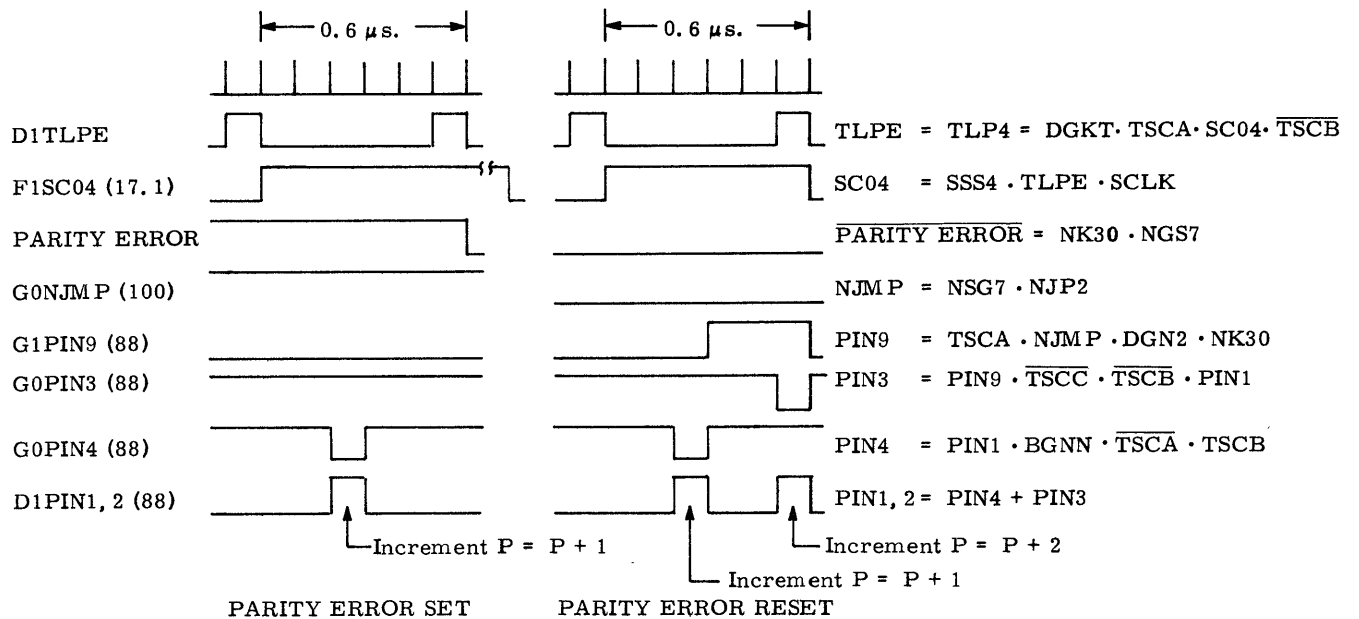
JNP transfers program control to the second sequential location (P + 2) if a core memory parity error does not exist (i. e. the Core Parity Error flip-flop is clear). If a core parity error exists, JNP transfers program control to the next sequential location (P + 1) and clears the Core Parity Error flip-flop.

Since the JNP command is an internal GEN 2 command, Sequence Control State 4 has a duration of 0.6 micro-seconds. The timing diagram and logic equations shown in Fig. GN2. 4 apply to the JNP command.

The P Register is always incremented during Time 2 ( $\overline{TSCA} \cdot TSCB$ ) of the JNP command. If the Parity Error flip-flop in the Core Memory Module is reset, the P Register is incremented for the second time (P + 2) at Time 5 ( $TSCA \cdot \overline{TSCB} \cdot \overline{TSCC}$ ). If the Parity Error flip-flop is set, the JNP command applies a signal (G1NCPE) to reset it.

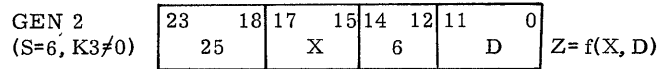


JNP Block Diagram

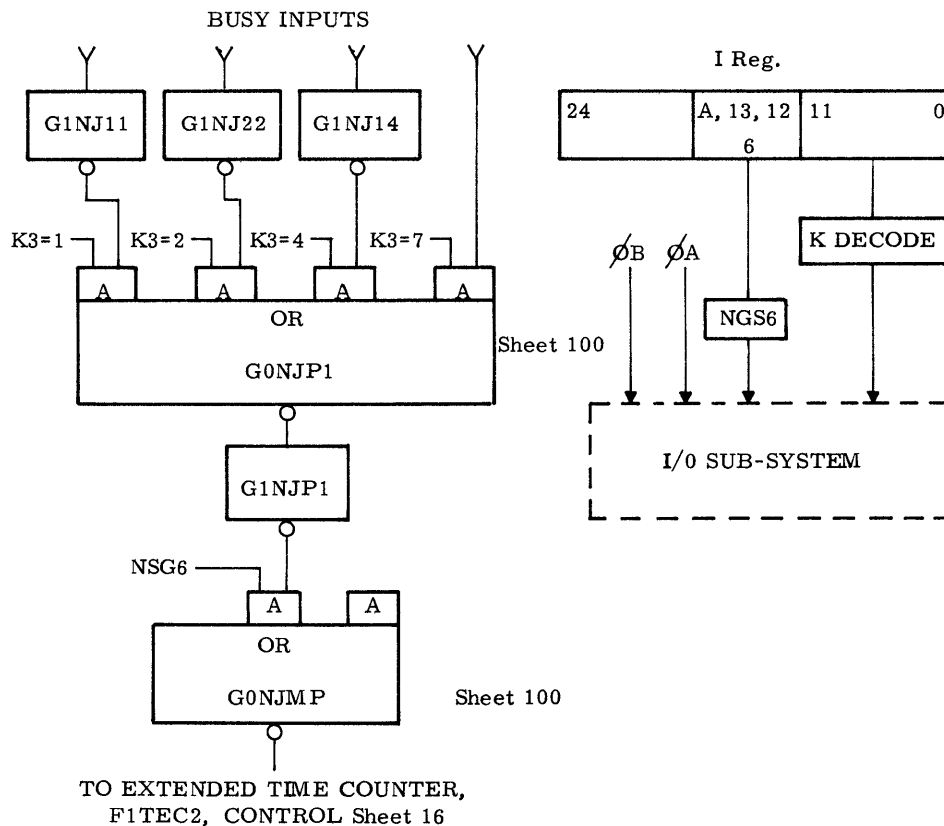


JNP Timing

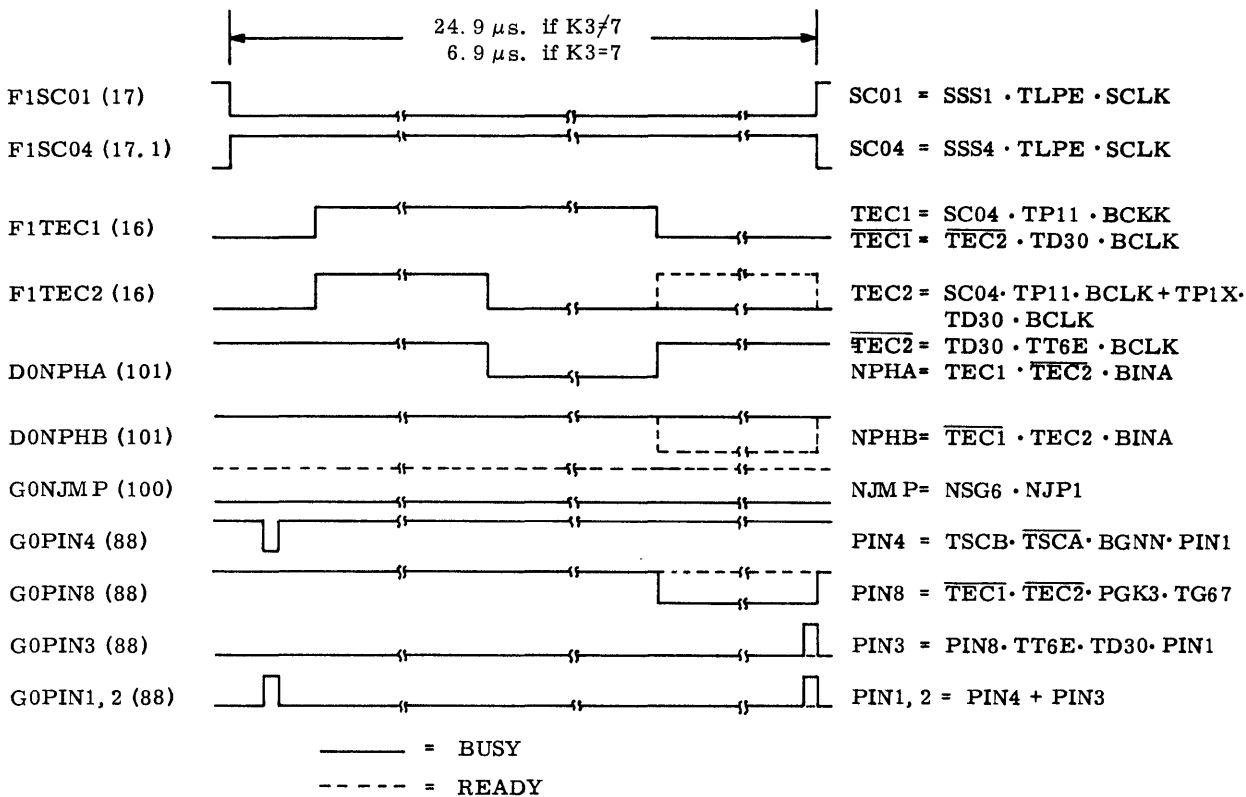
JNR-JUMP IF DEVICE D NOT READY



JNR transfers program control to the second sequential location (P + 2) if the addressed device (Z) is in operation. If the addressed device is "ready", JNR transfers program control to the first sequential location (P + 1).



JNR Block Diagram

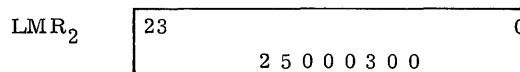
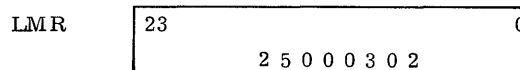


JNR Timing Diagram

The duration of Sequence Control State 4 is 24.9 microseconds if K3 (bits 11-9 of the JNR command) is not equal to 7 or the duration is 6.9 microseconds if K3 is equal to 7. Refer to the appropriate basic timing diagram contained in Fig. GN2.5 or GN2.6.

The S bits of the JNR command are decoded from  $I_{A, 13, 12}$  to enable NGS6. This signal, along with the decoded K3-0 signals and Phase A and Phase B, are applied to the I/O Sub-system. The P Register is incremented (GOPIN4) at Time 2 of State 4. The "busy" signal from the addressed device is applied through GONJP1. If the addressed device is busy (not ready), GONJP1 is enabled, enabling GONJMP. When enabled, GONJMP inhibits setting of F1TEC2 when the Delay Time Counter is equal to  $30_8$ . Inhibiting F1TEC2, inhibits generation of Phase B (DONPHB) and enables G1PIN8. G1PIN8 enables G1PIN3 causing the P Register to be incremented a second time (P + 2). If, however, the device is ready, GONJMP is inhibited. Inhibiting GONJMP enables setting F1TEC2 when the Delay Time Counter is equal to  $30_8$ . Setting F1TEC2 enables generation of Phase B and inhibits incrementation of the P Register for the second time.

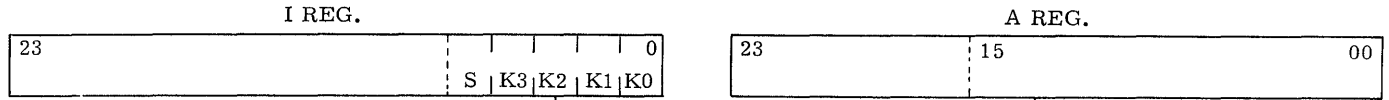
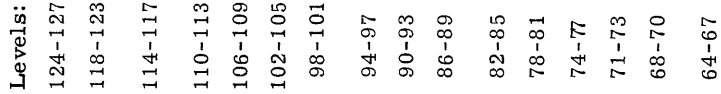
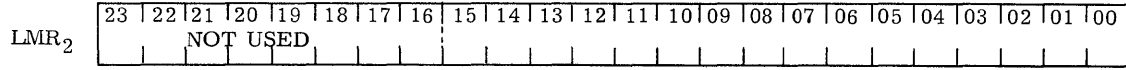
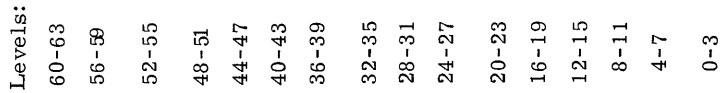
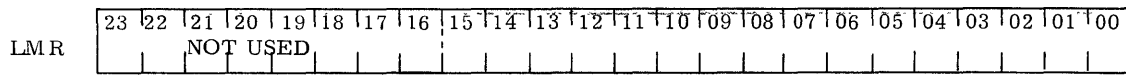
### LMR, LMR<sub>2</sub>-LOAD MASK REGISTER (Optional)



LMR transfers the contents of the A Register bits 15-0 to the first 16 bits of the optional API Mask Register. LMR<sub>2</sub> transfer the contents of the A Register bits 15-0 to the second 16 bits of the optional API Mask Register. LMR<sub>2</sub> is only used for systems having more than 64 interrupts.

Since LMR and LMR<sub>2</sub> are considered internal GEN 2 commands, Sequence Control State 4 has a duration of 0.6 microseconds. The timing diagram and logic equations shown in Fig. GN2.4 apply to the LMR and LMR<sub>2</sub> command.

Each bit of the A Register will specify if a specific group



API MASK REGISTER

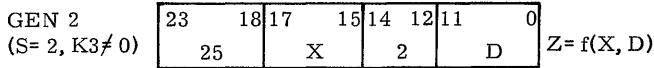
LMR, LMR<sub>2</sub> Block Diagram

of 4 interrupts are inhibited. If the A Register bit is a "one", the 4 associated interrupts are inhibited. If the A Register bit is a "zero", the 4 corresponding interrupts are allowed. The A Register bits and the interrupts affected are shown above.

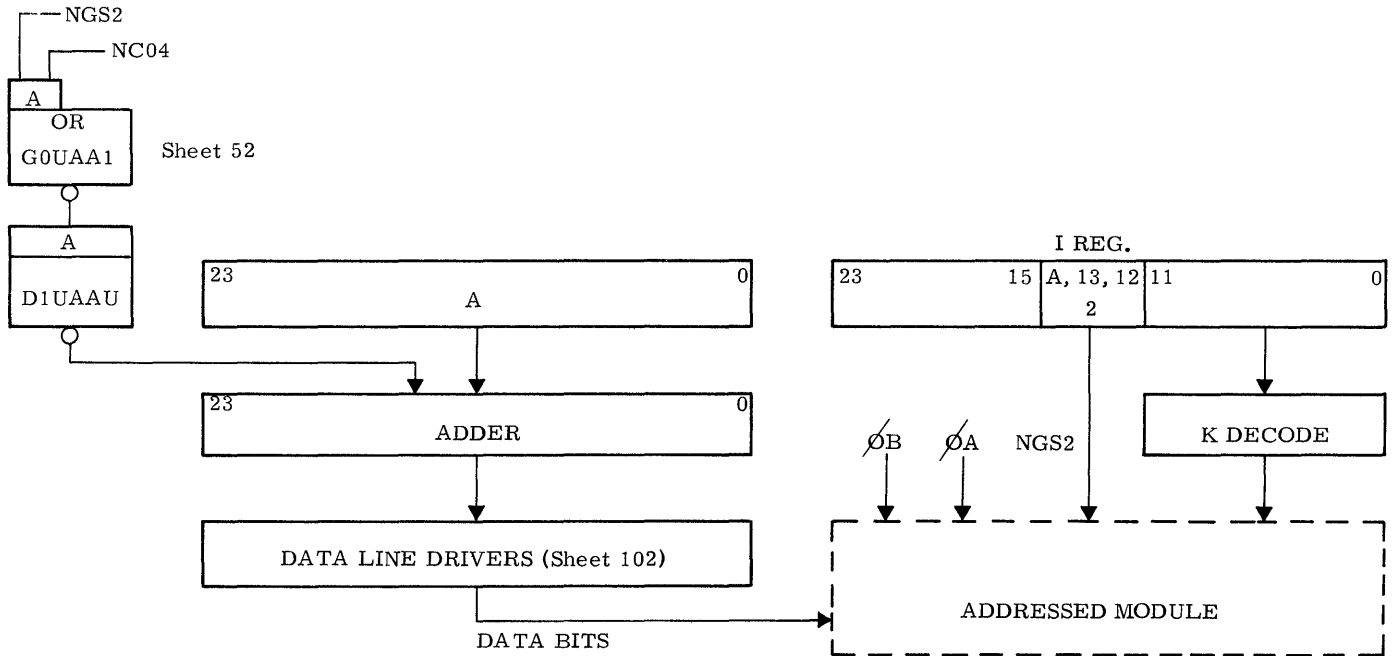
OPR is primarily used in conjunction with the Model 4202 I/O Buffer to:

1. Initiate the I/O operation or addressed channel.
2. Transfer the contents of the A Register to the addressed channel.
3. Set the addressed channel to the busy state.
4. Clear the addressed channels Error/Alarm line.

**OPR-OPERATE**



Refer to the command description for the OUT command



OPR Block Diagram

for details of operation of the OPR command within the Arithmetic Unit. The description and timing diagram of the OUT command applies to the OPR command except that the A Register is gated to the Adder Unit by enabling G0UAA1 by G1NGS2 · D1UC04 for the OPR command.

### OUT-OUTPUT TO DEVICE D

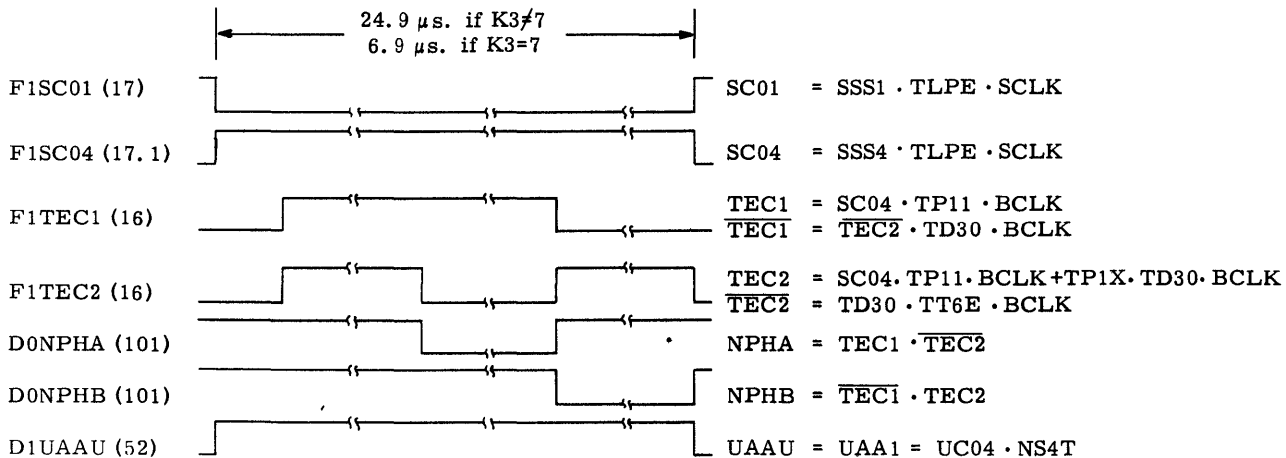
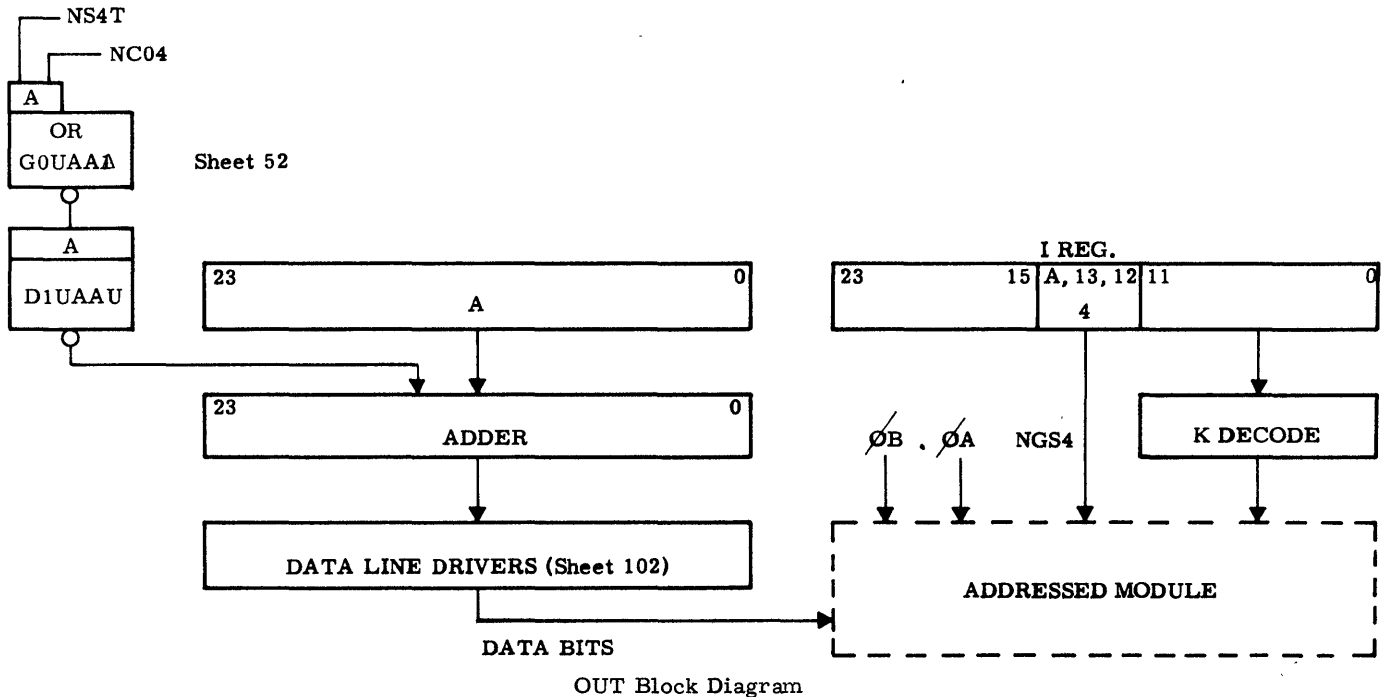
GEN 2	23	18	17	15	14	12	11	0	$Z = f(X, D)$
(S=4, K3≠0)	25		X		4		D		

OUT transfers data from the A Register to the addressed device. When the addressed device is one of the Memory Multiplexer channels, the contents of A are not used but the OUT instruction initiates the direct to memory I/O operation. Except for the Memory channels, OUT

performs two distinct functions: (1) specifies a device, (2) transfers data to that device. This data in some controllers may be instruction type data, i.e. point selection for analog input.

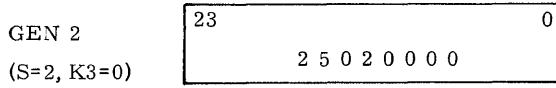
Since the OUT command is an external GEN 2 command, the duration of Sequence Control State 4 is 24.9 microseconds if the K3 bits are not equal to 7 or the duration is 6.9 microseconds if the K3 bits are equal to 7. The basic timing diagram contained in Fig. GN2.5 or GN2.6 applies to the OUT command.

The S and K bits of the command are decoded from the I Register and applied to the addressed module along with the Phase A and Phase B timing signals. The contents of the A Register are applied to the Adder unit throughout State 4. From the Adder, the contents of A are applied through the Data Line Drivers to the addressed Module.



OUT Timing Diagram

## PAI-PERMIT AUTOMATIC INTERRUPT

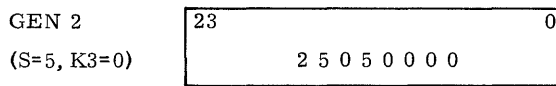


PAI sets the Permit Automatic Interrupt flip-flop (F1WPMT) to allow inhibitible program interrupts following the next interruptible command.

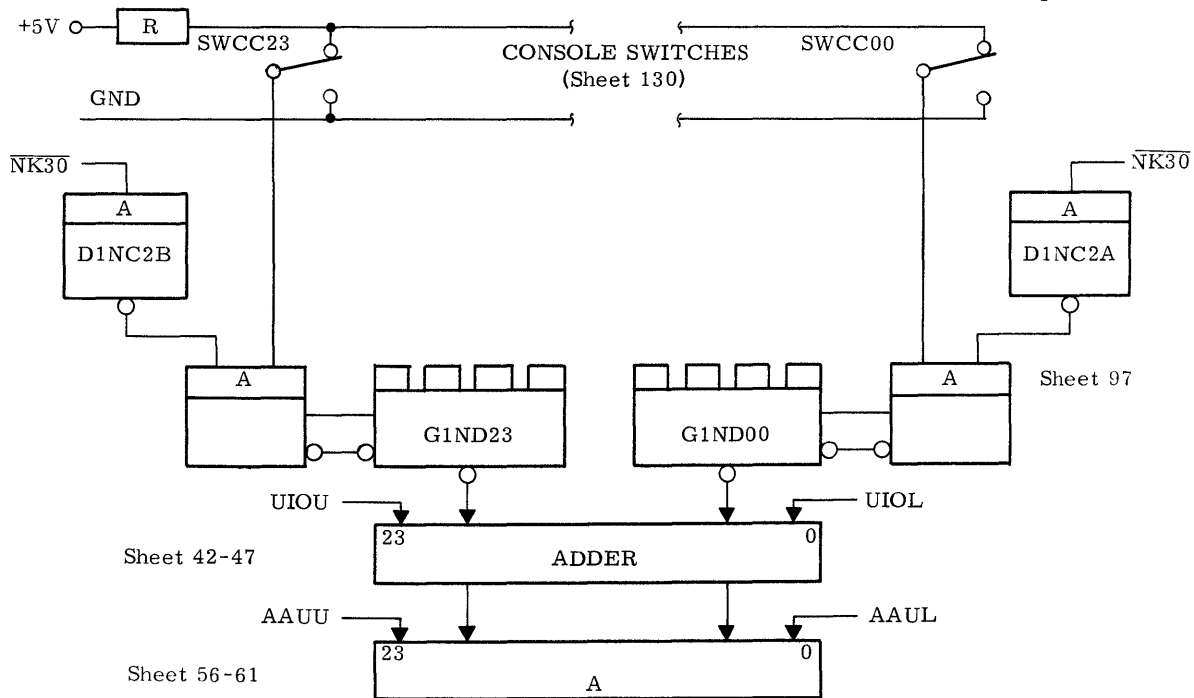
Since the PAI command is an internal GEN 2 command, Sequence Control State 4 has a duration of 0.6 microseconds. The timing diagram and logic equations shown in Fig. GN2.4 apply to the PAI command.

The S bits of the PAI command are decoded from I<sub>A,13,12</sub> to enable the NGS2 gates. The K3 bits of the PAI command are decoded from I<sub>11-9</sub> and enable the NK30 gates. At the Clock of Last Pulse of State 4, the Permit Automatic Interrupt flip-flop (F1WPMT) is set. When set, F1WPMT applies a signal to the Automatic Program Interrupt module to enable inhibitible type interrupts.

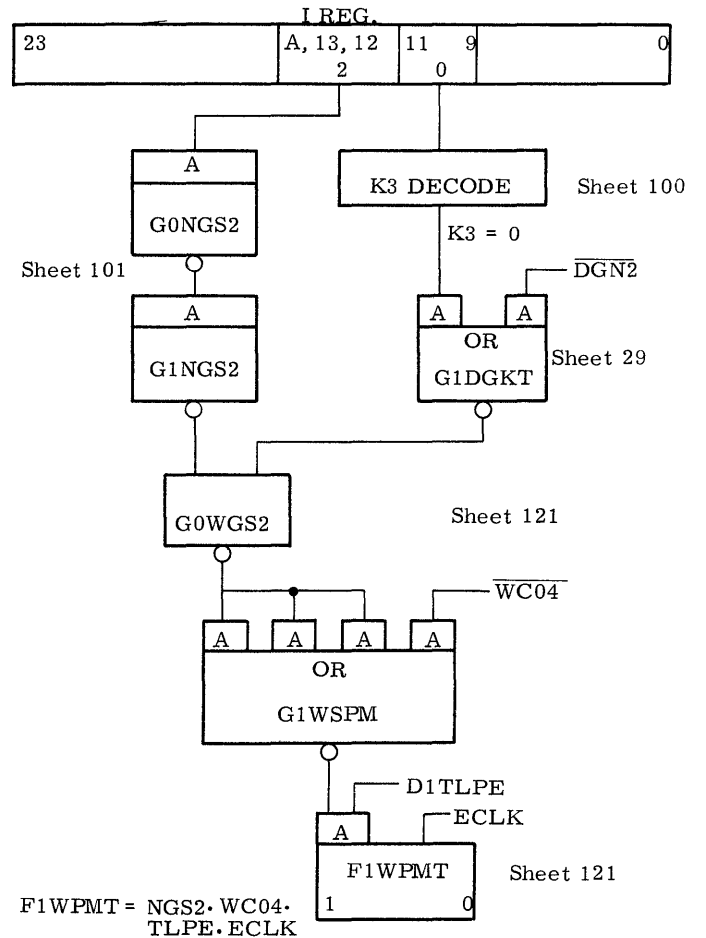
## RCS-READ CONSOLE SWITCHES



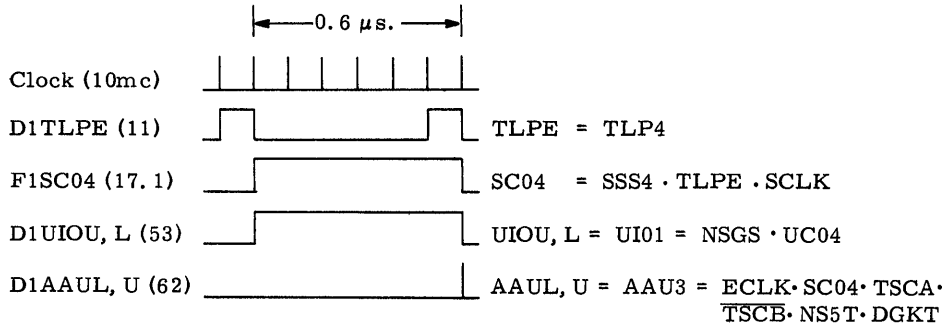
RCS places the contents of the console toggle switches in the A Register. A console toggle switch in the down position generates a "one"; a toggle switch in the up position generates a "zero".



RCS Block Diagram



PAI Block Diagram



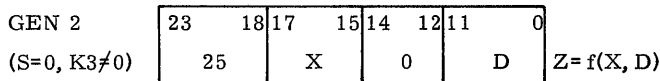
RCS Timing Diagram

position generates a "zero".

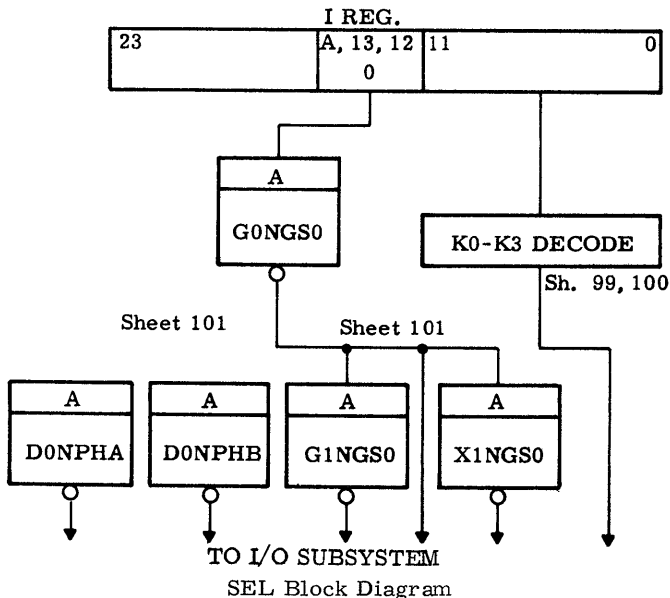
Since the RCS command is an internal GEN 2 command, Sequence Control State 4 has a duration of 0.6 microseconds. The timing diagram and logic equations shown in Fig. GN2.4 apply to the RCS command.

The S bits of the RCS command are decoded from  $I_{A,13,12}$  to enable the NGS5 gates. The K3 bits of the RCS command are decoded from  $I_{11-9}$  and enable the NK30 gates. The status of the Console Switches (SWCC23 - SWCC00) are applied through the I/O Input gates. From the I/O Input gates, the status of the Console Switches are gated to the Parallel Adder (D1UIOU, L) and at the Clock of Time 5 the contents of the Parallel Adder are gated to the A Register (D1AAU, L).

### SEL-SELECT DEVICE D



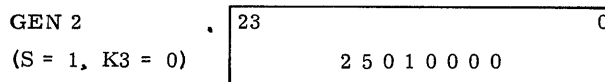
SEL enables a device (Z) in the input/output equipment, but performs no functional operation on, or with, the device. The SEL command is used only when timing requirements between the Arithmetic Unit and the device



are such that the time allotted for an IN, OUT, ACT, etc., is not enough to complete a functional operation. IN, OUT, ACT, etc., contain selection and functional operation capabilities within the command, but where timing requirements between the AU and the I/O module are restrictive, the SEL command precedes these commands. The SEL command is not normally required for I/O Modules currently in use.

Since SEL is an external GEN 2 command ( $K3 \neq 0$ ), the duration of Sequence Control State 4 is 6.9 microseconds if K3 is equal to 7 or 24.9 microseconds if K3 is not equal to 7. The timing diagram and logic equations contained in Fig. GN2.5 or Fig. GN2.6 apply to the SEL command. The S bits of the SEL command are decoded from  $I_{A,13,12}$  to enable the NGS0 gates.  $I_{11-0}$  is decoded to enable the corresponding K decode gates. These signals along with Phase A and Phase B are applied to the I/O Subsystem.

### SSA-SET STALL ALARM (Optional)



SSA resets a manually adjustable (1 to 5 second) timer in the optional Stall Alarm circuitry. If the timer is allowed to "time out", because the program does not execute another SSA command to again reset the timer, an output signal is available to light the Stall Alarm and Error indicators on the console and optionally halt computer sequencing, generate an interrupt, etc.

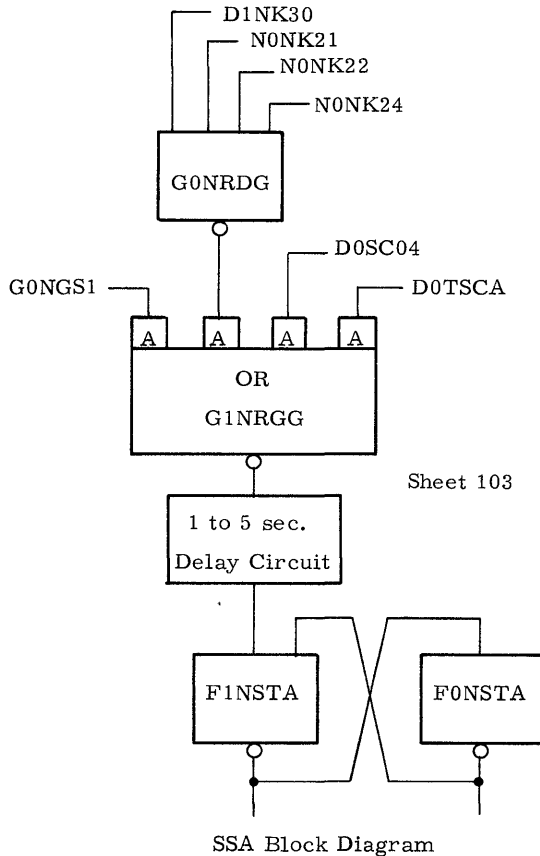
The Stall Alarm, as described in the Options section, is used to detect a malfunction in the computer program or system that causes a hang-up or stall condition in program sequencing. The stall alarm circuitry is inhibited in the manual mode of operation or when the Stall Lock-out switch is in the lockout position. Refer to the Options section for a detailed discussion of the Stall Alarm circuitry.

Since the SSA command is an internal GEN 2 command, Sequence Control State 4 has a duration of 0.6 microseconds. The timing diagram and logic equations shown in Fig. GN2.4 apply to the SSA command.

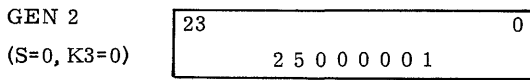
The S and K bits of the SSA command are decoded enabling G1NRRG at TSCA time of State 4. Enabling G1NRRG



initiates a new time period in the manually variable 1 to 5 second timer. If the timer is allowed to "time out", F1NSTA is set providing the error signal. Once set, F1NSTA may be cleared by pressing the ON (initialize) switch or by placing the STALL L/O switch in the lockout position with the console enabled.



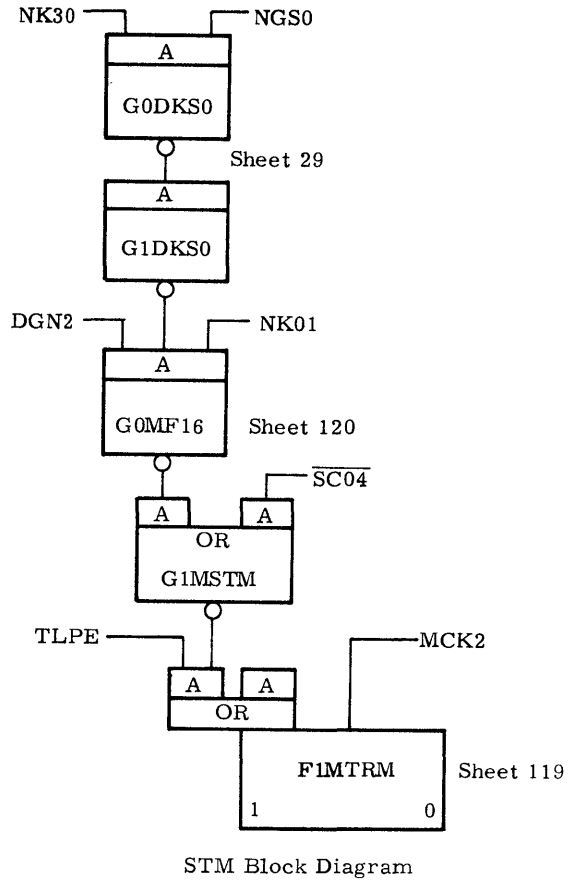
**STM-SELECT TRAPPING MODE (Optional)**



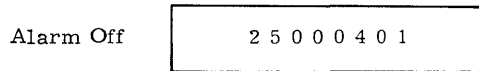
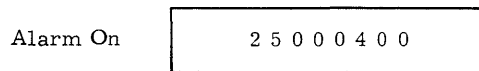
STM sets the Trapping Mode flip-flop (F1MTRM) to enable the protection functions of the optional Memory Protect (memory fence) capability. Refer to the Memory Protect discussion contained in the Options Description portion of this section for details.

Since the STM command is an internal GEN 2 command, Sequence Control State 4 has a duration of 0.6 microseconds. The timing diagram and logic equations shown in Fig. GN2.4 apply to the STM command.

The S bits of the STM command are decoded from I<sub>A</sub>, 13, 12 to enable NGS0. The K0 bits of the STM command are decoded from I<sub>2-0</sub> to enable NK01. These signals enable setting of the Trapping Mode flip-flop F1MTRM at the Clock of Last Pulse of State 4.



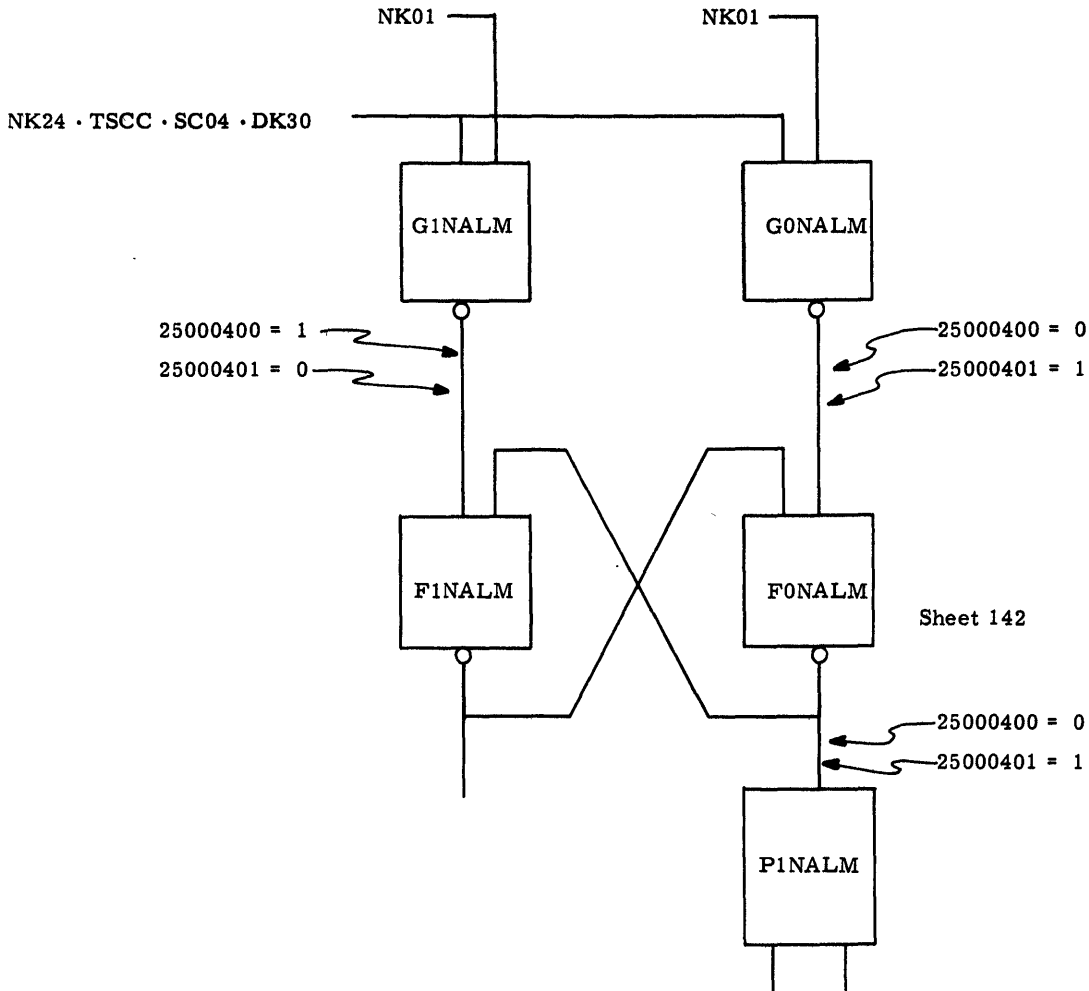
**PROGRAMMABLE ALARM**



The programmable alarm internal GEN 2 commands provide the capability of enabling a relay driver, under program control, for optional use by the system. The relay driver is turned on by executing 25000400 and turned off by executing 25000401.

Since the K3 bits of these commands are equal to 0, these commands are executed in 0.6 microseconds. The basic timing diagram contained in Fig. GN2.4 applies to these commands. The logic associated with these commands is shown on sheet 142 of the logic.

When the Alarm On command is executed, sets F1NALM enabling the relay driver P1NALM. F1NALM remains in the set state, enabling the relay driver, until the Alarm Off command is executed. When executed, the Alarm Off command resets F1NALM disabling the relay driver.



Programmable Alarm, Block Diagram

# GEN 3 COMMANDS

GEN 3 commands are microcoded instructions that provide serial shifting of the A and Q Registers, both left and right, and optionally affect the Overflow flip-flop. Actual shifting of the Q Register (memory cell 10<sub>8</sub>) occurs in the B Register.

GEN 3 commands are identified by the OP code 45<sub>8</sub> (bits 23-18). Fig. GN3.1 illustrates the GEN 3 format and explains the microcoded bit control.

GEN 3 commands "fetched" during Sequence Control State 1 are executed during Sequence Control States 4 and 5. The duration of State 4 is extended and controlled by the J Counter. The J Counter is preset from the shift constant (bits 4 - 0) of the GEN 3 command and then incremented at each shift of the B and/or A Register until it is equal to 37<sub>8</sub>. When J is equal to 37<sub>8</sub>, Last Pulse of State 4 is enabled. In this manner, the execution time of GEN 3 commands varies from 4.8 to 8.0 microseconds depending upon the value of the shift constant. GEN 3 commands may be interrupted following execution.

Due to the microcoding of GEN 3 commands, there are hundreds of unique operations that may be performed by

GEN 3 commands. At this time, 8 of these operations are considered to be used frequently and mnemonics have been assigned. These 8 commands and the microcoding are listed in Table GN1.1. Following the basic timing and sequence of events for GEN 3 commands, each of the 8 commands having assigned mnemonics is described.

## BASIC TIMING

Fig. GN3.2 contains a flow chart illustrating the basic operation of GEN 3 commands. Fig. GN3.3 contains a timing diagram with logic equations that applies to all GEN 3 commands. Refer to these aids during the following discussion.

Like all other commands, GEN 3 commands are "fetched" from memory during Sequence Control State 1. Non-indexed GEN 3 commands are then executed in Sequence State 4. During State 4, core memory cell 10<sub>8</sub> is addressed (GOMX03). As previously described, memory cell 10<sub>8</sub> is used as the Q Register. The contents of memory cell 10<sub>8</sub> are gated to the B Register. Also, during this time, the J Counter is cleared (D0JJE0) and the complement of I<sub>4-0</sub> (K) is gated to J.

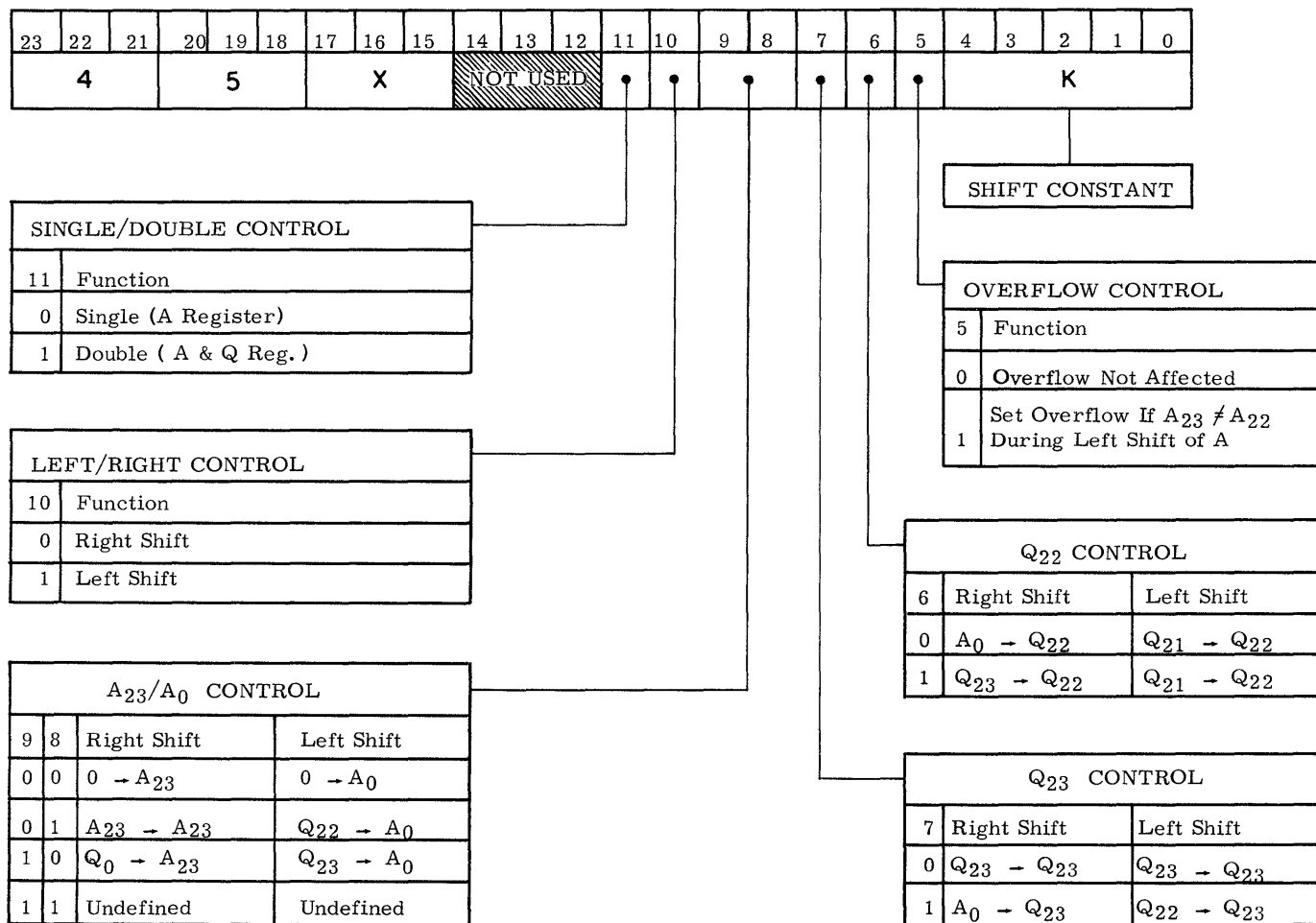


Fig. GN3.1. Action Control Bits of GEN 3 Commands

MNEMONIC	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DLA	0	0	0	1	1	0	1	0	0	1	← K →				
DLL	0	0	0	1	1	1	0	1	0	0	← K →				
DRA	0	0	0	1	0	0	1	0	0	0	← K →				
DRC	0	0	0	1	0	1	0	1	1	0	← K →				
DRL	0	0	0	1	0	0	0	1	1	0	← K →				
MAQ	0	0	0	1	0	0	0	1	1	0	1	1	0	0	0
SLA	0	0	0	0	1	0	0	0	0	1	← K →				
SLL	0	0	0	0	1	0	0	0	0	0	← K →				

Table GN3. 1. Microcoding of Defined GEN 3 Commands

During this portion of State 4, the Sequence Time Counter (F1TSCA - C) is incremented in the same manner as full operand commands. That is, the Sequence Time Counter is incremented and held in Time 2 ( $\overline{SCA} \cdot SCB \cdot SCC$ ) until Data Ready (MUD1MDRY) is received from memory. The Sequence Time Counter is then incremented until Time 5 ( $SCA \cdot \overline{SCB} \cdot \overline{SCC}$ ), where it is held until Memory Release (MUD1MRLS) is received from memory. After Memory Release is received, the Sequence Time Counter is incremented to Time 6 (F1TT6E).

During Time 6, the B and/or A Registers are shifted according to the microcoding of the GEN 3 command. These shifts occur at each clock pulse of Time 6. At each shift, the J Counter is incremented. When the J Counter is equal to  $37g$ , the number of shifts specified by the K bits (4-0) of the GEN 3 command have occurred, further shifting is inhibited and Last Pulse is enabled to end State 4.

Following State 4, State 5 is entered. During State 5, memory cell  $10g$  is again addressed (G0MX03) and the contents of B are stored back in the Q Register (cell  $10g$ ). The basic timing of State 5 is the same as for all full operand store commands. State 5 completes the

execution of the GEN 3 command and State 1 is entered to "fetch" the next instruction.

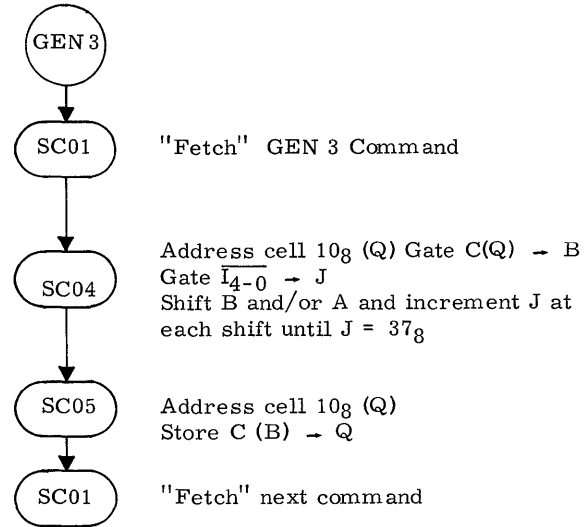


Fig. GN3. 2. Basic GEN 3 Flow Chart

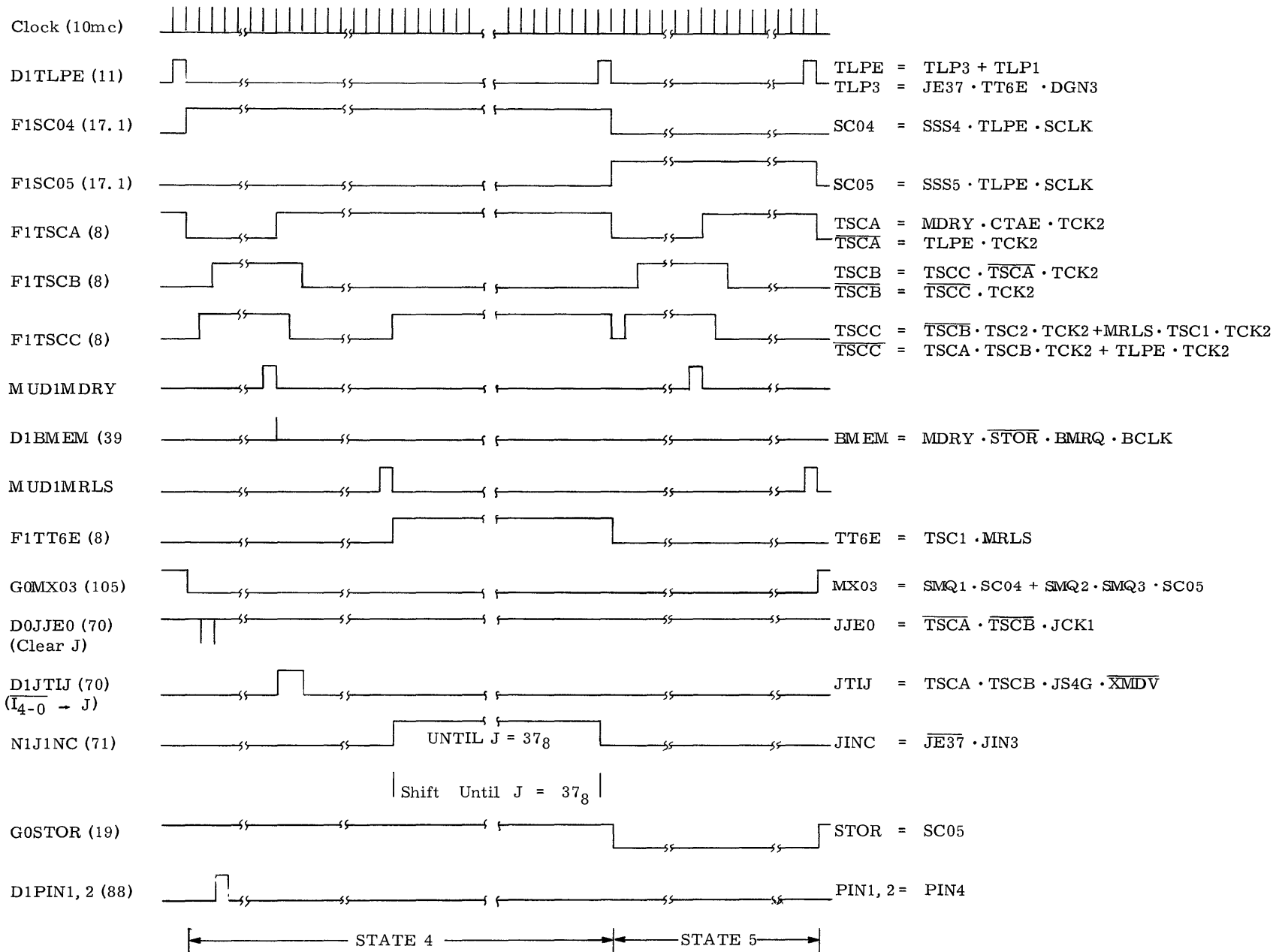
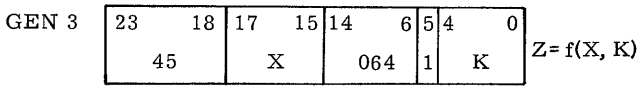


Fig. GN3.3. GEN 3 Basic Timing

## DLA-(SHIFT) DOUBLE LEFT ARITHMETIC

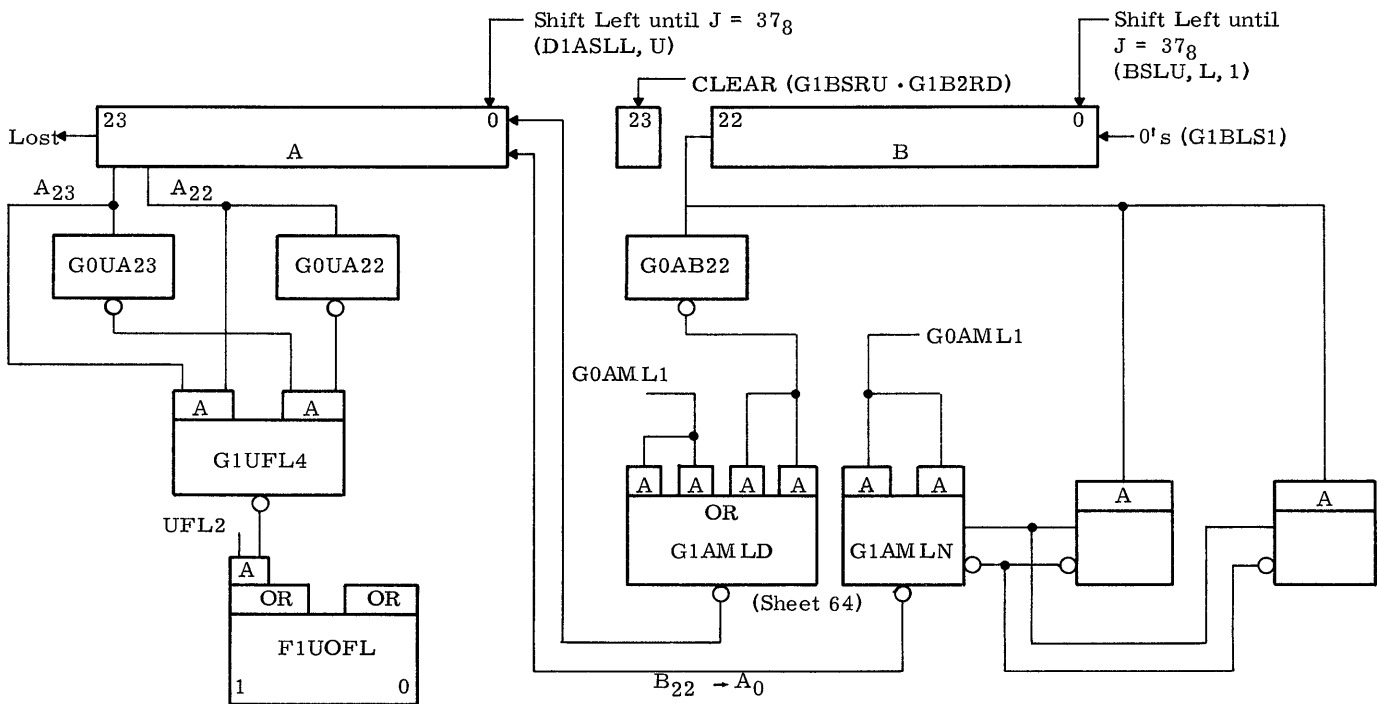


DLA shifts the contents of the A and Q Registers, Z places to the left. Bits shifted out of Q<sub>22</sub> enter A<sub>0</sub>. Bits shifted out of A<sub>23</sub> are lost. Bit Q<sub>23</sub> is cleared. If any of the bits shifted into A<sub>23</sub> are unlike A<sub>23</sub>'s original contents, the Overflow flip-flop (F1UOFL) is set.

During State 4, the contents of memory cell 10<sub>8</sub> are gated to the Arithmetic Unit B Register. The

complement of I<sub>4-0</sub> (Z) is gated to the J Counter. Both the A and B Registers are shifted left with bit 22 of the B Register applied to A<sub>0</sub> via G1AMLD and G1AMLN. "Zeros" are shifted into B<sub>0</sub> at each shift (G1BSL1). Bits shifted out of A<sub>23</sub> are lost. If B<sub>23</sub> is set, it is cleared by G1BSRU and G1B2RD. If at any shift of the A Register, bit 22 is not equal to bit 23, the Overflow flip-flop is set. This indicates that the sign bit of the A Register has been changed by the shift.

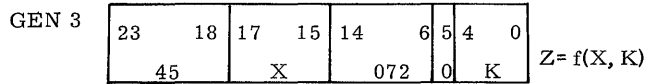
The J Counter is incremented at each shift of A and B and when J is equal to 37<sub>8</sub>, the number of shifts specified by Z have occurred and further shifting is inhibited. State 5 is then entered and the contents of B are stored in cell 10<sub>8</sub> (Q).



Logic Element	Logic Sheet	Logic Equation	Function
D1ASLL, U	63. 1	$G0ASL1 = AABL \cdot TT6E \cdot JE37$	Shift Left A <sub>23-0</sub>
D1BSLL, 1, U	40	$G0ASLB = TT6E \cdot IR11 \cdot JE37 \cdot AABL$	Shift Left B <sub>23-0</sub>
G1MALD G1MALN	64 64	$AB22 \cdot AML1$ $AML1 \cdot BR22$	B <sub>22</sub> → A <sub>0</sub>
G1BSRU G1B2RD	40 41	$B3RD \cdot BB23$ $B3RD \cdot BB23$	Clear B <sub>23</sub>
G1BSL1	38	$\overline{DDIV} \cdot BGN3$	0 → B <sub>0</sub>
G1UFL4 G1UFL2	54 54	$UA22 \cdot AR23 + UA23 \cdot AR22$ $ASL1 \cdot IR05 \cdot UGN3$	Set Overflow

DLA Block Diagram

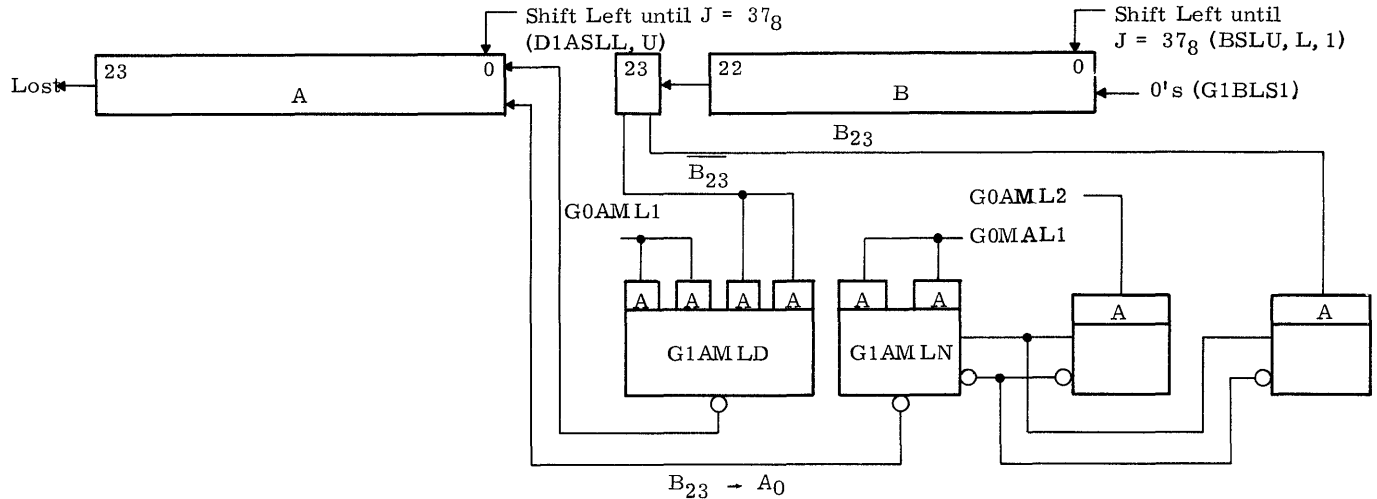
## DLL-(SHIFT) DOUBLE LEFT LOGICAL



DLL shifts the contents of the A and Q Registers Z places to the left. Bits shifted out of  $Q_{23}$  enter  $A_0$ . Bits shifted out of  $A_{23}$  are lost.

During State 4, the contents of memory cell  $10_8$  are

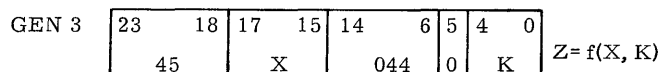
gated to the Arithmetic Unit B Register. The complement of  $I_{4-0}$  (Z) is gated to the J Counter. Both the A and B Registers are then shifted left. Bits shifted from  $B_{23}$  are applied to  $A_0$  by G1AMLN and G1AMALN. "Zeros" are shifted into  $B_0$  at each shift (G1BLS1). Bits shifted out of  $A_{23}$  are lost. The J Counter is incremented at each shift of A and B and when J is equal to  $37_8$ , the number of shifts specified by Z have occurred and further shifting is inhibited. State 5 is then entered and the contents of B are stored in memory cell  $10_8$  (Q).



Logic Element	Logic Sheet	Logic Equation	Function
D1ASLL, U	63.1	$G0ASL1 = AABL \cdot TT6E \cdot \overline{JE37}$	Shift Left $A_{23-0}$
D1BSLL, 1, U	40	$G0ASLB = TT6E \cdot IR11 \cdot \overline{JE37} \cdot AABL$	Shift Left $B_{23-0}$
G1BLS1	38	$\overline{DDIV} \cdot BGN3$	$0 \rightarrow B_0$
G1MALD	64	$\overline{AB23} \cdot AML2$	$B_{23} \rightarrow A_0$
G1MALN	64	$\overline{BR23} \cdot AML2$	
G1B2LD	38	$B22R \cdot BB22$	$B_{22} \rightarrow B_{23}$
G1B3LD	38	$BGN3 \cdot \overline{BR22}$	

DLL Block Diagram

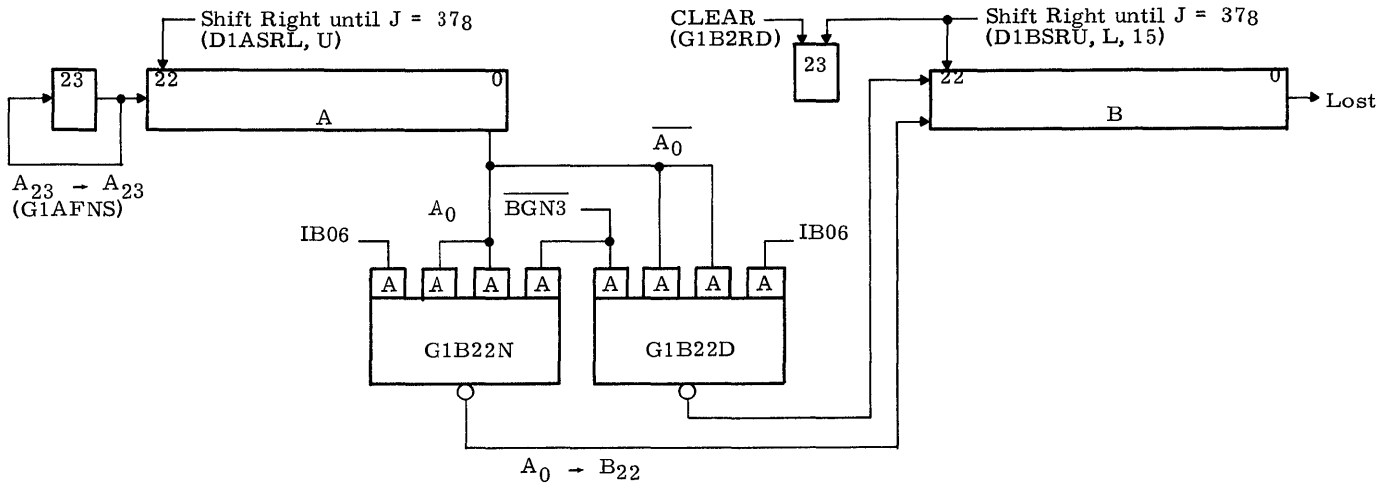
## DRA-(SHIFT) DOUBLE RIGHT ARITHMETIC



DRA Shifts the A and Q Registers Z places to the right. Bits shifted out of  $A_0$  are shifted into  $Q_{22}$ . Bits shifted out of  $Q_0$  are lost. Bits shifted into  $A_{22}$  are the same as  $A_{23}$ .  $A_{23}$  remains unchanged and  $Q_{23}$  is cleared.

During State 4, the contents of memory cell  $10_8$  are

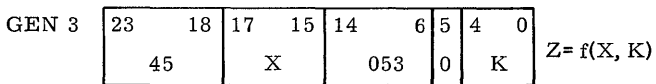
gated to the Arithmetic Unit B Register. The complement of  $I_{4-0}$  (Z) is gated to the J Counter. Both the A and B Registers are then shifted right. Bits shifted from  $A_0$  are shifted into  $B_{22}$  via G1B22N and G1B22D.  $A_{23}$  is shifted to  $A_{22}$  and through the Serial Full Adder back to  $A_{23}$  leaving  $A_{23}$  unchanged.  $B_{23}$  is cleared by G1B2RD. Bits shifted from  $B_0$  are lost. The J Counter is incremented at each shift of A and B. When the J Counter is incremented to  $37_8$ , the number of shifts specified by Z have occurred and further shifting is inhibited. State 5 is then entered and the contents of B are stored in cell  $10_8$  (Q) to complete execution of the DRA command.



Logic Element	Logic Sheet	Logic Equation	Function
D1ASRL, U	63.1	$G0ASR2 = AABR \cdot T6E3 \cdot \overline{JE37}$	Shift Right $A_{23-0}$
D1BSRU, L, 15	40	$G0ASRB = AABR \cdot \overline{JE37} \cdot IR11 \cdot TT6E$	Shift Right $B_{23-0}$
G1B22D G1B22N	38 38	$\overline{BI06} \cdot AR00 \cdot BGN3$ $\overline{BI06} \cdot BGN3 \cdot \overline{BA00}$	$A_0 \rightarrow B_{22}$
G1AFNS N0AFNS	67 67	$G0AFNA = ANA3 \cdot AR23$ G1AFNS	$A_{23} \rightarrow A_{23}$
G1B2RD	41	$B3RD \cdot BB23$	Clear $B_{23}$

DRA Block Diagram

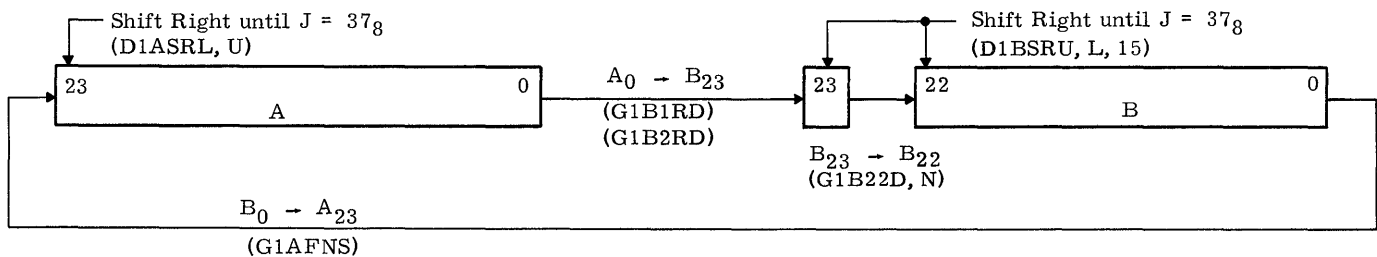
### DRC-(SHIFT) DOUBLE RIGHT CIRCULAR



DRC shifts all 48 bits of the A and Q Registers Z places in a right circular fashion with  $Q_0$  shifted into  $A_{23}$  and  $A_0$  shifted to  $Q_{23}$ .

During State 4, the contents of memory cell  $10_8$  (Q)

are gated to the B Register. The complement of  $I_{4-0}$  (Z) is gated to the J Counter. The A and B Registers are then shifted right with  $B_0$  applied through the Serial Full Adder to  $A_{23}$ .  $A_0$  is shifted through G1B1RD and G1B2RD to  $B_{23}$ . The J Counter is incremented at each shift of A and B. When the J Counter is equal to  $37_8$ , the number of shifts specified by Z have occurred and further shifting is inhibited. State 5 is then entered and the contents of the B Register are stored in memory location  $10_8$  (Q) to complete execution of the DRC command.



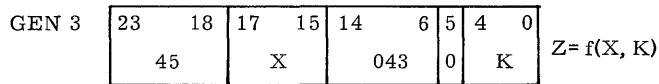
DRC Block Diagram



Logic Element	Logic Sheet	Logic Equation	Function
D1ASRL, U	63.1	$G0ASR2 = AABR \cdot T6E3 \cdot \overline{JE37}$	Shift Right $A_{23-0}$
D1BSRU, L, 15	40	$G0ASRB = AABR \cdot \overline{JE37} \cdot IR11 \cdot TT6E$	Shift Right $B_{23-0}$
G1B1RD G1B2RD	41 41	$AR00 \cdot B22R$ $B22R \cdot \overline{BA00}$	$A_0 \rightarrow B_{23}$
G1B22D G1B22N	38 38	$BB23 \cdot \overline{DMPY} \cdot BGN3 \cdot IR06$ $\overline{BR23} \cdot BGN3 \cdot \overline{GMPY} \cdot IR06$	$B_{23} \rightarrow B_{22}$
G1AFNS N0AFNS	67 67	$G0AFNA = DGN3 \cdot BR00 \cdot IR09$ G1AFNS	$B_0 \rightarrow A_{23}$

DRC Block Diagram

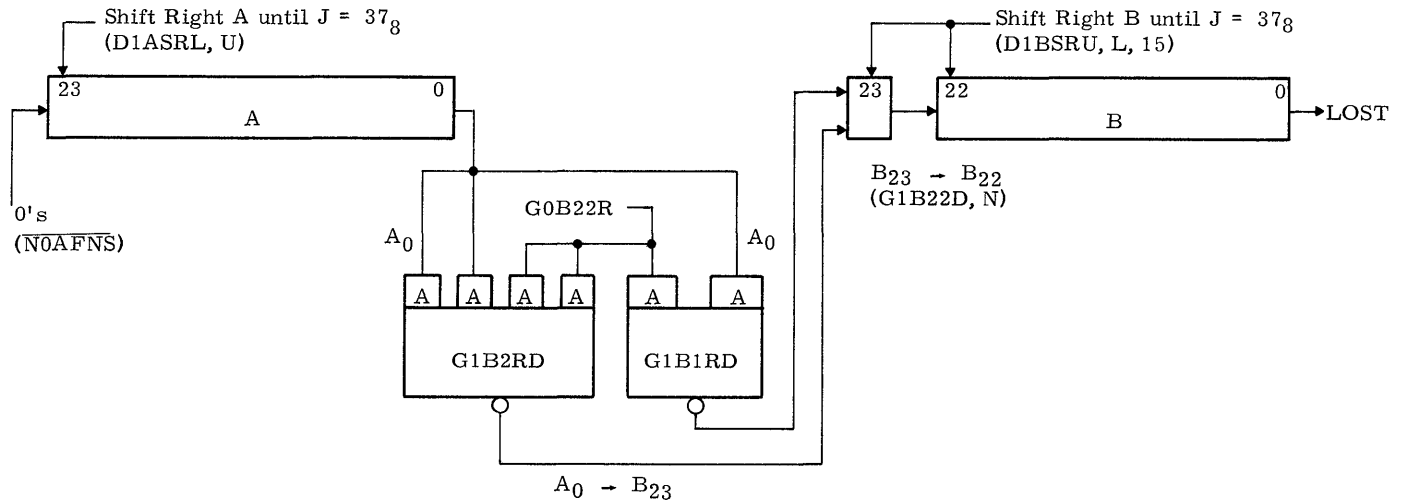
### DRL-(SHIFT) DOUBLE RIGHT LOGICAL



DRL shifts all 48 bits of the A and Q Registers Z places to the right.  $A_0$  is shifted into  $Q_{23}$ . Bits shifted out of  $Q_0$  are lost. Zeros are shifted into  $A_{23}$ .

During State 4, the contents of memory cell  $10_8$  (Q)

are gated to the B Register. The complement of  $I_{4-0}$  (Z) is gated to the J Counter. The contents of both the A and B Registers are then shifted right with the bits shifted from  $A_0$  applied to  $B_{23}$  via G1B1RD and G1B2RD. "Zeros" are shifted into  $A_{23}$  by inhibiting the Serial Full Adder. Bits shifted from  $B_0$  are lost. The J Counter is incremented at each shift of the A and B Registers. When the J Counter is equal to  $37_8$ , the number of shifts specified by Z have occurred and further shifting is inhibited. Sequence State 5 is then entered and the contents of the B Register are stored in memory cell  $10_8$  (Q) to complete the execution of the DRL command.

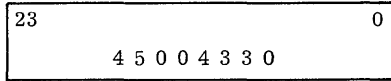


Logic Element	Logic Sheet	Logic Equation	Function
D1ASRL, U	63.1	$G0ASR2 = AABR \cdot T6E3 \cdot \overline{JE37}$	Shift Right $A_{23-0}$
D1BSRU, L, 15	40	$G0ASRB = AABR \cdot \overline{JE37} \cdot IR11 \cdot TT6E$	Shift Right $B_{23-0}$
G1B1RD G1B2RD	41 41	$AR00 \cdot B22R$ $B22R \cdot \overline{BA00}$	$A_0 \rightarrow B_{23}$
G1B22D G1B22N	38 38	$BB23 \cdot \overline{DMPY} \cdot BGN3 \cdot IR06$ $\overline{BR23} \cdot BGN3 \cdot \overline{DMPY} \cdot IR06$	$B_{23} \rightarrow B_{22}$

DRL Block Diagram

## MAQ-MOVE A TO Q

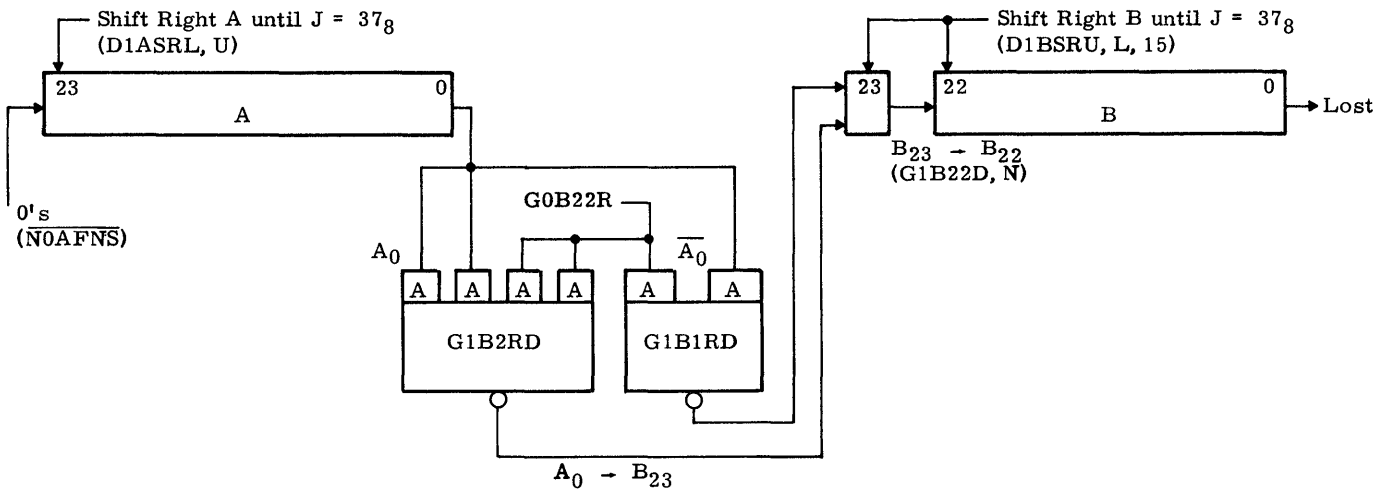
GEN 3



MAQ places the contents of the A Register into the Q Register. The original contents of Q are lost. Zeros are placed in the A Register.

During State 4, the contents of memory location 10g (Q) are gated to the B Register. The complement of  $I_{4-0}$  (bits 4 through 0 of the MAQ command) are gated to the

J Counter. Since bits 2 through 0 of the MAQ command are zero, this presets the J Counter equal to 7. The A and B Registers are then shifted right. Bits shifted from  $A_0$  are gated through G1B1RD and G1B2RD to  $B_{23}$ . Bits shifted from  $B_0$  are lost. "Zeros" are shifted to  $A_{23}$  from the disabled Serial Full Adder. The J Counter was preset to 7, 24 shifts will have occurred when the J Counter is equal to 37g and further shifting is inhibited. After 24 shifts, the contents of A have been shifted to B, "zeros" have been shifted to all 24 bits of A, and the original contents of B (Q) are lost. State 5 is then entered and the contents of B are stored in cell 10g (Q) to complete the MAQ command.

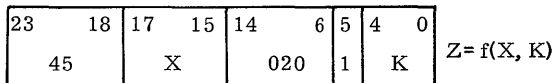


Logic Element	Logic Sheet	Logic Equation	Function
D1ASRL, U	63.1	$G0ASR2 = AABR \cdot T6E3 \cdot \overline{JE37}$	Shift Right $A_{23-0}$
D1BSRU, L, 15	40	$G0ASRB = AABR \cdot \overline{JE37} \cdot IR11 \cdot TT6E$	Shift Right $B_{23-0}$
G1B1RD G1B2RD	41 41	$AR00 \cdot B22R$ $B22R \cdot \overline{BA00}$	$A_0 \rightarrow B_{23}$
G1B22D G1B22N	38 38	$BB23 \cdot \overline{DMPY} \cdot BGN3 \cdot IR06$ $\overline{BR23} \cdot BGN3 \cdot \overline{DMPY} \cdot IR06$	$B_{23} \rightarrow B_{22}$

MAQ Block Diagram

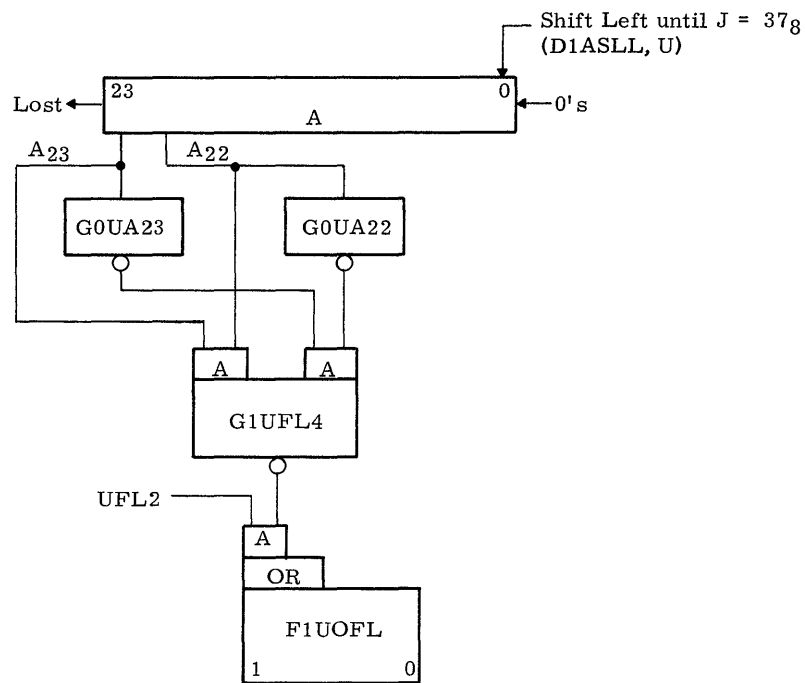
## SLA-SHIFT LEFT ARITHMETIC

GEN 3



SLA shifts the contents of the A Register Z places to the left. Bits shifted out of  $A_{23}$  are lost. Zeros are shifted into  $A_0$ . The Overflow flip-flop (F1UOFL) is set if any bit shifted into  $A_{23}$  is unlike the original content of  $A_{23}$ .

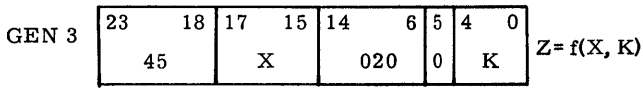
During State 4, the contents of memory location 10g (Q) are gated to the B Register. The complement of  $I_{4-0}$  (Z) is gated to the J Counter. The A Register is then shifted left with "zeros" shifted into  $A_0$  (G1AMLN). Bits shifted out of  $A_{23}$  are lost. If at any shift of A, bit 23 changes ( $A_{23} \neq A_{22}$ ) the Overflow flip-flop is set. At each shift of A, the J Counter is incremented. When the J Counter is equal to 37g, further shifting of A is inhibited. State 5 is then entered and the contents of B (original contents of Q) are gated back to cell 10g unchanged. This completes execution of the SLA command.



Logic Element	Logic Sheet	Logic Equation	Function
D1ASLL, U	63. 1	$G0ASL1 = AABL \cdot TT6E \cdot \overline{JE37}$	Shift Left $A_{23-0}$
G1AMLN	64	$G0AML3 = DGN3 \cdot \overline{WI08} \cdot \overline{AIR9}$	$0 \rightarrow A_0$
G1UFL4	54	$UA22 \cdot \overline{AR23} + UA23 \cdot \overline{AR22}$	Set Overflow
G1UFL2	54	$ASL1 \cdot IR05 \cdot UGN3$	

SLA Block Diagram

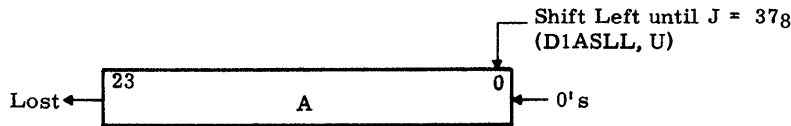
## SLL-SHIFT LEFT LOGICAL



SLL shifts the contents of the A Register Z places to the left. Bits shifted out of  $A_{23}$  are lost. Zeros are shifted into  $A_0$ .

During State 4, the contents of memory location  $10_8$  (Q)

are gated to the B Register although this data is not changed or used during the SLL command. The complement of  $I_{4-0}$  (Z) is gated to the J Counter. The A Register is then shifted left. "Zeros" are shifted into  $A_0$  from G1MALN. Bits shifted out of  $A_{23}$  are lost. The J Counter is incremented at each shift of A. When the J Counter is equal to  $37_8$ , the A Register has been shifted left the number of places specified by Z and further shifting is inhibited. State 5 is then entered and the contents of B are stored back in cell  $10_8$  (Q). The contents of B (Q) was not changed by the SLL command.



Logic Element	Logic Sheet	Logic Equation	Function
D1ASLL, U	63.1	$G0ASL1 = AABL \cdot TT6E \cdot \overline{JE37}$	Shift Left $A_{23-0}$
G1AMLN	64	$G0AML3 = DGN3 \cdot \overline{WI08} \cdot \overline{AIR9}$	$0 \rightarrow A_0$

SLL Block Diagram

## INDEXING ADDRESS MODIFICATION

The GE-PAC instruction format allocates three bits (15, 16, and 17) to specify Indexing Address Modification. When bits 15, 16, and 17 are not equal to zero, the 14 bits of the instruction operand (0 through 13) are added to the least significant 15 bits of the specified index register contents. The memory location (1 through 7) of the index register is specified by bits 15 through 17. The result of this addition allows the instruction to address one of  $32,768_{10}$  memory locations.

To illustrate the effect of indexing address modification, consider the following examples.

Example 1:

P =  $1000_8$   
Location  $1000_8$  =  $00500100_8$   
Index Reg. 5 =  $00000040_8$   
Following execution of the command (LDA) in location  $1000_8$ ;  
A Register = Contents of location  $140_8$   
Location  $1000_8$  =  $00500100_8$   
Index Reg. 5 =  $00000040_8$ .

Example 2:

P =  $2000_8$   
Location  $2000_8$  =  $32700100_8$   
Index Reg. 7 =  $00040000_8$   
Following execution of the command (STA) in location  $2000_8$ ;  
Location  $40100_8$  = Contents of the A Register  
Location  $2000_8$  =  $32700100_8$   
Index Reg. 7 =  $00040000_8$ .

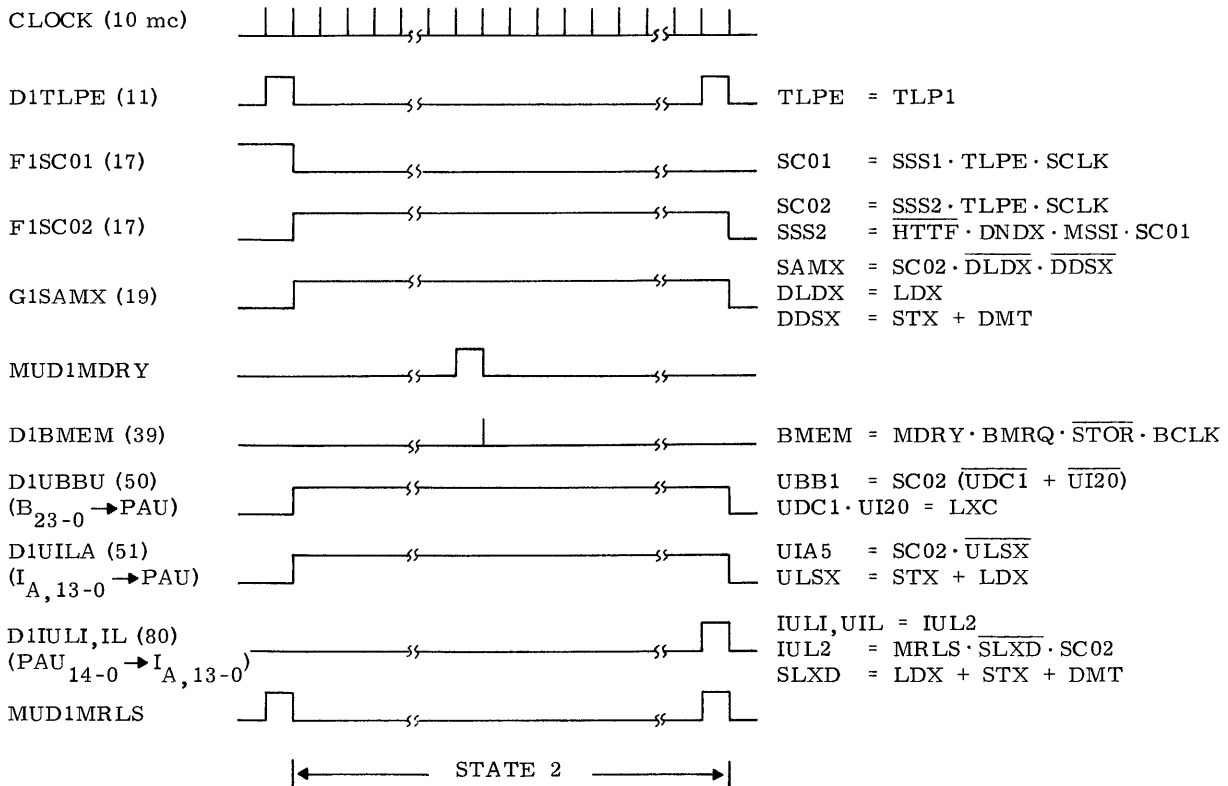
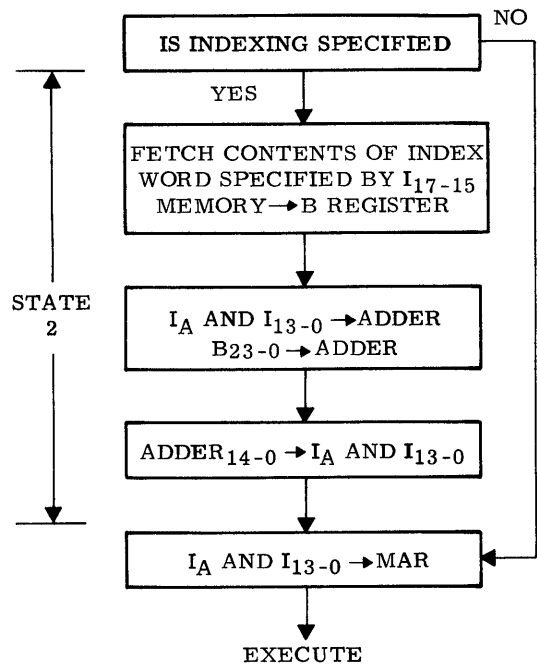
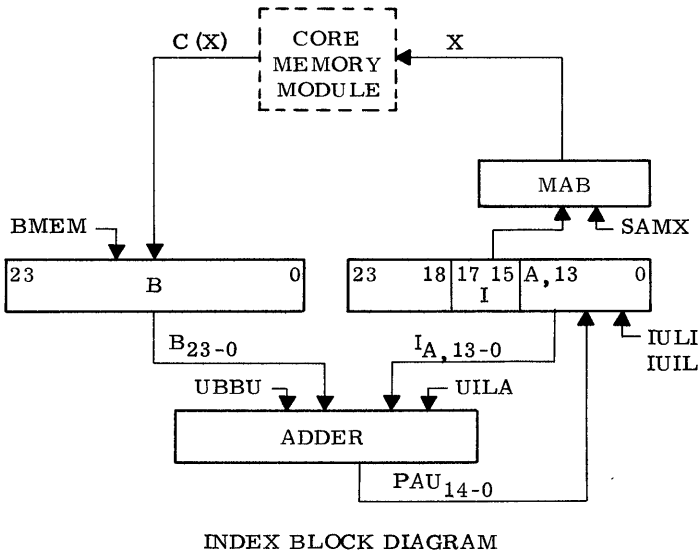
Some commands use the index bits (15, 16, and 17) for instruction control rather than for index address modification. These commands are:

INX  
LDX  
LXC  
LXX  
STX  
TXH.

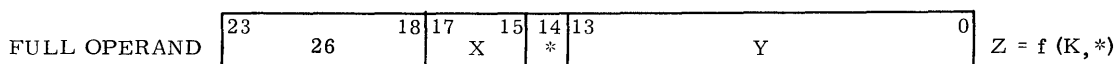
Refer to the command description for the use of the index bits for these commands.

Indexing of GEN 1, GEN 2, and GEN 3 commands may change the intelligence of the microcoding. Therefore, caution must be exercised when specifying index address modification of these commands.

Indexing of any command requires the use of Sequence Control State 2. The indexed command is "fetched" during a normal Sequence Control State 1. Bits 15, 16, and 17 not being equal to zero causes sequencing to State 2. During State 2, memory is addressed from bits 15, 16, and 17 (G1SAMX) and the contents of the addressed index register are gated to the B Register. From B, all 24 bits of the addressed index register are gated to the Adder (D1UBBU). At the same time,  $I_A, 13-0$  is gated to the Adder where summation with the index register contents occurs. The result of this summation (bits 14 through 0) is gated back to  $I_A, 13-0$  (D1IULI, IL), to complete the indexing address modification operation. The next Sequence State would then be entered and memory addressed from the I Register.



## INX - INCREMENT X



INX adds the numeric value of Z to the contents of the X core cell specified by bits 17 through 15 of the command. The value of K must have a value within the range of +8, 191 to -8, 192. As the INX instruction is performed, K<sub>13</sub> is extended into bit 14 (i. e. , if K<sub>13</sub> = 0, bit 14 is set to 0 and represents a positive incrementation; if K<sub>13</sub> = 1, bit 14 is set to 1 and represents a negative incrementation). Then, K<sub>14-0</sub> is added to the contents of X<sub>23-0</sub> and stored back in X. If the INX command is relative addressed, K<sub>14-0</sub> plus P<sub>14-0</sub> are added to X<sub>23-0</sub> and stored back in X.

**NOTE**

An INX command specifying to increment X cell 2 by zero (26200000) is NOP - No Operation. If the index field of the INX command (17 through 15) is equal to zero, the instruction is undefined.

Sequence Control State 1 operates as a normal "fetch" cycle for the INX command except that bit 13 of the INX command is applied to both bits 13 and 14 of the Adder Unit and the result transferred to IA<sub>13-0</sub>. In this manner, a negative value of K in 2's complement form will be extended from 14 bits (13 through 0) to 15 bits in the I Register (IA<sub>13-0</sub>).

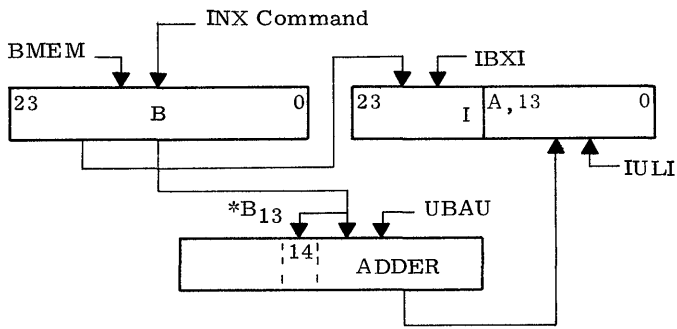
Following State 1, State 2 is entered. Memory is addressed from the X bits of the I Register (17 through 15) and the contents of the addressed X cell are gated to the B Register. From B, the contents of the addressed

Non-Indexed Word Times.	3 (S1, S2, S4)
Interruptable Following Execution?	Yes
<b>CHANGES FOLLOWING EXECUTION</b>	
A <sub>23-0</sub>	
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C (P) + 1
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	
Memory X	C (X <sub>23-0</sub> ) + Z

### COMMAND CHARACTERISTICS

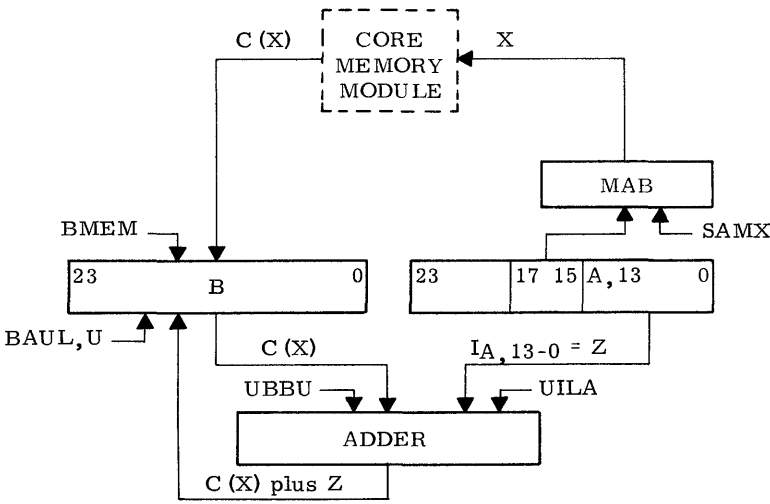
X cell are gated to the Adder Unit (D1UBBU). At the same time, the contents of IA<sub>13-0</sub> (Z) are gated to the Adder Unit (D1UILA). The sum of these two adder inputs is then gated back to the B Register at the end of State 2.

Following State 2, State 4 is entered. During State 4, memory is again addressed from the X bits of the I Register and the contents of the B Register (original contents of the X cell plus Z) are gated back to the X cell.

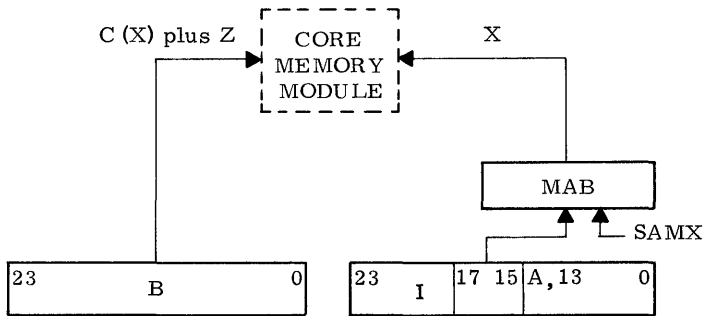


\*  $G1UB14(49) = UINX \cdot USIA \cdot BR13$

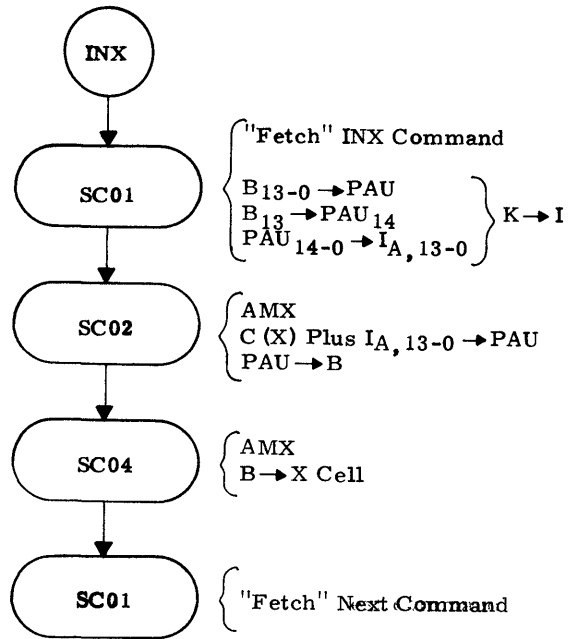
Sequence State 1



Sequence State 2

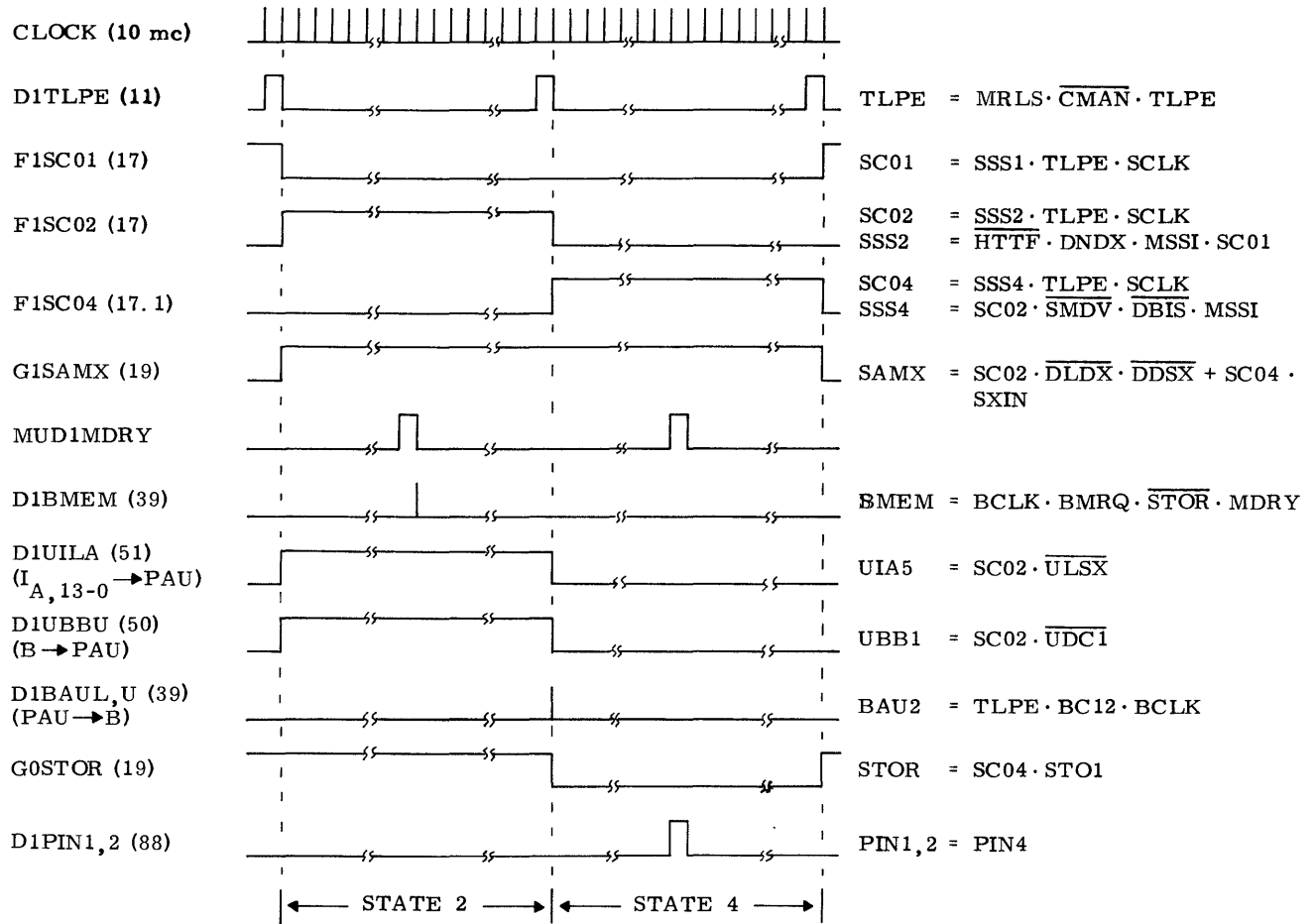


Sequence State 4



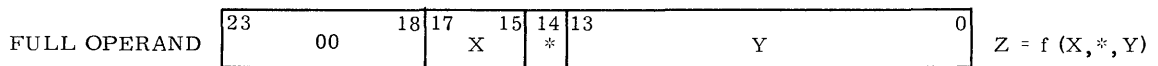
INX BLOCK DIAGRAM





INX TIMING DIAGRAM

# LDA - LOAD THE A REGISTER



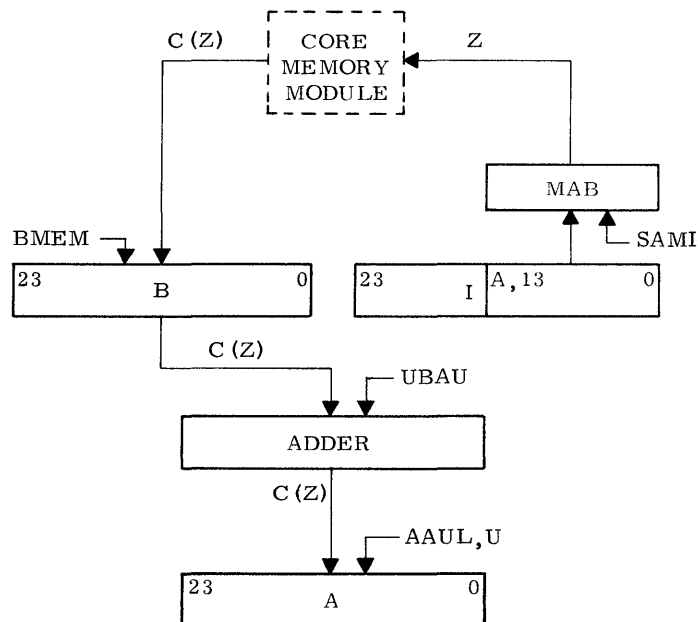
LDA places the contents of memory location Z into the A Register. The contents of memory location Z are unchanged.

A non-indexed LDA command is executed during Sequence Control State 4 (SC04). Memory location Z is addressed from  $I_A, 13-0$  (D1SAMI) during State 4. The contents of memory location Z are gated to the B Register by D1BMEM during the Clock Pulse of Memory Data Ready (MUD1MDRY).

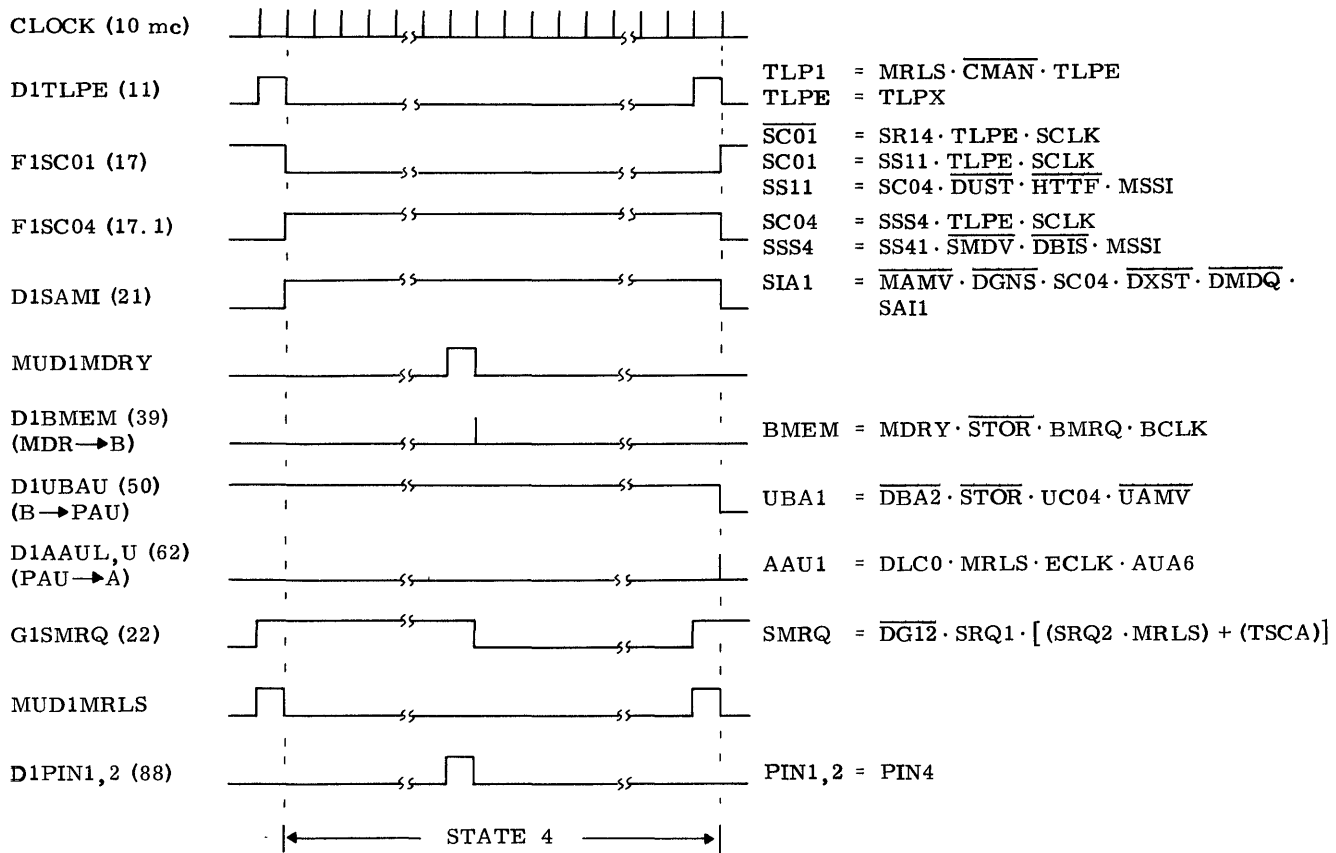
From B, the contents of memory location Z are gated to the Adder Unit (D1UBAU). At the Clock of Last Pulse, the contents of the Adder Unit are gated to the A Register (D1AAUL,U), completing execution of the LDA command. Sequence Control State 1 is then entered to "fetch" the next command.

Non-Indexed Word Times.	2 (S1, S4)
Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	C (Z)
$Q_{23-0}$	
$P_{14-0}$	C (P) + 1
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	
Memory Z	

COMMAND CHARACTERISTICS

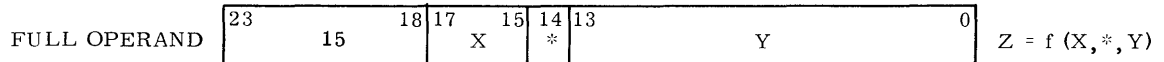


LDA BLOCK DIAGRAM



LDA TIMING DIAGRAM

# LDP - LOAD PLACE



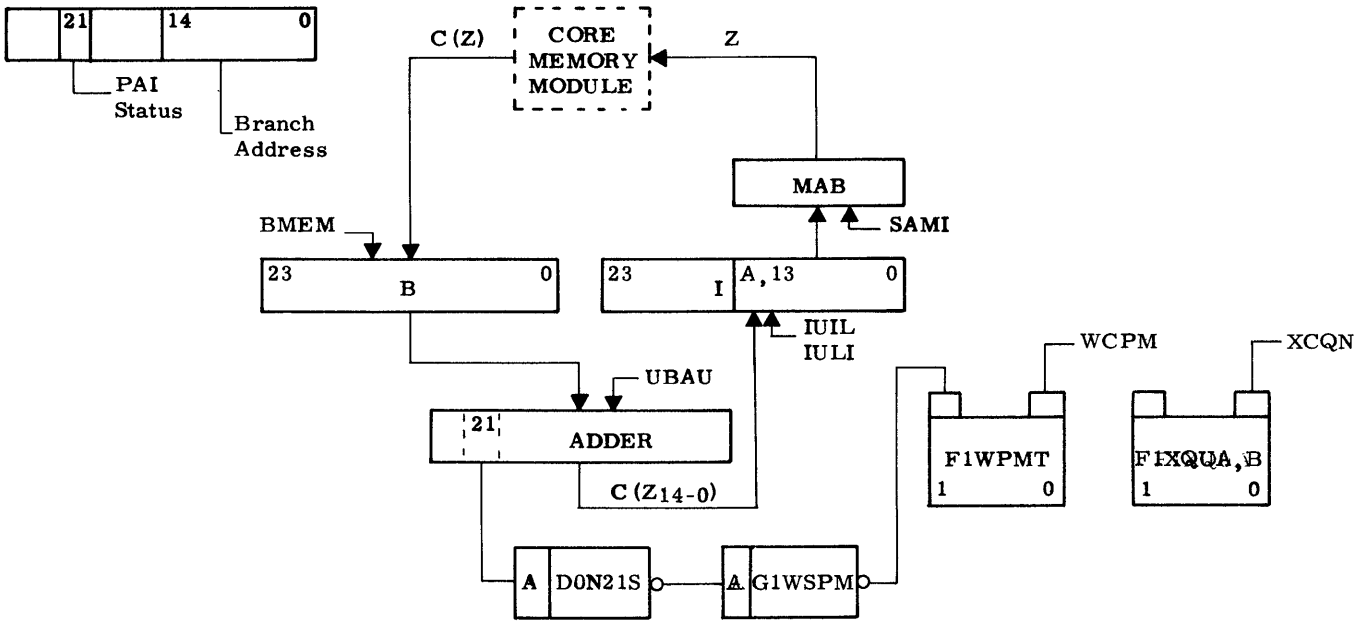
LDP transfers program control to the location specified by the contents of Z<sub>14-0</sub>. Bit 21 of Z controls the status of the Permit Automatic Interrupt flip-flop (F1WPMT); if bit 21 is a "one", F1WPMT is set; if bit 21 is a "zero", F1WPMT is cleared. The LDP command unconditionally clears the Quasi flip-flop.

The LDP command is "fetched" during a normal Sequence Control State 1. At the last pulse of this State 1, the Remember flip-flop (F1XRMF) is set. Following this State 1, memory is addressed from I<sub>A</sub>,<sub>13-0</sub> and the contents of core cell Z are gated to the B Register. From B, the contents of core cell Z are gated to the Adder Unit (D1UBAU). Bits 14-0 are then gated to I<sub>A</sub>,<sub>13-0</sub> at memory release (D1IULI,UIL). At Time 3, the Permit Automatic Interrupt flip-flop is cleared, and at the Clock of Last Pulse Envelope it is set if bit 21 in the Adder (i.e., bit 21 of Z) is a "one". Also, at Last Pulse Envelope the Quasi flip-flops (F1XQUA,B) are reset.

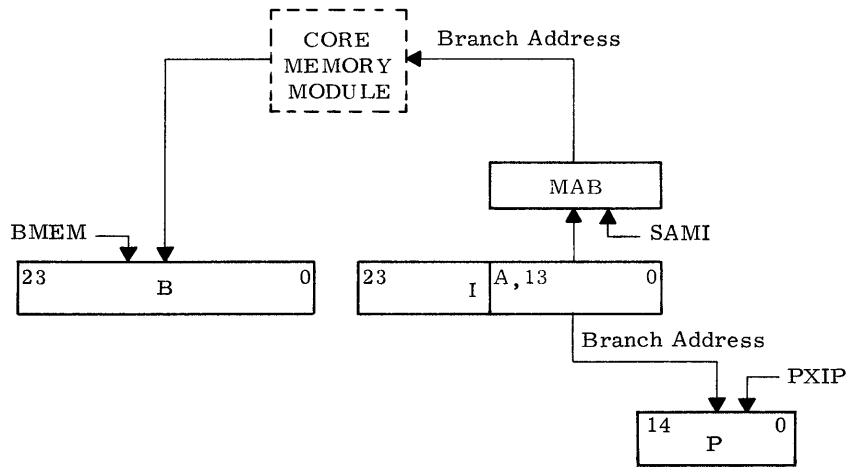
Following this State 4, State 1 is entered to "fetch" the next command. Memory is addressed from I<sub>A</sub>,<sub>13-0</sub> and the contents of I<sub>A</sub>,<sub>13-0</sub> are transferred to the P Register to complete the transfer of program control. This State 1 then operates in the normal manner except that at the Clock of Last Pulse Envelope, the Remember flip-flop is cleared. Sequencing then continues to execute this command.

Non-Indexed Word Times.	2 (S1,S4)
Interruptable Following Execution?	No
<b>CHANGES FOLLOWING EXECUTION</b>	
A <sub>23-0</sub>	
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C (Z <sub>14-0</sub> )
F1WPMT	Set if Z <sub>21</sub> = 1 Reset if Z <sub>21</sub> = 0
F1UOFL	
F1ETST	
J <sub>4-0</sub>	
Memory Z	
F1XQUA, B	Reset

COMMAND CHARACTERISTICS

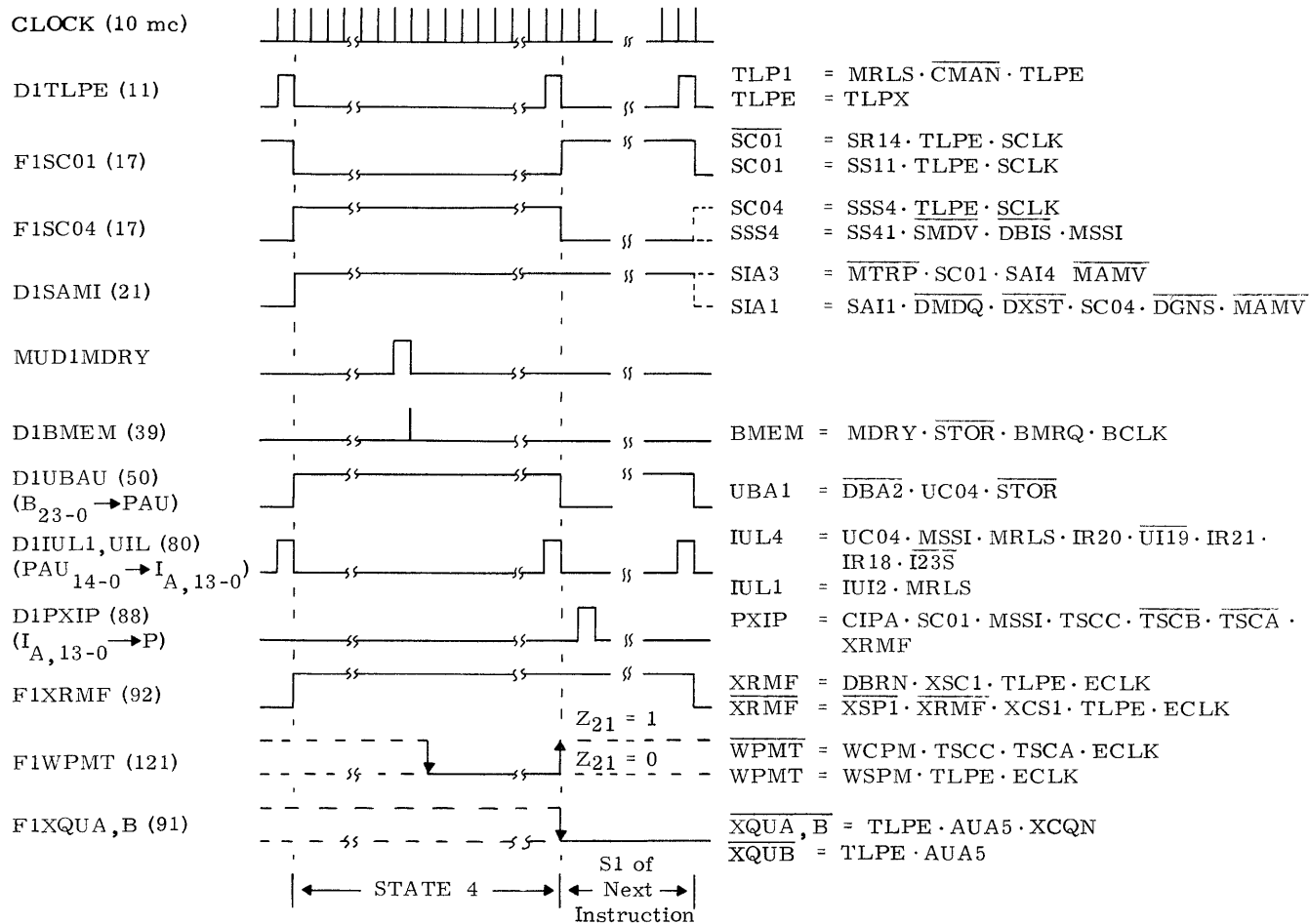


Sequence State 4



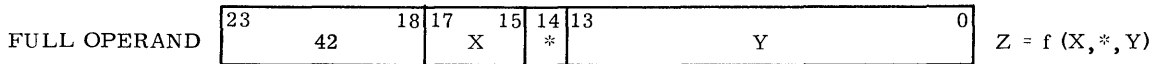
Next Sequence State 1

LDP BLOCK DIAGRAM



LDP TIMING DIAGRAM

# LDQ - LOAD THE Q REGISTER



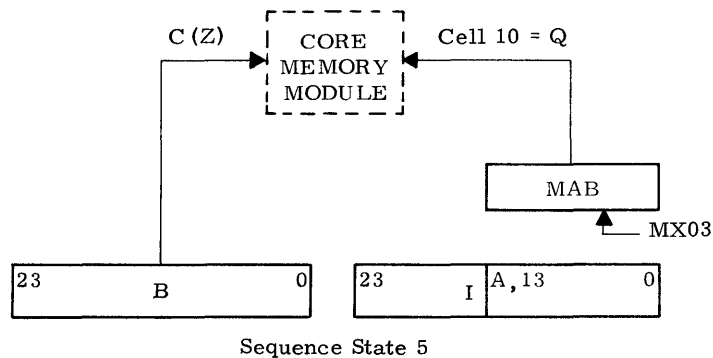
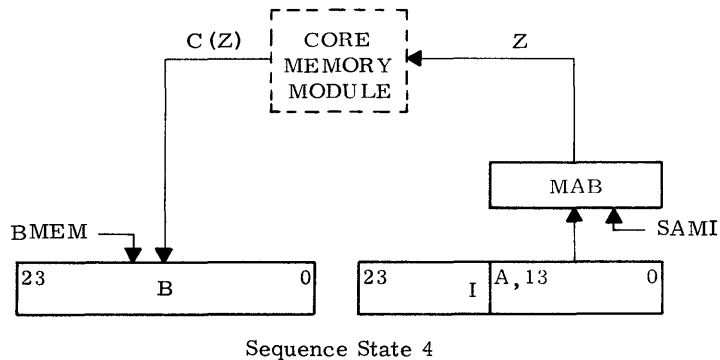
LDQ places the contents of memory location Z into the Q Register (memory location 10g). The contents of memory location Z are unchanged by the LDQ command.

The LDQ command is executed during Sequence Control States 4 and 5. During State 4, memory is addressed from I<sub>A, 13-0</sub> (SAMI) and the contents of memory location Z are gated to the B Register.

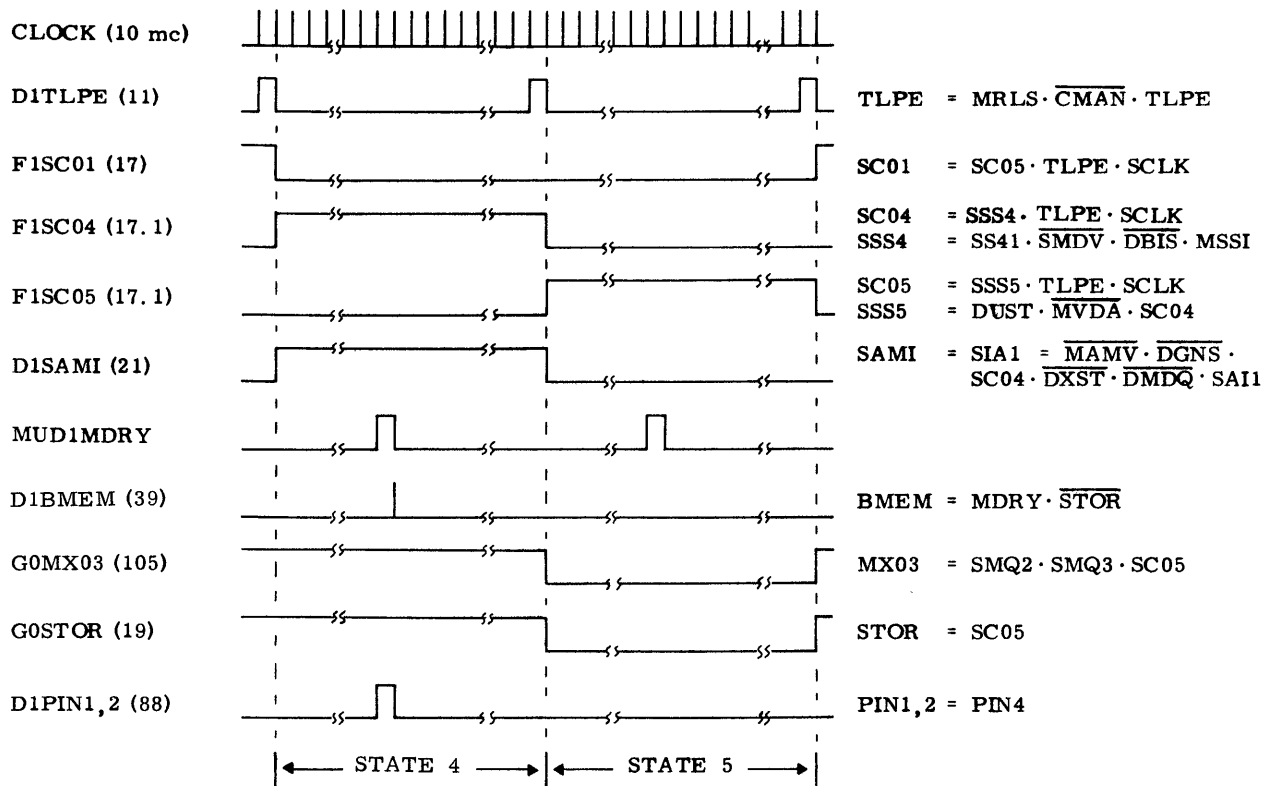
Following State 4, State 5 is initiated. During State 5, memory location 10g is addressed by enabling G0MX03, and the contents of the B Register are stored in cell 10g. In this manner, the contents of memory location Z are stored in the Q Register (memory cell 10g).

Non-Indexed Word Times.	3 (S1, S4, S5)
Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
A <sub>23-0</sub>	
Q <sub>23-0</sub>	C (Z)
P <sub>14-0</sub>	C (P) +1
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	
Memory Z	

## COMMAND CHARACTERISTICS



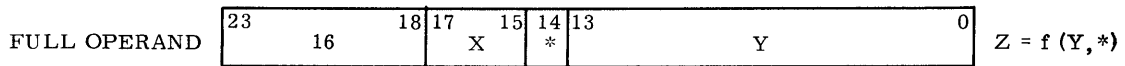
LDQ BLOCK DIAGRAM



LDQ TIMING DIAGRAM



# LDX - LOAD X LOCATION FROM Z

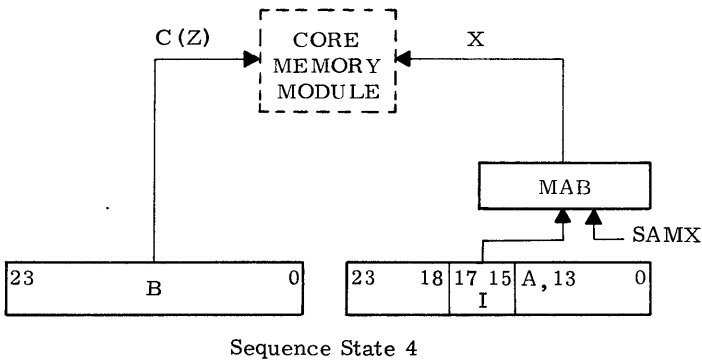
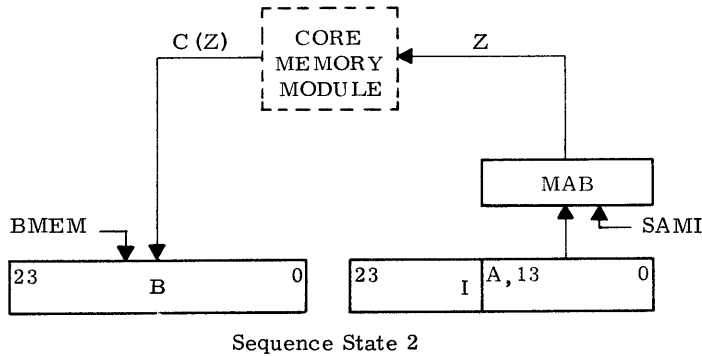


LDX places the contents of memory location Z into the specified (bits 17 through 15) index core cell. If the index field (bits 17 through 15) of the LDX command are zero, the command is undefined.

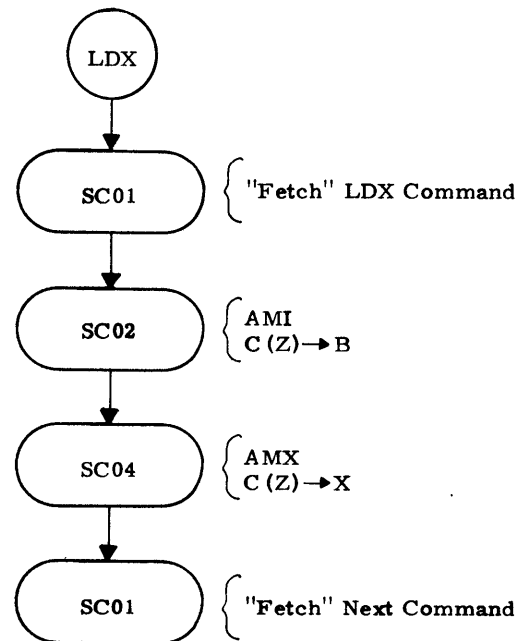
The LDX command is executed during Sequence Control States 2 and 4. During State 2, memory is addressed from  $I_{A,13-0}$  (D1SAMI) and the contents of memory location Z are gated to the B Register. Following State 2, State 4 is entered. During State 4, memory is addressed from  $I_{17-15}$  (G1SAMX) and the contents of the B Register are stored in the index core cell specified by bits 17 through 15 of the LDX command.

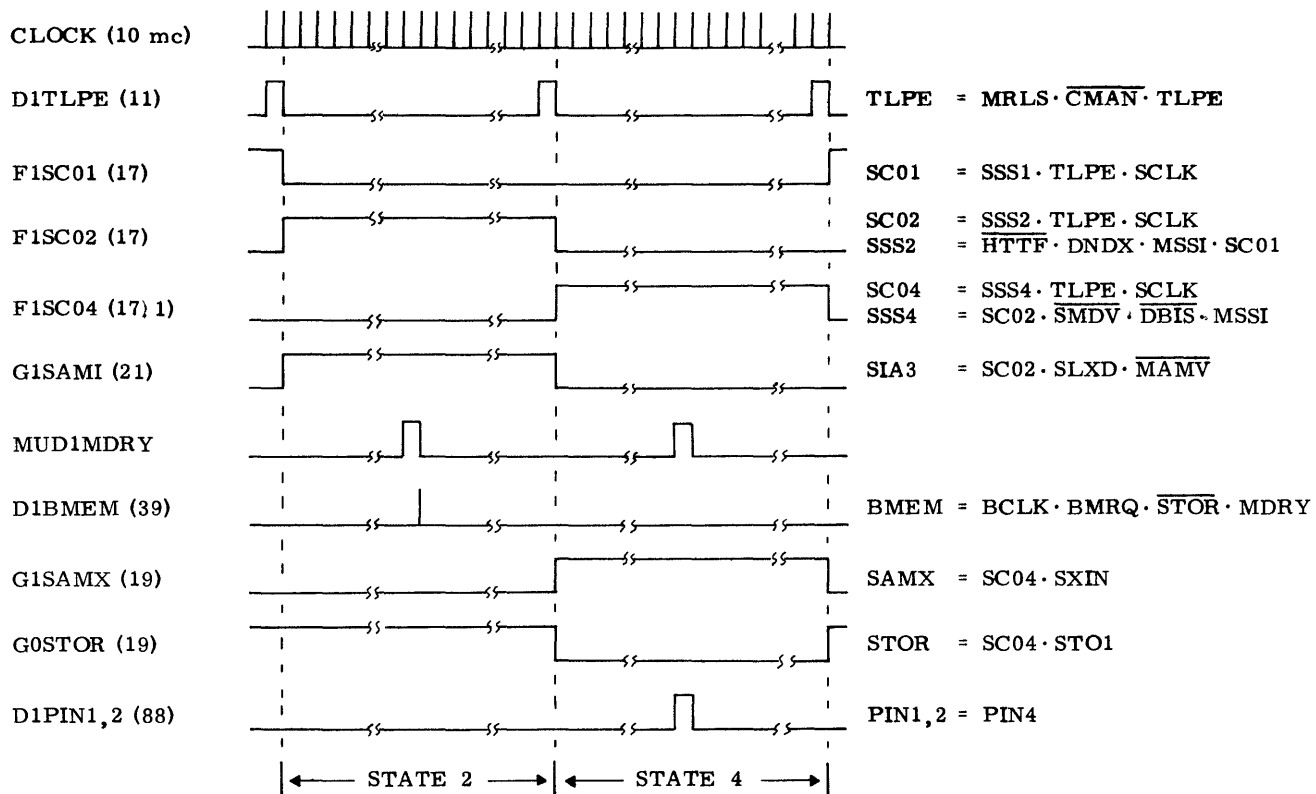
Non-Indexed Word Times.	3 (S1, S2, S4)
Interruptable Following Execution?	No
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	C (P) + 1
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	
Memory X	C (Z)

COMMAND CHARACTERISTICS



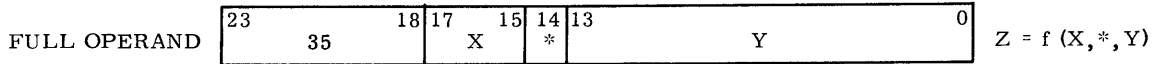
LDX BLOCK DIAGRAM





LDX TIMING DIAGRAM

## LPR - LOAD PLACE AND RESTORE



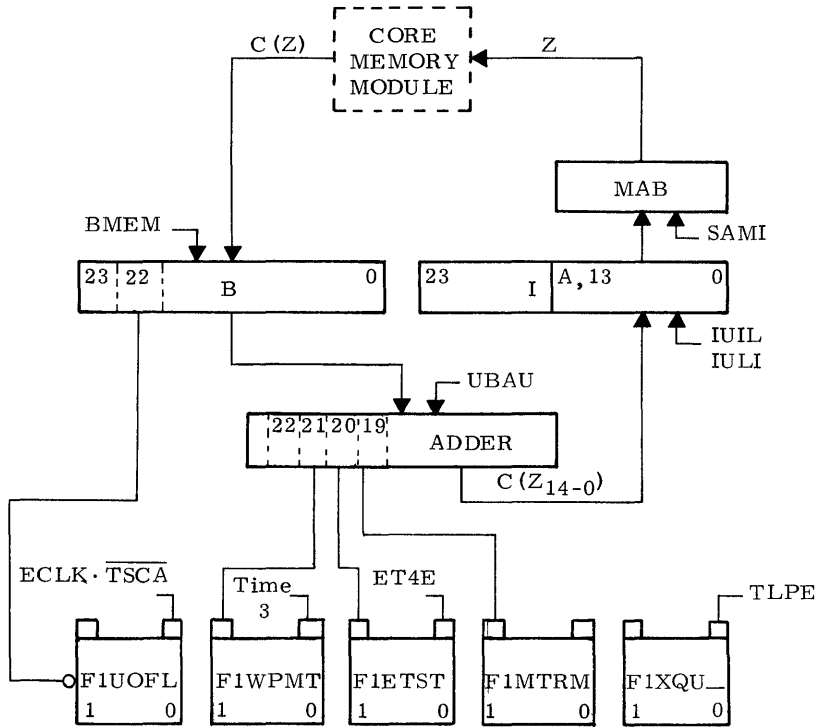
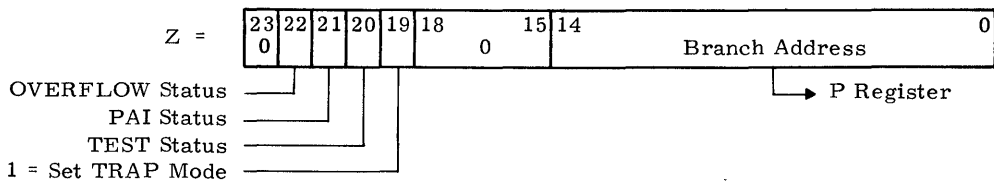
LPR restores the status of the Overflow flip-flop (F1UOFL), Permit Automatic Interrupt flip-flop (F1WPMT), and Test flip-flop (F1ETST) from the contents of bits 22, 21, and 20, respectively, of core location Z. The contents of  $Z_{14-0}$  are placed in the P Register to transfer program control. If bit 19 of Z is a "one", the Trapping Mode flip-flop (F1MTRM) is set. If bit 19 of Z is a "zero", the status of the Trapping Mode flip-flop is unchanged. The LPR command unconditionally resets the Quasi flip-flops (F1XQUA, B).

The LPR command is "fetched" during a normal Sequence Control State 1 (SC01). At Last Pulse Envelope of State 1, the Remember flip-flop (F1XRMF) is set. Following State 1, State 4 is entered and memory is addressed from  $I_{A,13-0}$ . The contents of memory location Z are gated from memory to B and from B to the Adder Unit. The Overflow, Permit Automatic Interrupt, Test, and Quasi flip-flops are cleared during State 4. The Overflow flip-flop is then set, if bit 22 of the Adder is a "one"; the Permit Automatic Interrupt flip-flop is set if Adder bit 21 is a "one"; the Test flip-flop is set if Adder bit 20 is a "one"; and the Trapping Mode flip-flop is set if Adder bit 19 is a "one". The contents of Adder bits 14-0 are transferred to  $I_{A,13-0}$ .

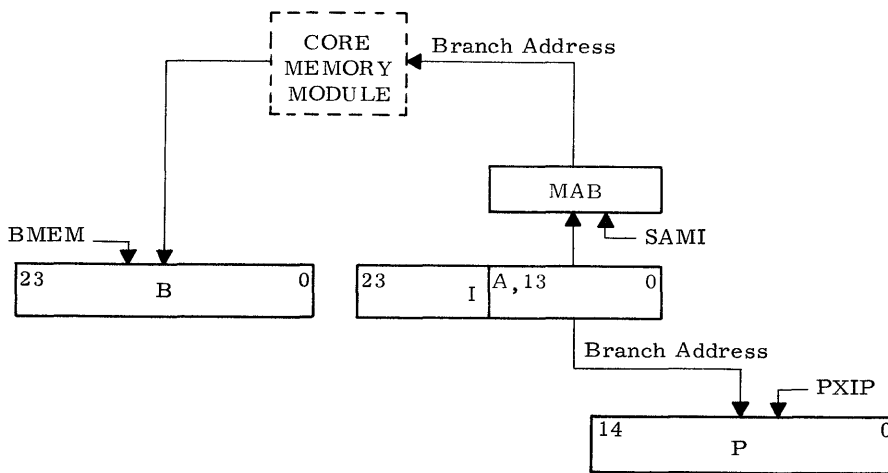
Following State 4, State 1 is entered to "fetch" the next instruction, as specified by the contents of  $Z_{14-0}$ . Memory is addressed from  $I_{A,13-0}$  to obtain this instruction. The contents of  $I_{A,13-0}$  are then transferred to the P Register during this State 1, thereby transferring program control to the address specified by the contents of  $Z_{14-0}$ . At Last Pulse Envelope of State 1, the Remember flip-flop is cleared. Sequencing then continues to execute the new command (i.e., the command located in the address specified by the contents of  $Z_{14-0}$ ).

Non-indexed Word Times.	2 (S1, S4)
Interruptable Following Execution?	No
<b>CHANGES FOLLOWING EXECUTION</b>	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(Z_{14-0})$
F1WPMT	Set if $C(Z_{21}) = 1$ Reset if $C(Z_{21}) = 0$
F1UOFL	Set if $C(Z_{22}) = 1$ Reset if $C(Z_{22}) = 0$
F1ETST	Set if $C(Z_{20}) = 1$ Reset if $C(Z_{20}) = 0$
F1MTRM	Set if $C(Z_{19}) = 1$
F1XQUA, B	Reset

COMMAND CHARACTERISTICS

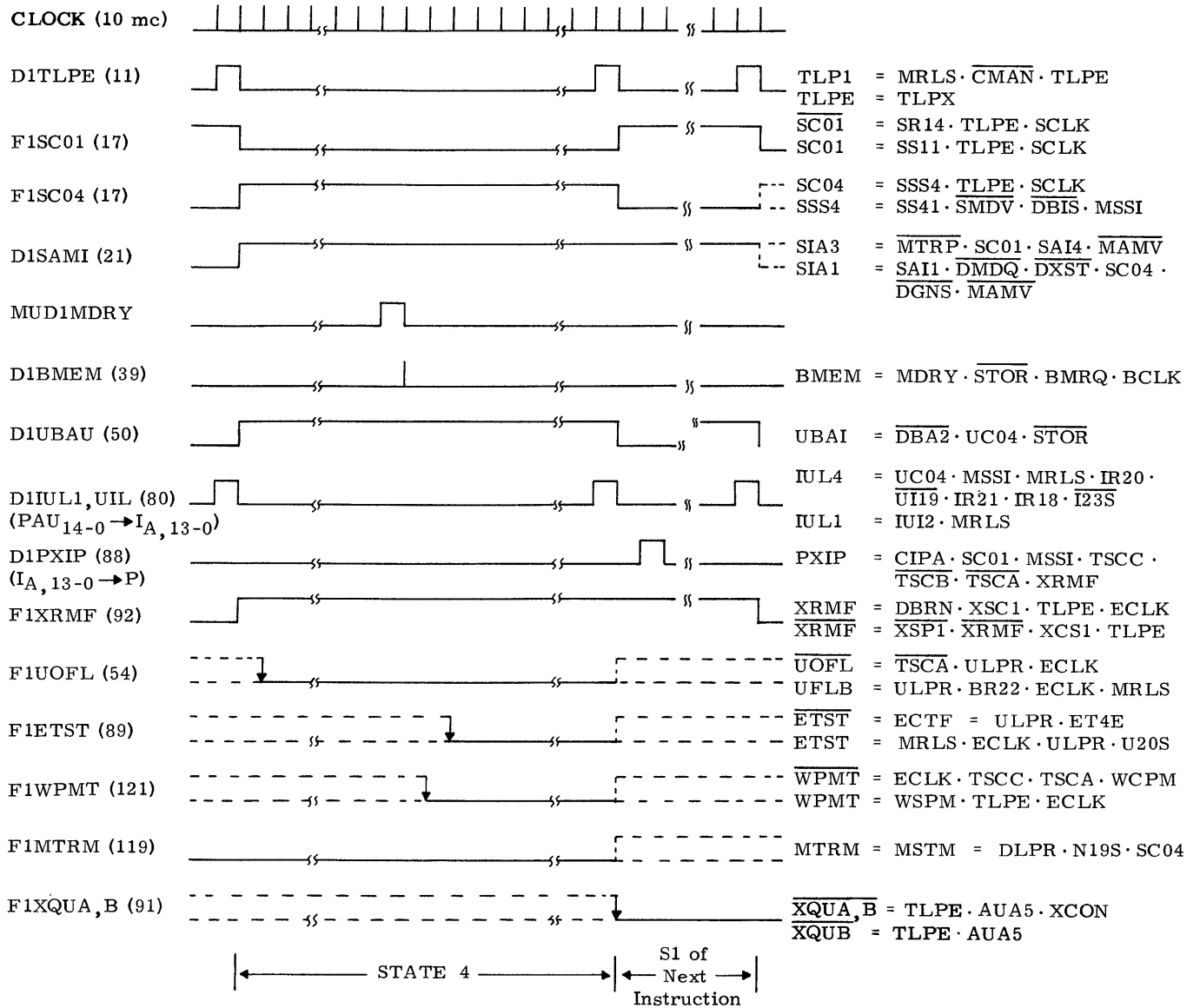


Sequence State 4



Next Sequence State 1

LPR BLOCK DIAGRAM



LPR TIMING DIAGRAM

# LXC - LOAD X WITH COUNT

FULL OPERAND	23	17	18	17	15	14	13	0
				X	*			SEE TEXT

The function performed by the LXC command depends upon the configuration of bits 14 through 0. If bits 14 through 0 of the LXC command are "zero", the contents of the J Counter are stored in bits 4 through 0 of the addressed X cell. Bits 23 through 5 of the addressed X cell are replaced by "zeros". If bits 14 through 0 are not "zeros", the LXC command causes the following to occur:

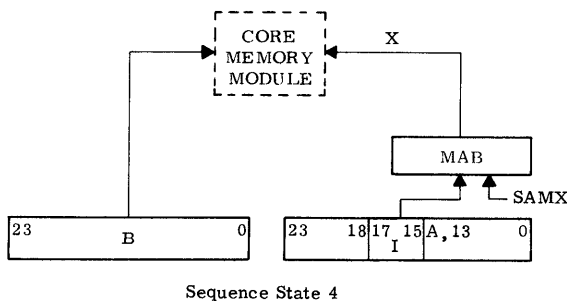
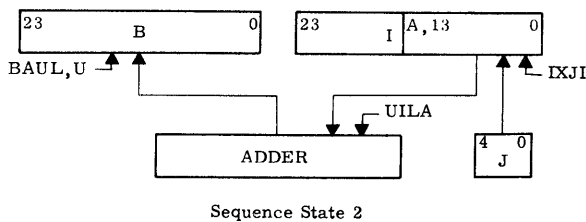
1. Bits 23 through 14 of the indicated X cell are cleared.
2. Bits 13 through 5 of the indicated X cell are replaced by bits 13 through 5 of the instruction operand. If the LXC command is relative addressed (bit 14 is a "one"), bits 15 through 5 are replaced by the sum of bits  $I_A$ , 13-5 of the instruction operand plus the core address (P) of the LXC command.
3. Bits 4 through 0 of the indicated X cell are replaced by the logical "OR" of  $J_{4-0}$  and  $I_{4-0}$ .

The LXC command is executed during Sequence Control States 2 and 4. During State 2, the contents of the J Counter are transferred (single-ended) to bits 4 through 0 of the I Register. Then,  $I_A$ , 13-0 is gated to the Adder Unit (D1UILA). The contents of the Adder (24 bits) are then transferred to the B Register.

**NOTE**

Although memory is requested during State 2 (G1SMRQ), operation of the LXC command is not affected. The contents of core cell 0 will be transferred to the B Register but this data will be destroyed by the double-ended transfer of the Adder to the B Register. This operation is implemented for memory protect error detection.

Following State 2, State 4 is entered. Memory is addressed from the X bits of the I Register (17 through 15). The contents of the B Register are then stored in the addressed X cell to complete the LXC operation.

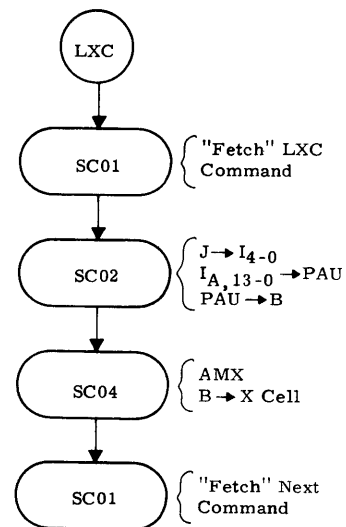


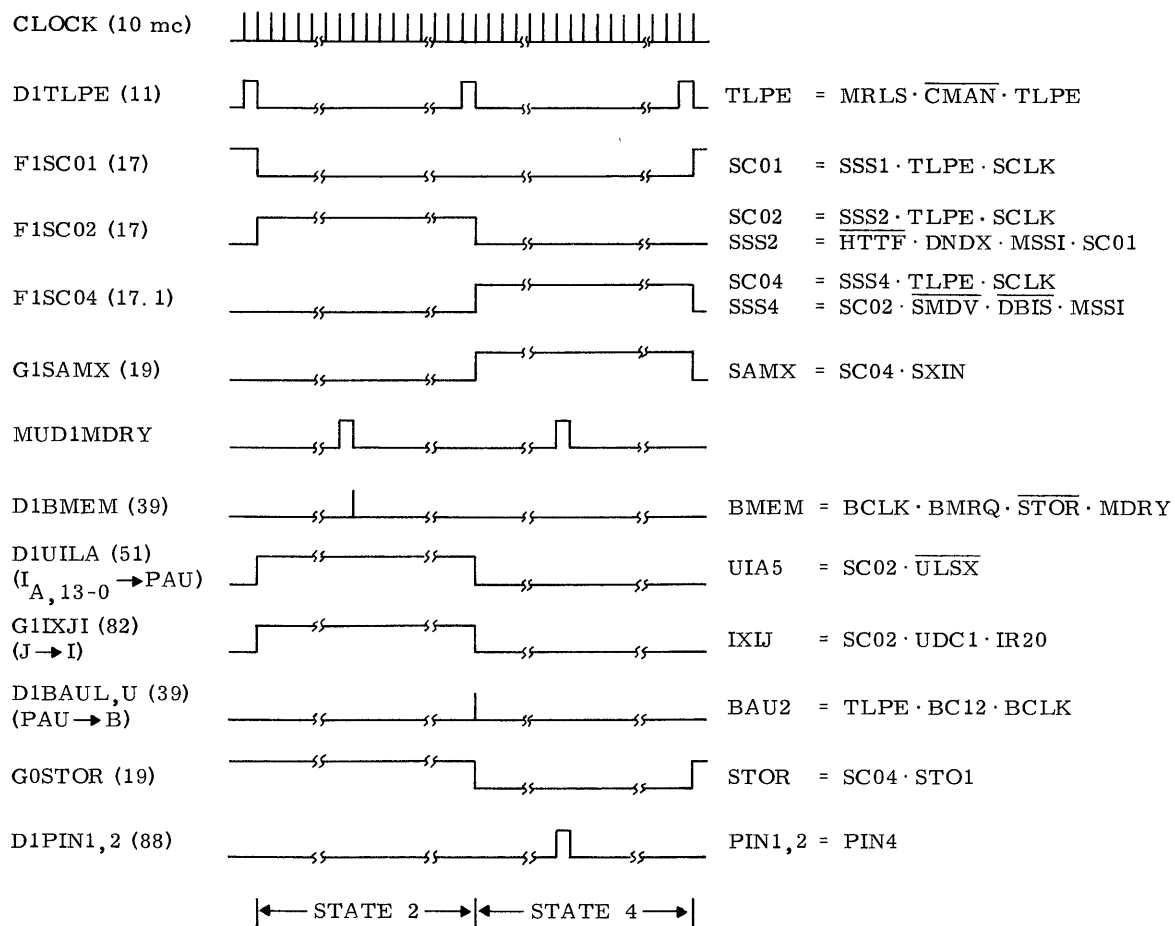
LXC BLOCK DIAGRAM

Word Times.	3 (S1, S2, S4)
Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	
Memory X	$^{23}$ Zero $^5$ $^4$ $C(J_{4-0}) * ^0$

COMMAND CHARACTERISTICS

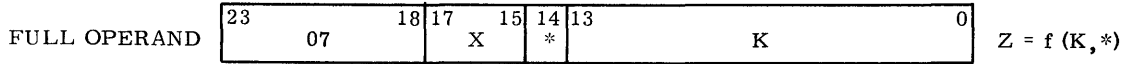
\* If LXC = 17X00000. See text for contents of X following other LXC commands.





LXC TIMING DIAGRAM

# LXK - LOAD X WITH K

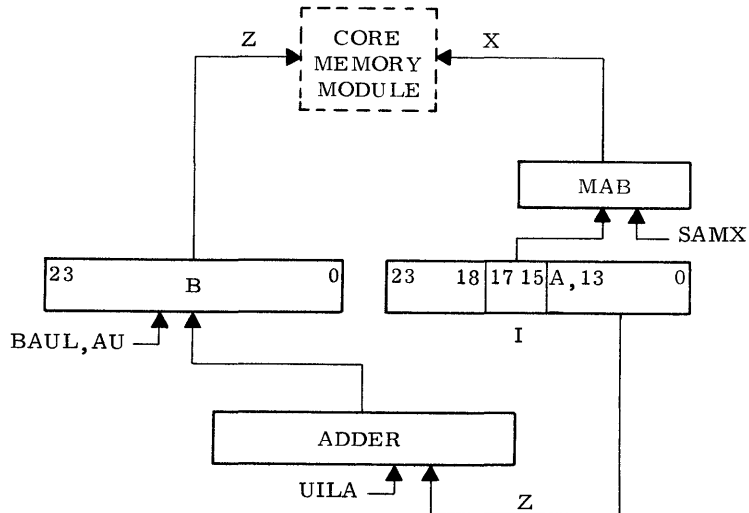


LXK stores the value Z into the addressed X core cell. Leading bits of the addressed X cell are set to "zero". The range of K, when not relative addressed, may vary from 0 to +16,383. The range of K when relative addressed may vary from -8,192 to +8,191 since bit 13 represents the sign of bits 12-0 when relative addressed. If bits 17, 16, and 15 of the LXK command are zero, the command is undefined (i.e., an X cell address must be specified).

The LXK command is executed during Sequence Control State 4 (SC04). During State 4, memory is addressed from  $I_{17-15}$  (G1SAMX). The contents of  $I_A$ , 13-0 are gated, via the Adder Unit (G1UILA), to the B Register (D1BAUL, AU). From B, this value is stored in the addressed X cell to complete the LXK command.

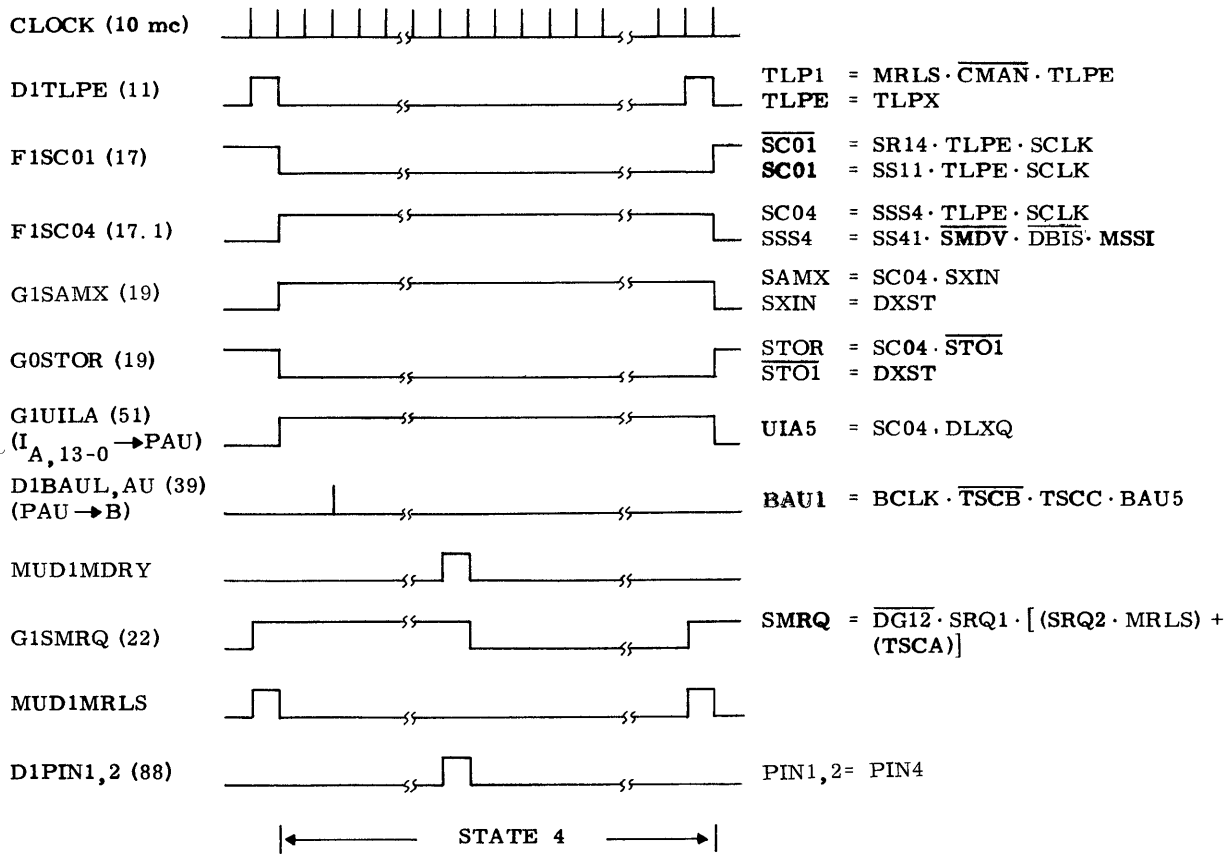
Non-Indexed Word Times.	2 (S1, S4)
Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	C(P) +1
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	
Memory X	$^{23}$ Zeros $^{15}$ $^{14}$ Z $^0$

## COMMAND CHARACTERISTICS



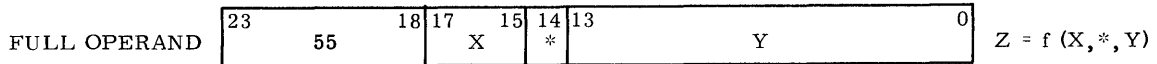
LXK BLOCK DIAGRAM





LXX TIMING DIAGRAM

# MPY - MULTIPLY



MPY forms the product of the contents of core cell Z (multiplicand) and the contents of the Q Register (multiplier). The contents of the A Register are added algebraically to the least significant half of the product. Thus, with proper scaling, it is possible to form the value QZ+A. The result is stored in A<sub>23-0</sub> and Q<sub>22-0</sub>, with the most significant half in A. Bit 23 of Q is set to "zero" and is not a part of the product. The sign of A (A<sub>23</sub>) applies to the entire product. Either positive or negative (2's complement) values may be multiplied, with the correct product, positive or negative (2's complement), contained in A and Q.

## BINARY MULTIPLICATION

For a better understanding of the following description, two important steps of multiplication must be recalled: two numbers are multiplied by a series of (1) additions, and (2) shifts. The following examples illustrate that the mechanics employed for decimal numbers is true also for binary numbers.

Decimal	Binary
3	(multiplicand) 00011 = 3
x15	(multiplier) 01111 = 15
3 Add	00011 Add & Shift
3 Add	00011 Add & Shift
3 Add	00011 Add & Shift
3 Add	00011 Add & Shift
3 Add & Shift	00000 Shift
3 Add	0000101101 = 45
45	

As illustrated in the example of binary multiplication, the following rules are obeyed:

- (1) If a multiplier digit is a "one", the multiplicand is added to obtain a partial product. Then, a shift one place to the left occurs for the next partial product.
- (2) If a multiplier digit is a "zero", the multiplicand is not added but a shift one place to the left occurs for the next partial product.
- (3) The partial products are then added to obtain the result.

Multiplication within the 4022 Arithmetic Unit utilizes the "string" concept of multiplication to reduce the number of additions required, thereby increasing the speed of the multiply cycle. Using the "string" concept, it is not necessary to form the partial product for each bit when the multiplier contains two or more successive "one" bits.

The following observations will be of assistance in understanding the "string" concept of multiplication. First, a binary number such as, 0100001000 may be written as  $2^8 + 2^3$ . Observe also, that a binary number

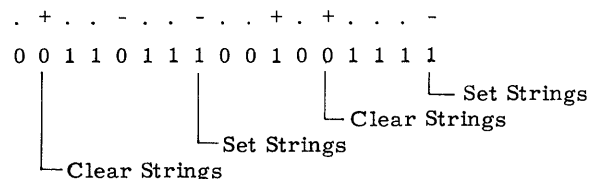
Non-Indexed Word Times.	S1, S3, S4, S5 8.9 - 12.1 $\mu$ s
Interruptable Following Execution?	Yes
<b>CHANGES FOLLOWING EXECUTION</b>	
A <sub>23-0</sub>	Most Significant Part of Product
Q <sub>23-0</sub>	23:22 Least Significant Part of Product
P <sub>14-0</sub>	C(P) + 1
F1WPMT	
F1UOFL	Set if Overflow Occurs
F1ETST	
J <sub>4-0</sub>	37 <sub>8</sub>
Memory Z	

## COMMAND CHARACTERISTICS

such as 00011111 may be written  $2^6 - 2^0$  (i.e., 00100000 minus 1 = 00011111). Thus, when a binary multiplier has a number of successive "one" bits, it is not necessary to form the partial product for each "one" bit, but, as in the case above, merely subtract  $2^0$  from  $2^6$ . Now consider a binary number such as 0011011, which can be written  $2^5 - 2^2 - 2^0$ , since it is the same as 0011111 reduced by  $2^2$ . A number such as this contains a "string" of "ones" with an included "zero". Note that the effect of an included "zero" is a subtraction of the corresponding power of two. This effect holds for more than one included "zero". Using this method, the least number of powers of 2 representing a binary number may be determined. For example:

$$\begin{array}{cccccccccccccccc}
 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
 = & 2^{15} & - & 2^{12} & - & 2^9 & + & 2^6 & + & 2^4 & - & 2^0.
 \end{array}$$

The following example is provided as an aid in determining when to add the multiplicand to the partial product and shift, when to subtract the multiplicand from the partial product and shift, and when to just shift the partial product. This example illustrates a multiplier with various combinations of "ones" and "zeros". A minus sign (-) above a digit indicates that a subtraction should be performed; a plus sign (+) above a digit indicates that an addition should be performed; and a dot (.) above a digit indicates that only a shift is required.



As the multiplier is shifted right from least toward more significant bits, each bit must be examined and the decision made to add, subtract, or shift.

These observations can best be summarized in terms of the concept of a string of "ones". Two or more adjacent "ones" constitute a string of "ones", and as noted above, the least significant "one" of a string is interpreted as  $\bar{1}$ , while the zero to the left of the most significant "one" is interpreted as  $^+0$ , for example, as in  $00^+111\bar{1}$ . A string may include isolated "zeros", i.e., "zeros" each of which is flanked by "ones", such "zeros" are interpreted as  $\bar{0}$ , for example, as in  $00^+1\bar{1}0\bar{1}1\bar{1}$ . Isolated "ones", i.e., "ones" each of which is flanked by "zeros" and is not a member of a string, are interpreted as  $^+1$ , for example, as in  $00\bar{1}0^+100$ . Thus, working from right to left, as a string of "ones" is met, a subtraction must take place, and only subtraction can take place within a string; in exiting from a string, an addition takes place and only additions can take place outside a string. Finally, note that the above discussion shows that, when working from right to left, a string is not initiated until at least two adjacent "ones" occur and does not end until at least two adjacent "zeros" occur.

### COMMAND DESCRIPTION

Fig. MPY.1 contains a basic flow chart of the Sequence Control States required to execute the MPY command and lists the basic functions performed within each Sequence State. Block diagrams of States 3, 4, and 5 are contained in Fig. MPY.2.

#### Sequence State 1

The MPY command is "fetched" from memory during a normal State 1. At the end of State 1, the MPY command is contained in the I Register. Following State 1, a non-indexed MPY command enters Sequence State 3.

#### Sequence State 3

A timing diagram with logic equations for State 3 is contained in Fig. MPY.3. During State 3, memory is addressed from  $I_A, 13-0$  (D1SAMI) and the contents of memory location Z (multiplicand) are gated to the B Register. From the B Register, the multiplicand is gated to the Adder (D1UBAU) and from the Adder to the I Register (D1IUIU, D1IUIL, D1IULI). At Time 3 of State 3, the Execute MPY or DVD flip-flop (F1XMDV) is set to indicate that the I Register does not contain the instruction, and to provide control for the MPY operation. The MPY command leaves the Execute flip-flop (F1XEXC) reset and the DVD command sets F1XEXC. Therefore, when F1XMDV is set and F1XEXC is reset, the MPY command is specified. Following State 3, State 4 is entered.

#### Sequence State 4

Sequence State 4 is used to bring the multiplier from the Q Register (memory cell  $10_8$ ) to the B Register and to perform the actual arithmetic operation. To provide these functions, the duration of State 4 is extended by entering Time 6 Envelope.

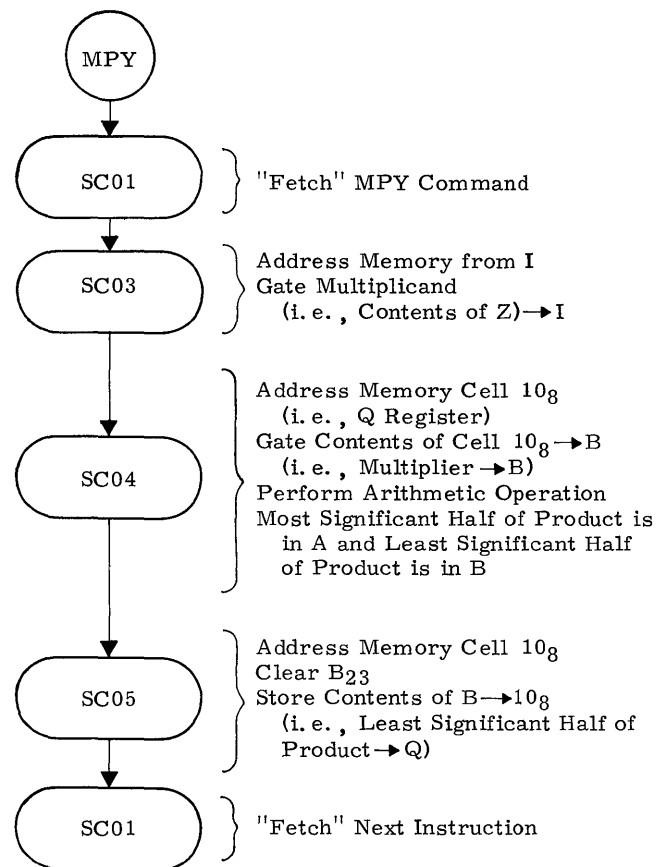


Fig. MPY.1. MPY Flow Chart

For ease of understanding, a detailed flow chart of State 4 is contained in Fig. MPY.4. The logic elements performing the functions listed in the flow chart are provided, along with the logic equations and the logic sheet numbers on which these elements may be found, in Table MPY.1.

An example of the arithmetic operation is provided in Table MPY.2. For simplicity, nine-bit registers are used. The timing diagram of State 4 contained in Fig. MPY.5, illustrates the timing associated with this example. Using these aids, little difficulty should be encountered in determining the operation of the Arithmetic Unit for any multiplier and multiplicand values.

During the first 1.6 microseconds of State 4, memory cell  $10_8$  is addressed ( $G0MX03$ ) and the multiplier is gated from the Q Register (cell  $10_8$ ) to the B Register.

At Time 3, the Delay Time Counter is preset to  $30_8$  and the J Counter is preset to  $7_8$ . The J Counter is incremented as each bit of the multiplier is considered to determine the final product. When J is equal to  $37_8$ , all 23 bits of the multiplier have been considered and the fix-up cycle, if required, is entered to end the MPY cycle. The Delay Time Counter is used to provide control for the addition or subtraction and shift required with each multiplier bit.

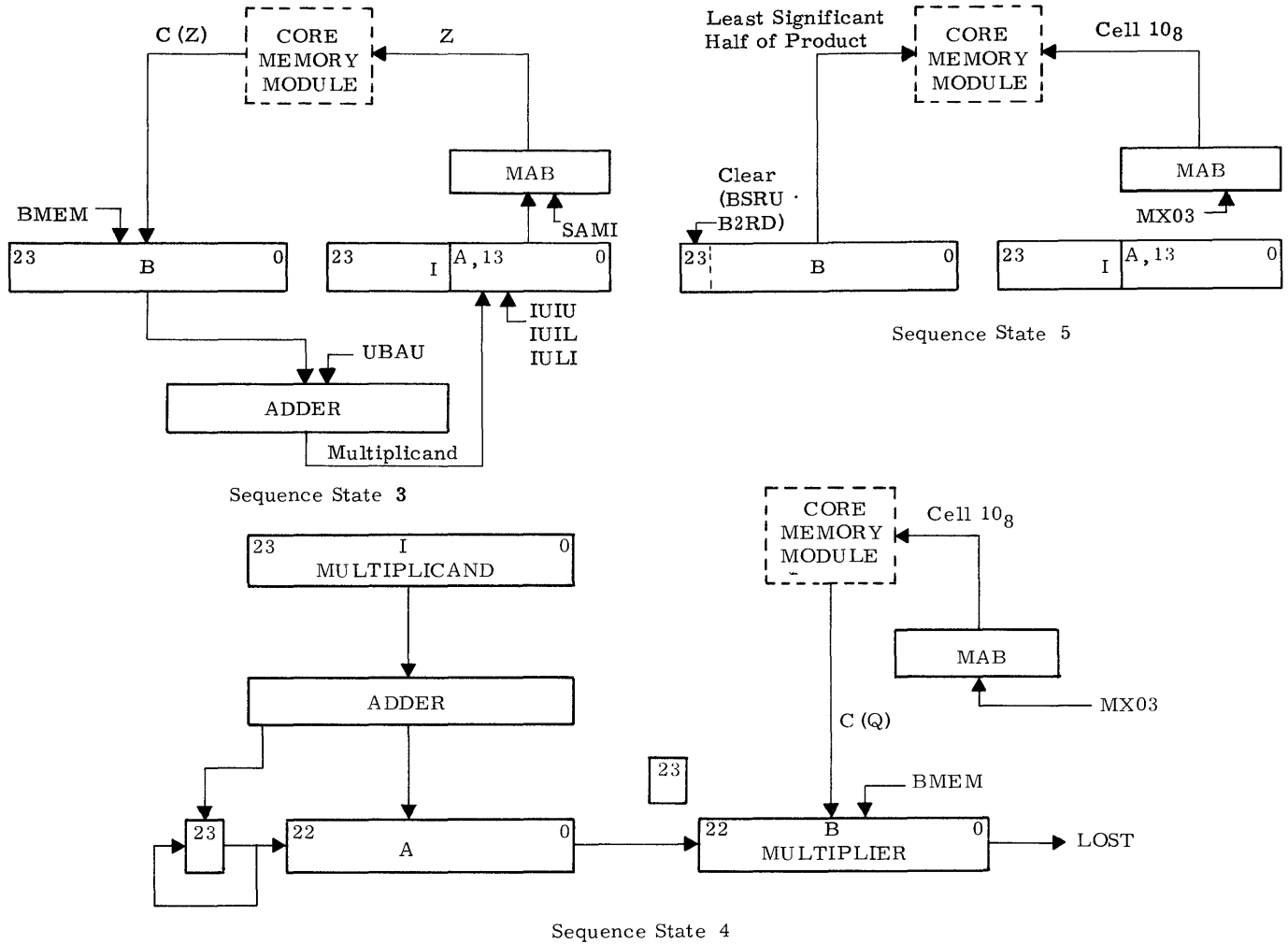


Fig. MPY.2. MPY Block Diagram

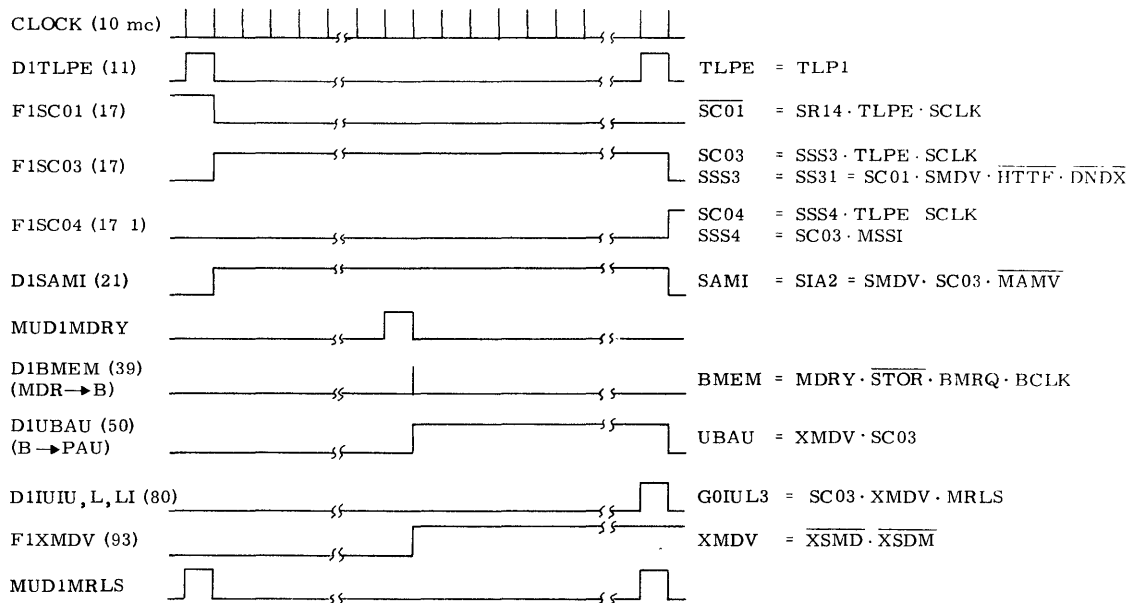


Fig. MPY.3. MPY Sequence State 3 Timing

Refer to Table MPY. 1 for Flow Chart Logic Equations

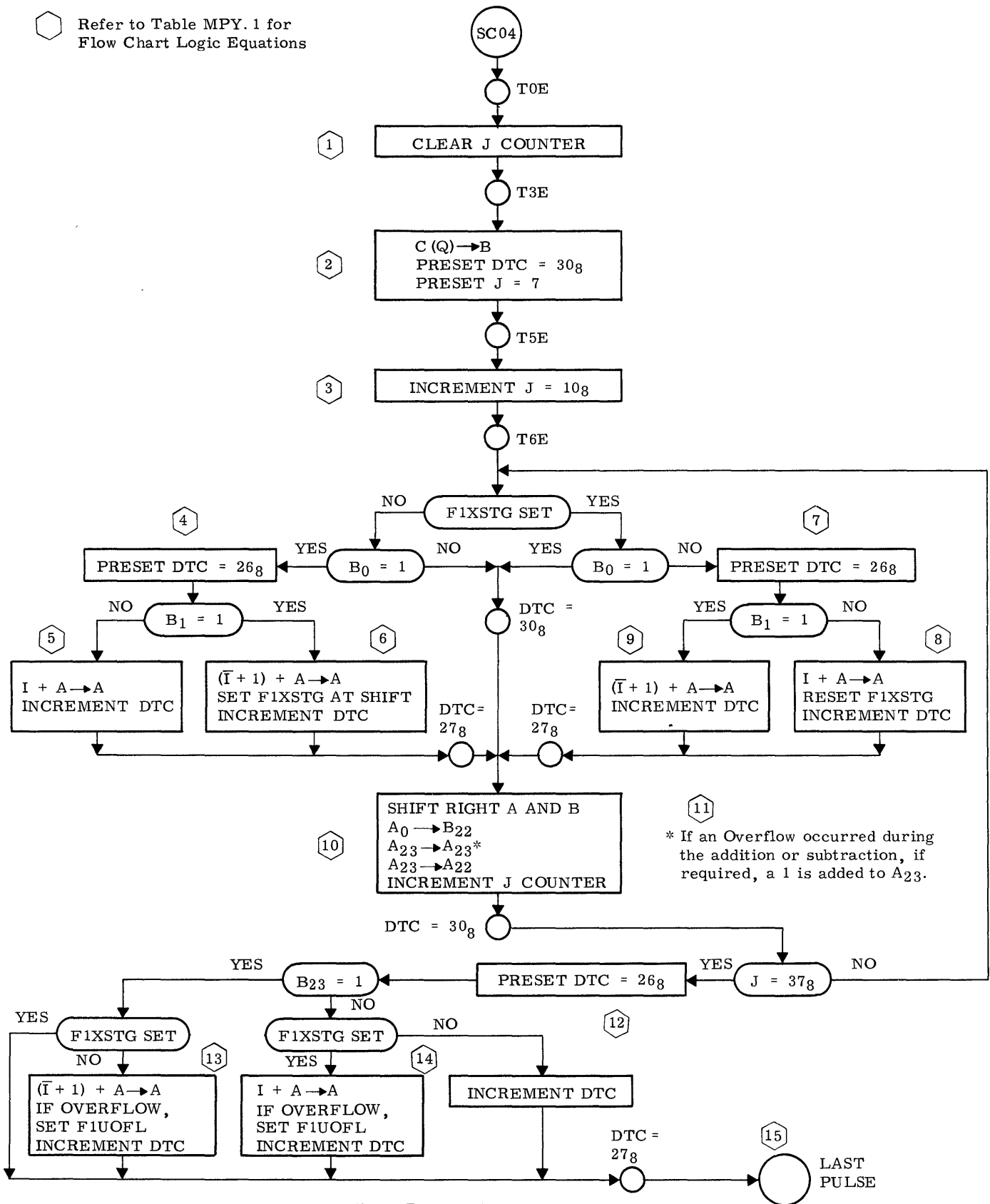


Fig. MPY. 4. State 4 Flow Chart

- 1 Clear J Counter:  
 $D0JJE0 (70) = JCK1 \cdot \overline{TSCA} \cdot \overline{TSCB} \cdot JS4G$   
 $G1JS4G (70) = SC04 \cdot DMD1$
- 2 Preset Delay Time Counter to 30g:  
 $G1TPAF (12) = TP30 + TD30$   
 $G0TP30 (12) = TSCB \cdot TSCA \cdot DMGT \cdot TSCC$   
Preset J Counter to 7:  
 $G1JP07 (70) = TP30 \cdot XMDV$
- 3 Increment J Counter:  
 $N1JINC (71) = G0JINC = XMDV \cdot TT5E \cdot MRLS \cdot JC04$
- 4 Preset Delay Time Counter to 26g:  
 $G1TP26 (12) = TD30 \cdot \overline{JE07} \cdot XMDV \cdot G0TP26$   
 $G0TP26 (12) = \overline{XSTG} \cdot BR00 \cdot DMPY$
- 5 Add I to A:  
 $D1UAAU (52) = XMDV \cdot UC04$   
 $D1UILA, AU (51) = UIA3 = \overline{JE37} \cdot UIAA \cdot \overline{XSTG} \cdot \overline{BR01} \cdot BR00$   
 $D1AAU, L (62) = ECLK \cdot TE26 \cdot AMDV$
- 6 Subtract I from A:  
 $D1UAAU (52) = XMDV \cdot UC04$   
 $D1UINA (51) = UIN2 = UC04 \cdot XMDV \cdot \overline{JE37} \cdot BR01 \cdot BR00 \cdot \overline{XSTG} \cdot \overline{XEXC}$   
 $G1UENC (51) = UIN2$   
 $D1AAU, L (62) = ECLK \cdot TE26 \cdot AMDV$   
Set Strings Flip-Flop at Shift of A and B:  
 $F1XSTG (94) = XDST \cdot XBSR \cdot ICK1$   
 $G1XDST (94) = BR01 \cdot BR00$
- 7 Preset Delay Time Counter to 26g:  
 $G1TP26 (12) = TD30 \cdot \overline{JE07} \cdot XMDV \cdot G0TP26$   
 $G0TP26 (12) = BR00 \cdot XSTG \cdot DMPY$
- 8 Add I to A:  
 $D1UAAU (52) = XMDV \cdot UC04$   
 $D1UILA, AU (51) = UIA2 = \overline{BR00} \cdot \overline{BR01} \cdot XSTG \cdot \overline{JE37} \cdot UIAA$   
 $D1AAU, L (62) = ECLK \cdot TE26 \cdot AMDV$   
Reset Strings Flip-Flop at Shift of A and B:  
 $\overline{F1XSTG} (94) = \overline{XNST} \cdot XBSR \cdot ICK1$   
 $G1XNST (94) = BR01 \cdot BR00$
- 9 Subtract I from A:  
 $D1UAAU (52) = XMDV \cdot UC04$   
 $D1UINA (51) = UIN3 = UC04 \cdot XMDV \cdot \overline{JE37} \cdot BR01 \cdot BR00 \cdot XSTG \cdot \overline{XEXC}$   
 $G1UENC (51) = UIN3$   
 $D1AAU, L (62) = ECLK \cdot TE26 \cdot AMDV$
- 10 Shift Right A and B:  
 $G0ABSR (63) = \overline{JE37} \cdot \overline{DMPY} \cdot TE27 \cdot TT6E + TT6E \cdot \overline{JE37} \cdot TE30 \cdot ABSG$   
Gate A<sub>0</sub> to B<sub>22</sub>:  
 $G1B22N (38) = \overline{BA00} \cdot BMPY$   
 $G1B22D (38) = BMPY \cdot AR00$   
Gate A<sub>23</sub> to A<sub>23</sub>:  
 $G1AFNS (67) = \overline{AFNA} = ANA3 \cdot AR23$   
 $N0AFNS (67) = G1AFNS$   
Increment J Counter:  
 $N1JINC (71) = G0JINC = JE37 \cdot ABSG \cdot TE30 \cdot TT6E + JE37 \cdot TT6E \cdot XMDV \cdot TEFF$
- 11 Add 1 to A<sub>23</sub> if Overflow Occurred During 5, 6, 8, or 9:  
 $F1AFNP (66) = TE26 \cdot UFL1 \cdot BCLK$   
 $G1UFL1 (48) = W23C \cdot \overline{U22C} \cdot \overline{U23S} \cdot \overline{ADIV} + U23S \cdot U22C \cdot \overline{U23C} \cdot ADIV$
- 12 Preset the Delay Time Counter to 26g:  
 $G1TP26 (12) = TD30 \cdot \overline{JE07} \cdot XMDV \cdot JE37$
- 13 Subtract I from A:  
 $D1UAAU (52) = XMDV \cdot UC04$   
 $D1UINA (51) = UIN4 = JE37 \cdot \overline{XSTG} \cdot UIAA \cdot BR23$   
 $G1UENC (51) = UIN4$   
 $D1AAU, L (62) = ECLK \cdot TE26 \cdot AMDV$   
If Overflow Occurs During Subtraction, Set F1UOFL:  
 $F1UOFL (54) = UFLB$   
 $G0UFLB (48) = JE37 \cdot TE26 \cdot ECLK \cdot UFL1$   
 $G1UFL1 (48) = W23C \cdot \overline{U22C} \cdot \overline{U23S} \cdot \overline{ADIV} + U23S \cdot U22C \cdot \overline{U23C} \cdot ADIV$
- 14 Add I to A:  
 $D1UAAU (52) = XMDV \cdot UC04$   
 $D1UILA, AU (51) = UIA4 = UIAA \cdot JE37 \cdot XSTG \cdot \overline{BR23}$   
 $D1AAU, L (62) = ECLK \cdot TE26 \cdot AMDV$   
If Overflow Occurs During Addition, Set F1UOFL:  
 $F1UOFL (54) = UFLB$   
 $G0UFLB (48) = JE37 \cdot TE26 \cdot ECLK \cdot UFL1$   
 $G1UFL1 (48) = W23C \cdot \overline{U22C} \cdot \overline{U23S} \cdot \overline{ADIV} + U23S \cdot U22C \cdot \overline{U23C} \cdot ADIV$
- 15 Enable Last Pulse:  
 $D1TLPE (11) = JE37 \cdot AMDV \cdot TE27$

Table MPY. 1. State 4 Flow Chart Logic Equations

At Time 5 (Memory Release), the J Counter is allowed to increment for the first time. Time 6 is then entered and the actual computation is performed.

As described in the basic multiply discussion, the two least significant bits of the multiplier (B<sub>1</sub> and B<sub>0</sub>) and the status of the Strings flip-flop (FIXSTG) are used to determine if the multiplicand is added or subtracted from the partial product in the A Register and then the partial product is shifted or if only a shift of the partial product is required.

After the J Counter is equal to 37<sub>8</sub>, the status of the Strings flip-flop and the sign bit of the multiplier (B<sub>23</sub>) is examined to determine if an addition or subtraction is required to "fix-up" the product. If B<sub>23</sub> is a "zero" and the Strings flip-flop is set, then an addition must be performed to end the within strings cycle. If B<sub>23</sub> is a "one" and the Strings flip-flop is reset, then the multiplier was negative and an additional subtraction

is required. Following this "fix-up", if required, State 4 is ended. Therefore, at the end of State 4, the most significant half of the product is in the A Register and the least significant half of the product is in the B Register.

### Sequence State 5

During State 5, the least significant half of the product contained in the B Register is stored in the Q Register (cell 10<sub>g</sub>). A timing diagram and logic equations for State 5 are contained in Fig. MPY.6.

At the first clock pulse of State 5, B<sub>23</sub>, if a "one", is reset to "zero". Memory cell 10<sub>g</sub> is addressed (G0MX03) and the contents of the B Register are stored in cell 10<sub>g</sub>.

During State 1 of the command following MPY, the Multiply/Divide flip-flop (FIXMDV) is reset.

S	
I =	0 0 0 0 0 1 1 1 1 = +15 = MULTIPLICAND
B =	0 0 0 0 0 1 1 0 0 = +12 = MULTIPLIER
A =	0 0 0 0 0 0 0 0 0 = 0

J COUNT	DTC	A REGISTER								B REGISTER								I REGISTER								XSTG	ACTION								
		S	7	6	5	4	3	2	1	0	S	7	6	5	4	3	2	1	0	S	7	6	5	4	3			2	1	0					
10	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	T5E of S4
11	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	Shift
12	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	Shift	
12	26	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	SUB
13	27	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	Set XSTG Shift	
14	30	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	Shift	
14	26	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	ADD	
15	27	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Clear XSTG Shift	
16	30	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Shift	
17	30	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Shift	
20	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Shift	

↑ Equivalent to J = 37<sub>8</sub>  
When Using 24-Bit Registers

↑ = 180<sub>10</sub>

Table MPY.2. MPY Example (+15 x +12)

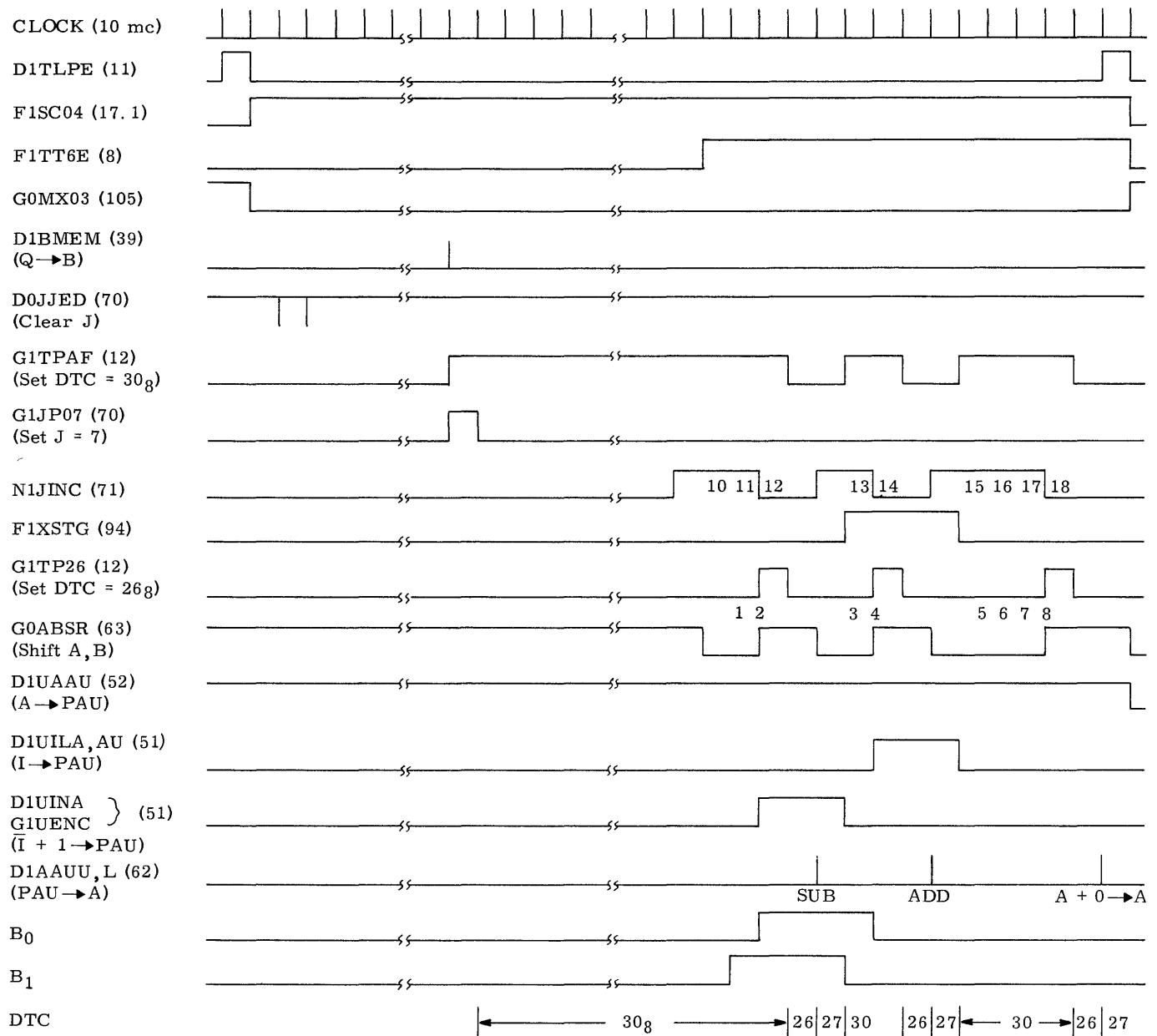


Fig. MPY. 5. MPY Sequence State 4 Timing Example (+15 x +12)



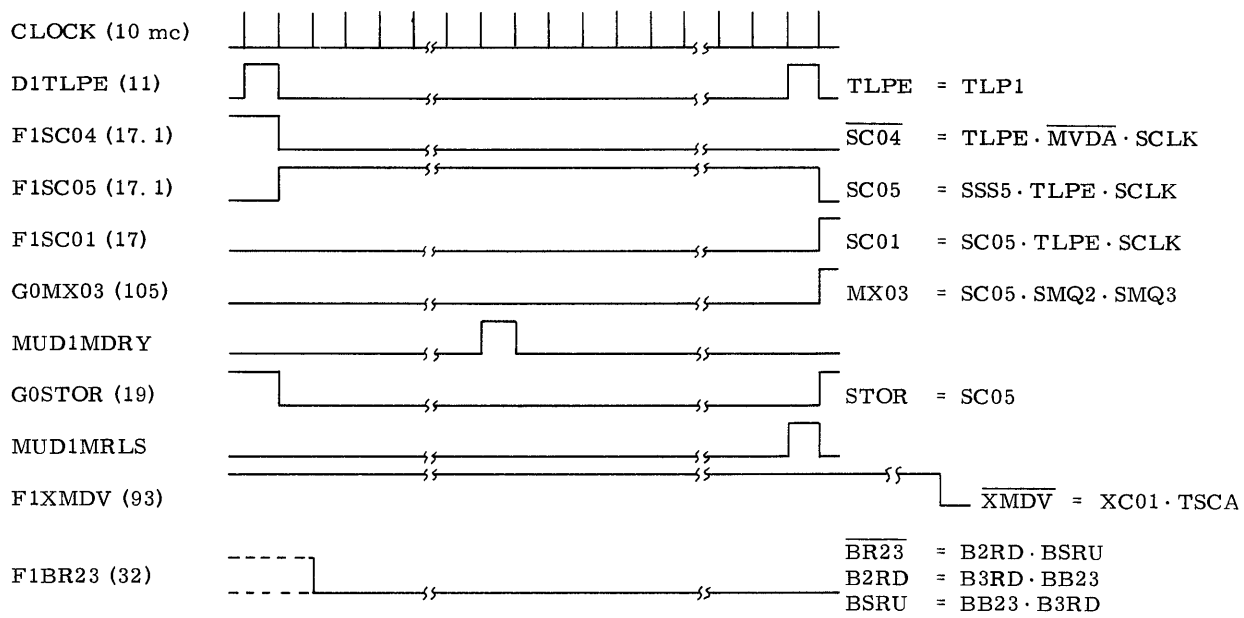


Fig. MPY. 6. MPY Sequence State 5 Timing

# NOP - NO OPERATION

<b>FULL OPERAND</b>	<b>23</b>	2	6	2	0	0	0	0	0	0	0
---------------------	-----------	---	---	---	---	---	---	---	---	---	---

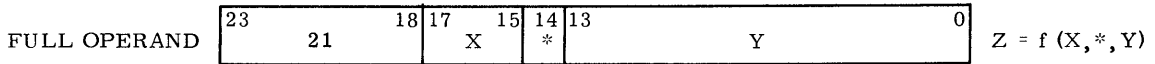
NOP transfers program control to the next sequential location (P+1). No operation is performed. NOP is useful as a programming tool to replace deleted commands or to provide room for the insertion of new commands in the program.

NOP is an INX (Increment X) command that specifies incrementing of index cell 2 by 0. Sequence States 1, 2, and 4 are required to "fetch" and "execute" the NOP command. Refer to the INX command description for details of the operation of the NOP command.

Word Times.	3 (S1, S2, S4)
Interruptable Following Execution?	Yes
<b>CHANGES FOLLOWING EXECUTION</b>	
A <sub>23-0</sub>	
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C (P) +1
F1WPMT	
F1UOFL	
F1ETST	
J <sub>4-0</sub>	
Memory X	

## COMMAND CHARACTERISTICS

# ORA - LOGICAL OR TO A



ORA performs the logical OR of the contents of core cell Z with the contents of the A Register. Each bit of Z is compared with the corresponding bit of A. When either or both is a "one", a "one" is placed in that position of A. When both bits are "zero", that position of A is not changed.

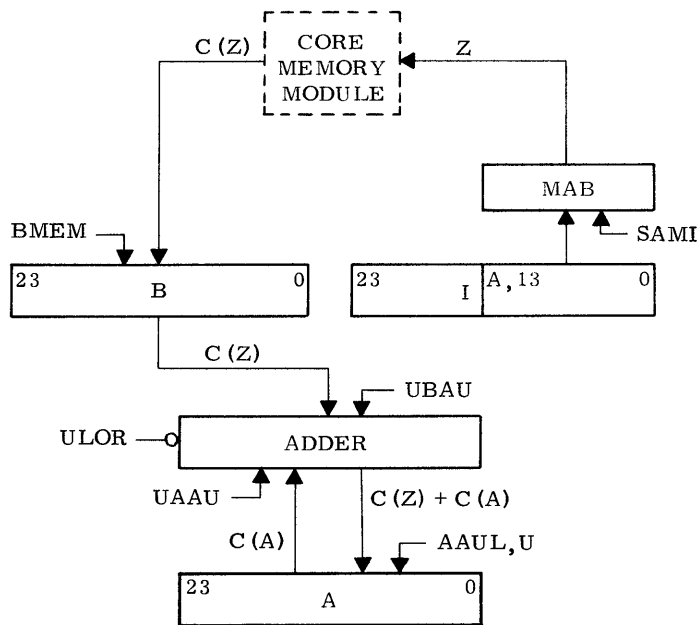
A non-indexed ORA command is executed during Sequence Control State 4 (SC04). Memory location Z is addressed from  $I_{A,13-0}$  (D1SAMI) during State 4. The contents of memory location Z are gated to the B Register by D1BMEM during the Clock pulse of Memory Data Ready (MUD1MDRY). From B, the contents of memory location Z are gated to the Adder Unit (D1UBAU). At the same time, the contents of the A Register are gated to the Adder Unit. Also, the control signal (D0ULOR) to enable the Logical OR function of the Adder Unit is applied to the Adder Unit. The result is then gated back to the A Register (D1AAUL,U) to complete the ORA execution.

To exemplify the ORA comparison, consider the following; 4 bits are used for simplicity:

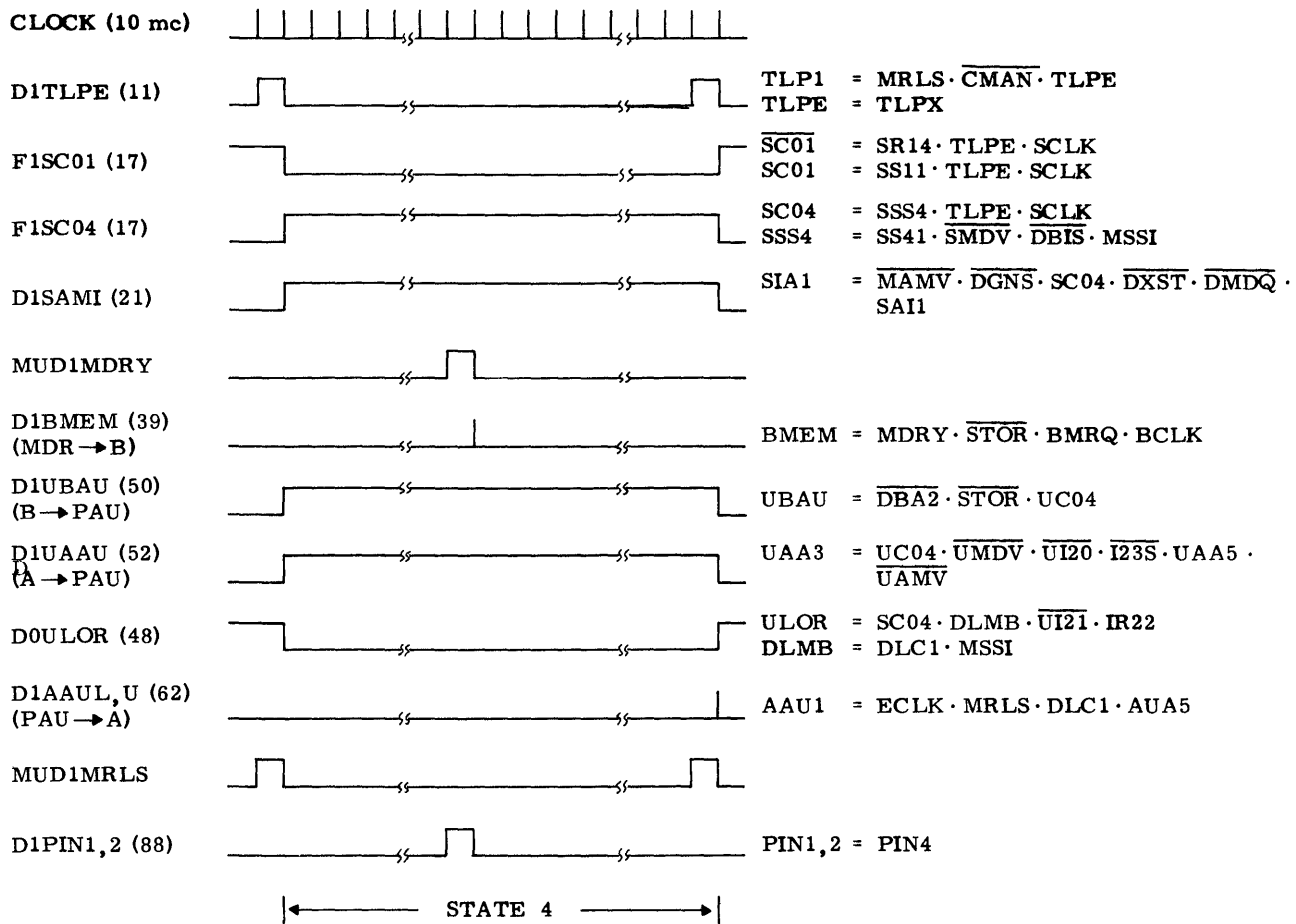
Contents of A Register            0011  
 Contents of B Register (Z)      0101  
 Result Placed in A Register    0111.

Non-Indexed Word Times.	2 (S1, S4)
Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	$C(Z_{23-0})$ ORed with $C(A_{23-0})$
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	
Memory Z	

## COMMAND CHARACTERISTICS

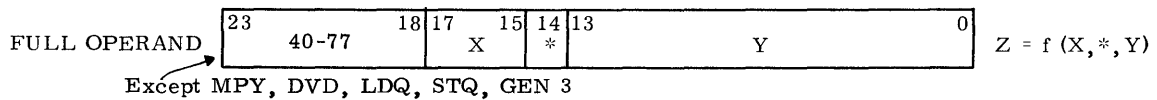


ORA BLOCK DIAGRAM



ORA TIMING DIAGRAM

# QUASI COMMANDS



Quasi commands provide operations not included in the hardware. These commands supply the programmer with a mnemonic which allows the running program to be linked to a subroutine. Quasi commands are identified by operation codes 40<sub>8</sub> through 77<sub>8</sub> if the command is not an MPY, DVD, LDQ, STQ, or GEN 3. Quasi commands store the operand portion of the command (Z) in memory location 2 and the next instruction is "fetched" from the memory location specified by the operation code (bits 23 through 18) of the Quasi command. The command located at the specified operation code address is normally an SPB which branches to a software subroutine associated with the Quasi command. Some Quasi commands, however, contain an instruction other than SPB in the location specified by the operation code. These instructions include AKA, LDK, and SKA. These instructions will contain an ADD, LDA, and SUB command with an operand address equal to 2, respectively, in the location specified by the operation code. The combination of the Quasi command storing the value Z in cell 2 and then executing the instruction in the operation code address completes the operation to be performed for these commands.

Although the function to be performed by the Quasi command depends on the instruction located in the operation code address, certain operation codes are normally reserved to perform specific Quasi functions, thereby, providing program compatibility with other GE-PAC computers. The following is a list of these commands:

Quasi Operation Code	Quasi Command Mnemonic
40	LDK
41	DLD
46	REL
47	AEL
50	SKA
51	DAD
52	LDI
53	STI
56	RBL
57	ABL
60	AKA
61	DSU
62	OOM
70	FAD
71	FSU
72	FMP
73	FDV
74	FIX/FLO/FMS

The Quasi command cannot be interrupted following execution. That is, the command located in the operation code address must be executed before an interrupt can occur. Commands within the Quasi subroutine, however, may be interrupted following execution provided the command is normally interruptible following execution. If the optional Memory Protect function is enabled relative addressed instructions within the Quasi subroutine are not subject to the protect criteria. Commands that are located within the Quasi subroutine and not relative addressed are subject to the established protect criteria.

To better understand the operation of Quasi commands, consider the following examples.

## Example 1 - FAD 70X\*Y

The Floating Add command (FAD) adds the single length floating point number contained in memory location Z to the floating point number contained in the A Register.

The FAD command is "fetched" from memory and since the Op Code is 70<sub>8</sub>, it is decoded as a Quasi command. If the FAD command is relative addressed and/or indexed, the computation is performed and the result (Z) is gated to I<sub>A,13-0</sub>. State 4 is then entered and index cell 2 is addressed. The contents of I<sub>A,13-0</sub> (Z) are then stored in index cell 2. The Op Code portion of the FAD command (70<sub>8</sub>) is gated from I<sub>23-18</sub> to I<sub>5-0</sub>. The P Register is not incremented during this command execution. State 1 is then entered and memory is addressed from the I Register and the command located in cell 70<sub>8</sub> is "fetched". Since the FAD function cannot be completed in one instruction, this will be an SPB command. During execution of the SPB command, the P Register will be incremented by 1. This provides a value of P equal to the location of the FAD command plus 1. This value is stored in index cell 1 and will be used to return to the running program following the completion of the subroutine required to perform the FAD operation. After storing the contents of the P Register in cell 1, the SPB command loads the P Register with the branch address specified by bits 13-0 of the SPB command. This will transfer program control to the subroutine that will perform the FAD operation.

The first instruction of this subroutine will contain an STX command to save the location of the running program that was stored in cell 1. The rest of the subroutine will perform the operations required to obtain the result of floating point addition. This routine will use the contents of index cell 2 to obtain the address (Z) of one of the operands.

The last instruction of the FAD subroutine will be an LDP or LPR command. This LDP or LPR command will restore the P Register with the location of the running program (i. e., with the value stored in cell 1 by the SPB command and saved in another location by the

STX command. This LDP or LPR command will also clear the Quasi flip-flop, FIXQUA, which is used for control of the optional Memory Protect circuitry.

**Example 2 AKA - 60X0K**

The Add K to A command adds the value Z to the contents of the A Register.

During State 1, the AKA command is "fetched" from memory. Since the Op Code of the AKA command is 60<sub>8</sub>, it is decoded as a Quasi command. If the AKA command is indexed, indexing occurs and the result (Z) is gated to I<sub>A,13-0</sub>. State 4 is then entered and index cell 2 is addressed. The contents of I<sub>A,13-0</sub> are stored in cell 2. The Op Code portion of the AKA command (60<sub>8</sub>) is gated from I<sub>23-18</sub> to I<sub>5-0</sub>. The P Register is not incremented during this command. State 1 is then entered. Memory is addressed from the I Register and the command in location 60<sub>8</sub> is "fetched". This command will normally be an ADD (11000002). This ADD command will sum the contents of the A Register with the contents of index cell 2 (value Z). During execution of this ADD command, the P Register will be incremented and the Quasi flip-flops (FIXQUA,B) will be cleared. Program control will then return to the location following the AKA command.

In this example, an SPB command was not located in the Quasi Op Code address because only one instruction was required, in addition to the AKA command, to complete the AKA operation.

**HARDWARE OPERATION**

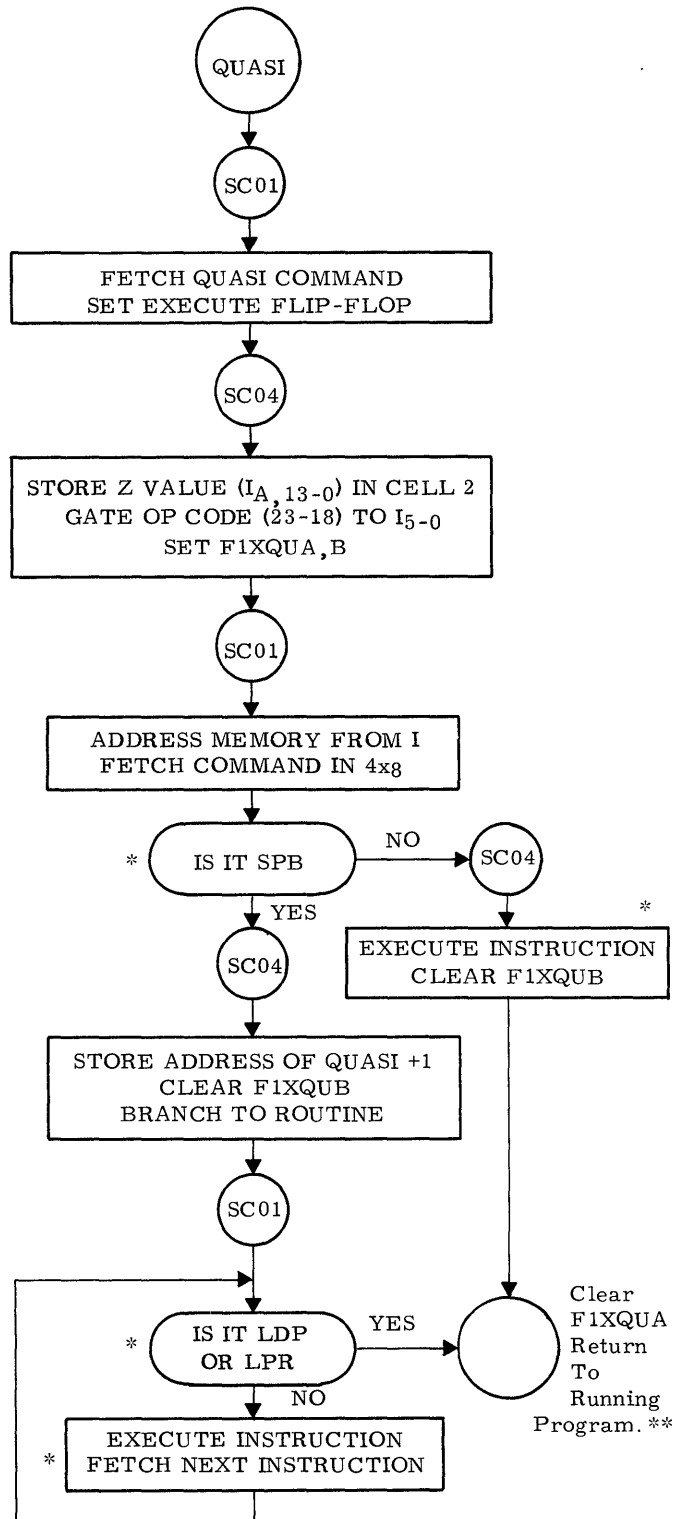
Although there are many Quasi commands that may be executed, the hardware operation for these commands is the same. Therefore, this discussion will describe only these functions.

Fig. QSI. 1 contains a flow chart of the basic hardware functions performed. Note that the only concern about the subroutine addressed by the Quasi command is to determine when the Quasi flip-flop FIXQUA is reset. This flip-flop is used by the optional Memory Protect logic, when enabled, to determine if a violation exists. Quasi commands within the subroutine are not subject to trapping if the command is relative addressed. A more detailed discussion of Memory Protect is contained in the Options (OPT) portion of this Arithmetic Unit Description.

Fig. QSI. 2 contains a block diagram of the functions performed by the Sequence States of the Quasi command. A timing diagram, including logic equations, is contained in Fig. QSI. 3.

During State 1, the Quasi command is "fetched" from memory. At Last Pulse of State 1, the Execute flip-flop is set. Following State 1, State 4 is entered.

During State 4, index cell 2 is addressed (G0MX01). I<sub>A,13-0</sub> (Z) is gated to the Adder and from the Adder to the B Register and stored in location 2. I<sub>23-18</sub> (Quasi Op Code) is gated to I<sub>5-0</sub>. The Quasi flip-flops (FIXQUA,B) are set. Following State 4, the second State 1 is entered.



\* All commands within the Quasi subroutine that are not relative addressed, are subject to memory protect trapping if enabled.

\*\* Running program starts at the location of the Quasi command plus 1.

Fig. QSI. 1. Quasi Flow Chart

During this State 1, memory is addressed from the I Register and the instruction located in the memory cell specified by the Quasi Op Code is gated to memory. This command is then executed. Normal incrementation of the P Register occurs. If the command is not

an SPB, both Quasi flip-flops (FIXQUA and FIXQUB) are cleared at Last Pulse of State 4. If the command is an SPB, FIXQUB will be cleared at Last Pulse of State 4 of the SPB command but, FIXQUA will not be cleared until an LDP or LPR command is executed.

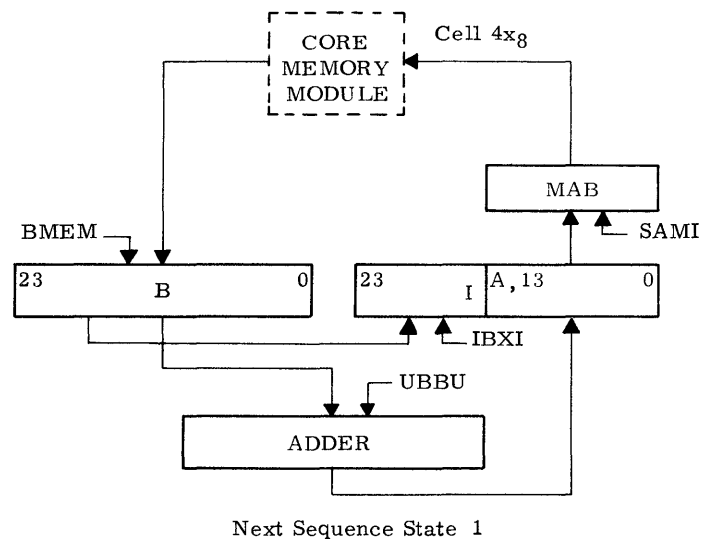
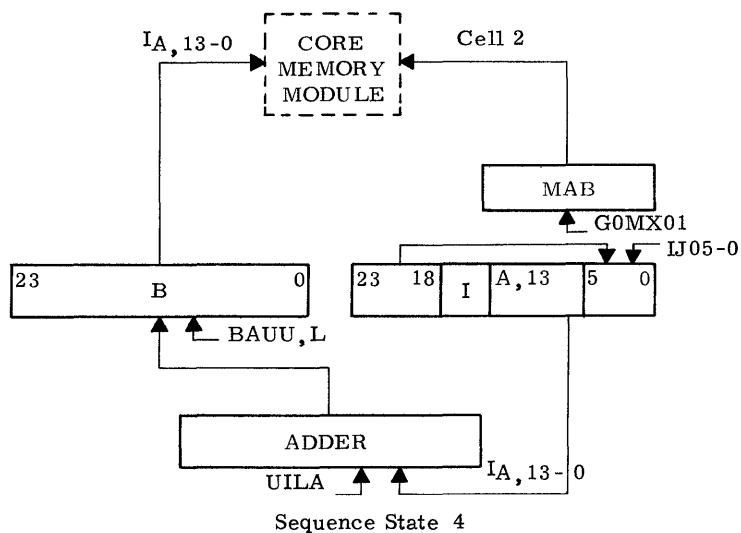
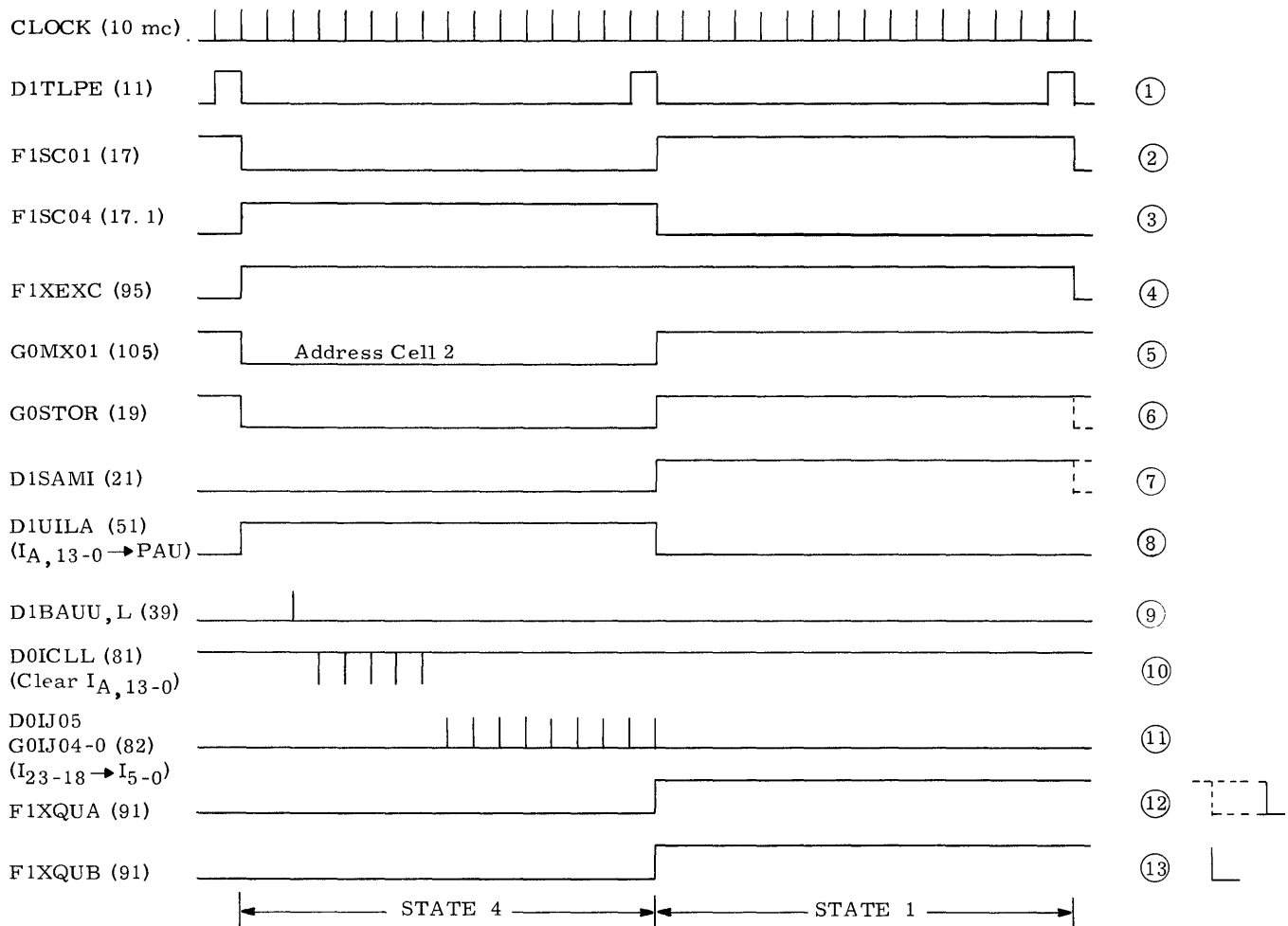


Fig. QSI. 2. Quasi Block Diagram



- ①  $D1TLPE = TLP1 = TLPE \cdot MSS1 \cdot SC04$
- ②  $F1SC01 = SSS1 \cdot TLPE \cdot SCLK$
- ③  $F1SC04 = SSS4 \cdot TLPE \cdot SCLK$
- ④  $F1XEXC = XXC1 \cdot TLPE \cdot ICK1$
- ⑤  $G0MX01 = SC04 \cdot DQUA$
- ⑥  $G0STOR = SC04 \cdot \overline{STO1}$
- ⑦  $D1SAMI = SIA3$
- ⑧  $D1UILA = UIA5 = SC04 \cdot DLXQ$
- ⑨  $D1BAUU, L = BCLK \cdot \overline{TSCB} \cdot TSCC \cdot BAU5$   
 $G1BAU5 = BLXQ \cdot \overline{TSCA} \cdot BC04$
- ⑩  $D0ICLL = \overline{ISCA} \cdot ICK1 \cdot TSCB \cdot \overline{ICL1}$
- ⑪  $D0IJ05 = IQUA \cdot ISCA \cdot ICK1 \cdot IR23$   
 $G0IJ04-0 = IQUA \cdot ISCA \cdot ICK2 \cdot IR22-18$
- ⑫  $F1XQUA = DQUA \cdot SC04 \cdot TLPE \cdot ECLK$   
 $F1XQUB = AUA5 \cdot TLPE \cdot ECLK \cdot [(XQUB \cdot \overline{XSPB}) + (XLPR \cdot \overline{XSPB}) + (DSPB \cdot XTRP \cdot WC04)]$
- ⑬  $F1XQUA = DQUA \cdot SC04 \cdot TLPE \cdot ECLK$   
 $F1XQUB = AUA5 \cdot TLPE \cdot ECLK$

Fig. QSI. 3. Quasi Timing Diagram



# RELATIVE ADDRESSING

Relative addressing modifies the operand (bits 13-0) portion of the command "fetched" from memory by adding the core cell address of the command to the operand bits. The core cell address of most commands is the bit information contained in the P Register. However, for the object instruction of XEC and Quasi (i. e., locations 40 through 77) commands, the bit information contained in  $I_A, 13-0$  is the address of the command. Relative addressing of a command is specified by a "one" in bit 14 (\*), of the command. All commands may be relative addressed with the exception of the following:

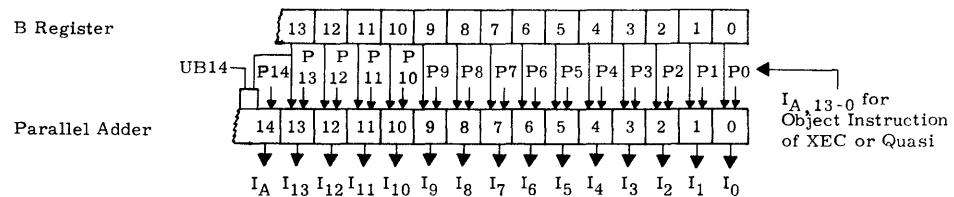
- GEN 1
- GEN 2
- GEN 3
- TXH

Relative addressing, as well as indexing and the normal incrementation of P, allows addressing of up to  $32,768$  memory locations. Relative addressing permits the program to address a memory location in the range of plus  $8,191_{10}$  to minus  $8,192_{10}$  with respect to the instruction address. The plus or minus direction is determined by bit position 13 of the instruction operand address. When bit 13 is a "one", the operand address is in 2's complement form and relative addressing occurs in the negative direction.

Relative addressing for all commands occurs during Sequence Control State 1. The relative addressed command is "fetched" from memory and gated to the B Register. The relative addressed command will normally be addressed from the P Register. However, if the relative addressed command is the object instruction of XEC or Quasi, it will be addressed from the I Register. Although BTS, BTR (with the jump condition true), and BRU commands address memory from I for the next instruction, the P Register will also contain the address of this next instruction and this value is used for relative addressing.

From the B Register, bits 23 through 14 of the relative addressed command are gated to the I Register for decoding. With bit 14 of the command a "one" and if the command is not a GEN, then IR14 will be set indicating that the command is relative addressed.  $B_{23-00}$  is gated to the Parallel Adder. Because it is relative addressed,  $B_{13}$  is also applied to bit 14 of the Adder ( $G1UB14$ ). The contents of the P Register (or I Register for XEC and Quasi) are also gated to the Adder where the summation occurs. The result of this summation is then gated to  $I_A, 13-0$  as the effective operand address of the command. The command is then executed in the normal manner using this effective operand address.

The following examples illustrate the summation and the expansion of  $B_{13}$  to bit 14 of the Adder. Both positive and negative addressing is illustrated.



Example 1:

Command Operand in B Register

13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	0	0	0	0	0	0	0	1	1

= +  $6,147_{10}$

Command Address in P (or I) Register

13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0

=  $14,336_{10}$

Parallel Adder Result

13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	0	0	0	0	0	0	0	1	1

=  $20,483_{10}$   
Gated to  $I_A, 13-0$  as Effective Operand Address (Z)

Example 2:

Command Operand in B Register

13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0

= -  $8,192_{10}$

Command Address in P (or I) Register

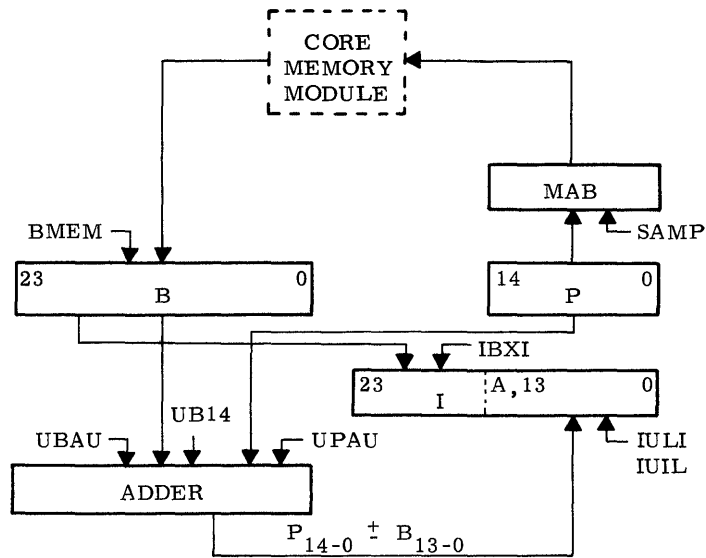
13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0

=  $12,228_{10}$

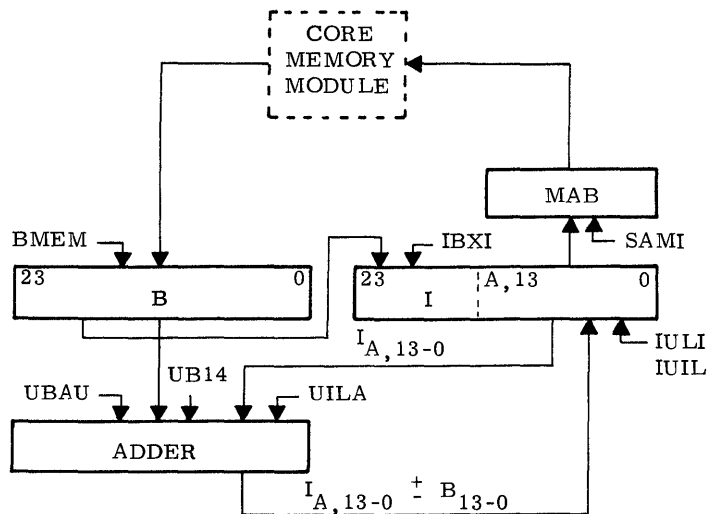
Parallel Adder Result

13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0

=  $4,096_{10}$   
Gated to  $I_A, 13-0$  as Effective Operand Address (Z)

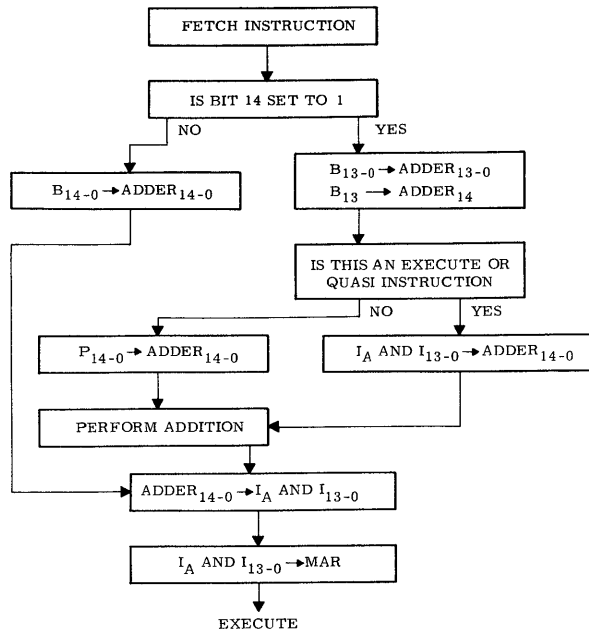


(1) All Relative Addressed Commands Except Quasi and XEC Object Instructions.

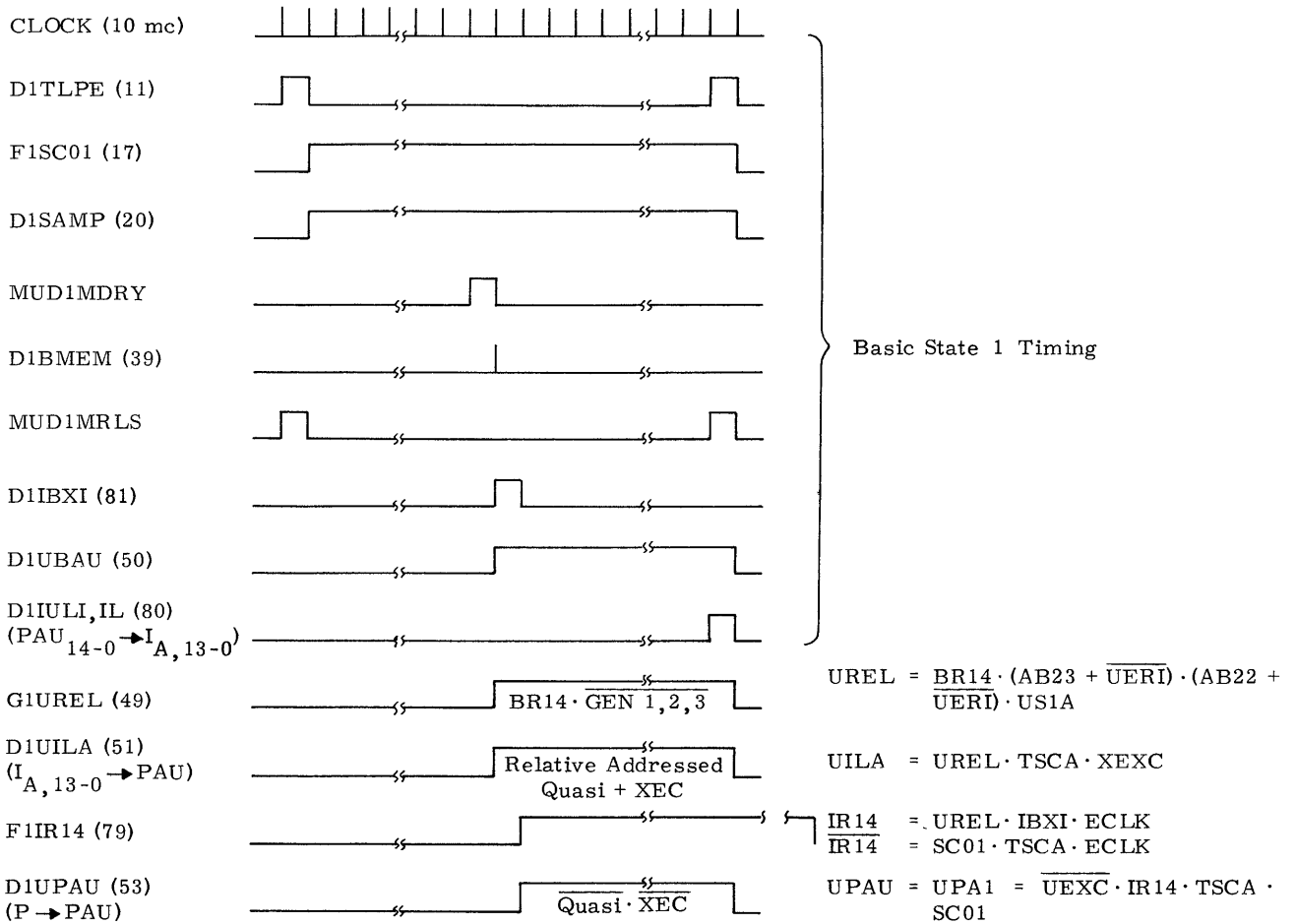


(2) Object Instruction of XEC or Quasi.

RELATIVE ADDRESSING BLOCK DIAGRAM

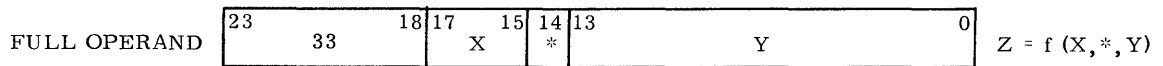


RELATIVE ADDRESSING FLOW CHART



RELATIVE ADDRESSING TIMING DIAGRAM

# SPB - SAVE PLACE AND BRANCH



SPB places the status of the Overflow (F1UOFL), Permit Automatic Interrupt (F1WPMT), Test (F1ETST), Trapping Mode (F1MTRM), and Quasi (F1XQUA) flip-flops in bit position 22 through 18, respectively, of index cell 1. The contents of the P Register, plus 1\*, are stored in bits 14 through 0 of index cell 1. The SPB command resets the Permit Automatic Interrupt flip-flop, inhibiting inhibitable interrupts. The Trapping Mode flip-flop is cleared if the SPB command is executed due to a trap error or if the SPB command is executed as a result of an Automatic Program Interrupt.

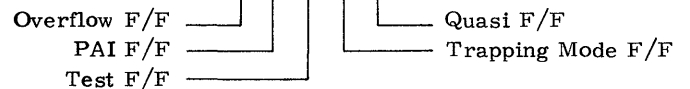
\* NOTE

The address that is stored by the SPB command is normally the address of the SPB command plus 1. However, there are exceptions to this. If an SPB command is performed due to the use of XEC or Quasi commands, the address of the XEC or Quasi plus 1 is stored. If SPB is executed immediately following an acknowledged Automatic Program Interrupt or memory protect trap, P contains the address of current program control. In these cases only, the value saved in index word 1 is P, the first unexecuted instruction in the interrupted program (i. e., the address to which program control will return when the interrupt has been serviced).

The SPB command is executed during Sequence Control State 4 (SC04). During the next Sequence Control State 1 (SC01) memory is addressed from I and the contents of IA<sub>13-0</sub> are gated to the P Register to complete the transfer of program control.

During State 4, memory cell 1 is addressed (G0MX00). The contents of the P Register (D1UPAU) and the status of the aforementioned flip-flops (D1UIOU) are gated to the Parallel Adder Unit. From the Adder, this data is gated to the B Register (D1BAUL, AUU), and from B, to memory cell 1. At Time 3 of State 4, the Permit

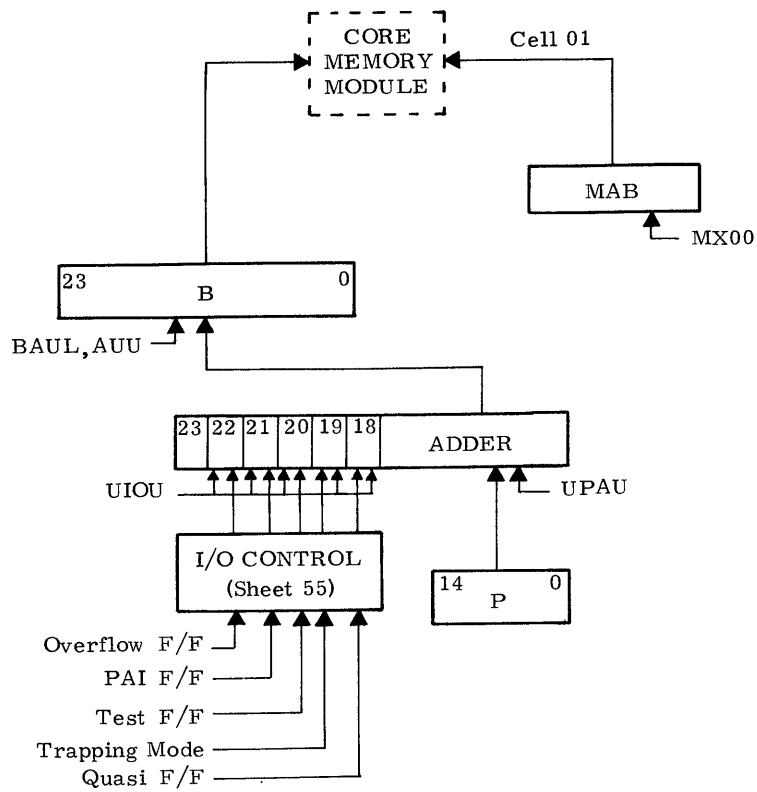
Non-indexed Word Times.	2 (S1, S4)																				
Interruptable Following Execution?	No																				
CHANGES FOLLOWING EXECUTION																					
A <sub>23-0</sub>																					
Q <sub>23-0</sub>																					
P <sub>14-0</sub>	Z																				
F1WPMT	Reset																				
F1UOFL																					
F1ETST																					
F1MTRM	Reset if SPB is Executed Because of Trap Error or API																				
J <sub>4-0</sub>																					
Memory Cell 1	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 5%;">23</td><td style="width: 5%;">22</td><td style="width: 5%;">21</td><td style="width: 5%;">20</td><td style="width: 5%;">19</td><td style="width: 5%;">18</td><td style="width: 5%;">17</td><td style="width: 5%;">15</td><td style="width: 5%;">14</td><td style="width: 5%;">0</td> </tr> <tr> <td style="text-align: center;">0</td><td></td><td></td><td></td><td></td><td></td><td style="text-align: center;">0</td><td></td><td style="text-align: center;">C (P) + 1 *</td><td></td> </tr> </table>	23	22	21	20	19	18	17	15	14	0	0						0		C (P) + 1 *	
23	22	21	20	19	18	17	15	14	0												
0						0		C (P) + 1 *													



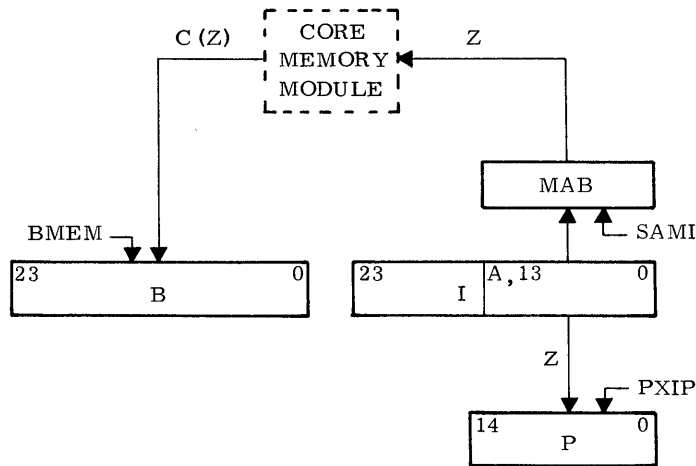
## COMMAND CHARACTERISTICS

Automatic Interrupt flip-flop (F1WPMT) is cleared. Last Pulse of State 4 will clear the Quasi flip-flop, F1XQUB. The Trapping Mode flip-flop (F1MTRM) is cleared at Last Pulse Envelope if the SPB command was executed as the result of a trapping mode error, or if the SPB command was executed as the result of an Automatic Program Interrupt.

Following State 4, State 1 is entered to "fetch" the command located in memory cell Z. Memory is addressed from IA<sub>13-0</sub> (D1SAMI) and the contents of IA<sub>13-0</sub> are transferred to the P Register. In this manner, core cell Z is addressed and Z is transferred to the P Register for subsequent program control.

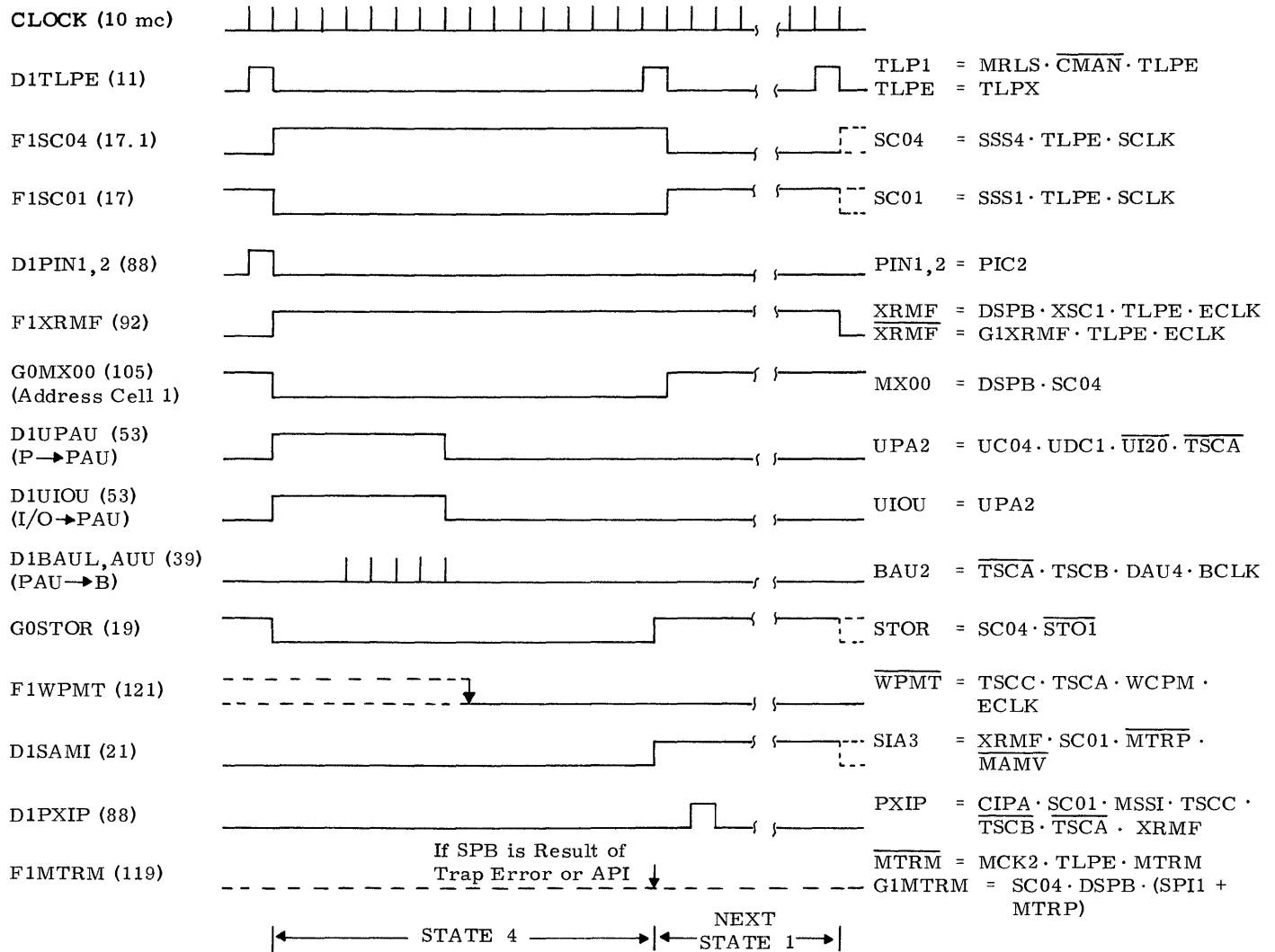


Sequence State 4



Next Sequence State 1

SPB BLOCK DIAGRAM



SPB TIMING DIAGRAM

# STA - STORE CONTENTS OF A

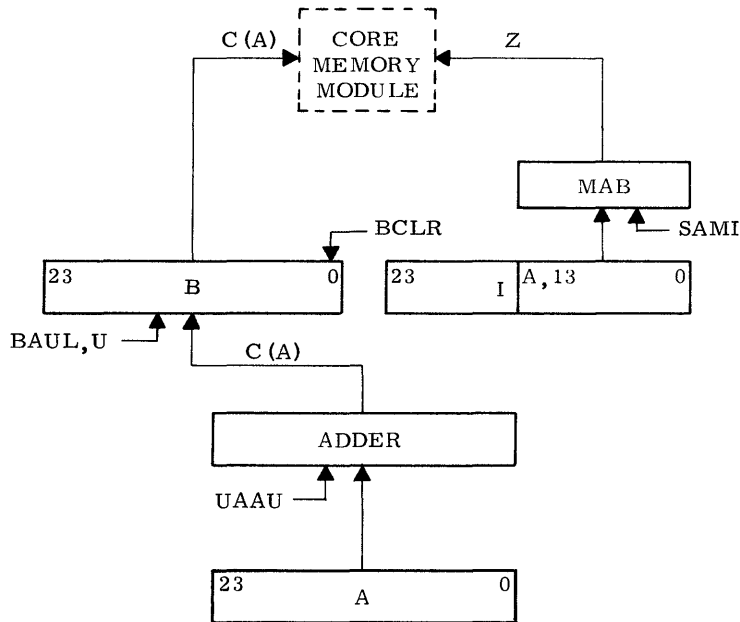
FULL OPERAND	23	18	17	15	14	13	Y	0	$Z = f(X, *, Y)$
	32	X	*						

STA places the contents of the A Register into memory location Z. The contents of the A Register are unchanged.

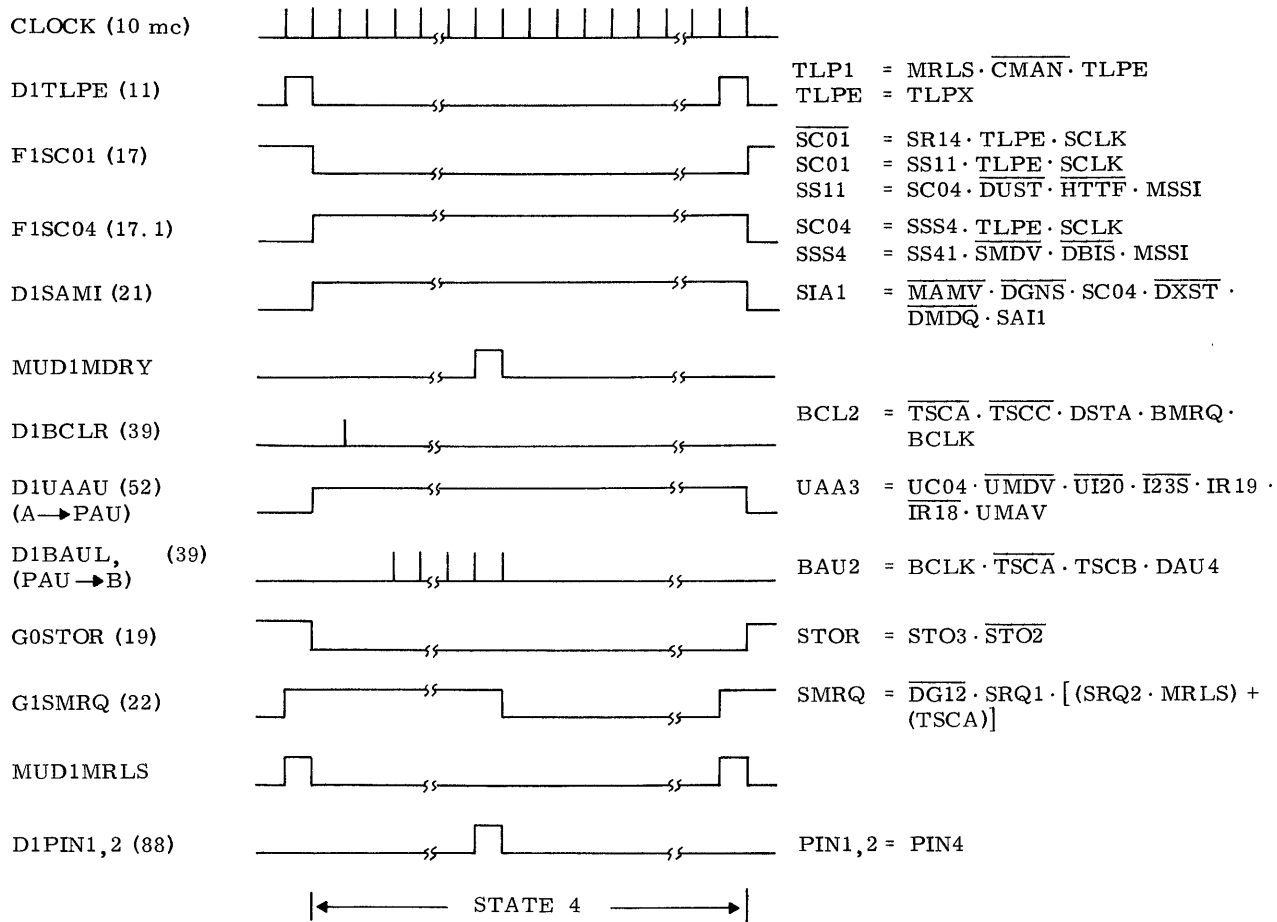
A non-indexed STA command is executed during Sequence Control State 4 (SC04). Memory location Z is addressed from  $I_{A, 13-0}$  (DISAMI) during State 4. The B Register is cleared (BCLR) and then the contents of the A Register are gated (UAAU), via the Adder Unit, to the B Register (BAUL,U). From the B Register, the data from the A Register is transferred to the memory module where it is stored in location Z. Following Sequence Control State 4, Sequence Control State 1 is entered to "fetch" the next command.

Non-Indexed Word Times.	2 (S1,S4)
Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	
Memory Z	$C(A)$

COMMAND CHARACTERISTICS



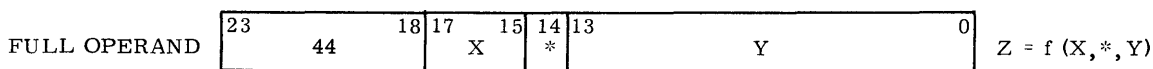
STA BLOCK DIAGRAM



STA TIMING DIAGRAM



# STQ - STORE CONTENTS OF Q

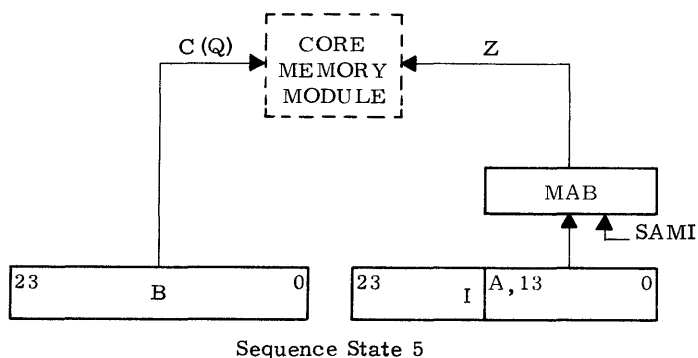
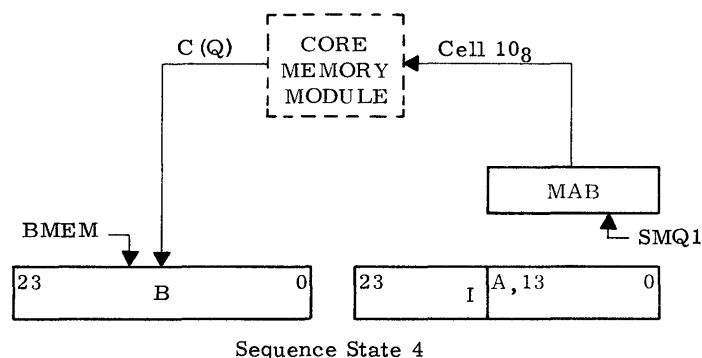


STQ places the contents of the Q Register (memory location  $10_8$ ) into memory location Z. The contents of the Q Register are unchanged by the STQ command.

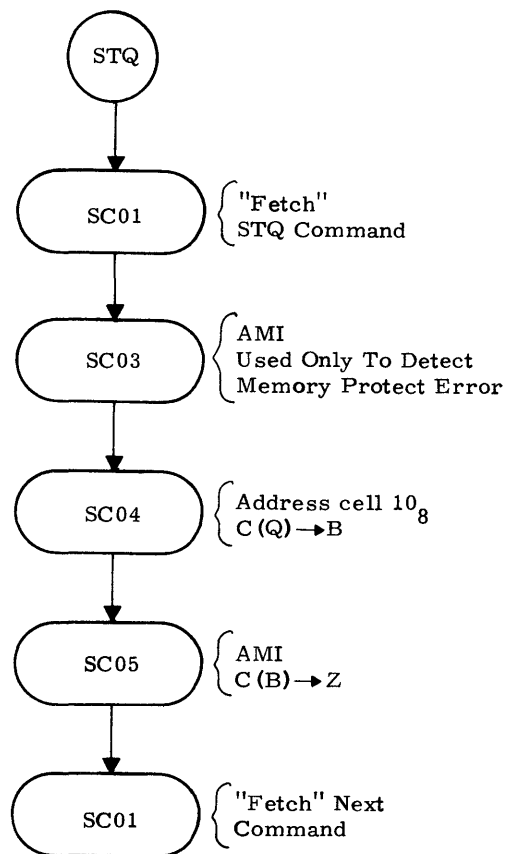
Sequence Control States 3, 4, and 5 are required to execute the STQ command. During State 3, memory is addressed from  $I_A, 13-0$  and the contents of memory location Z are gated to the B Register, but this is only used to detect a Memory Protect Error. Following State 3, State 4 is entered. During State 4, memory location  $10_8$  is addressed by enabling G0MX03. The contents of cell  $10_8$  (Q Register) are gated to the B Register. Following State 4, State 5 is entered. During State 5, memory is addressed from  $I_A, 13-0$  (SAMI) and the contents of the B Register are stored in memory location Z. In this manner, the contents of the Q Register are stored in memory location Z.

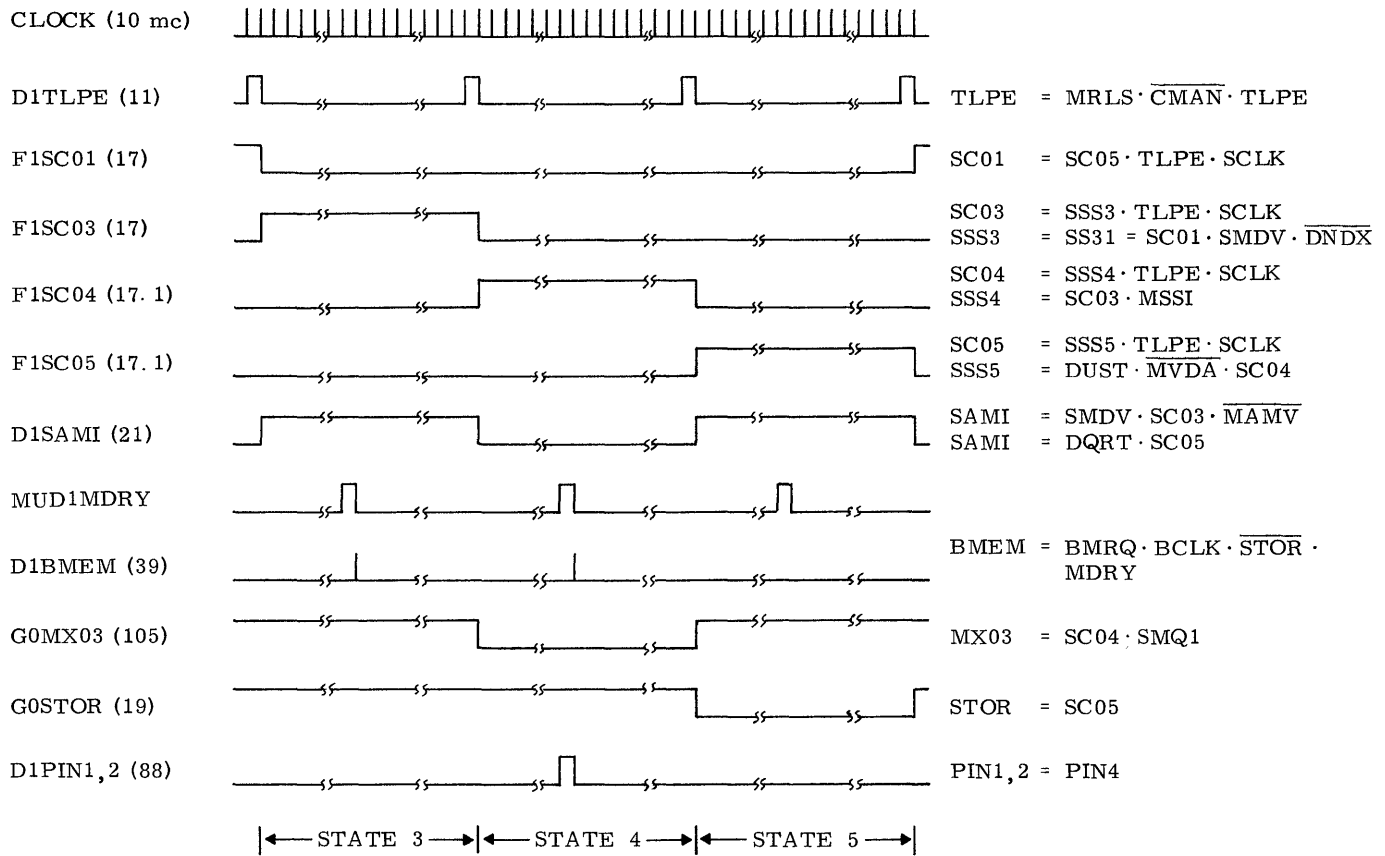
Non-Indexed Word Times.	4 (S1, S3, S4, S5)
Interruptable Following Execution?	Yes
<b>CHANGES FOLLOWING EXECUTION</b>	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	C (P) +1
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	
Memory Z	C (Q) i. e., memory cell $10_8$

COMMAND CHARACTERISTICS



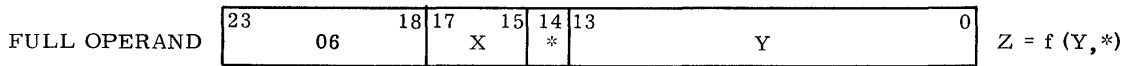
STQ BLOCK DIAGRAM





STQ TIMING DIAGRAM

# STX - STORE X LOCATION INTO Z

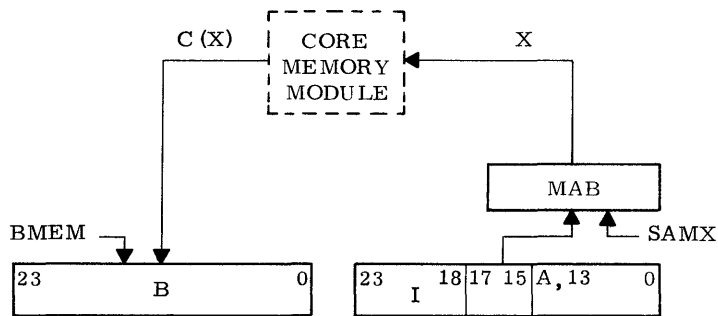


STX stores the contents of the indicated X core cell into memory location Z. The STX command may not be automatically modified since bits 15, 16, and 17 are used to specify the X cell to be stored. If no index address is indicated, the STX command is executed as a DMT command.

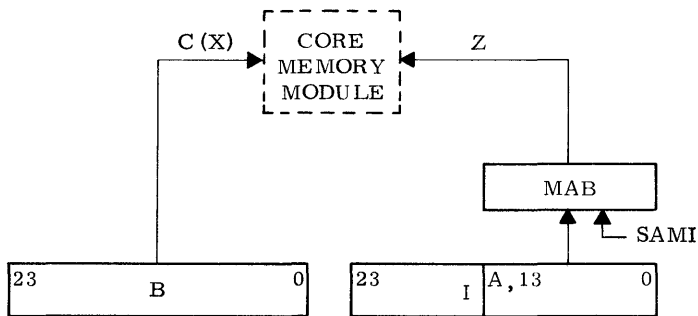
The STX command is executed during Sequence Control States 2, 3, and 4. During State 2 memory is addressed from  $I_{A,13-0}$  for detection of a Memory Protect (optional) violation. Following State 2, State 3 is entered. During State 3, memory is addressed from  $I_{17-15}$  ( $G1SAMX$ ) and the contents of the addressed X cell are gated to the B Register. Following State 3, State 4 is entered. During State 4, memory is addressed from  $I_{A,13-0}$  and the contents of the B Register are stored in memory location Z. In this manner, the contents of the addressed X cell are stored in location Z.

Word Times.	4 (S1, S2, S3, S4)
Interruptable Following Execution?	Yes
CHANGES FOLLOWING EXECUTION	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	$C(P) + 1$
F1WPMT	
F1UOFL	
F1ETST	
$J_{4-0}$	
Memory Z	$C(X)$

## COMMAND CHARACTERISTICS

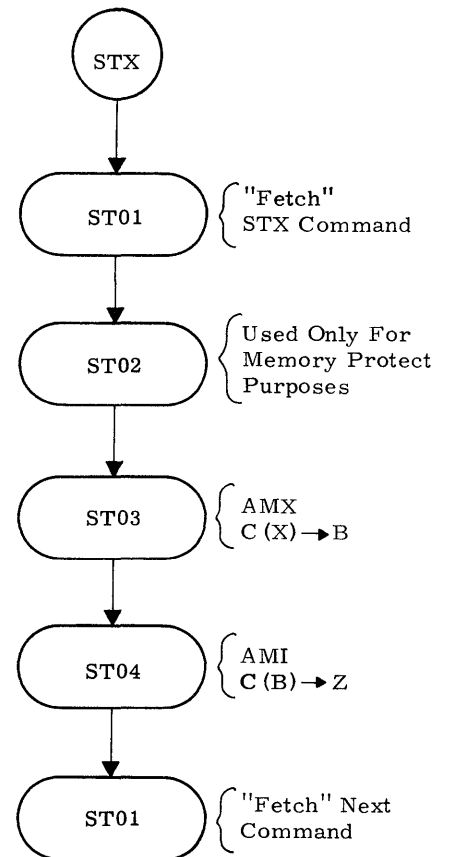


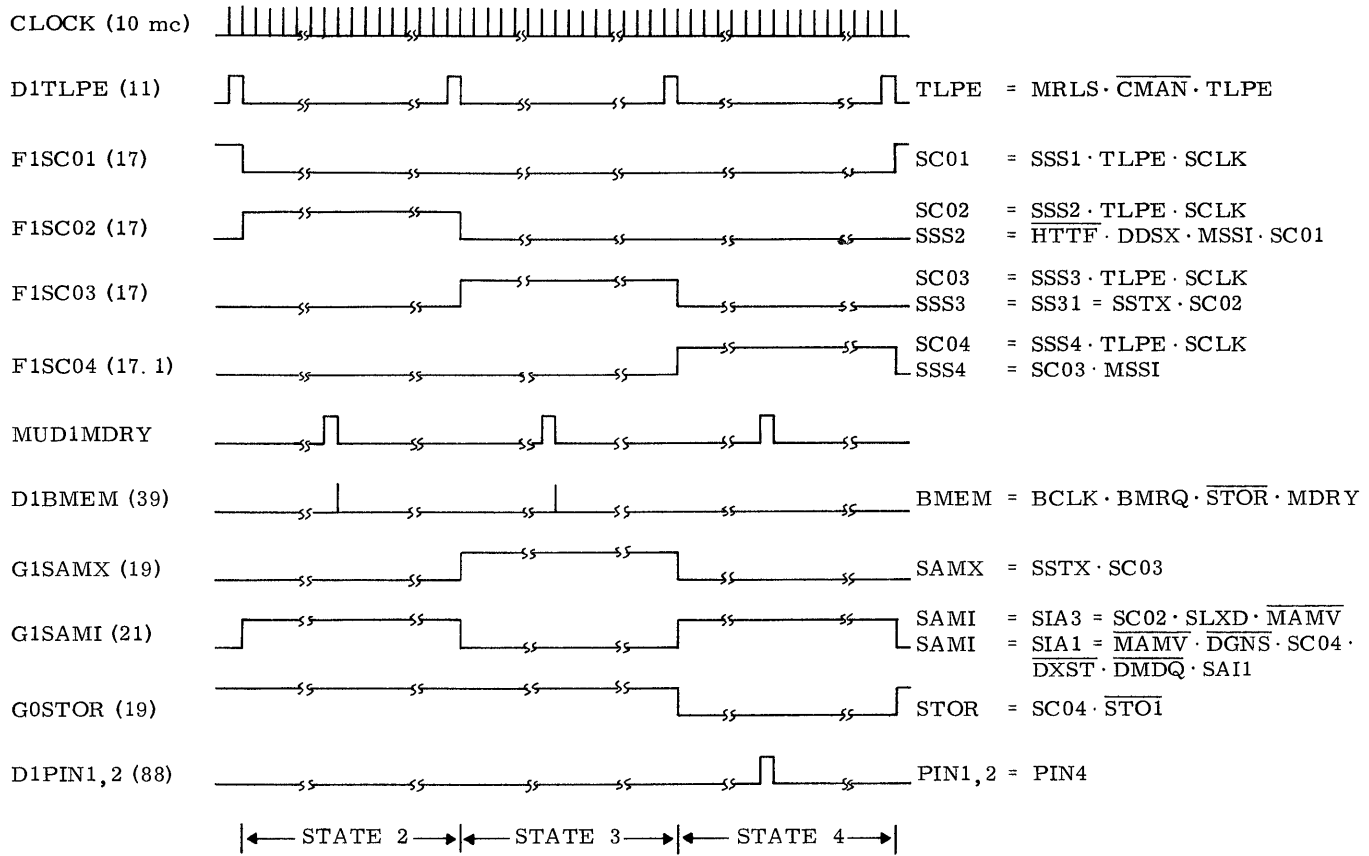
Sequence State 3



Sequence State 4

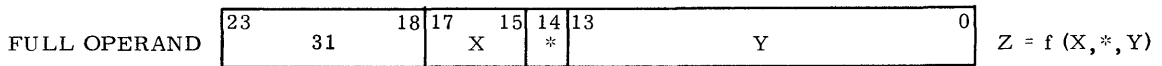
## STX BLOCK DIAGRAM





STX TIMING DIAGRAM

## SUB - SUBTRACT Z FROM A



SUB performs algebraic subtraction of the contents of core cell Z from the contents of the A Register. The result of the subtraction is stored in the A Register. If the result is too large to be stored in the 23 bits of A (i. e. , more negative than  $-2^{23}$  or more positive than  $2^{23}-1$ ), the Overflow flip-flop (F1UOFL) is set.

A non-indexed SUB command is executed during Sequence Control State 4 (SC04). Memory location Z is addressed from IA<sub>13-0</sub> (DISAMI) during State 4. The contents of memory location Z are gated to the B Register by D1BMEM during the Clock pulse of Memory Data Ready (MUD1MDRY). From B, the complement of the contents of memory location Z and the Enable Carry signal (GIUENC) are gated to the Adder Unit. The Enable Carry signal adds 1 to the complemented contents of core memory cell Z forming the 2's complement of Z. The contents of the A Register are also gated to the Adder Unit where the summation of A and the 2's complement of Z occurs. The result is gated (DIAAUL,U) back to the A Register. This is the difference between the contents of the A Register and core cell Z.

If arithmetic overflow - either positive or negative occurs during the summation, the Overflow flip-flop (F1UOFL) is set. The following examples illustrate (a) a positive overflow, and (b) a negative overflow. For simplicity, five bit registers are illustrated. Consider the most significant bit as bit 23, and the next most significant bit as bit 22.

Consider:

(A = +8) - (Z = -11) = +19. Nineteen is too large to be contained in 4 bits plus a sign bit; thus, the Overflow flip-flop is set.

Contents of Z	1 0101 (-11)	
1's Complement of Z	0 1010	
+ Enable Carry	1 (+1)	
2's Complement of Z	0 1011	
+ Contents of A	0 1000 (+8)	
Total	1 0011	Overflow Set = 23S · 23C · 22C.

(a) Positive Overflow.

Consider:

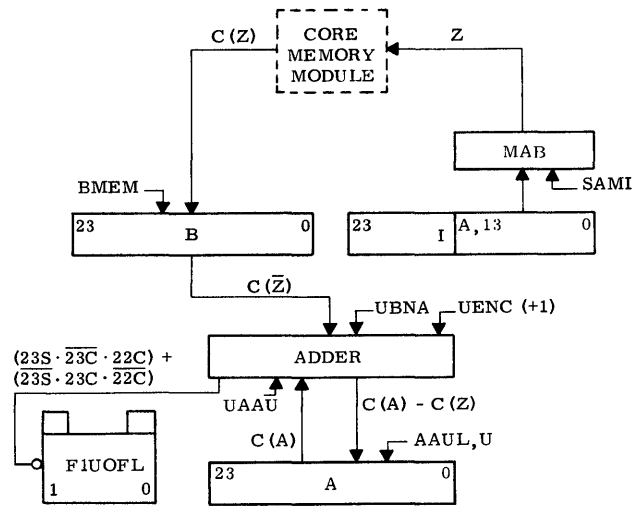
(A = -7) - (Z = +13) = -20. Negative twenty is too large to be contained in 4 bits plus a sign bit; thus, the Overflow flip-flop is set.

Contents of Z	0 1101 (+13)	
1's Complement of Z	1 0010	
+ Enable Carry	1 (+1)	
2's Complement of Z	1 0011	
+ Contents of A	1 1001 (-7)	
Total	0 1100	Overflow Set = 23S · 23C · 22C.

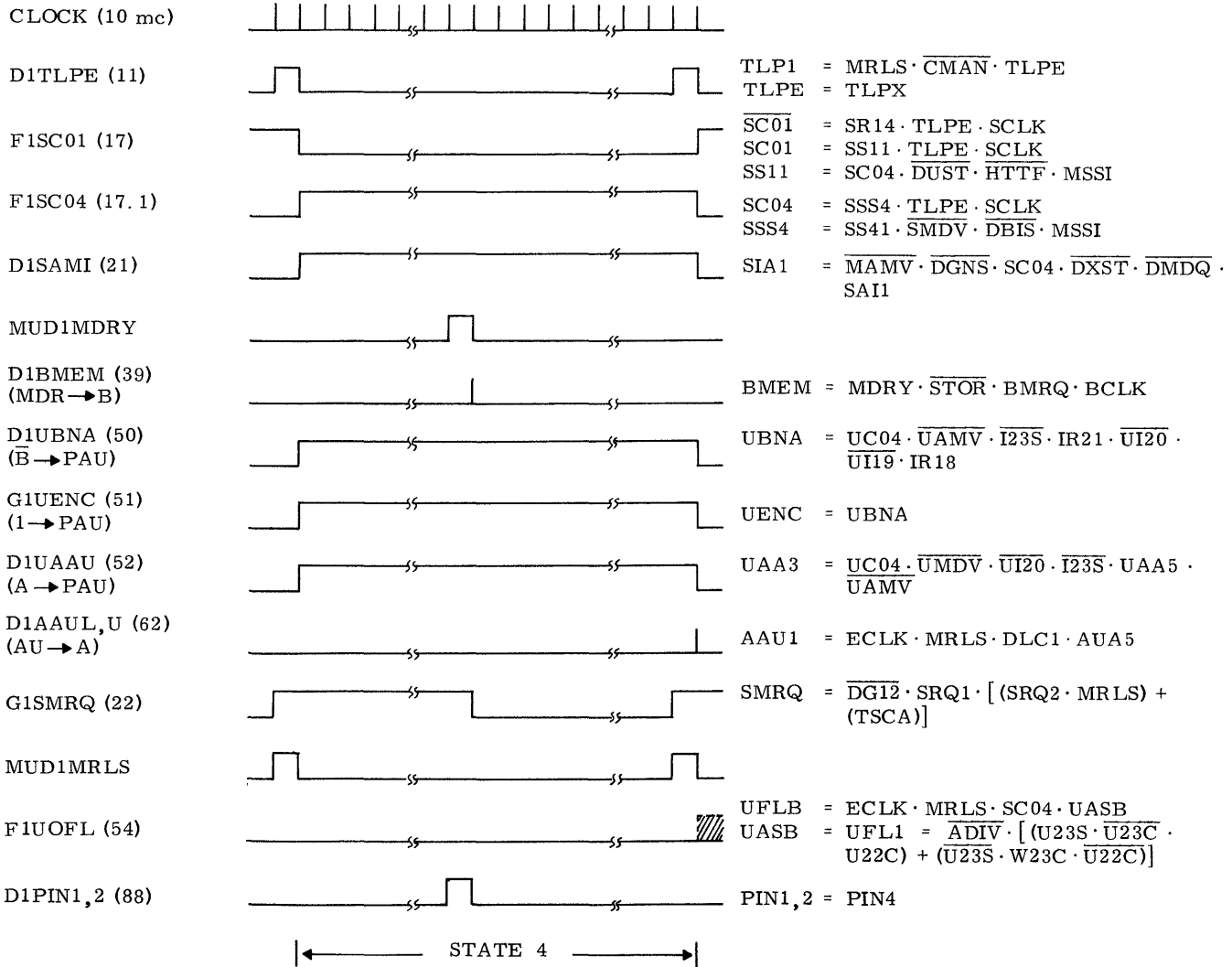
(b) Negative Overflow.

Non-Indexed Word Times.	2 (S1, S4)
Interruptable Following Execution?	Yes
<b>CHANGES FOLLOWING EXECUTION</b>	
A <sub>23-0</sub>	C (A) - C (Z)
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C (P) +1
F1WPMT	
F1UOFL	Set if Overflow Occurs
F1ETST	
J <sub>4-0</sub>	
Memory Z	

### COMMAND CHARACTERISTICS



SUB BLOCK DIAGRAM



SUB TIMING DIAGRAM

# TIM/TOM - TABLE INPUT TO MEMORY/TABLE OUTPUT FROM MEMORY

The Table Input to Memory (TIM) and Table Output from Memory (TOM) is an optional capability that provides a faster I/O rate and eliminates the normal software "housekeeping" chores required when using GEN 2 commands. This is a semi-direct to memory I/O scheme where data flow is from the I/O Device, to the B Register, to memory, or from memory, to the B Register to the I/O Device. The TIM/TOM method of input/output does not disturb the A, P, Q, J, or Index Registers.

TIM/TOM is not an instruction in the true sense of the word (i. e., such as GEN 2). It is a hardware sequence that is activated by an interrupt from the I/O Device. Each TIM or TOM channel is operated in conjunction with a unique interrupt that is activated by the device using the TIM or TOM function. Each of these interrupts is wired to a matrix board which determines the device code and the operation, TIM or TOM, for that device.

When an interrupt from an I/O Device using the TIM/TOM function occurs, the word stored in the interrupt response address is "fetched" from memory. The word stored in this address will be a TIM/TOM Control Word. This Control Word is used to define how the data is to be organized (i. e., how many characters or bytes are contained in each 24 bit word), where in the first 16K of core memory the table of words is located or to be located, and how many words are to be transferred. The format of the TIM/TOM control word is shown below:

23	18	17	16	15	14	13		0
	N		C		P		Y	

## N Field - Word Count:

This field contains the current number, in 1's complement form, of words remaining to be read (TOM) or written (TIM). The maximum number of words is 63.

## C Field - Character Count:

This field specifies at any given point its time, the number of characters still to be packed (TIM) or unpacked (TOM) in the word being operated on. Initially, C is normally set equal to P.

## P Field - Packing Mode:

This field specifies the number, in 2's complement form, of characters to be packed or unpacked in each word.

- 00 = 4 characters per 24 bit word
- 01 = 3 characters per 24 bit word
- 10 = 2 characters per 24 bit word
- 11 = 1 character per 24 bit word.

## Y Field - Starting Address:

This field initially specifies the starting address (-1) of the data table. This field is then incremented to address the word being operated on at any given point in time.

## NOTE

Because the N field may specify up to 63 words and 1, 2, 3, or 4 characters may be packed or unpacked in each word, the number of characters that may be transferred in a block is 252 six-bit characters, 189 eight-bit characters, 126 twelve-bit characters, or 63 twenty-four-bit characters.

As many as 48 different TIM/TOM channels may be included as an option in the 4022A System. Each channel has a defined interrupt response address. Associated with each interrupt response address, is a defined device code and input or output enable. The device code and the input or output enable is obtained from a wired matrix board. Therefore, individual interrupts are required for each TIM and each TOM channel.

Each TIM/TOM function requires 12.7 or 30.7  $\mu$ s to execute depending on whether a high-speed or low-speed channel (K3 = 7) is used. Therefore, burst mode transfer rates of up to 62,000 words or characters per second is possible.

## BASIC OPERATION

Although the TIM and TOM methods of inputting and outputting data are very similar, the general operation of TIM and TOM are listed separately to provide a clearer understanding.

## TIM

1. The program loads the assigned API Response Address with the Control Word.
2. When an interrupt occurs from the input device, a matrix board generates the associated device code and enables the input function.
3. The Control Word is "fetched" from the interrupt response address. The C field (bits 16 and 17) of the Control Word is compared with the P field, and if they are equal, the Y address field is incremented by one. The C field is always incremented by one to indicate that one character will be transferred during this cycle. If a carry occurs from the C field (C incremented to 00) it is applied to the N field. A carry will occur from the C field during the cycle that the last character of a word is read in from the device. In this manner, the word count is incremented when the last character of a word is received. When the last character of the last word is received, the N field is incremented to 77<sub>8</sub> and an "Echo" interrupt is generated. When C is equal to 00, C is set equal to the P field to prepare for the next word.
4. The updated Control Word is stored back in the API response address. This updated Control

Word will be addressed when the next interrupt occurs.

5. The contents of the Y Address specified by the Control Word is gated from memory to the B Register. If this is the first character of this word to be obtained, the B Register is cleared. If more than one character is to be stored in the word, the B Register is then shifted left in a circular fashion. That is, if 2 characters are to be stored in this word, B is shifted left circular 12 places; if 3 characters are to be stored in this word, B is shifted left circular 8 places, etc.
6. The contents of the B Register are gated to the Parallel Adder where the character from the device is ORed into the least significant bits. The result is then gated back to the B Register.
7. The contents of B are then stored back into the memory location specified by the Y field of the Control Word.
8. The above sequence will occur each time the interrupt occurs until the proper number of words have been read. When the proper number of words have been read, an "Echo" interrupt is generated, as described in step 3. After the "Echo" interrupt is generated, the program will normally load the interrupt response address with another TIM control word in preparation for the next input function.

## TOM

1. The program loads the API response address with the Control Word.
2. When an interrupt occurs from the output device, a matrix board generates the associated device code and enables the output function.
3. The Control Word is "fetched" from the interrupt response address. The C field of the control word is compared with the P field and if they are equal, the Y address is incremented. The C field is always incremented by one to indicate that one character will be transferred during this cycle. If a carry occurs from the C field (i. e. , C is incremented to 00), it is applied to the N field. A carry will occur from the C field during the cycle that the last character of a word is transferred to the device. In this manner, the word count is incremented when the last character of a word is transferred to the device. When the last character of the last word is transferred, the N field will be incremented to 77<sub>8</sub> and an "Echo" signal is generated. Whenever the C field is equal to 00, the C field is set equal to the P field, to prepare for the next word.
4. The updated Control Word is stored back in the API response address. This updated Control

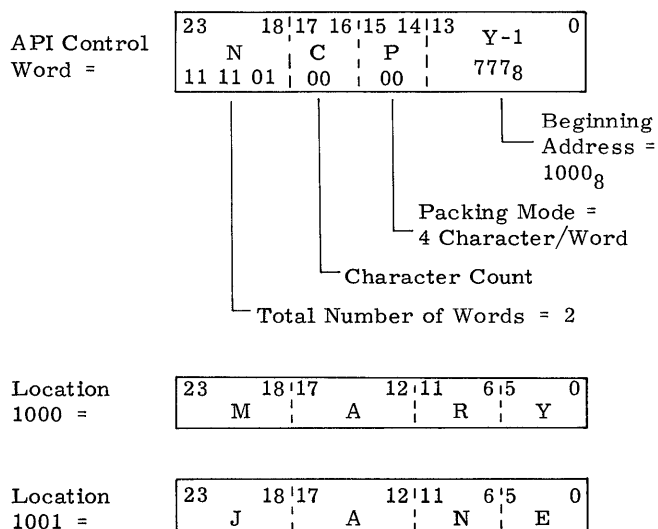
Word will be addressed when the next interrupt occurs.

5. The contents of the Y Address specified by the Control Word is gated from memory to the B Register. If more than one character is contained in a word, the B Register is then shifted left circular. That is, if 2 characters are contained in a word, the B Register is shifted left circular 12 places; if 3 characters are contained in the word, B is shifted left circular 8 places, etc.
6. The contents of the B Register are gated to the Adder where the least significant bits (1 character) is transferred to the addressed output device. As previously mentioned, the device address (code) is generated by a matrix board from the interrupt input.
7. The contents of B are then stored back in the memory location specified by the Y Address of the Control Word.
8. The above sequence will occur each time the interrupt occurs, until an "Echo" interrupt is generated, as described in step 3. After the "Echo" interrupt is generated, the program will normally load the interrupt response address with another TOM control word in preparation for the next function.

To further illustrate the operation of the TOM function, the following example is provided. This example illustrates the organization of data to be transferred, how the Control Word is updated, and the general sequence of events during a TOM operation.

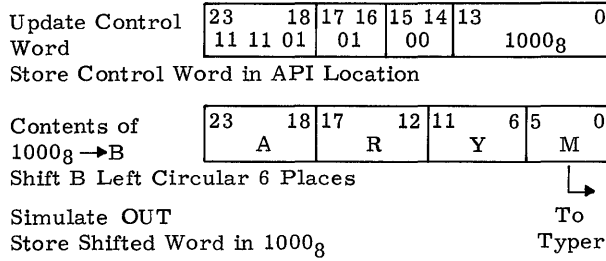
TOM Example:

This example illustrates the basic operation of using a TOM capability to type out the words MARY JANE coded and previously stored in memory.

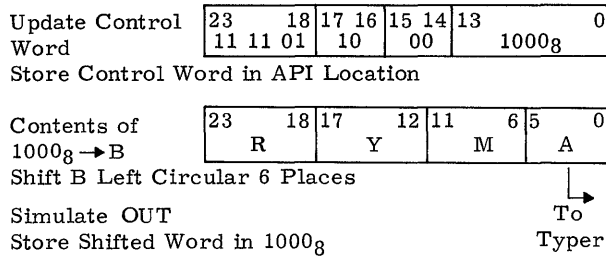




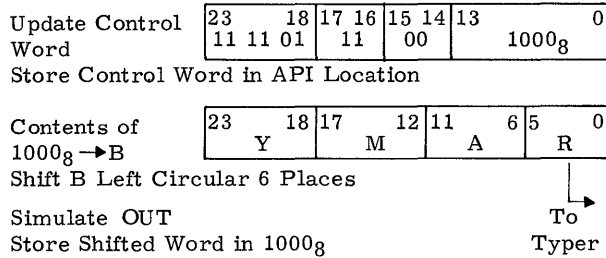
1st INTERRUPT:



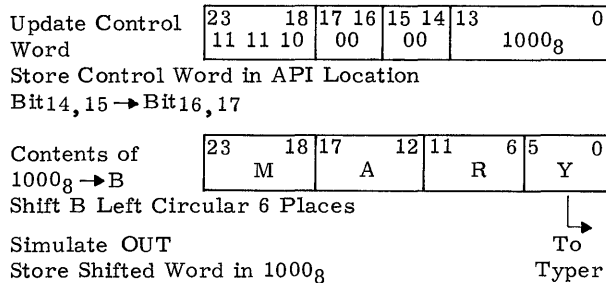
2nd INTERRUPT:



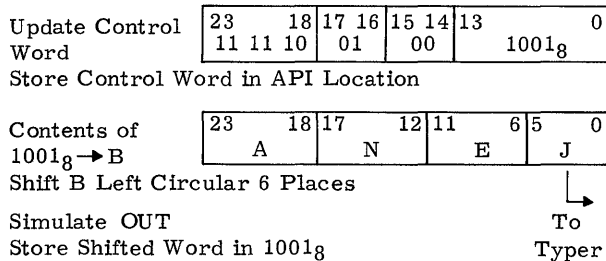
3rd INTERRUPT:



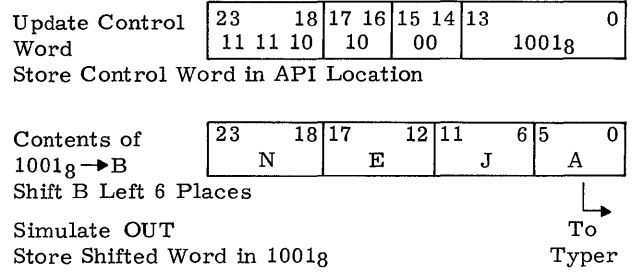
4th INTERRUPT:



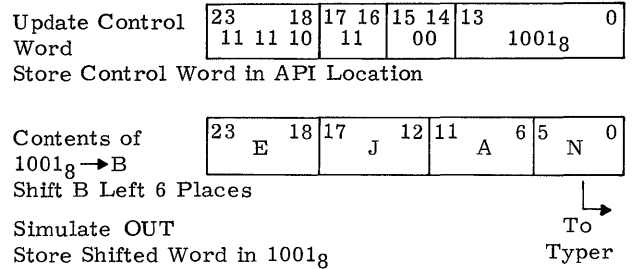
5th INTERRUPT:



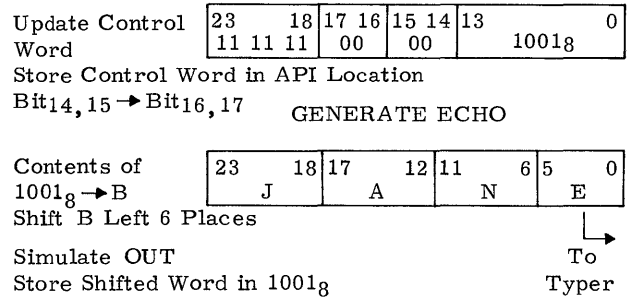
6th INTERRUPT:



7th INTERRUPT:



8th INTERRUPT:



The ECHO signal generated, because this is the last character of the table, causes another interrupt. If desired, this interrupt will change the Control Word for additional data outputs. If, however, another interrupt occurs for the Control Word shown above, the operation will be terminated before data is transferred.

## THEORY OF OPERATION

The TIM/TOM function is initiated upon receipt of an interrupt at an interrupt channel wired for the TIM/TOM function. This interrupt is wired to a matrix board to define the operation of TIM or TOM (S bit = 5 or 4) and to define the device code (K3, K2, K1, and K0) address of the device causing the interrupt. The TIM or TOM function is then executed in Sequence Control States 1, 3, 4, and 5. Fig. TIM/TOM.1 illustrates the basic functions performed in each of these Sequence States. A detailed flow chart of the TIM/TOM function is contained in Fig. TIM/TOM.2. Refer to these aids during the following discussion.

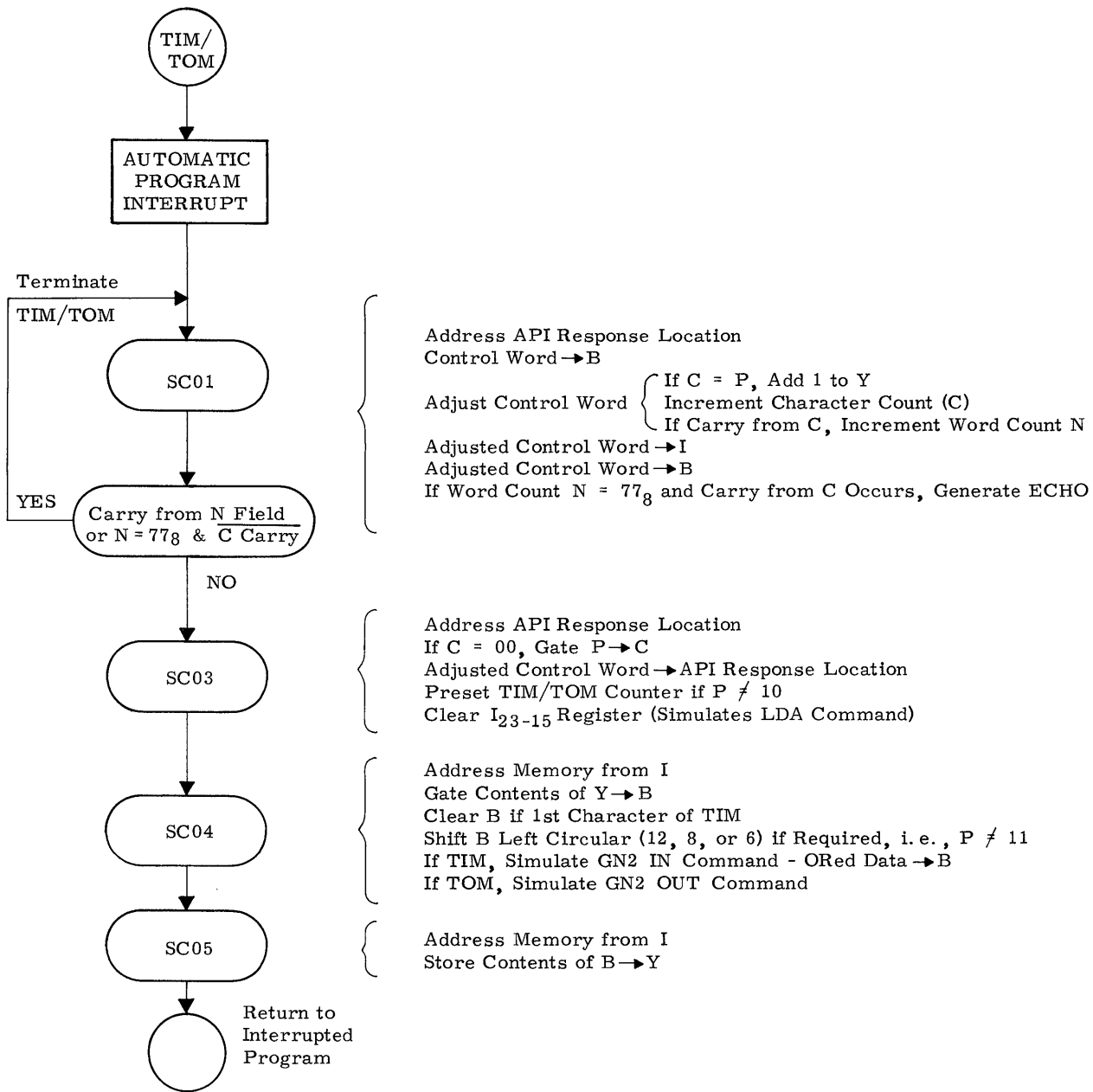


Fig. TIM/TOM. 1. Sequence State Flow Chart

### Device Code Matrix

A diode matrix board (RTTA1) is used to generate the device and operation code from an interrupt input defined for use as a TIM or TOM function. This matrix board is shown on sheet 133 of the Arithmetic Unit logic.

Each TIM and TOM interrupt is wired to one of the input drivers, D0HL01 through D0HL24 (future expansion will provide an additional 24 input drivers). From each of these drivers, diodes are wired to enable only the inverters (N1HT00 through N1HT11) corresponding to the

K bits of the device code. Also, all input drivers used for TIM functions will have a diode wired to enable N1HT12. The following list of inverters illustrates the K bit decode pattern. This K bit decode corresponds to the device code specified by the GEN 2 commands.

N1HT00	}	K0
N1HT01		
N1HT02		
N1HT03	}	K1
N1HT04		
N1HT05		

N1HT06	}	K2
N1HT07		
N1HT08		
N1HT09	}	K3
N1HT10		
N1HT11		
N1HT12	}	S

As an example of the operation of the diode board, consider that a TIM function for a device having a device code (K3 - K0) of 1100, has an interrupt line connected to D0HL01. Diodes would then be jumpered from the output of D0HL01 to enable N1HT09 and N1HT06. Since it is an input function (TIM), a diode will also be jumpered from D0HL01 to enable N1HT12. GOHTTA is enabled when any TIM or TOM interrupt occurs and is used to control the Arithmetic Unit for the TIM/TOM function.

With HT09 and HT06 enabled and all other device code inverters disabled, the device code of K3 - K0 = 1100 is enabled, in the same manner as decoding for GEN 2 commands. Refer to logic sheets 99 and 100.

Any TIM operation enables N1HT12. N1HT12 is then used in S bit decoding to enable an S = 5 (IN) function. Whenever N1HT12 is disabled, during a TIM/TOM function, an S = 4 (OUT) function is enabled indicating a TOM operation.

### Sequence State 1

Block and timing diagrams of State 1 operation are contained in Figs. TIM/TOM. 3 and 4. During State 1, memory is addressed from the API (D1SAMW) and the TIM/TOM Control Word from the interrupt response address is gated to the B Register. The I Register is cleared (D0ICU1, CUL). If C (bits 16 and 17) is equal to P (bits 14 and 15), a "one" is set in I<sub>0</sub>. A "one" is unconditionally set into I<sub>16</sub>. The contents of I (I<sub>16</sub> = 1 and I<sub>0</sub> = 1 if C = P) are gated to the Adder (UIAU, LA). The contents of B (Control Word) are also gated to the Adder (D1UBBU) where it is summed with the contents of I. The result of this summation will then add 1 to C, add 1 to N if a carry results from C, and add 1 to Y if C was equal to P. The result of this summation is then gated to both the I and B Registers.

If during the summation, a carry resulted from C (PAU<sub>17</sub>) causing N to be incremented to 778, an Echo signal (G1WEKO) is generated to indicate that this is the last character of the last word to be transferred. This Echo signal is then used by the program to determine that the transfer is complete and that a new Control Word must be stored in the Interrupt response address if further data transfers are desired.

If during the summation, a carry resulted from PAU<sub>23</sub>, indicating that N was incremented beyond 778, or if N is equal to 778 and no carry results from C (i.e., PAU<sub>17</sub>) the operation is terminated (G1HSTP). This indicates that an interrupt occurred after the last word specified by the Control Word was transferred.

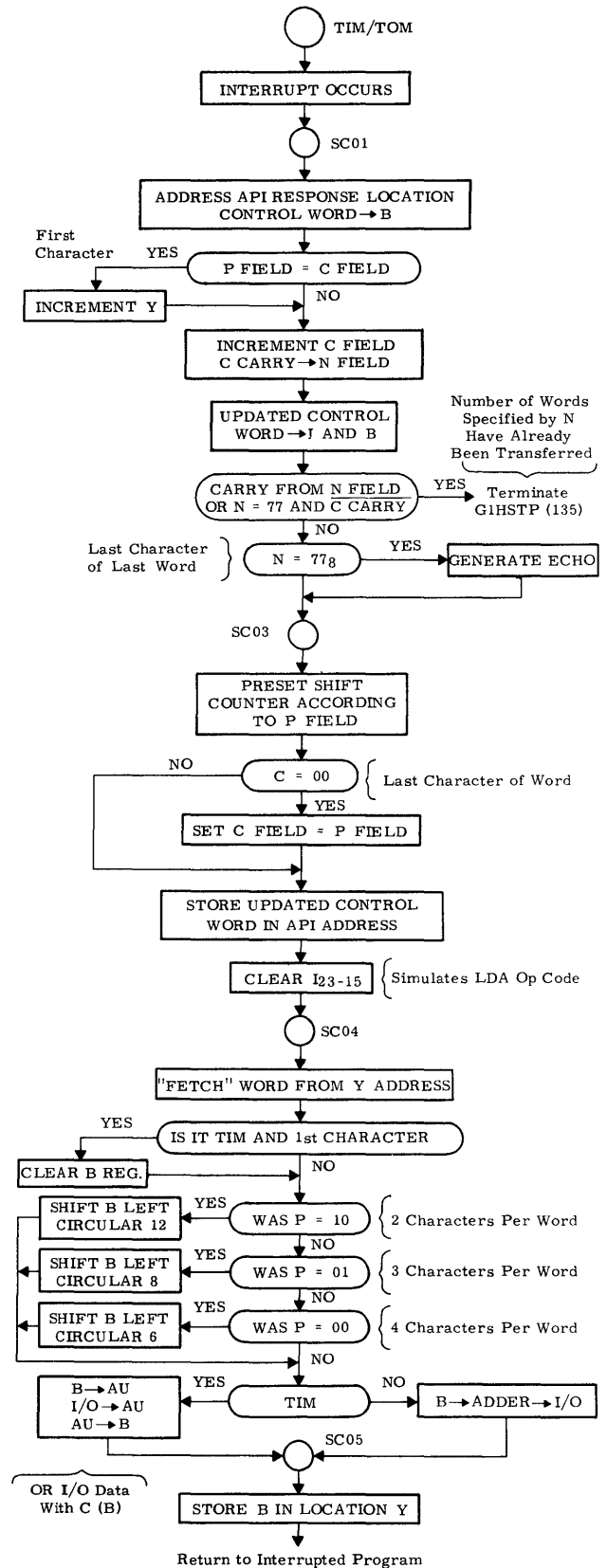


Fig. TIM/TOM. 2. TIM/TOM Flow Chart

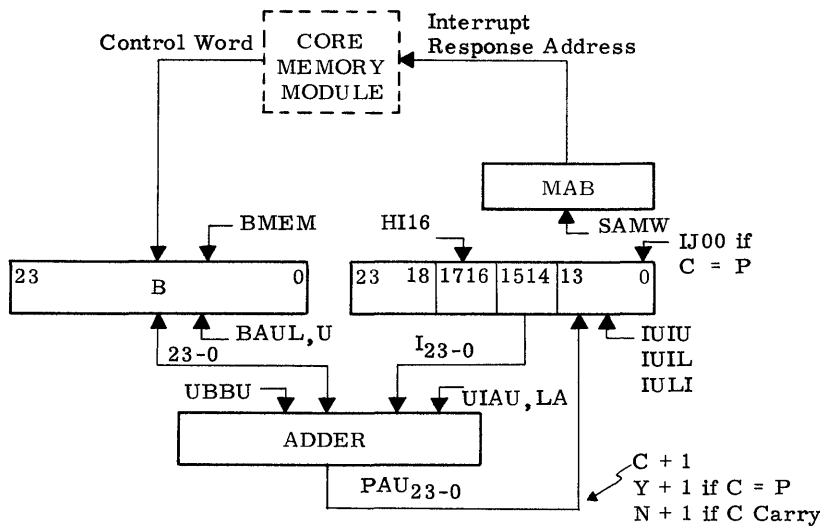


Fig. TIM/TOM. 3. State 1 Block Diagram

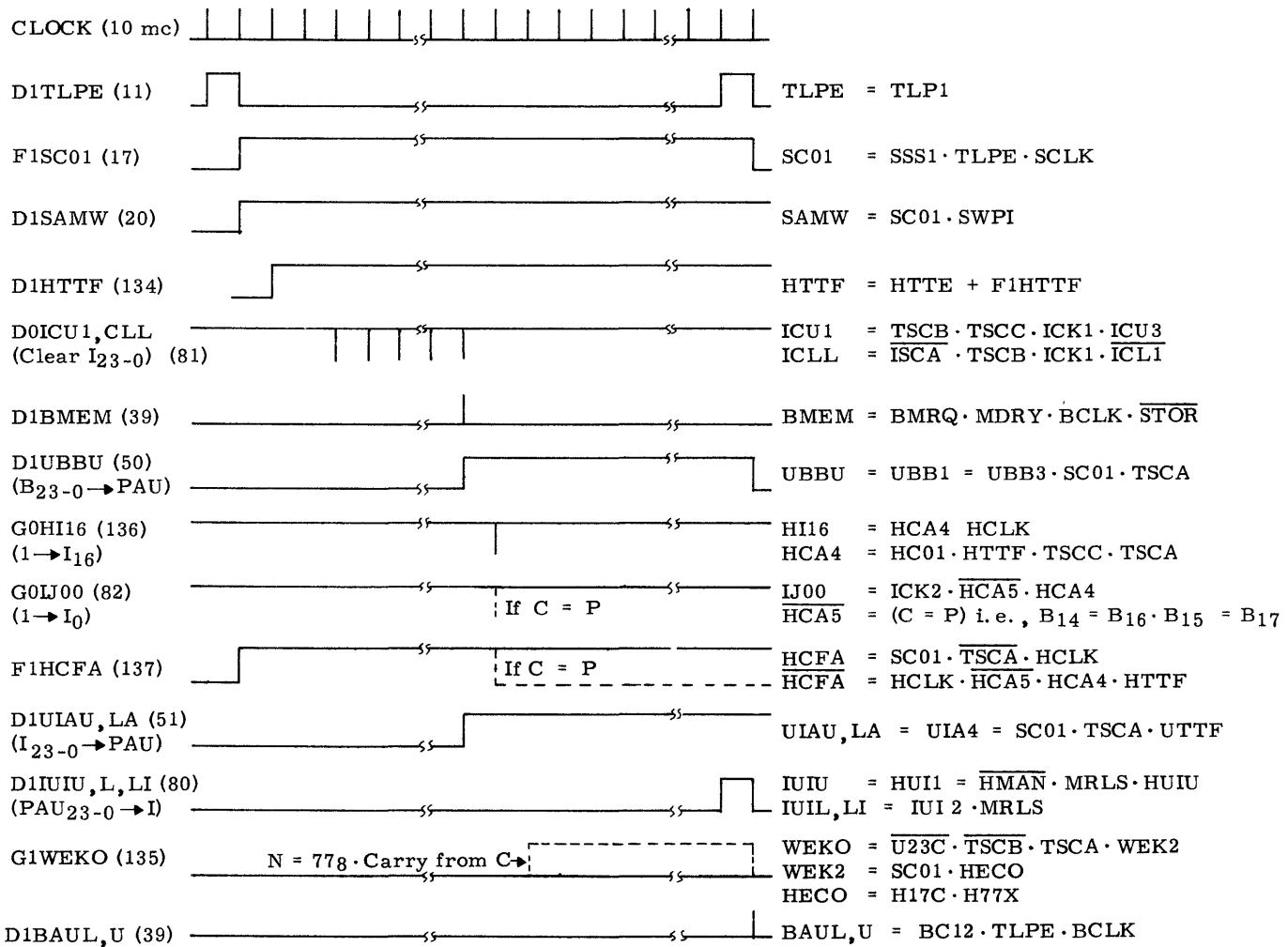


Fig. TIM/TOM. 4. State 1 Timing Diagram

Termination is implemented by holding sequencing in State 1 and allowing the interrupted program to continue.

Following State 1, State 3 is entered.

### Sequence State 3

Timing and block diagrams of State 3 operation are contained in Figs. TIM/TOM. 5 and 6. During Sequence State 3, the C field ( $I_{17,16}$ ) is examined and if it is equal to 00, the C field is set ( $GOHI_{17,16}$ ) equal to the P field ( $I_{15,A}$ ). This indicates that this is the last character of the present word and presets the C field for the next word. The contents of I (updated Control Word) are then gated to the B Register via the Adder. Memory is addressed from the API Module (DISAMW) and the updated Control Word is stored back in the API Response Address. The next time an interrupt occurs for this address, this Control Word will again be used.

$I_{23-15,A}$  is cleared (D0ICU1). Clearing  $I_{23-15,A}$  simulates an LDA command for use during the next Sequence State.

Also during State 3, the TIM/TOM Counter (F1HCFA-E) is preset, if required, to control the number of shifts required to position the data word. A discussion of this control is provided later in this section.

Following State 3, State 4 is entered.

### Sequence State 4

Block and timing diagrams of State 4 operation are contained in Figs. TIM/TOM. 7 and 8. During State 4, the actual data transfer occurs. To accomplish this function, the duration of State 4 is extended in the same manner as are GEN 2 commands.

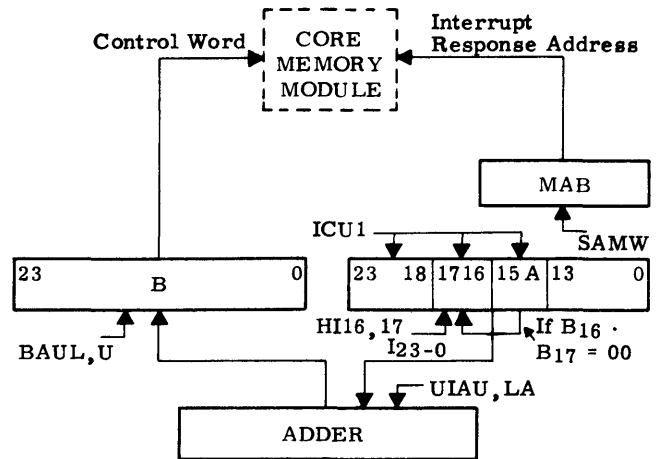


Fig. TIM/TOM. 5. State 3 Block Diagram

During the first portion of State 4, an LDA command is simulated, except that the contents of the AU are not gated to the A Register. This is accomplished because the Op Code portion of the I Register was cleared during State 3. Memory is addressed from Y and the contents of the memory cell are gated to the B Register. This is the contents of the memory cell that data is to be transferred from (TOM) or that data is to be transferred to (TIM).

If this is the first character of a TIM word, the B Register is cleared (D1BCLR). Clearing of B is necessary since the data bits received from the input device are Ored with B in the Adder.

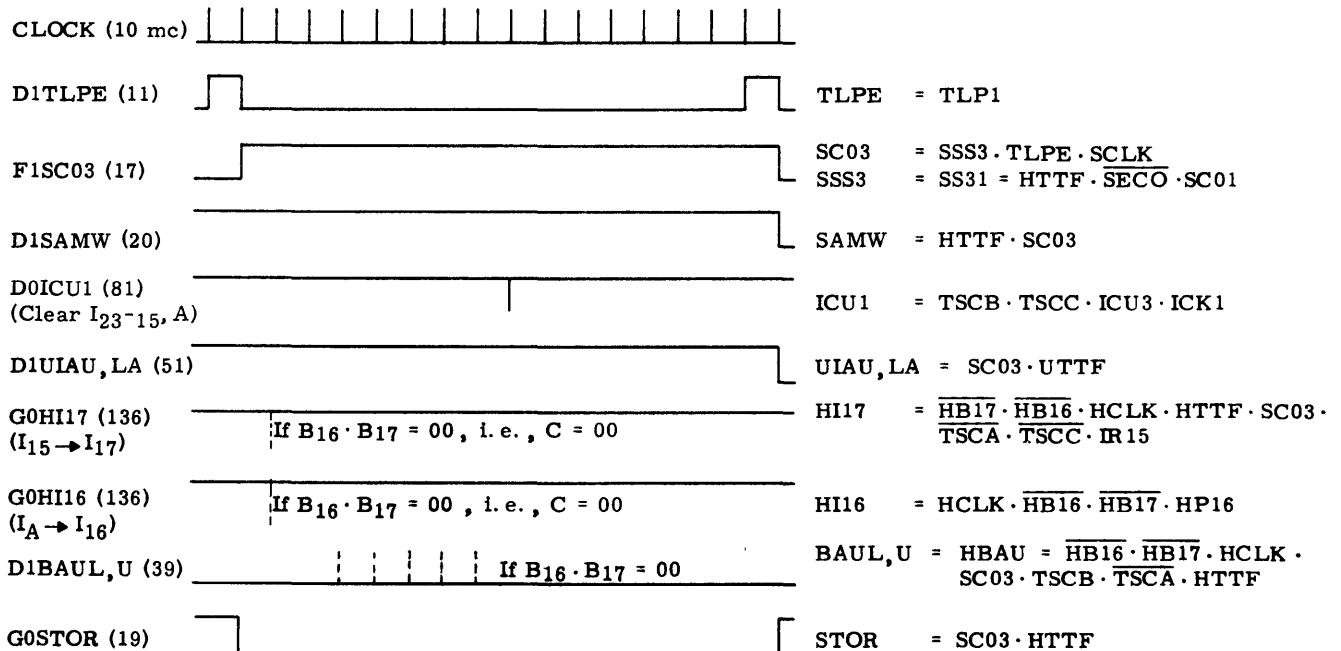


Fig. TIM/TOM. 6. State 3 Timing Diagram

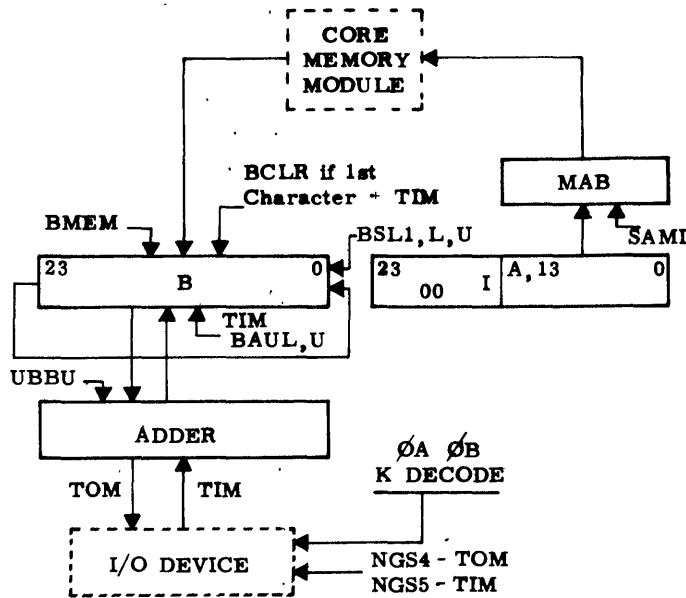


Fig. TIM/TOM. 7. State 4 Block Diagram

Time 6 Envelope is then entered. The basic timing of Time 6 is the same as that for an IN (TIM) or OUT (TOM) command. First, the data word must be positioned so that the character to be transferred is either in the least significant bits of the B Register or that space is reserved in the least significant bits of the B Register for the character to be received. This positioning of the B Register is accomplished by shifting the B Register left in a circular fashion, the number of bit

positions that a character requires. This operation is described below.

After the shifting is complete, data is transferred from B to the output device via the Adder (TOM) or the data received from the input device is gated to the Adder where it is Ored with the least significant bits of the B Register. In either case, the contents of the Adder are gated back to the B Register.

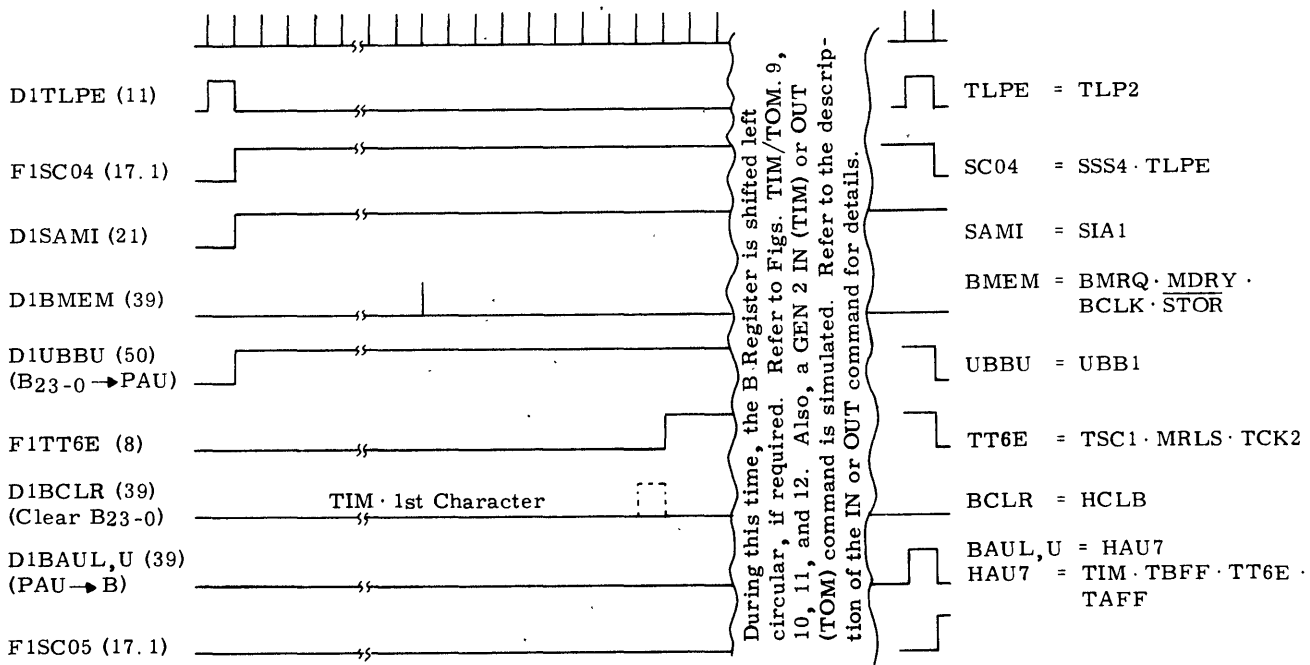


Fig. TIM/TOM. 8. State 4 Timing Diagram

Following State 4, State 5 is entered to store the data word back in memory location Y. Prior to discussing State 5, however, a discussion of positioning the character in the B Register is provided.

**NOTE**

When a TOM function is executed specifying 4 characters per word (i. e., P = 00), the complement of bit 5 of the data word is transferred to bit 6 of the output device by X1ND06 (sheet 101). This feature is primarily used by the paper tape punch. Using this technique, ASCII coded tape may be punched from six bit characters. Examination of the ASCII character code reveals that the seventh bit is always the complement of the sixth bit. This gating structure, therefore, conserves memory space by generating a seven bit ASCII code from six bits stored in memory. For example:

6-bit code to be punched = 0 1 0 1 0 1  
7-bit ASCII code punched = 1 0 1 0 1 0 1

X1ND06 = NGN5  
G1NGN5 = N1NGN5 · N1N05S

### Character Positioning

Since more than one character may be contained by a word, the B Register must be positioned so that the character being operated on will be in the least significant bits. This is accomplished by shifting the B Register left circular the number of places that the character contains, except when one character is equal to a word. The P field of the Control Word specifies

the number of characters in a word. This value is used to determine the number of shifts required.

- P = 11 - 1 character per word, no shifting required.
- P = 10 - 2 characters per word, 12 shifts required.
- P = 01 - 3 characters per word, 8 shifts required.
- P = 00 - 4 characters per word, 6 shifts required.

The TIM/TOM Counter (F1HCFA-E) provides control of the number shifts. During State 3, the Counter is preset, if necessary. During the first clock pulses of Time 6 Envelope of State 4, the B Register is shifted, if required. At each shift of B, the TIM/TOM Counter is incremented. When the required number of shifts have occurred, further shifting is inhibited by the TIM/TOM Counter.

Figs. TIM/TOM. 9, 10, 11, and 12 contain timing diagrams and logic equations of the shift control logic.

### Sequence State 5

During Sequence State 5, the contents of B (word being operated on) is stored back in memory location Y, as specified by the Control Word. Figs. TIM/TOM. 13 and 14 contain block and timing diagrams of the operation of State 5.

During State 5, memory is addressed from I (DISAMI), and the contents of the B Register are stored in memory location Y. The TIM/TOM flip-flop (F1HTFF) is cleared to end the TIM/TOM operation.

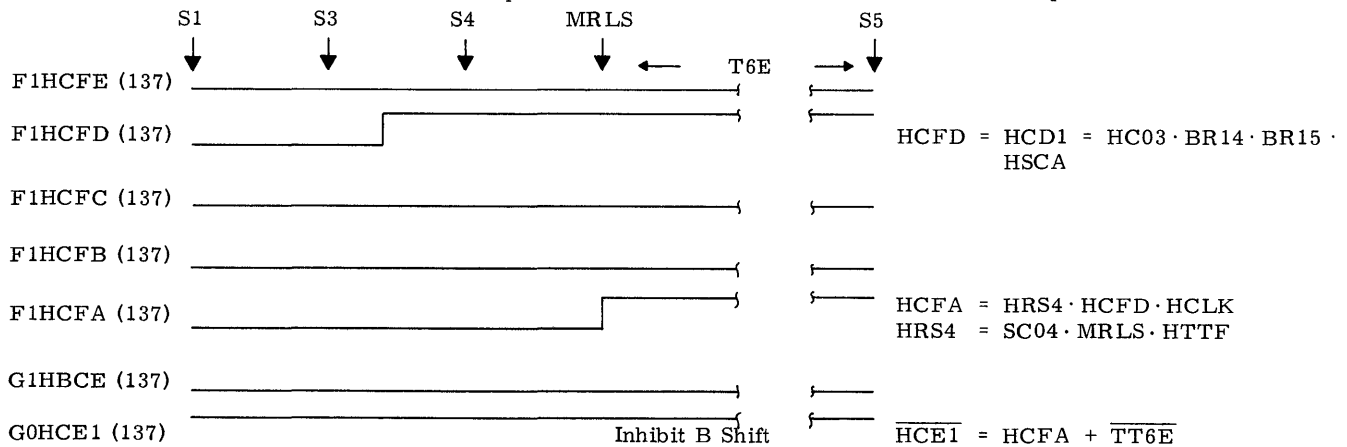
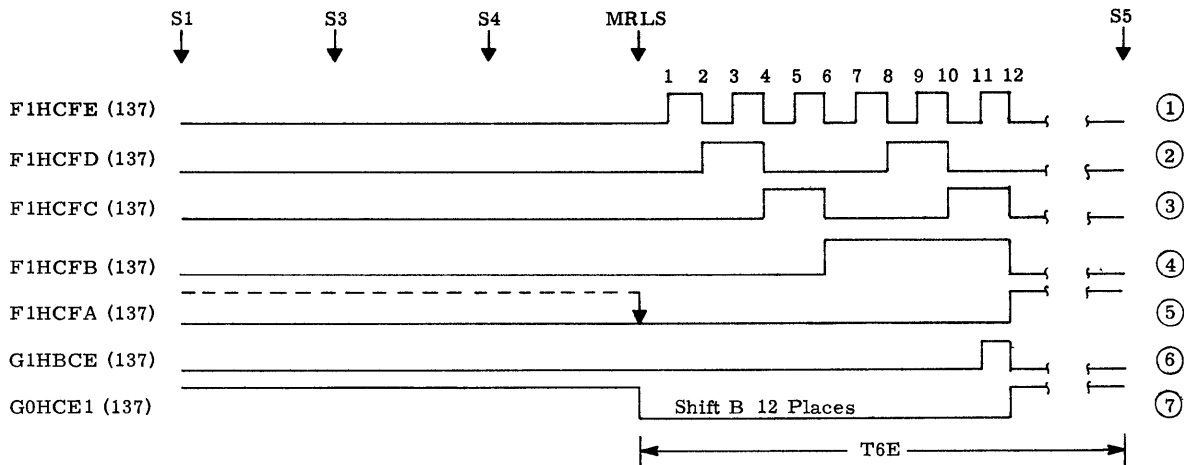
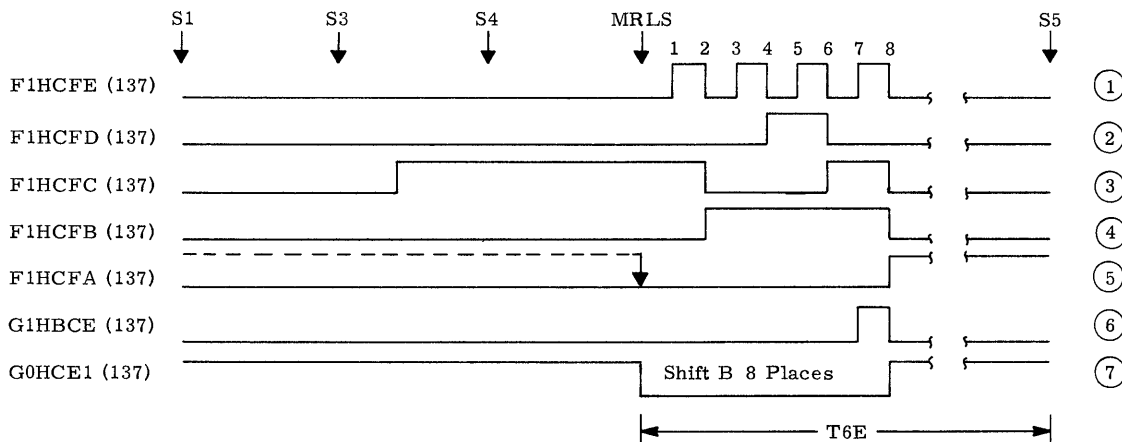


Fig. TIM/TOM. 9. P = 11 - 1 Character/Word Timing Diagram



- ①  $\overline{HCFE} = \overline{HCFA} \cdot TT6E \cdot HCLK \cdot \overline{HCFE}$   
 $HCFE = \overline{HCFA} \cdot TT6E \cdot HCLK \cdot HCFE + SC01 \cdot HCLK$
- ②  $\overline{HCFD} = HCFE \cdot \overline{HCFC} \cdot HCLK$   
 $HCFD = HCFE \cdot HCLK + SC01 \cdot HCLK$
- ③  $\overline{HCFC} = HCFE \cdot \overline{HCFD} \cdot HCLK$   
 $HCFC = HCFE \cdot HCFD \cdot HCLK + SC01 \cdot HCLK$
- ④  $\overline{HCFB} = HCFC \cdot HCFE \cdot HCLK$   
 $HCFB = HCFC \cdot HCFE \cdot HCLK + SC01 \cdot HCLK$
- ⑤  $HCFA = HBCE \cdot HCLK$
- ⑥  $HBCE = HCFB \cdot HCFC \cdot HCFE$
- ⑦  $HCE1 = \overline{HCFA} \cdot TT6E$

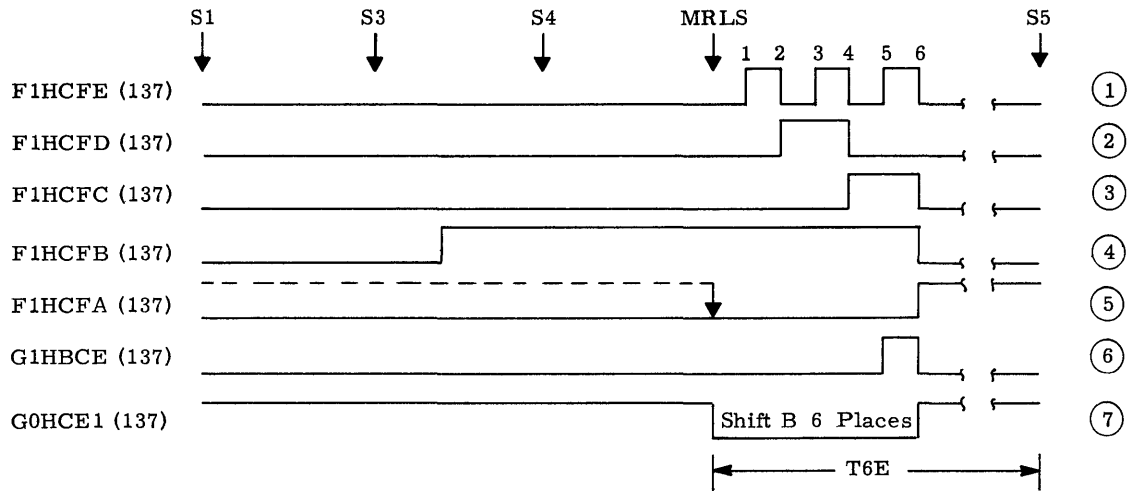
Fig. TIM/TOM. 10. P = 10 - 2 Character/Word Timing Diagram



- ①  $\overline{HCFE} = \overline{HCFA} \cdot TT6E \cdot HCLK \cdot \overline{HCFE}$   
 $HCFE = \overline{HCFA} \cdot TT6E \cdot HCLK \cdot HCFE + SC01 \cdot HCLK$
- ②  $\overline{HCFD} = HCFE \cdot \overline{HCFC} \cdot HCLK$   
 $HCFD = HCFE \cdot HCLK + SC01 \cdot HCLK$
- ③  $\overline{HCFC} = \overline{BR15} \cdot \overline{HB14} \cdot SC03 \cdot HSCB + HCFE \cdot HCFD \cdot HCLK$   
 $HCFC = HCFE \cdot HCFD \cdot HCLK + SC01 \cdot HCLK$
- ④  $\overline{HCFB} = HCFC \cdot HCFE \cdot HCLK$   
 $HCFB = HCFC \cdot HCFE \cdot HCLK + SC01 \cdot HCLK$
- ⑤  $HCFA = HBCE \cdot HCLK$
- ⑥  $HBCE = HCFB \cdot HCFC \cdot HCFE$
- ⑦  $HCE1 = \overline{HCFA} \cdot TT6E$

Fig. TIM/TOM. 11. P = 01 - 3 Character/Word Timing Diagram





- ①  $\overline{HCFE} = \overline{HCFA} \cdot \overline{TT6E} \cdot \overline{HCLK} \cdot \overline{HCFE}$   
 $HCFE = \overline{HCFA} \cdot \overline{TT6E} \cdot \overline{HCLK} \cdot HCFE + SC01 \cdot HCLK$
- ②  $\overline{HCFD} = HCFE \cdot \overline{HCFC} \cdot HCLK$   
 $HCFD = HCFE \cdot HCLK + SC01 \cdot HCLK$
- ③  $\overline{HCFC} = HCFE \cdot \overline{HCFD} \cdot HCLK$   
 $\overline{HCFD} = HCFE \cdot \overline{HCFD} \cdot HCLK + SC01 \cdot HCLK$
- ④  $\overline{HCFB} = \overline{BR15} \cdot \overline{BR14} \cdot \overline{HSCB} \cdot \overline{SC03} + HCFE \cdot \overline{HCFD} \cdot HCLK$   
 $\overline{HCFB} = HCFE \cdot \overline{HCFD} \cdot HCLK + SC01 \cdot HCLK$
- ⑤  $HCFA = HBCE \cdot HCLK$
- ⑥  $HBCE = HCFB \cdot HCFC \cdot HCFE$
- ⑦  $HCE1 = \overline{HCFA} \cdot \overline{TT6E}$

Fig. TIM/TOM. 12. P = 00 - 4 Character/Word Timing Diagram

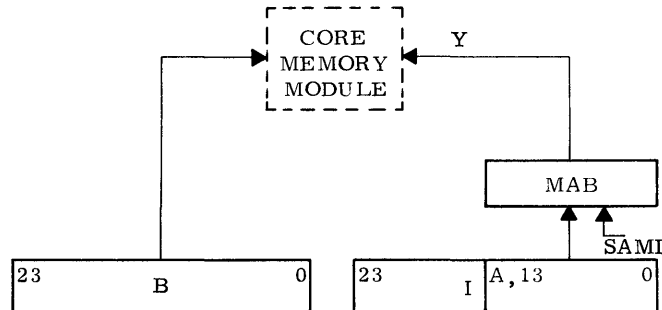


Fig. TIM/TOM. 13. State 5 Block Diagram

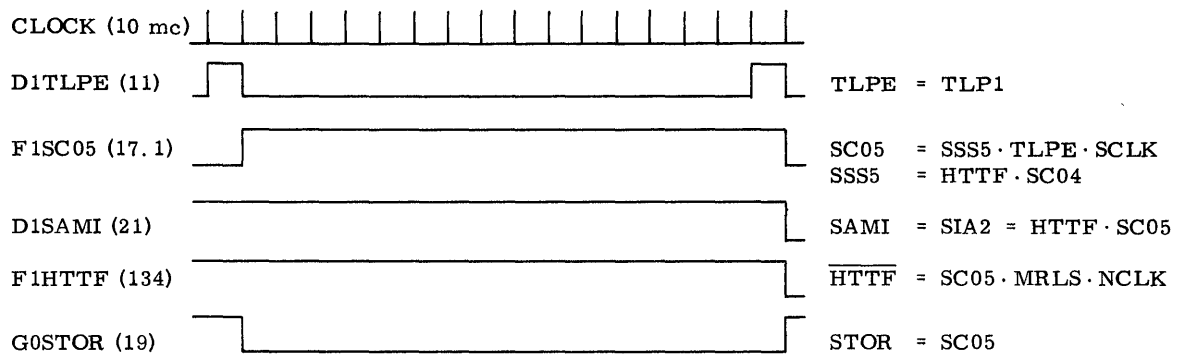
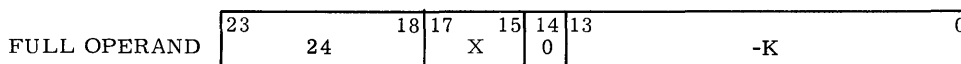


Fig. TIM/TOM. 14. State 5 Timing Diagram

## TXH - TEST X HIGH OR EQUAL



TXH sets the Test flip-flop (F1ETST) if the contents of the specified X cell (bits 14-0) are greater than or equal to the value K. If the contents of X<sub>14-0</sub> are less than K, the Test flip-flop is cleared. Bits 15, 16, and 17 of the TXH command specify the address of the X cell to be compared. The K value of the TXH command must be specified in 2's complement form. The value of the index cell contents may vary between 0 and 32,767. The value of K may range between 16,383 and 1. The contents of the addressed X cell are not changed by the TXH command. If the value of K is zero, the Test flip-flop cannot be set. If bits 15, 16, and 17 are "zero", the command is undefined, i. e., an X cell address must be specified.

During Sequence Control State 1, the TXH command is "fetched" from memory in the normal manner. However, as bits 13-0 of the command are gated from B to the Adder Unit, a "one" is forced to Adder Unit bit 14 by G1UB14. The result is then transferred from the Adder Unit to I<sub>A</sub>,13-0. In this manner, bit I<sub>A</sub> is forced to a "one".

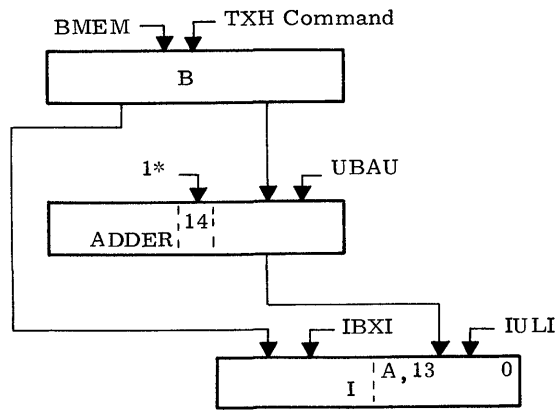
During Sequence Control State 2, memory is addressed from I<sub>17-15</sub> (G1SAMX) and the contents of the addressed X cell are gated to the B Register (D1BMEM). The contents of the X cell are then gated from the B Register to the Adder Unit (UBBU). At the same time, the contents of I<sub>A</sub>,13-0 are gated to the Adder Unit (UILA). The Test flip-flop is unconditionally cleared by the TXH command during Time 4 Envelope. At the Clock pulse of Memory Release (MUD1MRLS), the Test flip-flop is set if the summation of B (contents of X) and I<sub>A</sub>,13-0 (-K) results in a carry output from bit 14 of the Adder Unit.

The following examples are provided to illustrate that only when the contents of the addressed X cell are equal to or greater than the value K, is the carry output of Adder bit 14 true. For simplicity, a 5 bit K value and a 6 bit X value is used. Assume the most significant bit of the summed values to be PAU<sub>14</sub>.

Non-Indexed Word Times.	2 (S1, S2)
Interruptable Following Execution?	No
<b>CHANGES FOLLOWING EXECUTION</b>	
A <sub>23-0</sub>	
Q <sub>23-0</sub>	
P <sub>14-0</sub>	C (P) + 1
F1WPMT	
F1UOFL	
F1ETST	Set if $K \leq C(X)$ Reset if $K > C(X)$
J <sub>4-0</sub>	
Memory X	

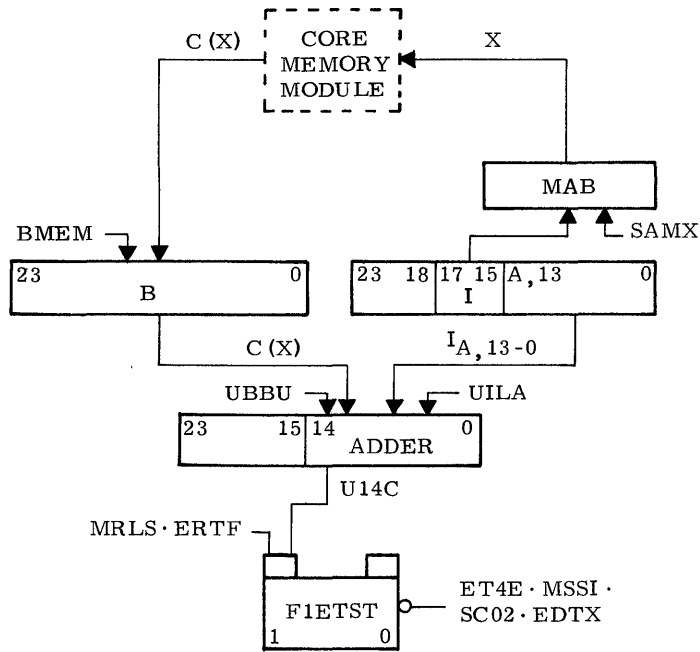
COMMAND CHARACTERISTICS

- (a) X = K = 5:
- |                         |   |       |                       |
|-------------------------|---|-------|-----------------------|
| 2's complement of K     | = | 1011  |                       |
| Forced "one" bit        | = | 1     |                       |
| C (X)                   | = | 00101 |                       |
|                         |   | 00000 |                       |
| PAU <sub>14</sub> Carry |   |       | = Test flip-flop set. |
- (b) K = 5:  
X = 4:
- |                         |   |       |                                   |
|-------------------------|---|-------|-----------------------------------|
| 2's complement of K     | = | 1011  |                                   |
| Forced "one" bit        | = | 1     |                                   |
| C (X)                   | = | 00100 |                                   |
|                         |   | 11111 |                                   |
| PAU <sub>14</sub> Carry |   |       | = Test flip-flop remains cleared. |



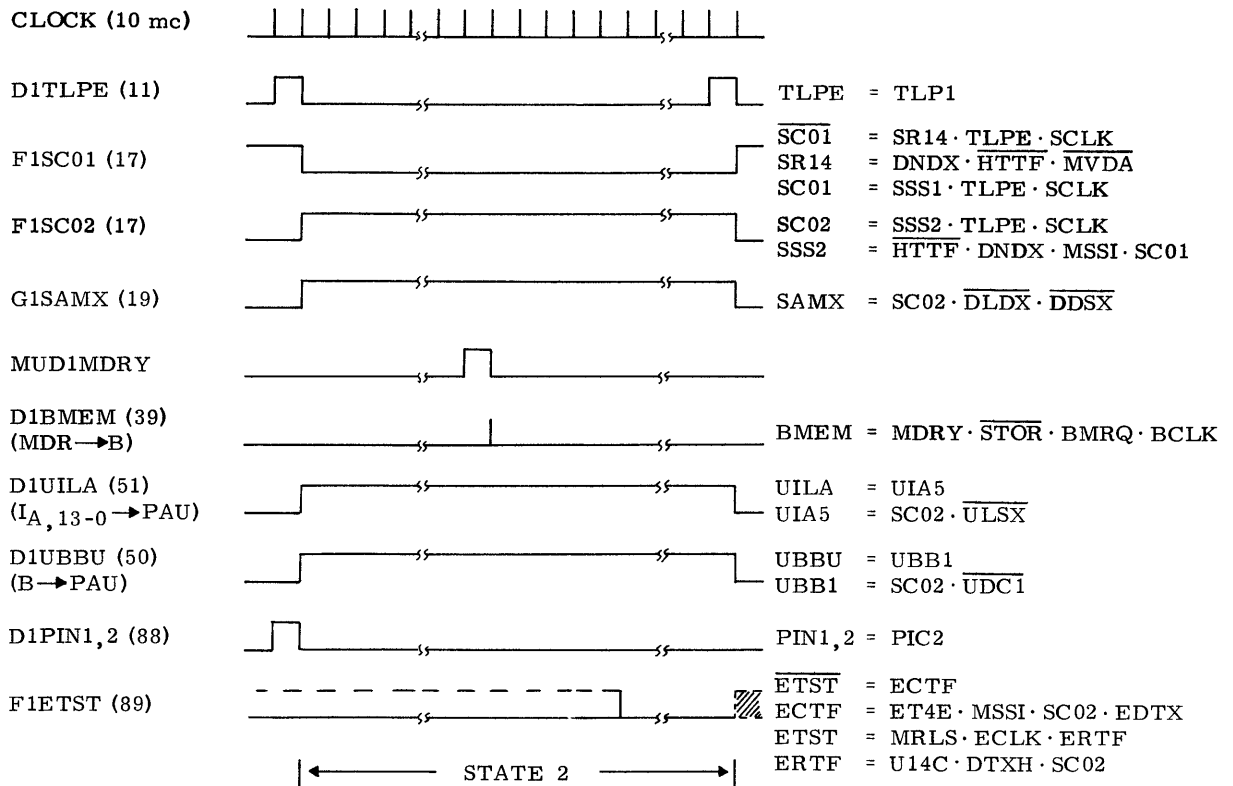
\*  $G1UB14(49) = ETXN \cdot \overline{UIXR} \cdot USIA$

Sequence State 1



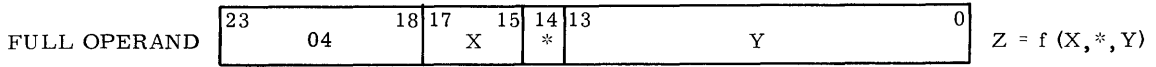
Sequence State 2

TXH BLOCK DIAGRAM



TXH TIMING DIAGRAM

# XEC - EXECUTE

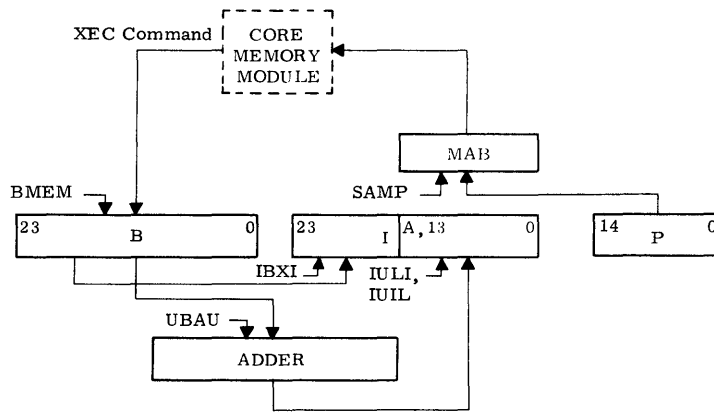


XEC indicates the address Z of the next instruction to be executed. Program control does not change, that is, the P Register is not incremented and the program continues in sequence after executing the instruction located at the effective operand address. All instructions including XEC, may be executed. If the object instruction (contents of cell Z) is relative addressed, the effective operand of the object instruction is computed from the location of the object instruction rather than from the contents of the P Register.

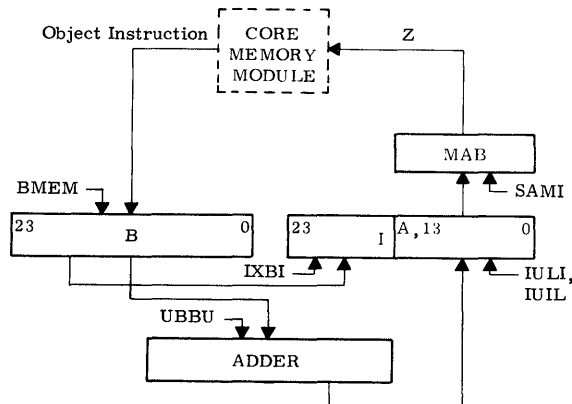
Two successive Sequence Control State 1's are required to illustrate the operation of the XEC command. During the first State 1, a normal "fetch" cycle occurs, except that at Last Pulse Envelope the Execute flip-flop (F1XEXC) is set. During the next State 1, memory is addressed from  $I_{A,13-0}$  (DISAMI) and the normal "fetch" cycle for the object instruction occurs except that at Last Pulse the Execute flip-flop is cleared. Sequencing then continues to execute the object instruction.

Non-Indexed Word Times.	1 (S1)
Interruptable Following Execution?	No
<b>CHANGES FOLLOWING EXECUTION</b>	
$A_{23-0}$	
$Q_{23-0}$	
$P_{14-0}$	
FIWPMT	
FIUOFL	
FIETST	
$J_{4-0}$	
Memory Z	

COMMAND CHARACTERISTICS

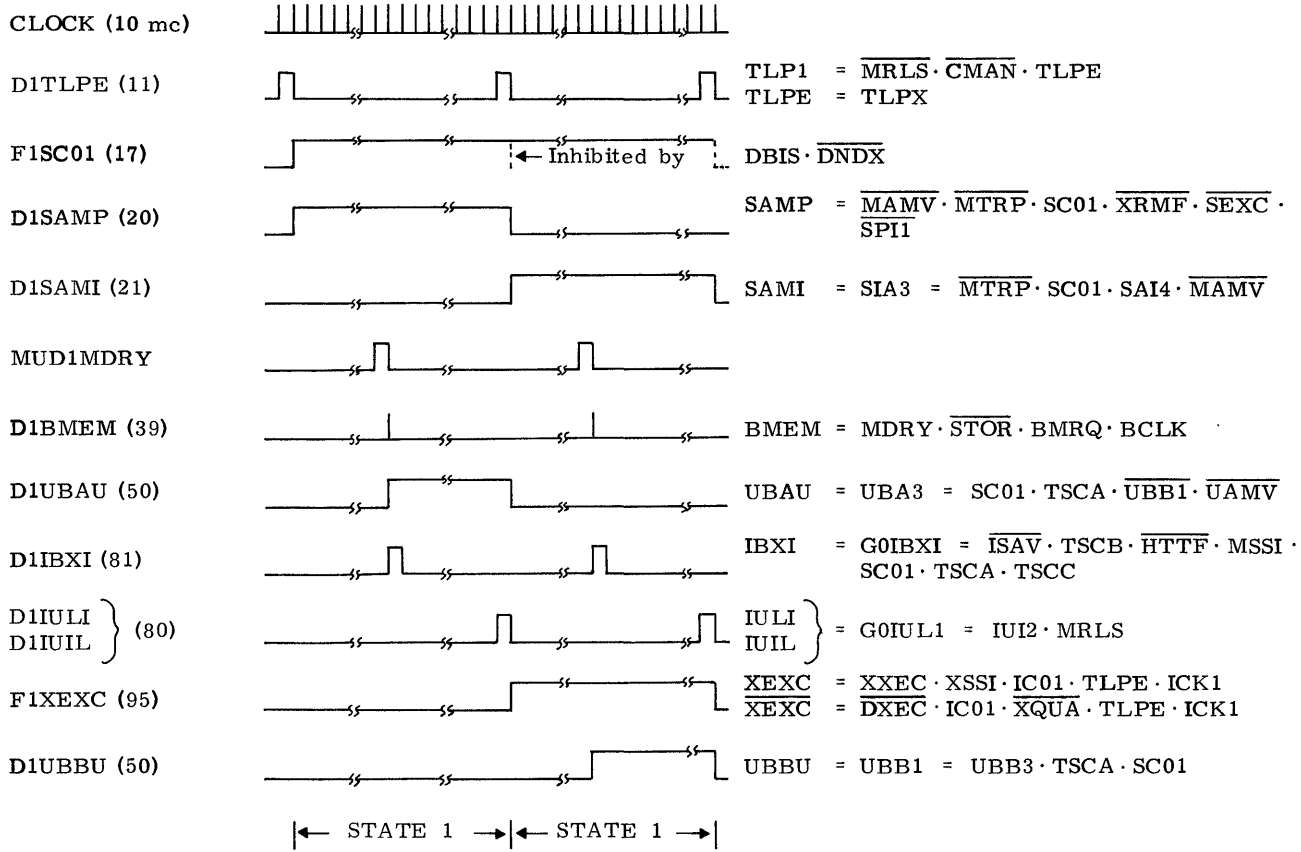


First Sequence State 1



Second Sequence State 1

XEC BLOCK DIAGRAM



XEC TIMING DIAGRAM

# CORE MEMORY

## 4015A

Portions of this work are reprinted  
by permission from "ICM-40 Magnetic  
Core Systems Instruction Manual."

COPYRIGHT 1966 by Honeywell Inc., Computer Control  
Division, Framingham, Massachusetts. Contents of this  
publication may not be reproduced in any form in whole or in  
part, without permission of the copyright owner. All rights  
reserved.

Printed in U. S. A.



# TABLE OF CONTENTS

## CORE MEMORY 4015A

### INTRODUCTION

#### LOGIC SYMBOLS AND NOMENCLATURE

F-02, QUAD NAND Gate

F-01, DUAL NAND Gate

Paralleled NAND Gates with Common Load Resistors

Type F-03 Power Amplifier Equivalent Logic Symbol

Type F-04 Flip-Flop

Logic Nomenclature

#### BASIC CORE THEORY

### SYSTEM DESCRIPTION

#### MEMORY MODULE ARRANGEMENT

#### PRIORITY ACCESS

#### ADDRESS DECODING AND SELECTION

#### ARITHMETIC UNIT ACCESS

Memory Control Timing

ICM-40 Memory Module Operation

#### USER 1, 2, 3 ACCESS

Memory Control Timing

ICM-40 Memory Module Operation

Read-Regenerate Mode

Clear-Write Mode

#### PARITY GENERATION

#### PARITY CHECKING

#### TRANSMISSION PARITY ERROR CHECKING

#### ICM-40 DC POWER

#### MEMORY RETENTION

Memory Wrap Around

### MAINTENANCE

#### RECOMMENDED TEST EQUIPMENT

#### RECOMMENDED SPARE PARTS

#### ICM-40 CIRCUIT BOARD HANDLING AND REPAIR

Inserting and Removing

Troubleshooting

Component Checking

Component Replacement

#### OPERATIONAL CHECK

#### CALIBRATION PROCEDURES

Drive Voltage Calibration

Logic Voltage Calibration

Strobe Timing Calibration

#### CORRECTIVE MAINTENANCE

Corrective Maintenance Inspection

Troubleshooting Procedures

#### MAGNETIC CORE STACK MAINTENANCE

Sense Windings

Drive Windings

#### POWER DISTRIBUTION

AC Power

DC Power

### APPENDIX A

### APPENDIX B

## LIST OF ILLUSTRATIONS

Fig. INT. 1.	4015A Memory System, Block Diagram	Fig. SYS. 7.	AU WRITE, Block Diagram
Fig. INT. 2.	Logic Nomenclature	Fig. SYS. 8.	Memory Control Timing-AU USER
Fig. INT. 3.	Typical B-H Characteristics	Fig. SYS. 9.	ICM-40 Timing - AU USER
Fig. INT. 4.	Coincident Current 2 1/2D Selection, Core Memory of Six Words, 2 Bits	Fig. SYS. 10.	ICM-40 Timing, AU USER
Fig. SYS. 1.	Functional Block Diagram	Fig. SYS. 11.	USER 1, 2, 3, READ Block Diagram
Fig. SYS. 2.	ICM-40 Memory Arrangement	Fig. SYS. 12.	USER 1, 2, 3, WRITE Block Diagram
Fig. SYS. 3.	Core Stack	Fig. SYS. 13.	Memory Control Timing, USER 1, 2, 3
Fig. SYS. 4.	Address Decoding and Selection	Fig. SYS. 14.	ICM-40 Timing, USER 1, 2, 3
Fig. SYS. 5.	Simplified Decoding and Selection Matrix	Fig. SYS. 15.	ICM-40 Timing, USER 1, 2, 3
Fig. SYS. 6.	AU READ, Block Diagram	Fig. SYS. 16.	Parity Generate
		Fig. SYS. 17.	Parity Check

## LIST OF TABLES

Table SYS. 1.	X-Decoding and Selection	Table MAINT. 3.	Recommended Spare Parts - Power Supply
Table SYS. 2.	Y-Decoding and Selection	Table MAINT. 4.	Operational Failures
Table SYS. 3.	Memory DC Power	Table MAINT. 5.	Partial Word Failures
Table MAINT. 1.	Recommended Test Equipment	Table MAINT. 6.	Address, Decoding, and Selection Failures
Table MAINT. 2.	ICM-40 Recommended Spare Parts		

# INTRODUCTION

The 4015A Core Memory is a high-speed digital storage device capable of storing 24 (24 data and 1 parity) bits of information in 8, 192 or 16, 384 randomly accessible locations. Data is stored in arrays of 30 mil OD x 18 mil ID ferrite cores, and selection is accomplished with three-wire, coincident-current techniques. A full memory cycle requires only 1.6 microseconds.

Features of the core memory are as follows:

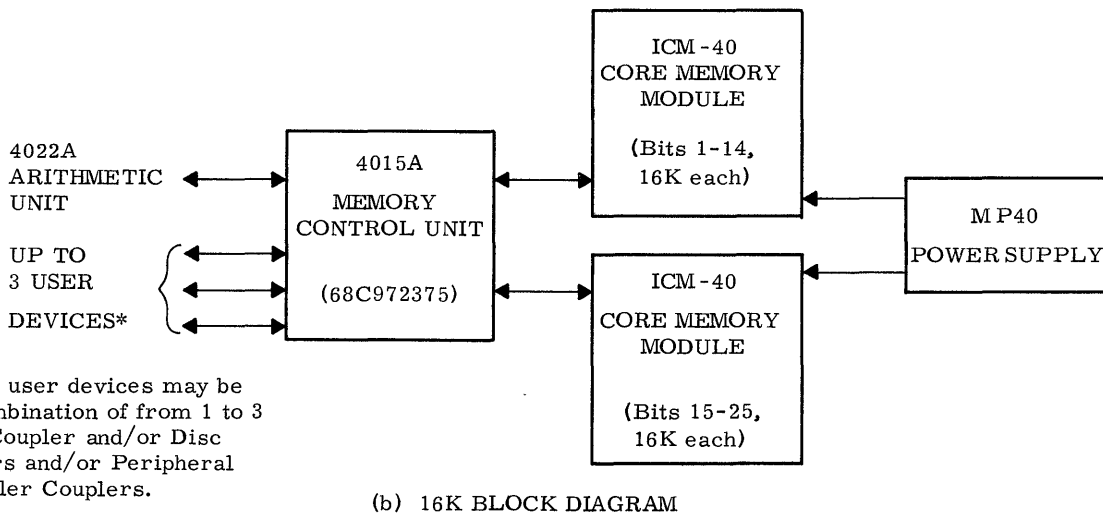
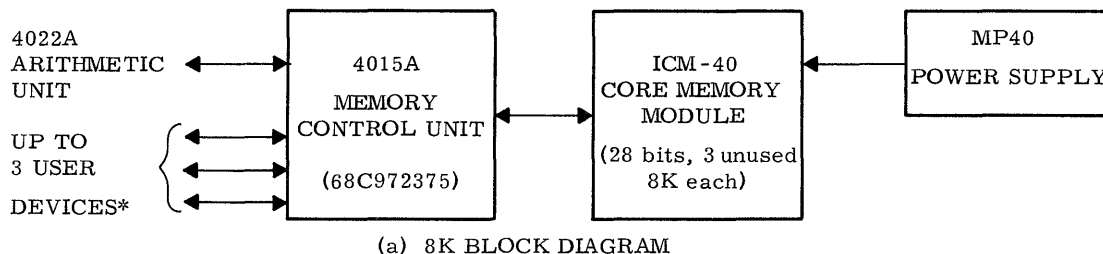
- Type - coincident current, destructive read.
- Capacity - 8, 192 (8K) or 16, 384 (16K).
- Word size - 24 data bits plus 1 parity bit.
- Communications - Parallel
- Temperature range - 0° C to 50° C ambient.
- Basic clock frequency - 10 megacycles.
- Error checking - Odd parity.
- Operating Modes -
  - Read - Regenerate
  - Clear - Write
  - Read - Modify - Write
- Multiplexed User Devices -
  - Up to 4 (4022 Arithmetic Unit and any combination of from 1 to 3 Drum Couplers, and/or Disc Couplers and/or Peripheral Controller Couplers).

The 4015A Core Memory consists of the 4015A Memory

Control Unit (GE Drawing 68C972375), one or two ICM-40 Core Memory Modules, and a MP40 Power Supply. The ICM-40 Memory Module is manufactured by Computer Control Company, Inc. Framingham, Massachusetts. The MP-40 Power Supply is manufactured by Ault Incorporated, 3501 48th Ave. North, Minneapolis, Minnesota. A basic block diagram of an 8K and 16K memory system is contained in Fig. INT. 1. The 4015A Memory Control Unit establishes priority of the user devices, multiplexes data and addresses to/from the user devices, checks or generates parity, and provides timing and operation control of the memory module. The ICM-40 Memory Module contains the ferrite cores for data storage, address decoding for selecting the addressed location, and the timing and control required to read and write data. Two ICM-40 Modules are required for 16K words of memory. When two ICM-40 Modules are required, the upper module contains bits 1 thru 15 and the lower module contains bits 16 thru 25 of the data word. The MP-40 Power supply provides operating power of +24vdc, +6vdc, and -6vdc to the ICM-40 memory module(s).

## LOGIC SYMBOLS AND NOMENCLATURE

This section illustrates the logic symbols and nomenclature used in the logic drawings for the ICM-40 Core Memory Module. Since the logic symbols for this logic differs from the standard General Electric drawings, a thorough understanding of these symbols is required to understand the operation of the ICM-40. Logic for the ICM-40 is contained in appendix A of this section.



\* The 3 user devices may be any combination of from 1 to 3 Drum Coupler and/or Disc Couplers and/or Peripheral Controller Couplers.

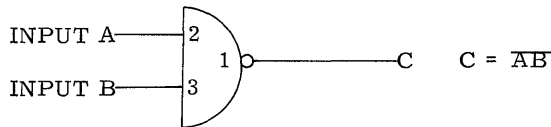
Fig. INT. 1. 4015A Memory System, Block Diagram

Each logic symbol used is presented in the following figures. Included with each symbol is the logic function and truth table for the symbol. Using these aids, little difficulty should be encountered in interpreting these symbols.

Positive logic with a binary "one" equal to positive 6 volts and a binary "zero" equal to zero volts is used within the ICM-40 logic. Binary "ones" applied to the ICM-40 from the Memory Control Unit are amplified from the 3.5 volt level to the 6 volt level. Binary "ones" at 6 volt level from the ICM-40 are applied through dropping resistors for application to the Memory Control Unit at the 3.5 volt level.

### F-02, QUAD NAND Gate

#### A. Logic Symbol



#### B. Logic Function

$$C = \overline{AB}$$

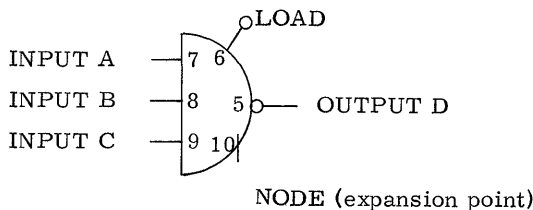
#### C. Truth Table

INPUT A	INPUT B	OUTPUT C
0	0	1
0	1	1
1	0	1
1	1	0

0 = GROUND  
1 = +6V

### F-01, DUAL NAND Gate

#### A. Logic Symbol



#### B. Logic Function

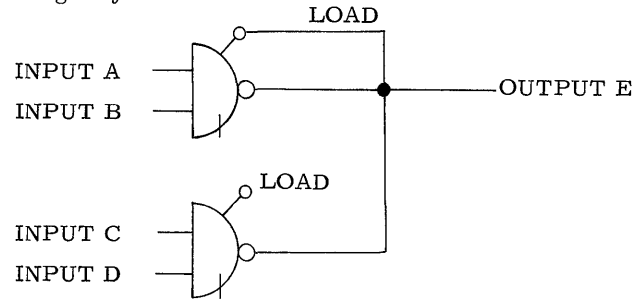
$$D = \overline{ABC}$$

#### C. Truth Table

INPUT A	INPUT B	INPUT C	OUTPUT D
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

### Paralleled NAND Gates with Common Load Resistors

#### A. Logic Symbol



#### B. Logic Function

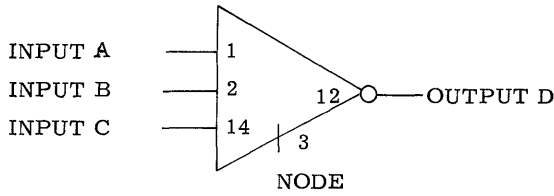
$$E = \overline{AB + CD}$$

#### C. Truth Table

INPUT A	INPUT B	INPUT C	INPUT D	OUTPUT E
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

## Type F-03 Power Amplifier Equivalent Logic Symbol

### A. Logic Symbol



### B. Logic Function

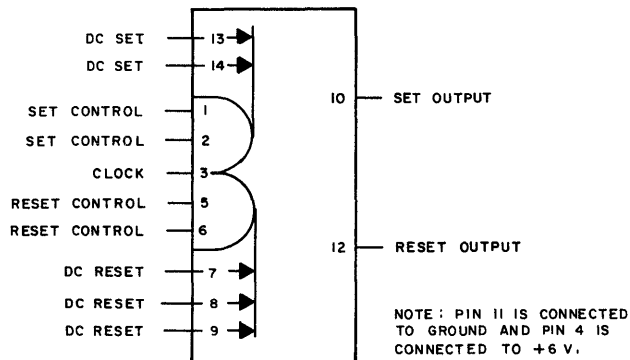
$$D = \overline{ABC}$$

### C. Truth Table

INPUT A	INPUT B	INPUT C	OUTPUT D
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

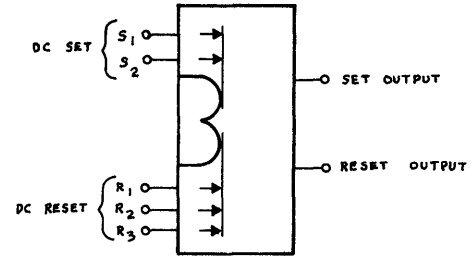
## Type F-04 Flip-Flop

### A. Logic Symbol



### B. DC Operation

### 1. Logic Diagram



### 2. Truth Table and Boolean Equations

$S_D$  - AND result of the dc set inputs.  
 $S_D = S_1 \cdot S_2$

$R_D$  - AND result of the dc reset inputs.  
 $R_D = R_1 \cdot R_2 \cdot R_3$

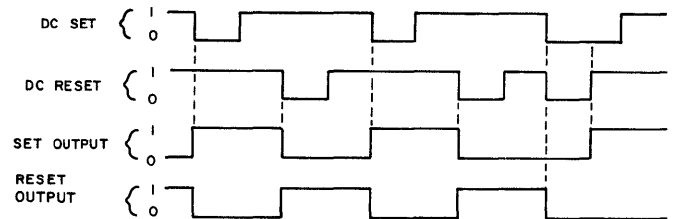
$F$  - state of the flip-flop (set output)

$F'$  - previous state of the flip-flop

$S_D$	$R_D$	$F$
0	0	(Both set and reset outputs are 0's.)
0	1	1
1	0	0
1	1	$F'$ (no change)

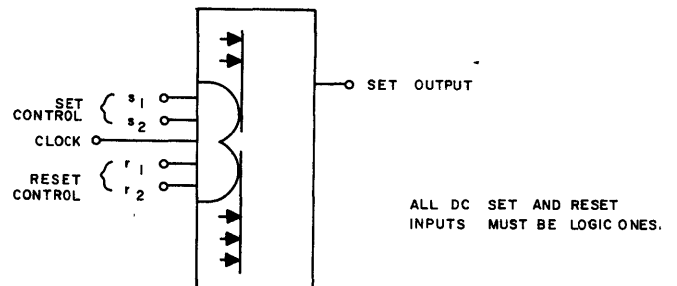
$$F = R_D (\overline{S_D} + F')$$

### 3. Timing Diagram



### C. AC Operation

#### 1. Logic Diagram



## 2. Truth Table and Boolean Equations

$S_C$  - AND result of the set control inputs,  
 $S_C = s_1 \cdot s_2$

$R_C$  - AND result of the reset control inputs,  
 $R_C = r_1 \cdot r_2$

$F'$  - previous state of the flip-flop

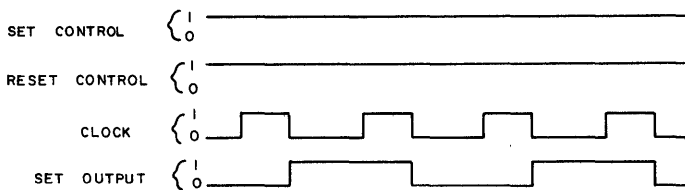
$F$  - state of the flip-flop after the clock pulse

S	R	F	F'	F
0	0	0	0	NO CHANGE
0	0	1	1	
0	1	0	0	RESET
0	1	1	0	
1	0	0	1	SET
1	0	1	1	
1	1	0	1	COMPLEMENT
1	1	1	0	

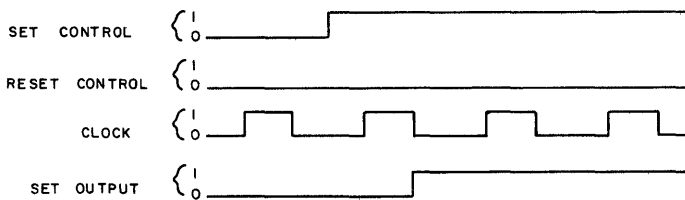
$$F = S_C \overline{F'} + \overline{R_C} F'$$

## 3. Timing Diagrams

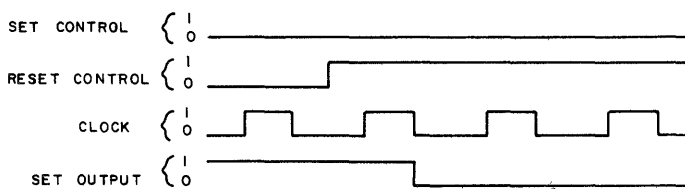
### COMPLEMENTING



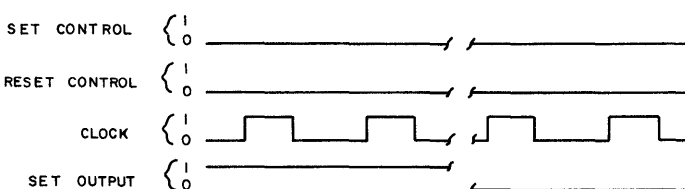
### SET



### RESET



### NO CHANGE



## Logic Nomenclature

Fig. INT. 2 illustrates a portion of the ICM-40 logic drawings and the meaning of the nomenclature used. Appendix A and B of this section contains a mnemonic dictionary of the logic signals, an input/output connector list, a circuit board location map, detailed information on each circuit card, and the ICM-40 logic.

Signal mnemonics contain the designation - (minus) to indicate that the signal function is true when at the zero volt level or the designation + (plus) to indicate that the signal function is true when at the + 6 volt level.

## BASIC CORE THEORY

Information is stored in a three-dimensional array of 18 x 30 mil ferrite cores. Each core may be individually set to one of two possible magnetic states thereby representing one bit of binary information. Nonvolatile storage is made possible by core B-H characteristics which approximate a rectangular hysteresis loop. The core state is identified by the polarity of flux within the toroid structure. The switching mechanism may be qualitatively understood by examination of the B-H loop shown in Figure INT. 3.

The H represents magnetizing force proportional to current magnitudes linking the toroid, and B symbolizes magnetic flux density within the core. The device characteristic is useful since the B-H relationship is extremely nonlinear and irreversible. For example, if a core is initially in state 0, as magnetizing force is increased, B is only slightly affected until H approaches  $H_1$ . As H increases from  $H_1$  to  $H_2$ , total flux reversal occurs (path a). At  $H_2$ , the core may be considered saturated in the opposite state such that an additional increase in H cannot significantly alter B. Irreversibility is shown by the fact that B returns to state 1 rather than starting point 0 as H is relaxed from  $H_2$  to 0. However, state 0 may again be realized by applying sufficient H of opposite polarity to traverse path b.

The threshold characteristic of the device permits its use in a coincident-current selection scheme as shown in Figure INT. 4. Each core in the array is linked by an X- and Y- drive line. Subscripts R and W refer to read and write, and + and - are polarity (direction) indicators. The current magnitude of  $IY$  or  $IX_R$  corresponds to  $H_1$ , or less, and their sum corresponds to a field exceeding  $H_2$ . Currents of this magnitude entering a core from the same side will produce a field exceeding  $H_2$ . Currents entering from opposite sides will have mutually cancelling fields resulting in  $H = 0$ . Currents  $IX_R$  and  $IX_W$  have the same amplitude but opposite polarity, as do  $IY+$  and  $IY-$ .

Each X- and Y- drive line links two cores in one bit array. The X-drive line is common to all bits. However, each bit has its own Y-drive lines. The prefixes 01 or 05 on the Y-drive lines correspond to bit 1 and 5, respectively. Bits 01 and 05 are used in this example since they are physically located in the top and second planes in 4K and 8K memories.

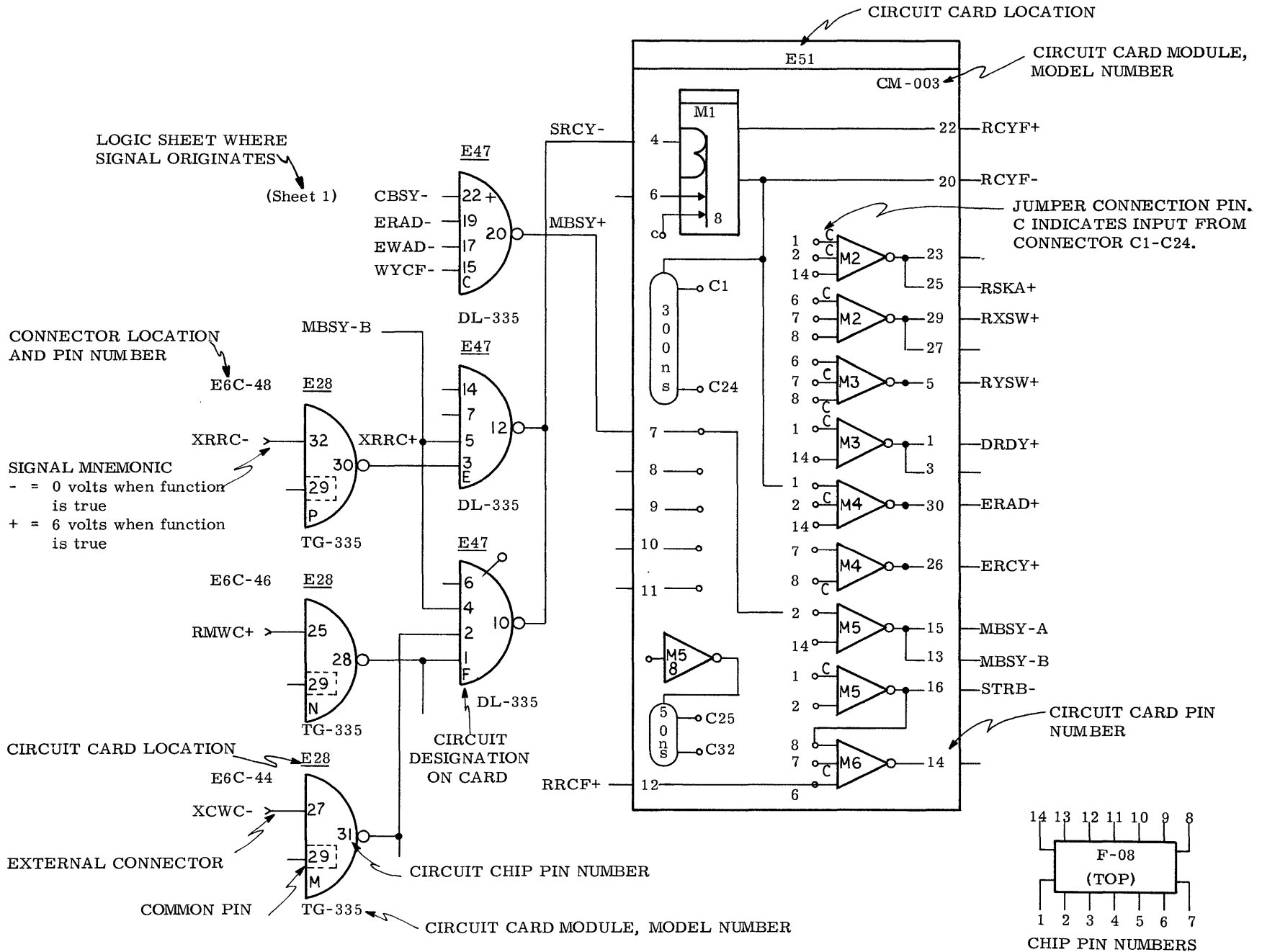


Fig. INT. 2. Logic Nomenclature

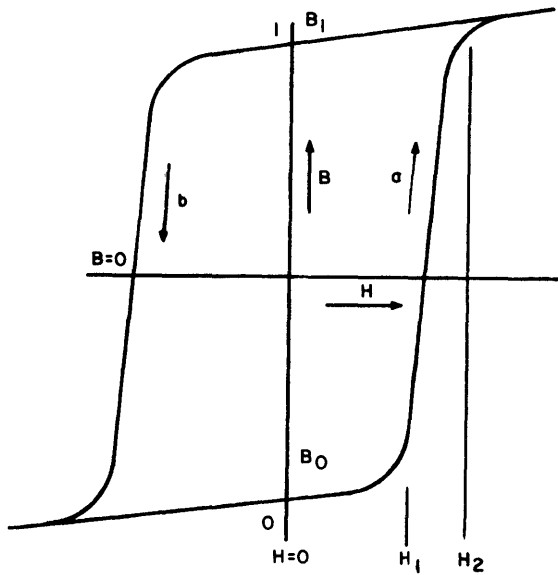


Fig. INT.3. Typical B-H Characteristics

If one X-drive line and one Y-drive line are energized, only one core in the entire array will see a magnetic field strong enough to cause it to change state. For example, if X1 and 01Y1 are energized with  $I_{X_R}$  and  $I_{Y+}$ , core  $C_{111+}$  will be switched to the clockwise magnetic state (if it is not already in that state). Core  $C_{111-}$  is subjected to a net field of zero and all other cores linked by 01Y1 or X1 are subjected to half currents. Consequently, a unique core address can be selected by energizing an X-drive line and one Y-drive line per bit.

Consider a read operation with all cores in array one (bit 1) in the counterclockwise magnetic state (ONE) and all cores in array five (bit 5) in the ZERO state. Assume  $I_{Y+}$  drive currents flow in lines 01Y1 and 05Y1 and  $I_{X_R}$  current flows in line X1. Core  $C_{111+}$  is switched to the ZERO state and the resultant flux change appears as a differential voltage at the sense

winding terminals (SW01). This voltage is amplified, strobed, and standardized, setting the corresponding data register stage to a ONE. If the direction of the Y-drive current is reversed ( $I_{Y-}$ ), core  $C_{111-}$  will be switched. Core  $C_{511+}$  was in the ZERO state so the currents drive the core further into saturation. This results in a flux change too small to set the associated data register flip-flop. Core  $C_{511-}$  is subjected to a net field of ZERO.

After the read operation, both cores at the selected address have been interrogated and stored information has been transferred to the data register. Although the read-out was destructive, the stored information may be reinserted during the write portion of the cycle. The data register flip-flop controls the Y-drive currents. Bit-1 currents  $I_{X_W}$  and  $I_{Y-}$  flow but there is no current in 05Y1 because the bit-5 data register is in the ZERO state. The result is that core  $C_{111+}$  is switched to the ONE state and core  $C_{511+}$  is left in the ZERO state. All other cores in the array remain in their original state.

The memory shown in Fig. INT. 4 can be expanded to an array of 8192 cores comprising one bit of an 8192-word memory. There are 256 X-drive lines and 16 re-entrant Y-drive lines, each Y-drive line linking 512 cores. The array of 8192 cores has two sense windings, each of which links 4096 cores in conjunction with 128 X-lines. One core plane consists of four arrays on a 5 x 9 inch printed circuit board. The core stack consists of one plane for every four bits stacked together with the X-lines threading all arrays. In addition to the core planes, printed circuit boards are located on the top (X-selection diode Flat Packs) and bottom (X-selection bus wiring) of the core stack.

In a 16K core stack, two 8K bits are used as one 16K bit. This is accomplished in the wiring of the decoding and selection circuitry. The 16K core stack contains one plane for every two bits. Four sense windings per bit are used according to the bit location. Since 25 bits are required for each word, two ICM-40 Memory Modules are used for 16K words of memory.



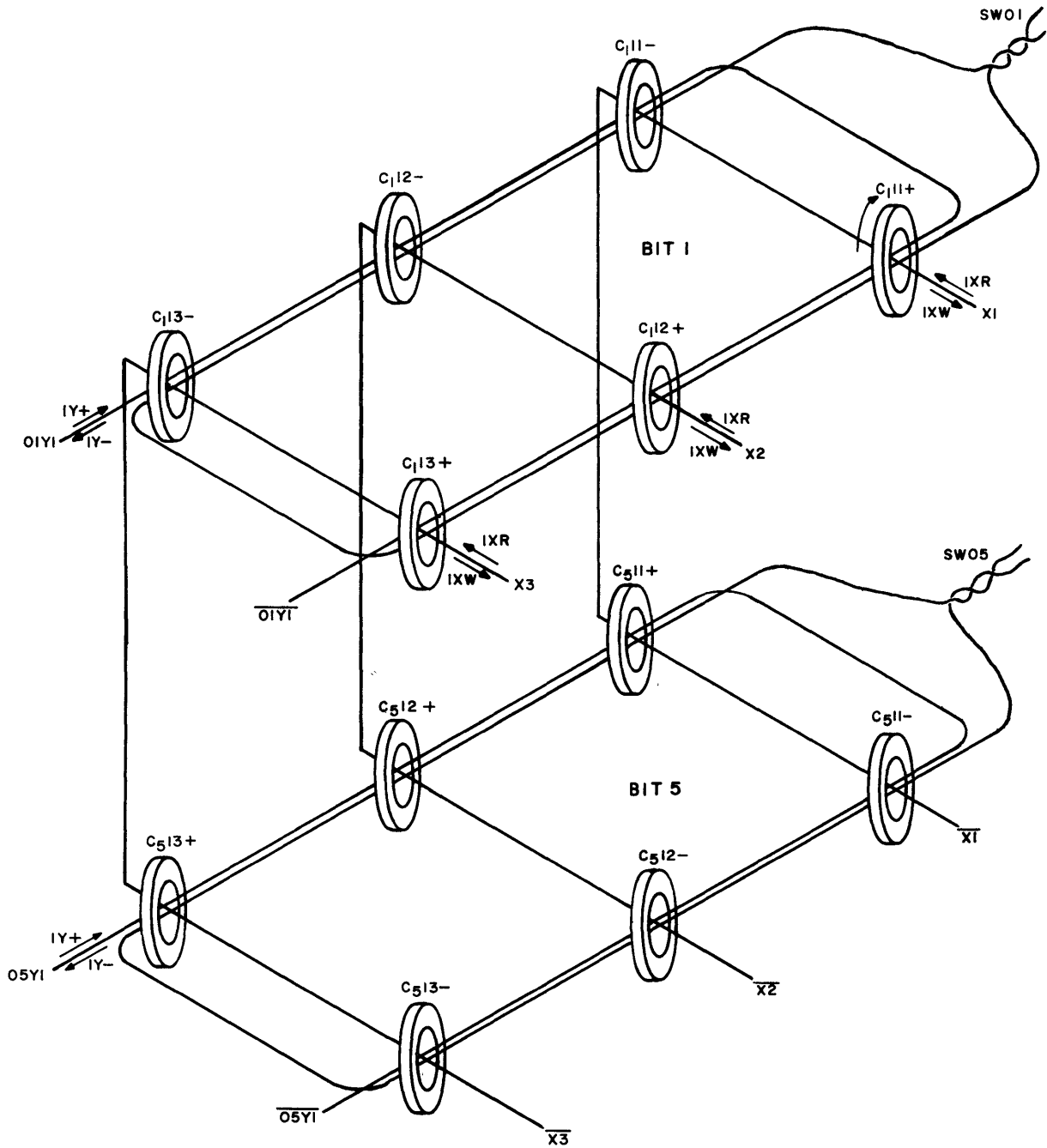


Fig. INT. 4. Coincident Current  $2\frac{1}{2}D$  Selection, Core Memory of Six Words, 2 Bits

# SYSTEM DESCRIPTION

A functional block diagram of the memory system is contained in Fig. SYS. 1. As shown in the diagram, as many as 3 devices, in addition to the Arithmetic Unit, may communicate directly with the memory. These 3 devices may be any combination of from one to three Drum Couplers, and/or Disc Couplers, and/or Peripheral Controller Couplers.

The user device applies a memory request signal, the operation (read or write) to be performed, and the memory location address to be affected to the Memory Control Unit. If more than one device requests memory at the same time, the Memory Control logic determines which device has priority and grants access to that device only. The device connected to the Channel 1 input has priority over the Arithmetic Unit or devices connected to Channel 2 or Channel 3 inputs. Channel 2 devices have priority over the Arithmetic Unit and the Device connected to Channel 3. Channel 3 requests have priority over Arithmetic Unit requests.

The Arithmetic Unit has the lowest priority since its operation speed is the fastest. The Memory Control Unit applies a device granted signal to the Device that obtains priority.

After the access priority has been established, a signal is applied to the core memory module to indicate the type of operation and to begin the memory cycle. The Arithmetic Unit always initiates the Read-Modify-Write operation within the memory regardless of a read or write operation. Other user devices initiate the Read-Regenerate mode if it is a read operation or initiate the Clear-Write mode if it is a write operation.

If the operation is write into core, the data transferred from the user device is gated through the Multiplexer Data Gates. The parity bit, if required, is generated and the data is transferred to the Data Register of the memory module. The address of the core location to be affected is multiplexed from the user device to the Address Register of the memory module. The address is decoded and the corresponding X and Y lines are enabled. Enabling the X and Y lines will cause the cores in this memory location to clear. Since this is a write into core operation, the data contained in this location is not used. The "one" data bits contained in the Data Register then controls the decoded Y lines to set the corresponding data bits of the addressed location. Setting the core cells of the addressed location to the configuration of "one" bits in the Data Register completes the memory cycle. When the Arithmetic Unit has access to memory, a Memory Release signal is applied from the Memory Control logic to the AU at the end of the cycle.

Read or bring from core memory operations require the same determination of priority after receipt of the memory request signal. The multiplexed address is again decoded to clear the bits of the corresponding memory location. The data sensed during this clearing operation is strobed through the sense amplifiers into the Data Register. The data contained in the Data Register is then used to control the Y lines and, thereby, store the same data back in the addressed location. The data in the Data Register is also applied through the Memory

Input Buffer for application to the user device. A Data Ready signal is provided by the Memory Control Unit to set this data in the user device. The data read from core is checked for odd parity. If a parity error exists, a parity error signal is provided. This parity error signal lights the Parity Error indicator on the console, provides for program detection of the error (JNP), and may be optionally used to halt further accesses to memory by the AU using the Stop On Parity switch. If the user is the Arithmetic Unit, a memory release signal is applied to the Arithmetic Unit upon completion of the memory cycle.

## MEMORY MODULE ARRANGEMENT

Fig. SYS. 2 illustrates the physical organization of the ICM-40 Memory Module. The Memory Module coordinates are marked for identification of printed circuit boards and component locations.

Fig. SYS. 3 illustrates the organization of the memory stack for both an 8K and a 16K memory system. An 8K memory system consists of one ICM-40 Memory Module with 8K words of storage. The 8K core stack consists of 7 planes of data bits. Each plane contains 4 bits of storage for 8K words. Therefore, 7 planes provide for storage of up to 28 bit words. The 4015A Memory System only utilizes 25 of this bits, providing a 24 bit data word plus a parity bit. Two additional planes are required by the stack; one X-Diode Board and one X-Buss Board.

A 16K memory system requires two ICM-40 modules. One module has a stack containing 14 bits of 16K words and the other module has a stack containing 12 bits of 16K words. The 14 bit stack contains data bits 1 thru 14 and the 12 bit stack contains data bits 15 thru 24, a parity bit, and one bit is unused.

## PRIORITY ACCESS

The Memory Control logic (68C972375, sheet 7) contains the logic required to enable access to memory of Channel 1 requests over AU, Channel 2, or Channel 3 requests; Channel 2 access to memory over AU or Channel 3 requests; and Channel 3 access to memory over AU requests. The Arithmetic Unit requests have the lowest priority for access to memory due to its operational speed. Channel 1, 2, and 3 requests are assigned as a system option for the devices (Drum, Disc, etc.) used by the system.

Requests from the user devices are applied to the Priority Request flip-flops F1MPR1 thru F1MPR4. Request from the Arithmetic Unit (G1SMRQ) are applied to F1MPR4. With no other request present, or at the last clock pulse of a previous memory cycle, the priority flip-flop corresponding to the channel request input is set. Priority of the Priority Request flip-flops is then established in gates G0M BR1 thru G0M BR4. Only one of these gates will be enabled according to the priority scheme.

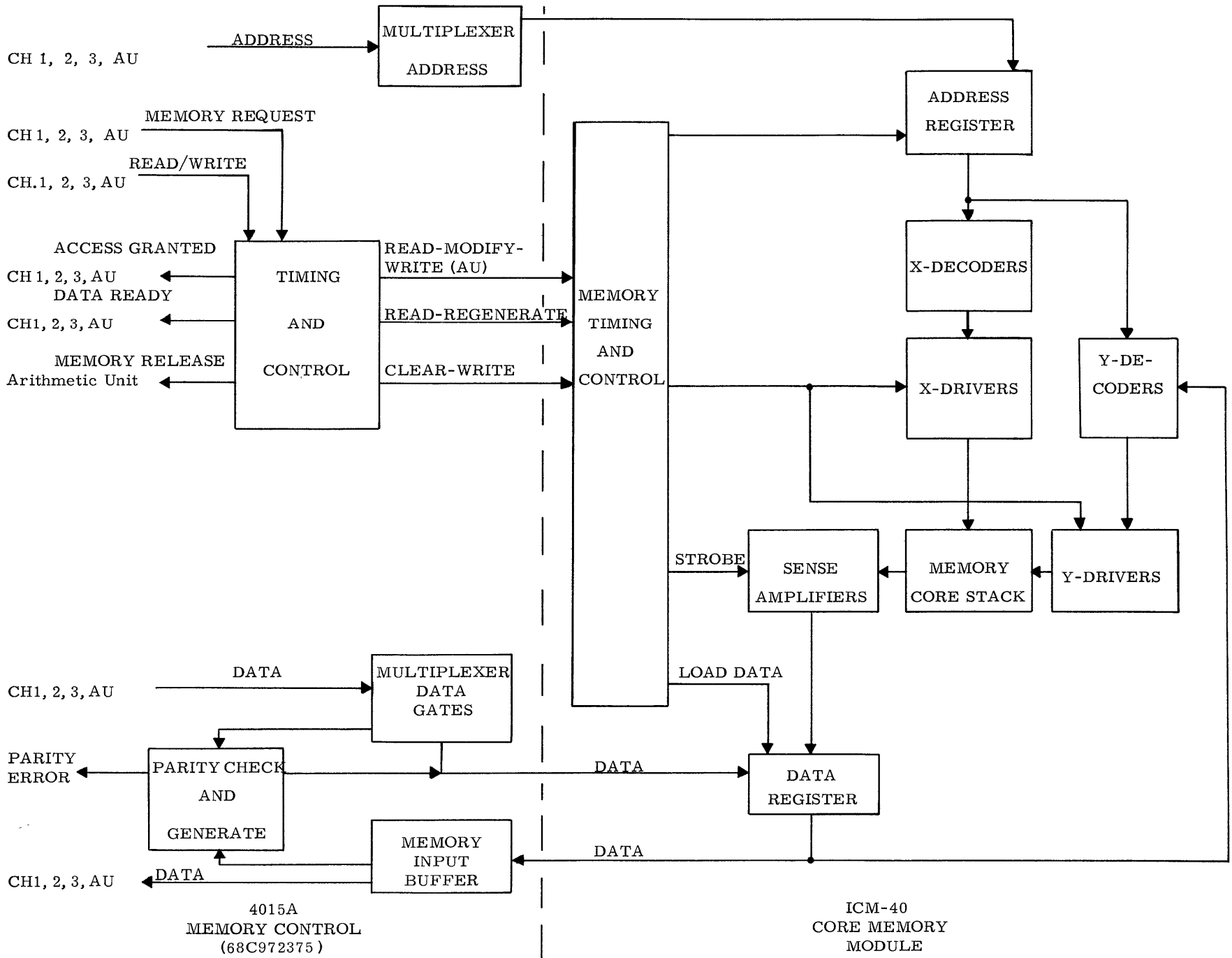


Fig. SYS. 1. Functional Block Diagram

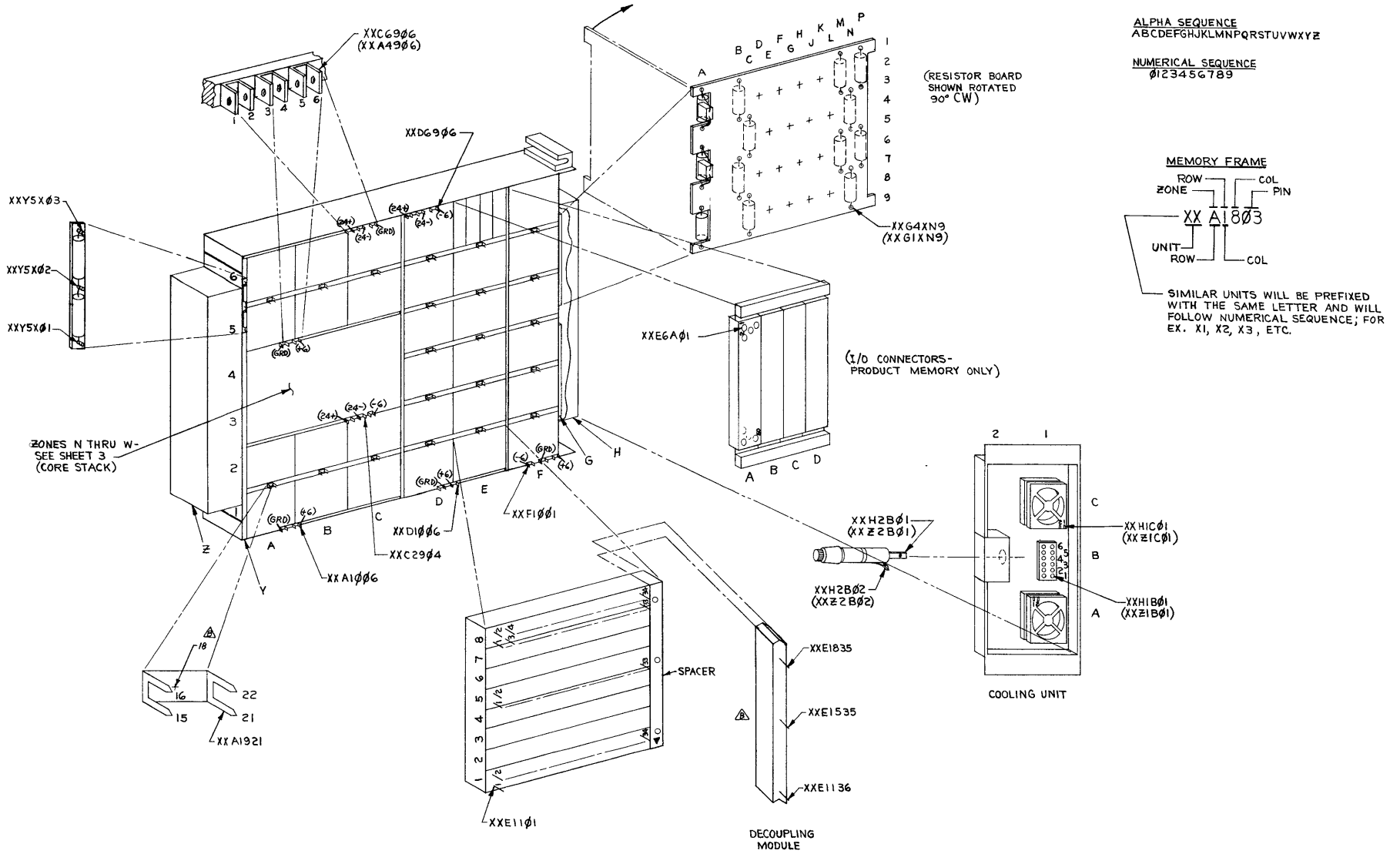


Fig. SYS. 2. ICM-40 Memory Arrangement

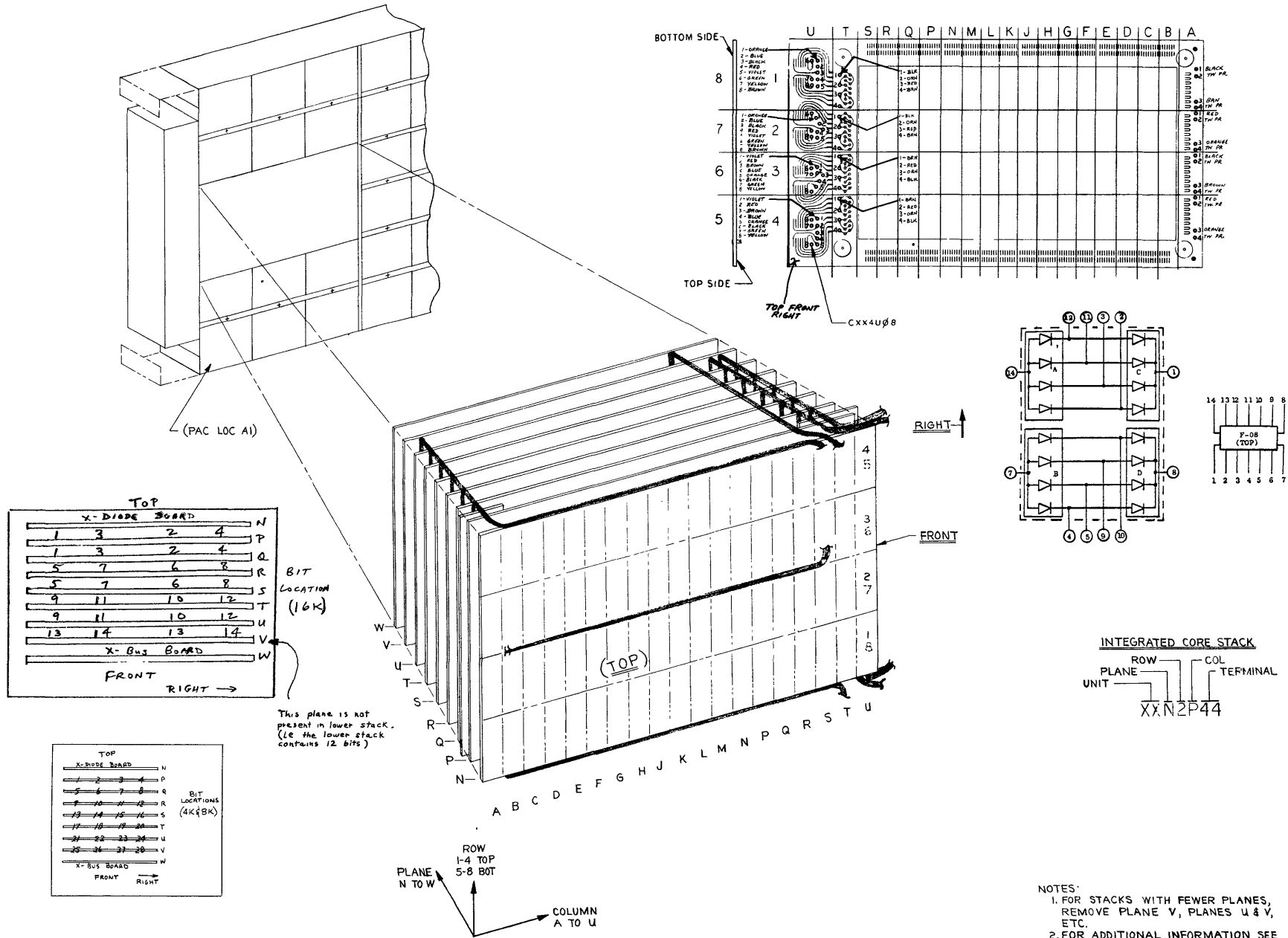


Fig. SYS. 3. Core Stack

$$\begin{aligned}
GOMBR1 &= F1MPR1 \\
GOMBR2 &= F1MPR2 \cdot \overline{F1MPR1} \\
GOMBR3 &= F1MPR3 \cdot \overline{F1MPR2} \cdot \overline{F1MPR1} \\
GOMBR4 &= F1MPR4 \cdot \overline{F1MPR3} \cdot \overline{F1MPR2} \cdot \overline{F1MPR1}
\end{aligned}$$

Enabling one of these gates starts the memory cycle and enables access to memory for the corresponding device. Address and data control of the memory is then allocated to the user device granted access.

A detailed description of the operation of the memory system for an Arithmetic Unit request and a User Device request is contained in this section. Both a read and a write operation of each request is discussed.

### ADDRESS DECODING AND SELECTION

The memory location address from the device granted access to memory is multiplexed via G1MA00 through G1MA13 of the Memory Control logic (68C972375, sheets 14, 15) to the Address Register of the ICM-40 Memory Module. These logic elements invert the address bits providing a "one" input for each "zero" address bit from the user device. The Address Register (See Appendix A, sheet A-6) bits are numbered from 1 thru 14 instead of 0 thru 13 as labeled in the Memory Control logic. Therefore, Memory Control address bit 0 is applied to Address Register bit 1, etc. When examining the address decoding, these two variations must be kept in mind.

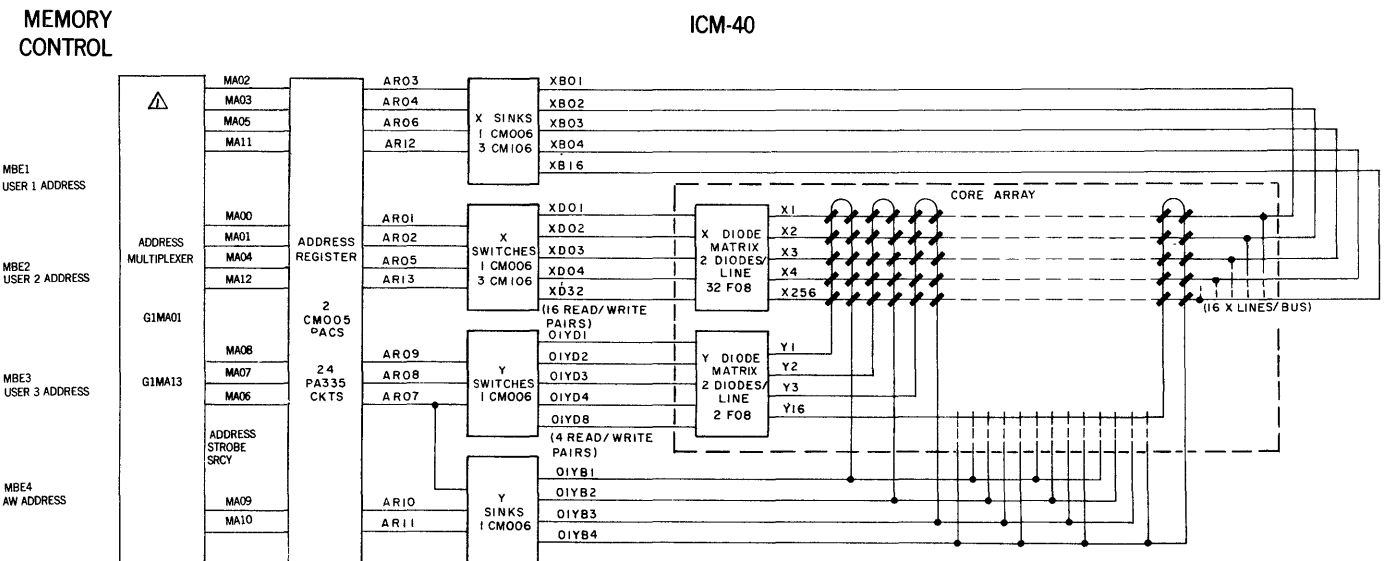
Fig. SYS. 4 illustrates a simplified diagram of the address decoding and selection for a typical bit (bit 1) of an 8K memory. The address bits are applied from the Memory Control logic to the Address Register in single-ended fashion and gated into the register at the start of

a memory cycle by the SRCY - signal. The Address Register outputs are double-ended and used to control the X and Y selection circuits.

Each Address Register output line shown represents a binary bit. Four bits are transferred to the X-switches and four others go to the X-sinks. The X-switches uniquely enable one of 16 read/write output pairs going to the diode matrix; the X-sinks select one of 16 read/write buses. The selected bus enables one end of 16 drive lines, only one of which has its other end connected to an enabled diode matrix. Thus, only one of the 256 X-drive lines will be selected.

The Y-selection is accomplished in a similar manner but with the inclusion of a read-write interchange, which is the operation of changing the polarity (direction) of read and write Y-currents with respect to X-currents. This saves decoding circuitry and is accomplished by having the read and write timing inputs a function of AR07. Selection of one of 32 effective Y-drive lines is made by selecting one of 16 wires. When the AR07 flip-flop is reset, the SWRW+ and SKRW+ signals are +6v during the read portion of the cycle and the SWWR+ and SKWR+ signals are +6v during the write portion. If the AR07 flip-flop is set, SWWR+ and SKWR+ are +6v at read time and SWRW+ and SKRW+ are +6v at write time. An X-drive line intersects with a Y-drive line at two cores as shown in Figure INT. 4. The relationship of X- and Y-currents (controlled by AR07) defines which of the two cores is addressed.

The 16K decoding and selection method is similar to that for the 8K, except two Y-switches are used per bit to select one of 32 Y-lines in 16K systems.



1 The address bits are complemented in the Address Multiplexer

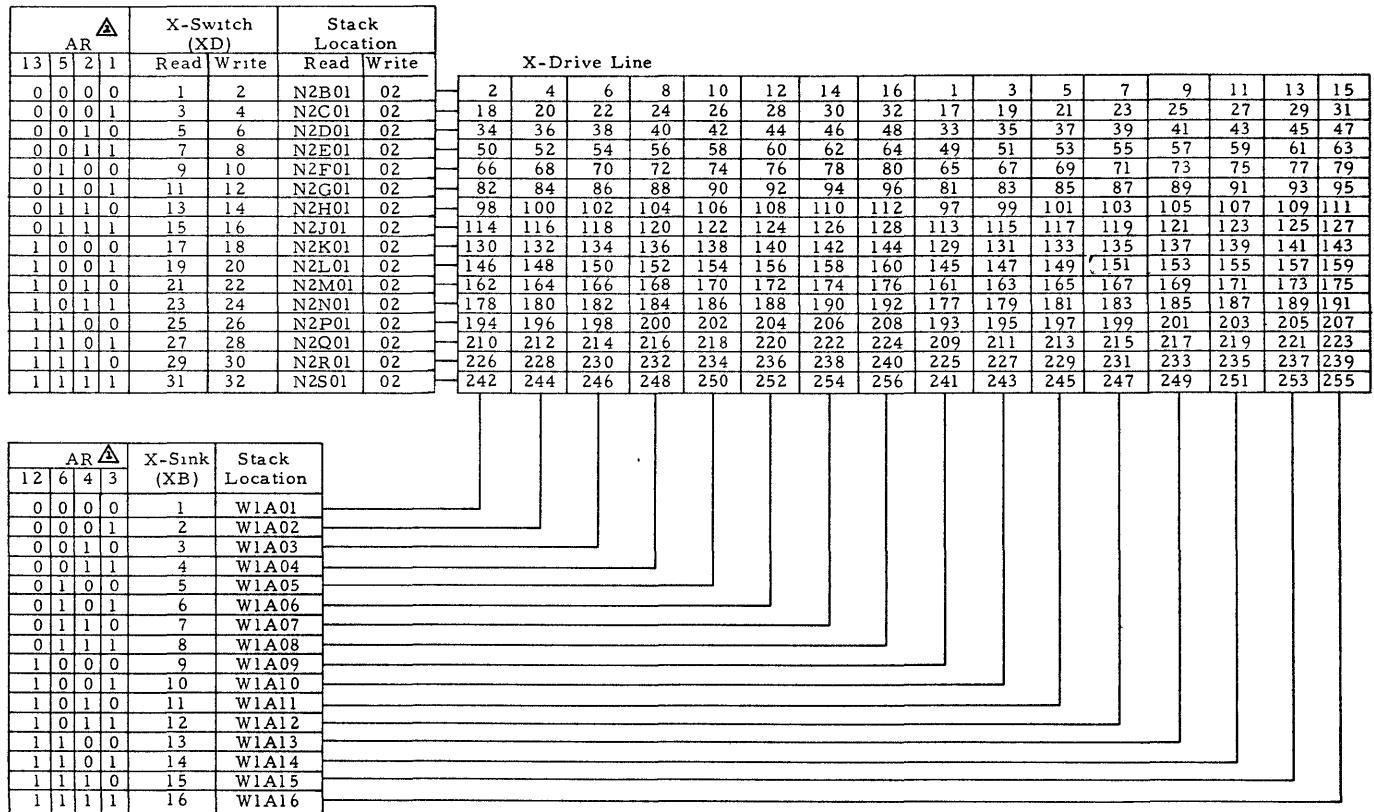
Fig. SYS. 4. Address Decoding and Selection

Table SYS. 1 shows the address decoding and X-drive line selection for 8K and 16K memories. The table shows the X-switch (XD01-XD32), sink (XB01-XB16) and drive line (X1-X256) selected by a given address. The stack location that is shown refers to the core stack drawing (Figure SYS. 3).

For example, if address register flip-flop outputs 13, 5, 2, 1 are in the "one" state (+6v) and 12, 6, 4, 3 are in the "zero" state (0v), read switch XD31 (stack location N2S01) and sink XB01 (stack location W1A01) are activated and X read current flows in drive line X242.

Table SYS. 2 shows address decoding and Y-drive line selection for 4K, 8K, and 16K memories. The plane locations are the connections of the Y-selection switch and sink outputs to the Y-core plane. The stack connections for a given bit can be determined from the decoding tables and the bit location diagrams (Figure SYS. 3). For example, read switch YD07 and sink YB03 are selected in bit 5 of an 8K memory by address AR11+, AR10-, AR09+, AR08+, and AR07- (10 110). Activating switch YD07 (stack location Q1U01) and sink YB03 (stack location Q1T03) selects drive line Y19.

Figure SYS. 5 is a simplified diagram of the Y-selection



$\Delta$  When using the multiplexer in the Memory Control Unit, these address bits are the complement of the address from the Arithmetic Unit or User Device.

Table SYS. 1. X-Decoding and Selection

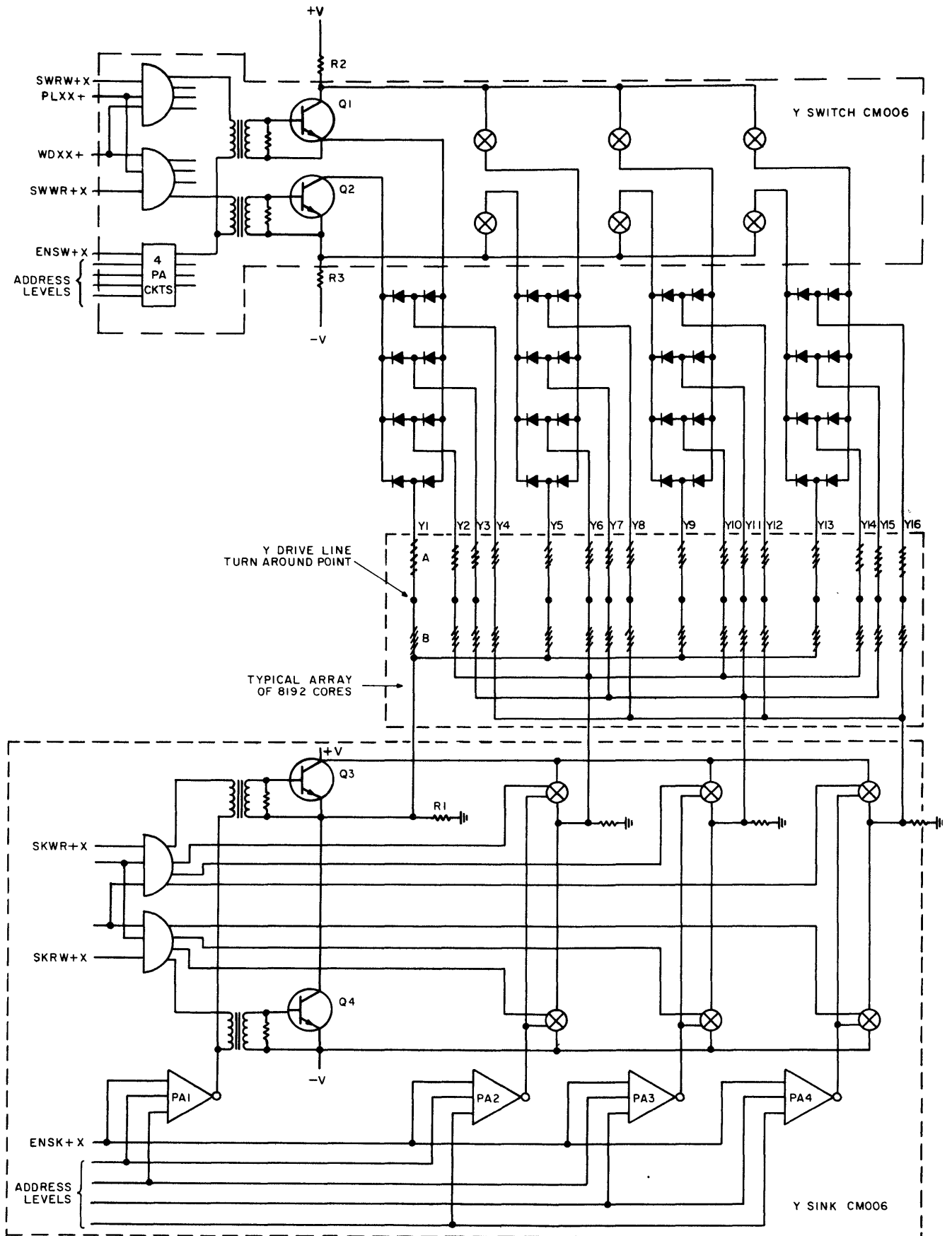
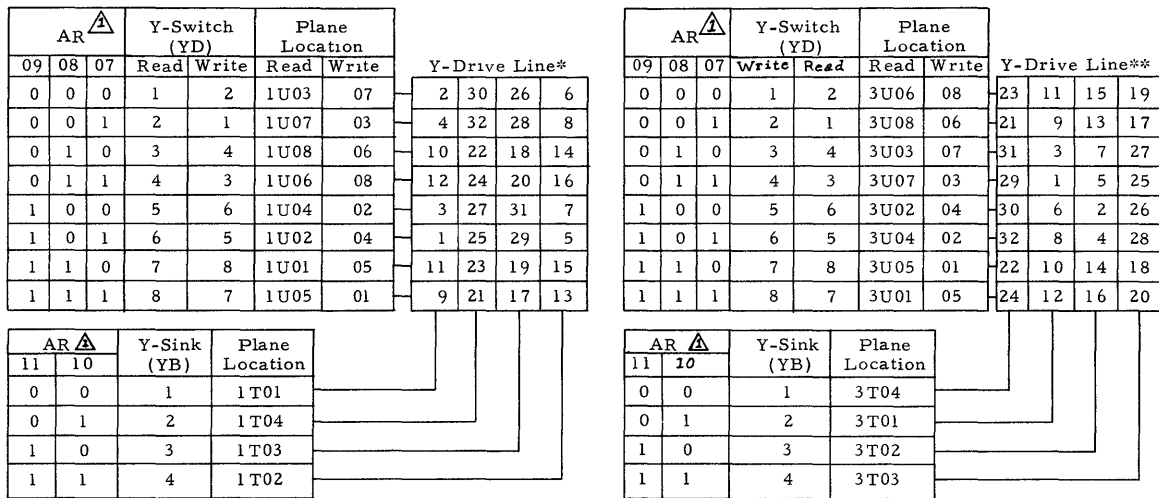


Fig. SYS. 5. Simplified Decoding and Selection Matrix





\*Use table for 4K and 8K bits 1, 2, 5, 6, 9, 10, 13, 14, 17, 18, 21, 22, 25, and 26 and 16K bits 1, 3, 5, 7, 9, 11, 13A, and 14A.

\*\*Use table for 4K and 8K bits 3, 4, 7, 8, 11, 12, 15, 16, 19, 20, 23, 24, 27, and 28 and 16K bits 2, 4, 6, 8, 10, 12, 13B, and 14B.

$\Delta$  When using the multiplexer in the Memory Control Unit, these address bits are the complement of the address from the Arithmetic Unit or User Device.

Table SYS. 2. Y-Decoding and Selection

The address levels and enabling inputs ENSK+X and ENSW+X do not change during the write portion of the cycle. When timing inputs SWWR+X and SKWR+X are generated, transistors Q2 and Q3 are turned on. This reverses the polarity of Y-drive current and write current flows from +V through Q3, Y1, CR1, Q2 and R3 to -V. The coincidence of X- and Y-write currents results in the generation of a "one" in the selected core.

Write Y-drive current flows only if the data register is a "one". If a "zero" was read out of the selected core during the read portion of the cycle, the Y-switch is prevented from turning on during write time by the WDX+ input. The selected core receives only an X-write current so it remains in the "zero" state.

If AR07 is in the set state, read current flows through R3, Q2, CR1, Y1, and Q3 and write current flows through R2, Q1, CR2, Y1, and Q4. This results in the selection of a core in the B-half of Y1.

The selection method shown in Figure SYS. 5 is used for X- and Y-selection in 8K and 16K memories. Y- and X-selection is the same for 8K and 16K memories but, 16K memories use more Y-selection switches than 8K memories.

The following statements summarize the X and Y line selection and control.

- Address Register bits 13, 5, 2, and 1 control the selection of the X Switch Lines (XD).

- Address Register bits 12, 6, 4, and 3 control the selection of the X Sync lines (XB).
- Address Register bits 8 and 9 control the selection of the Y Switch lines (YD).
- Address Register bits 10 and 11 control the selection of the Y Sync lines (YB).
- Address Register bit 7 is used to control the direction of current flow in the Y Switch (YD) and Y Sync (YB) lines.
- During a write into core cycle, the Y lines are controlled by the corresponding bits of the Data Register. Only those Y lines corresponding to a "one" in the Data Register are enabled to flip the core to the "one" state.

## ARITHMETIC UNIT ACCESS

The Arithmetic Unit utilizes the Read-Modify-Write mode of memory operation. The timing and basic operation of the memory for both a store (write) or read operation is the same (See Fig. SYS. 6, 7). In either a write or read operation the addressed memory location is cleared and the data contained therein is strobed through the Sense Amplifier into the memory data register. If it is a read operation (STORE), this data is gated to the Arithmetic Unit B Register (D1BMEM). The contents of the B Register are then written back in the memory location during the write portion of the memory

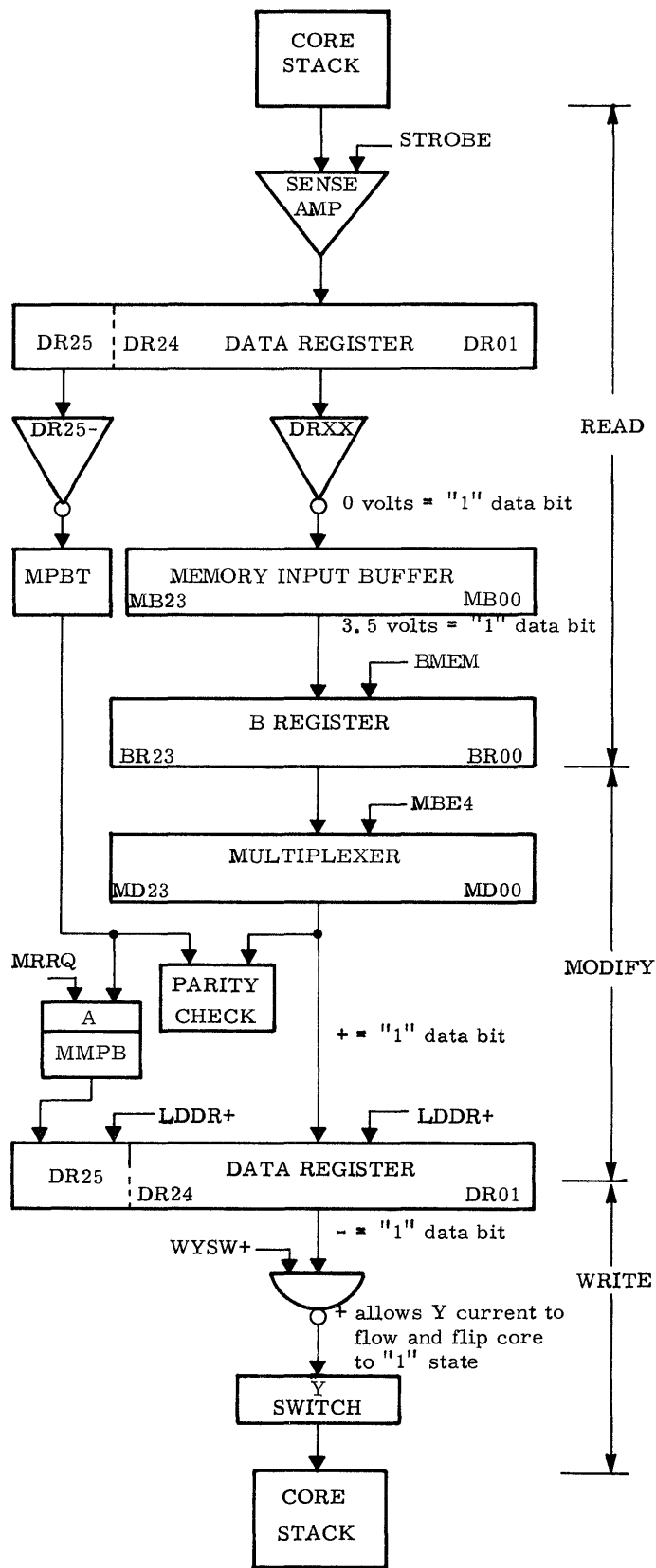


Fig. SYS. 6. AU Read, Block Diagram

cycle. If it is a read operation, the contents of the B Register is the data read from the addressed core location. If it is a write operation, the data from memory is not gated to the B Register and the B Register contains the new data to be stored in the addressed core cell. During a read operation, the data read from the addressed memory location is checked for odd parity. If an odd number of one bits is not read from memory an error indication is provided and the word read is stored back with incorrect parity. If the operation is write, the data to be stored from the B Register is checked by the parity generator. If an even number of ones is contained by the B Register data, a one parity bit is generated. Therefore, an odd number of data bits is always stored in the addressed location during a write operation.

### Memory Control Timing

Fig. SYS. 8 contains a timing diagram with logic equations of the memory control unit for an Arithmetic Unit memory access operation. The memory control unit timing is initiated by a memory request signal from the Arithmetic Unit (AUG1SMRQ) and operates independent of the ICM-40 Memory Module. The Memory Control unit applies control signals to the ICM-40 to control the Read-Modify-Write cycle (DIMBE4) and to start the read (GOMROW) and write (GOMRW) portions of the cycle.

The first clock pulse following G1SMRQ is used to determine the priority of memory access. If no other requests are present, F1MPR4 is set and D0MBR4 is enabled. D0MBR4 enables D1MBE4. D1MBE4 is applied to the ICM-40 to enable the Read-Modify-Write mode of operation. D1MBE4 also gates the memory location address from the Arithmetic Unit through the multiplexer (G1MA00 - G1MA13) to the Address Register of the ICM-40. D0MBR4 also enables D1MAE4 to gate the contents of the B Register thru the multiplexer (G1MD00 - G1MD23) to the Data Register of the ICM-40.

After the Arithmetic Unit has been granted access to memory, the memory time counter (F1MTC1, 2, 3, 4) is enabled and the read signal (GOMROW) is applied to the ICM-40. The memory time counter is a binary counter incremented by the 10 megacycle clock pulses. The Time Counter controls the timing within the memory cycle.

After a count of one, the time counter disables the read control signal (GOMROW) to the ICM-40. When the counter is incremented to a count of six, a Data Ready signal (D1MDRY) is applied to the Arithmetic Unit indicating that the addressed memory location has been read out and sequencing can continue in the AU. The next clock pulse following Data Ready enables GOMRW to start the write portion of the memory cycle in the ICM-40. When the time counter has been incremented to a count of 17, a Memory Release signal (F1MRLS) is applied back to the Arithmetic Unit to indicate the end of the memory cycle. This Memory Release signal may be used by the AU to end the current Sequence State and to enable a new Memory Request (G1SMRQ)

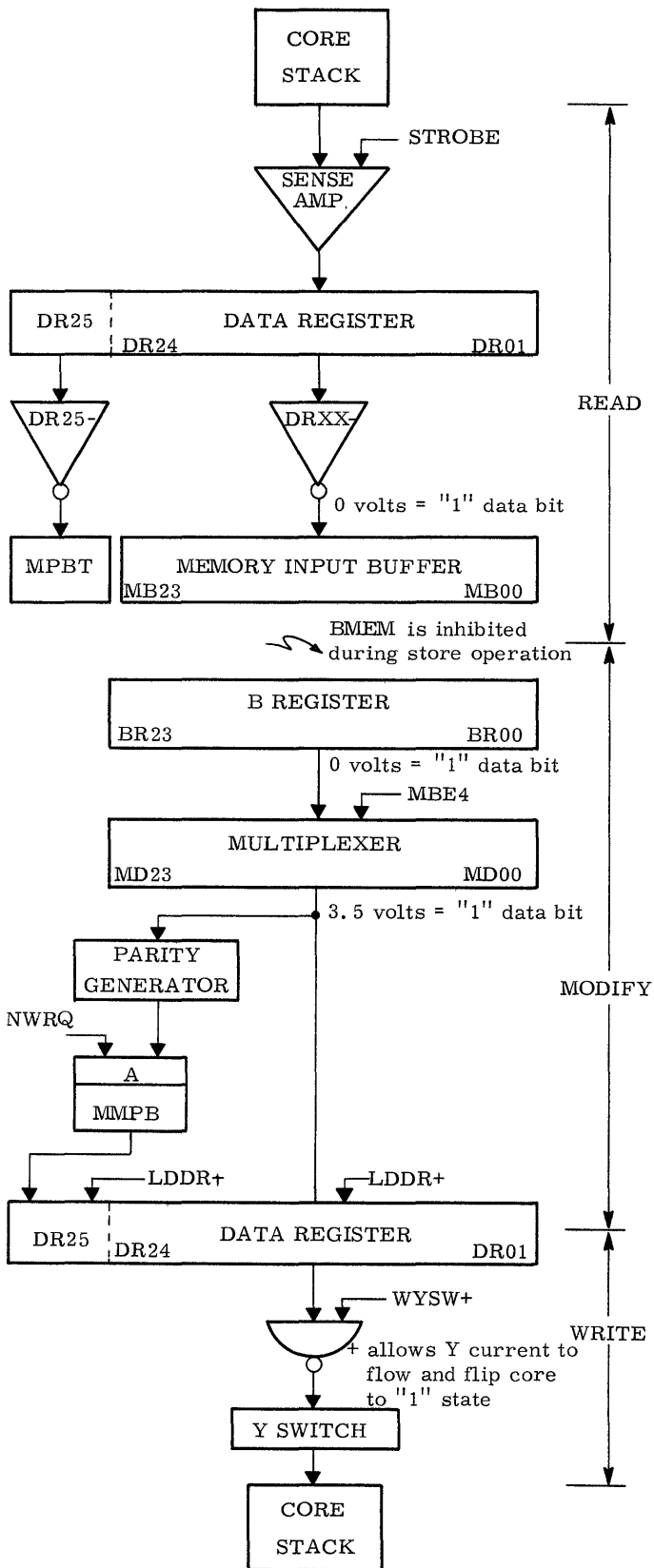


Fig. SYS. 7. AU Write, Block Diagram

for the next Sequence State.

The operation of the Memory Control unit described above applies to either a store (write) or read (store) memory request.

### ICM-40 Memory Module Operation

Timing diagrams of the memory module control signals are contained in Figures SYS.9 and 10. Figure SYS.9 compares the time duration and control signal levels required for the ICM-40 operation with the actual control signals provided by the memory control logic. Fig. SYS.10 illustrates the relationship of the various control signals and the delays encountered. This drawing illustrates the timing of the memory as close as possible. The exact memory timing will vary slightly from unit to unit. The strobe delay setting that provides optimum operation of each memory is labeled on the memory module. Refer to these timing diagrams and the ICM-40 logic contained in Appendix A during the following discussion.

The Arithmetic Unit access granted signal, DIMBE4, is applied to the Read-Modify-Write input, RMWC+, of the memory module. The read control signal, GOMROW, is applied to the Read-Regenerate control input, XRRC-, of the memory module. This XRRC- signal is inverted and then ANDed with the memory busy not signal, MBSY-. Assuming that the memory is not busy, since the memory control logic will not permit a request until after the previous memory cycle would be completed, the Start Read Cycle signal, SRCY-, will be enabled. SRCY- sets the Read Control flip-flop, RCYF. SRCY- also enables the Reset Data Register pulse, RDRP-, to reset the Data Register flip-flops in preparation for the data to be read from the memory cells. SRCY- also gates the memory location address to the Address Register, AR01 - AR14. XRRC+ and MBSY- also sets the Read-Regenerate Control flip-flop RRCF.

With the Read Control flip-flop set, RCYF- is at zero volts. This signal is applied to the input of a 300 nanosecond delay line. The delay line has 24 taps each of which are 12 nanoseconds apart. Fig. SYS.10 indicates the typical connections of this delay line. Tap C6 of the delay line is connected to M4 pin 2 enabling ERAD+ for approximately 72 nanoseconds. ERAD+, in conjunction with RSKA+, RYSW+, and RXSW+, are used to enable decoding of the address, enable the Read X and Y Switches, and enable the Read Sink to flip the addressed core cell bits to the zero state. The RSKA+, RYSW+, and RXSW+ signals are generated by delaying RCYF- in the delay line and jumpering the outputs to the gates as shown in Fig. SYS.10.

The Strobe signal, STRB-, is generated from the 300 nanosecond delay line and a 50 nanosecond delay line to provide more precise increments for strobe adjustment. The delayed signal is ANDed with RRCF+ (Read-Regenerate) which was enabled by XRRC+ and MBSY-. The exact setting of the strobe is labeled on each memory module. The strobe signal gates the "one" bits sensed from the addressed core cell into the Data Register, DRXX, causing the Data Register bit to set.

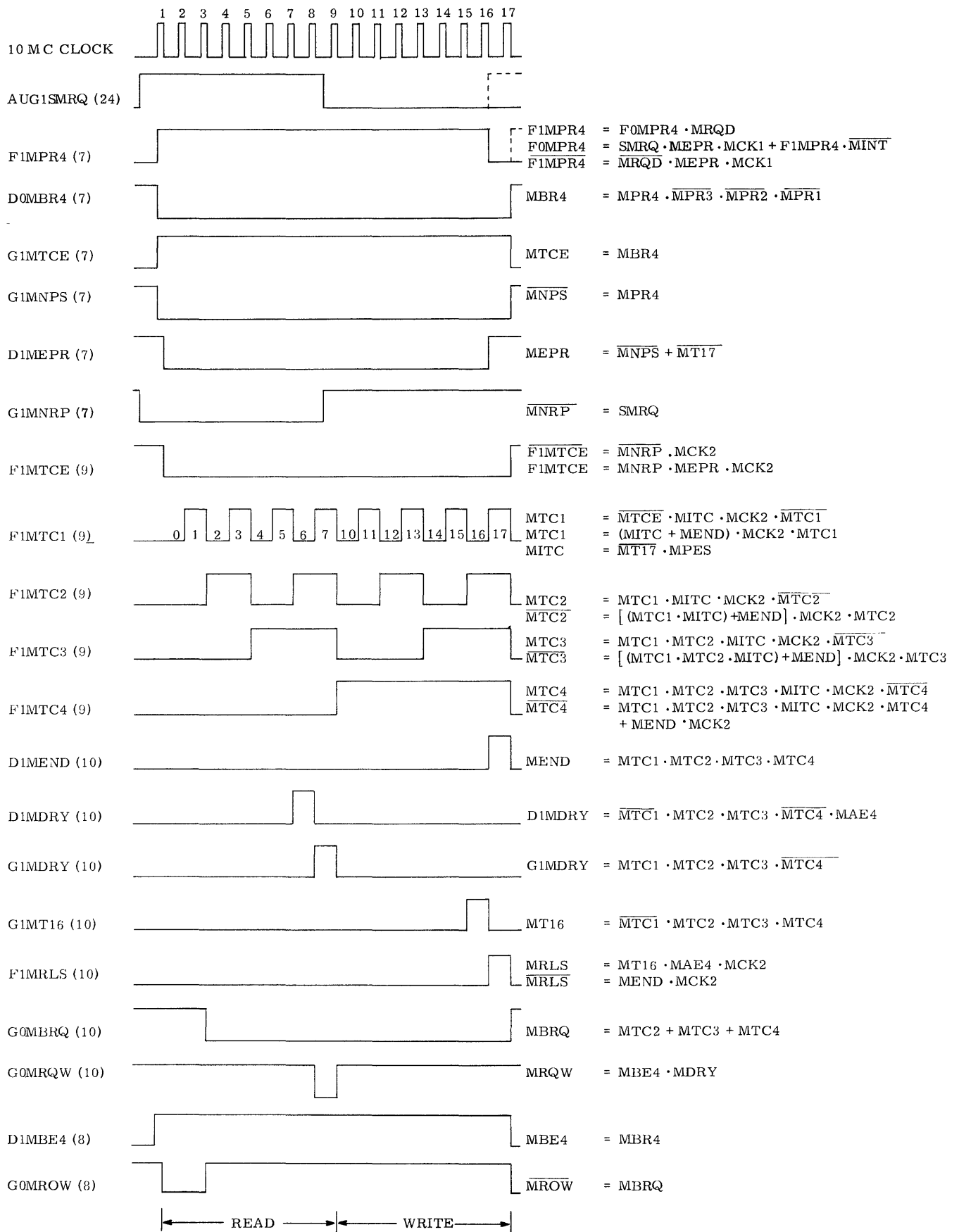


Fig. SYS. 8. Memory Control Timing - AU USER

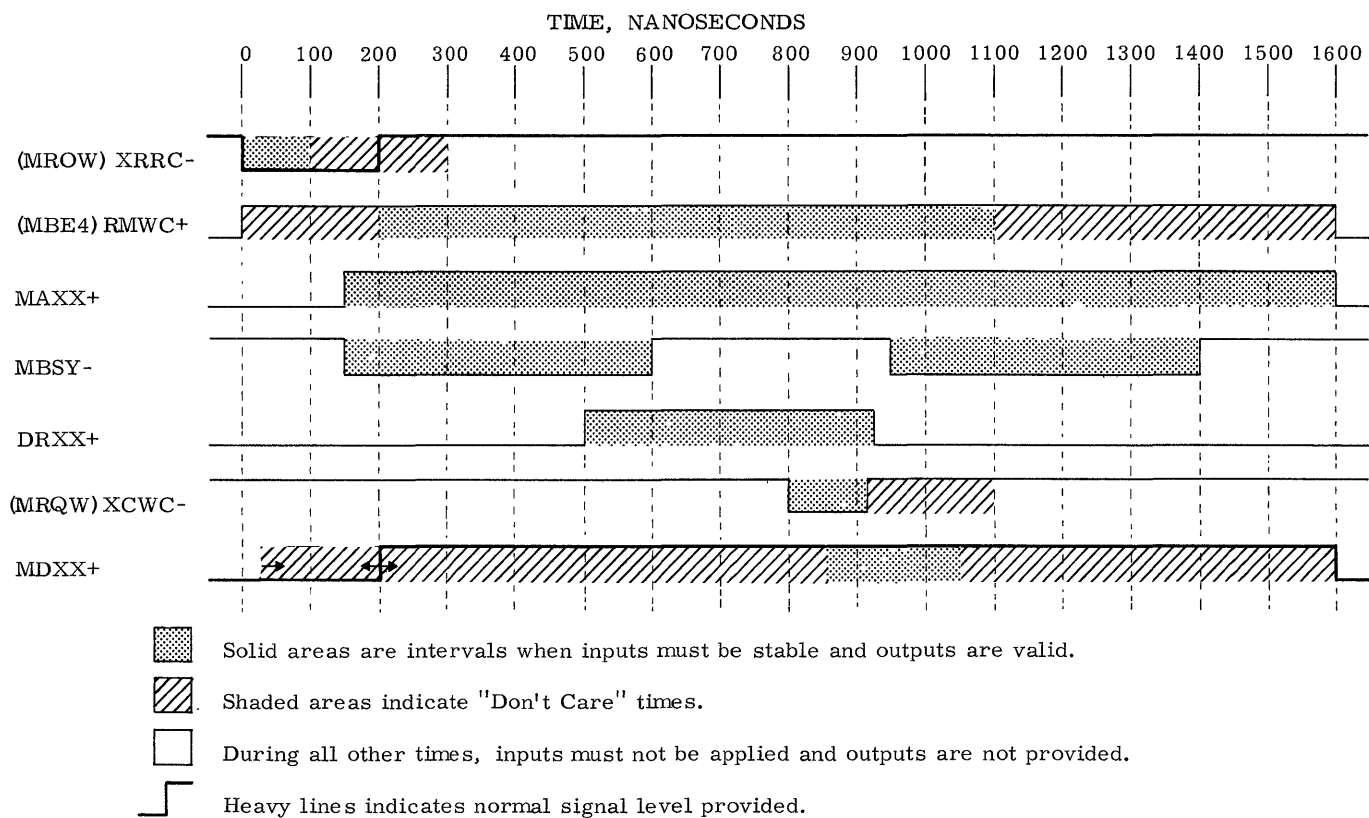


Fig. SYS. 9. ICM-40 Timing - AU User

From the Data Register, this data is applied thru the Memory Input Buffer of the Memory Control logic. If this is a read operation (Store) the data is gated to the B Register (D1BMEM). If this is a store operation, D1BMEM is inhibited and the B Register contains the new data to be stored in the addressed cell. The data read from the addressed cell is lost. During read operations, the data read from the addressed cell is checked for a parity error. Parity checking is discussed at the end of this section.

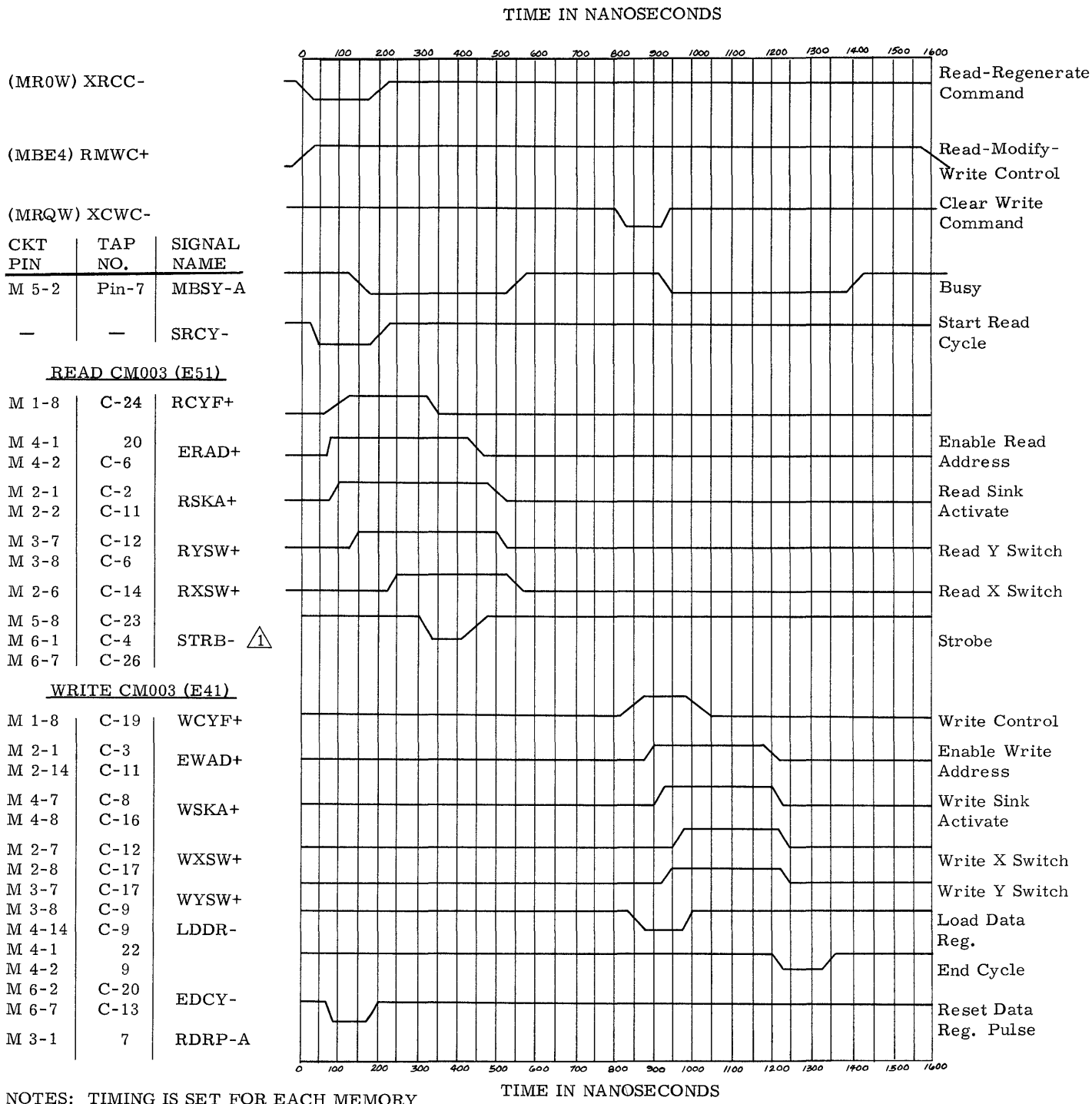
Following this read portion of the Read-Modify-Write memory cycle, the memory control logic applies a Request Write control signal (MRQW) to the Clear Write input (XCWC-) of the ICM-40 module. This XCWC-signal sets the write control flip-flop (WCYF) and resets the Read Regenerate Control flip-flop (RRCF). Setting WCYF and resetting RRCF generates LLDR which is used to gate the contents of the B Register into the Data Register. The Data Register will contain the data to be stored in the addressed memory location. The Enable Write Address (EWAD+), Write Sink Activate (WSKA+), Write X Switch (WXSX+) and Write Y Switch (WYSW+) signals are enabled by delaying the WCYF- signal. Fig. SYS. 10 indicates the delay line connections and timing relationships associated with these signals. During this write cycle, the Y lines are controlled (WDXX+) by the data bits in the Data Register. Each data bit in the Data Register that is a "one" enables the corresponding addressed Y line to

set the core in the "one" state. "Zero" bits in the Data Register inhibit the Y line (WDXX+) leaving the corresponding core in the "zero" state. After the data bits are stored in the addressed location, the memory cycle is ended.

### USER 1, 2, 3 ACCESS

User 1, 2, and 3 devices may include from one to three Drum Controllers, and/or Disc Controllers and/or Peripheral Controller Couplers. When a user device is accessing core memory to store new data in a selected cell, the ICM-40 Memory Module operates in the Clear-Write mode. The addressed memory cell is first cleared and then the new data to be stored controls the Y Switch to flip those cores corresponding to the "one" data bits. When the user device is accessing core memory to obtain data (read), the ICM-40 module operates in the Read-Restore mode. The addressed memory cell is cleared and the "one" bits sensed from the cell set the corresponding Data Register flip-flops. This data is then transferred to the user device and is also used to control the Y Switch to restore the core cells back to their original configuration. Figures SYS. 11 and 12 illustrate the basic operation of the read and write operations.

Parity is checked during all read operations and parity is generated, if required, during all write operations. A detailed discussion of parity checking and generation



NOTES: TIMING IS SET FOR EACH MEMORY TO GIVE OPTIMUM OPERATING MARGINS. TAPS AND TIMING ARE NOMINAL AND MAY VARY SLIGHTLY FROM THE VALUES SHOWN.

(READ-MODIFY-WRITE TIMING)

Fig. SYS. 10. ICM-40 Timing, AU USER

is provided later in this section.

### Memory Control Timing

Fig. SYS. 13 illustrates the basic timing of the Memory Control unit (68C972375) for a Device 1, 2, or 3 access to memory. This basic timing of the Memory Control Unit for a user 1, 2, or 3 access is very similar to that of an Arithmetic Unit access. The differences exist primarily in the control signals applied to the ICM-40 Memory Module. If the user device is requesting a write operation, signals (GOMWRQ and GOMRQW) are generated and applied to the ICM-40 to enable the Clear-Write operation. If the user device is requesting a read operation, signals (GOMRRQ and GOMROW) are generated and applied to the ICM-40 Module to enable the Read-Regenerate mode of operation.

The Memory Control Unit timing is initiated by the request signal from the user device. The first clock pulse following the request signal is used to determine the priority access. If no other higher priority access is present, F1MPR1, 2, or 3 is set. Setting F1MPR1, 2, or 3 enables the corresponding access gate GOMBR1, 2, or 3. This signal is inverted (D1MAE1, 2, 3 and D1MBE1, 2, 3) and used to gate the memory location address from the device through the multiplexer (G1MA00 - G1MA13) to the Address Register of the

ICM-40, and gates the data from the device through the multiplexer (G1MD00 - G1MD23) to the Data Register of the ICM-40. This data will only be set in the Data Register, however, if a write operation is requested. D1MBE1, 2, or 3, is also AND with the READ/WRITE signal from the device. If it is a read request, GOMROW and GOMRRQ are enabled to control parity checking and to initiate the Read-Regenerate mode of operation within the ICM-40. If it is a write request, GOMRWQ and GOMRQW are enabled to control parity generation and to initiate the Clear-Write mode of operation within the ICM-40.

After access has been granted, the Memory Time Counter (F1MTC1, 2, 3, 4) is allowed to increment at each clock pulse. This time counter provides control of the duration of the GOMROW and GOMRQW control signals so as not to exceed 300 nanoseconds. When the time counter is incremented to 7, a Data Ready (GOMRD1, 2, 3) signal is enabled. This signal is applied to the user device indicating that the request signal may be terminated and that if this is a read operation, the data read from core is available.

The time counter continues to be incremented in a binary manner until it is equal to 17<sub>8</sub>. When the time counter is equal to 17<sub>8</sub>, the memory cycle is complete.

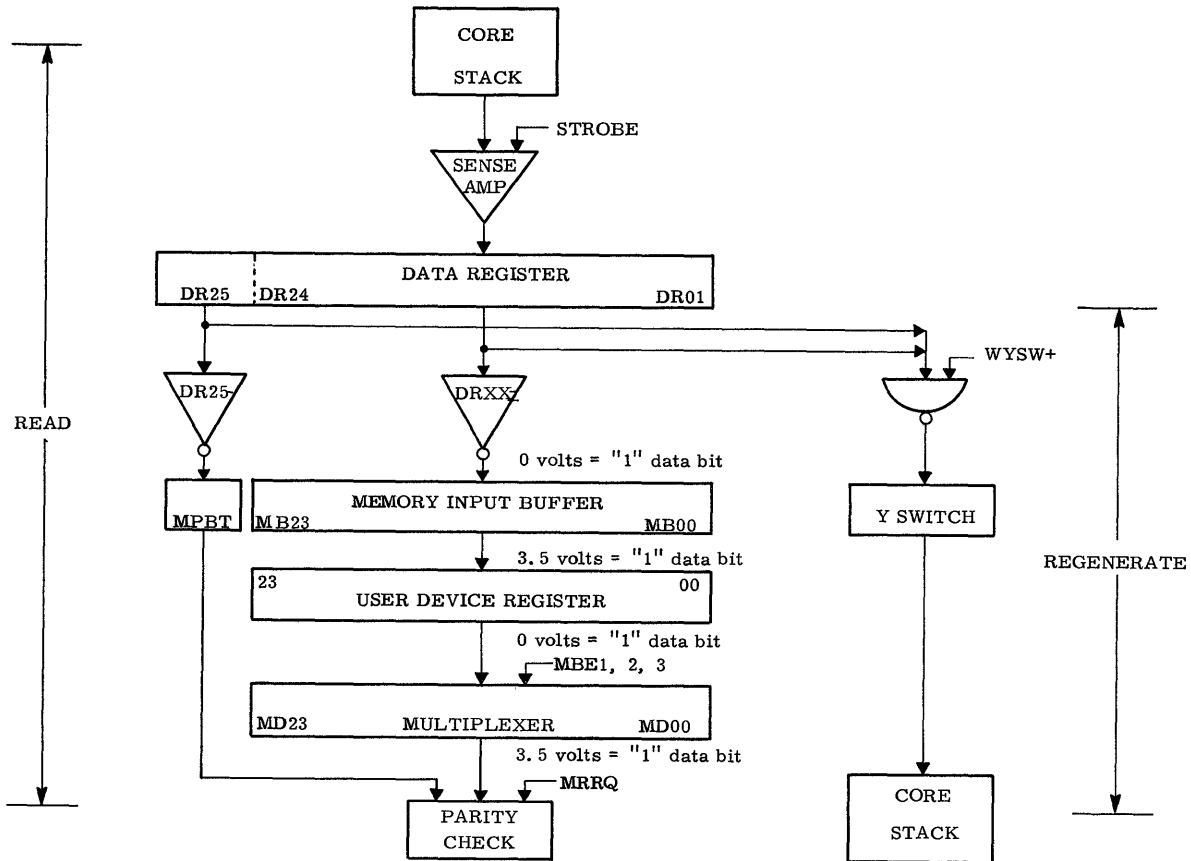


Fig. SYS. 11. USER 1, 2, 3, Read Block Diagram

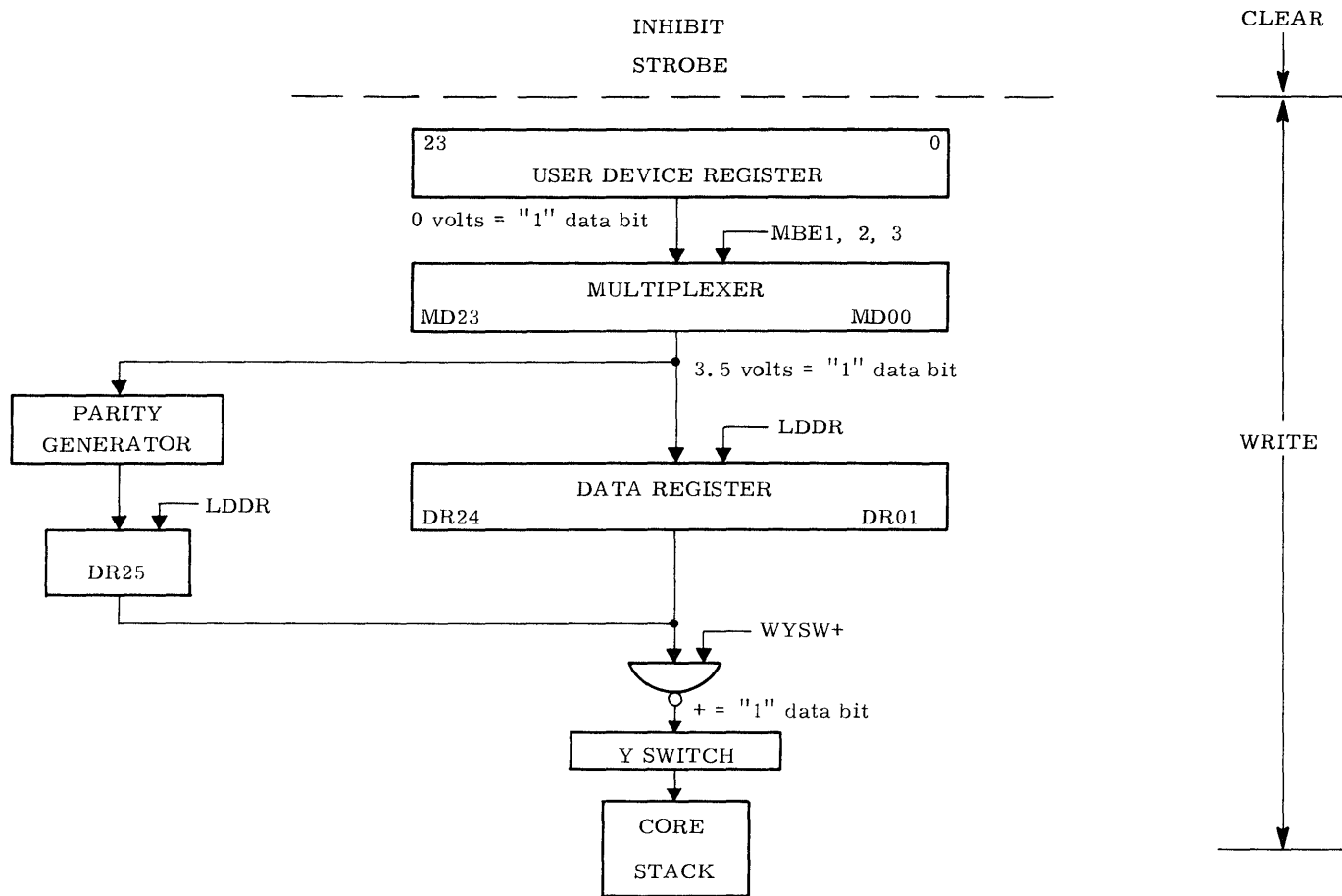


Fig. SYS. 12. USER 1, 2, 3, Write Block Diagram

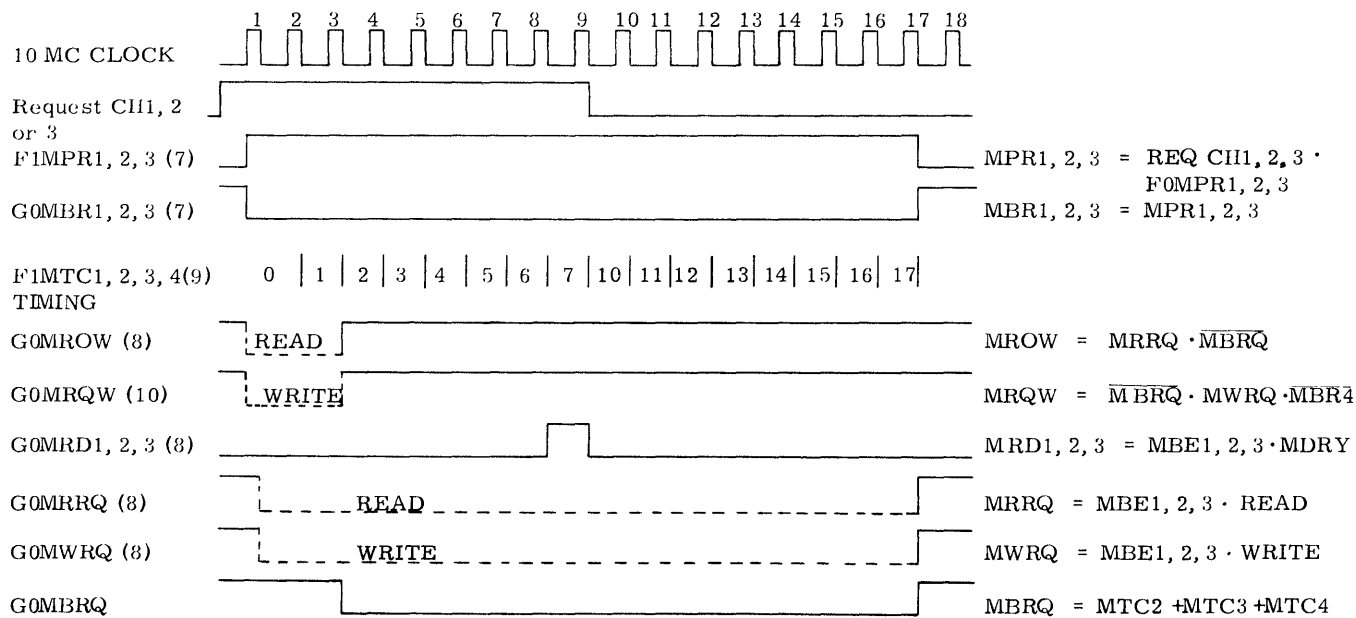


Fig. SYS. 13. Memory Control Timing, USER 1, 2, 3



## ICM-40 Memory Module Operation

Timing diagrams of the primary control signals within the ICM-40 Memory Module are presented in Figures SYS.14 and 15. These timing diagrams illustrate the timing for both a Read-Regenerate and a Clear Write mode of operation. The differences between these two modes involve the Strobe (STRB-) and the Load Data Register (LDDR-) signals. The strobe is enabled only during the Read-Regenerate mode. This strobe signal gates the data sensed from core into the Memory Data Register for a read operation. During a write (Clear-Write) operation, the strobe is inhibited and the data sensed from core is lost. The load data register signal gates the new data to be stored into the Memory Data Register during a write operation. During a read operation, the load data register signal is inhibited to retain the data sensed from core so that it may be restored in the addressed core cell.

Fig. SYS.14 illustrates the timing limits of the ICM-40 module control signals and relationship of the actual control signals from the Memory Control Unit.

Refer to these aids as well as the ICM-40 logic contained in Appendix A at the end of this section.

### Read-Regenerate Mode

The Read-Regenerate Mode of operation within the ICM-40 is initiated by the GOMROW signal from the Memory Control Unit. As previously described, this signal is enabled within the Memory Control Unit when a user device has been granted access for a read from core operation. GOMROW is applied to the Read-Regenerate control input, XRRC-, of the ICM-40 Module. This XRRC- signal is inverted and then ANDed with the not busy signal (MBSY-) to set (SRCY-) the Read Control flip-flop (RCYF) and to set the Read-Regenerate Control flip-flop (RRCF). Due to the timing of the Memory Control Unit, the ICM-40 will not be busy when the Read-Regenerate control signal is applied. RRCF+ is used to enable the Strobe pulse (STRB-) and to disable the Load Data Register pulse since this is a read operation.

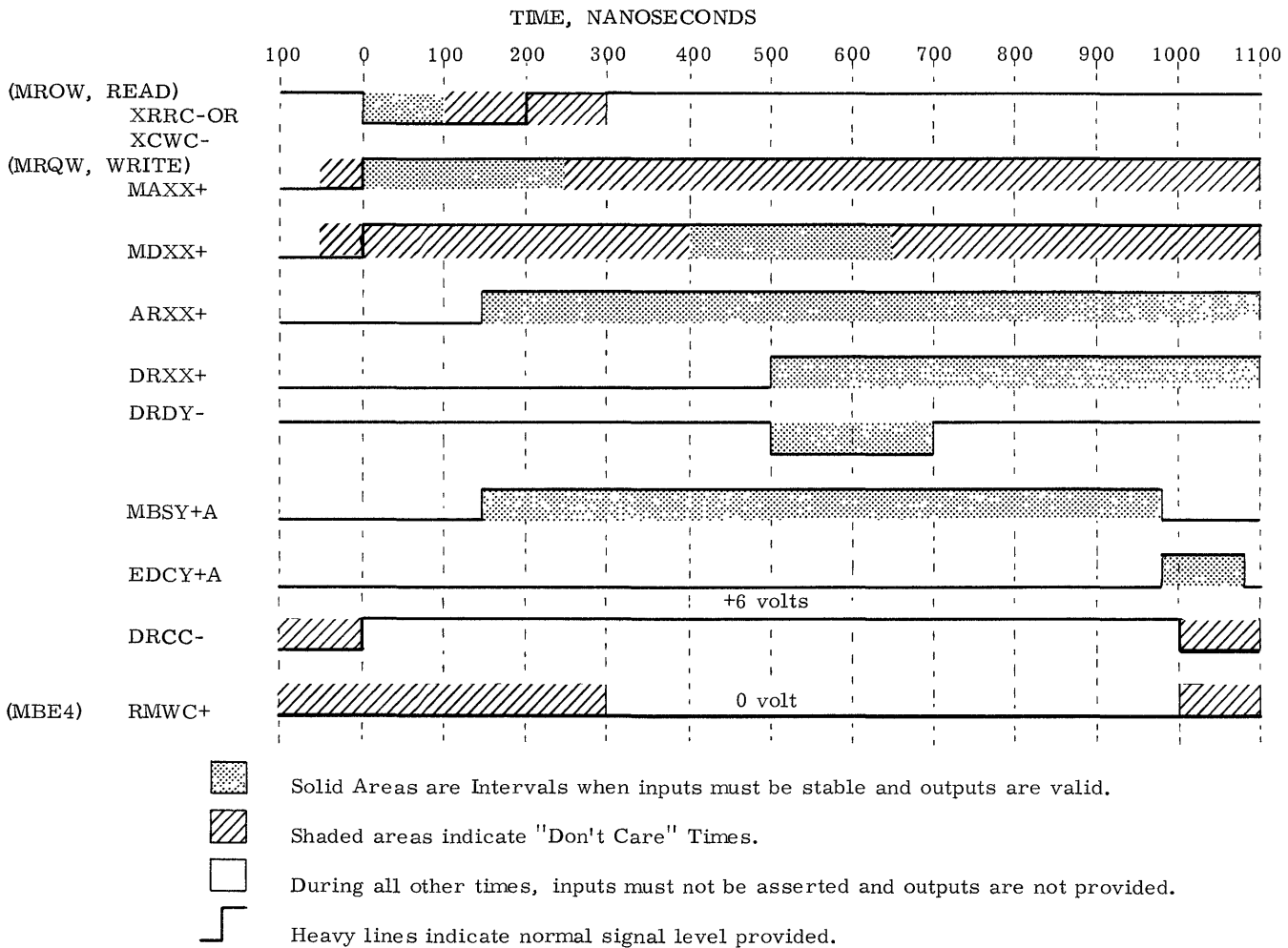
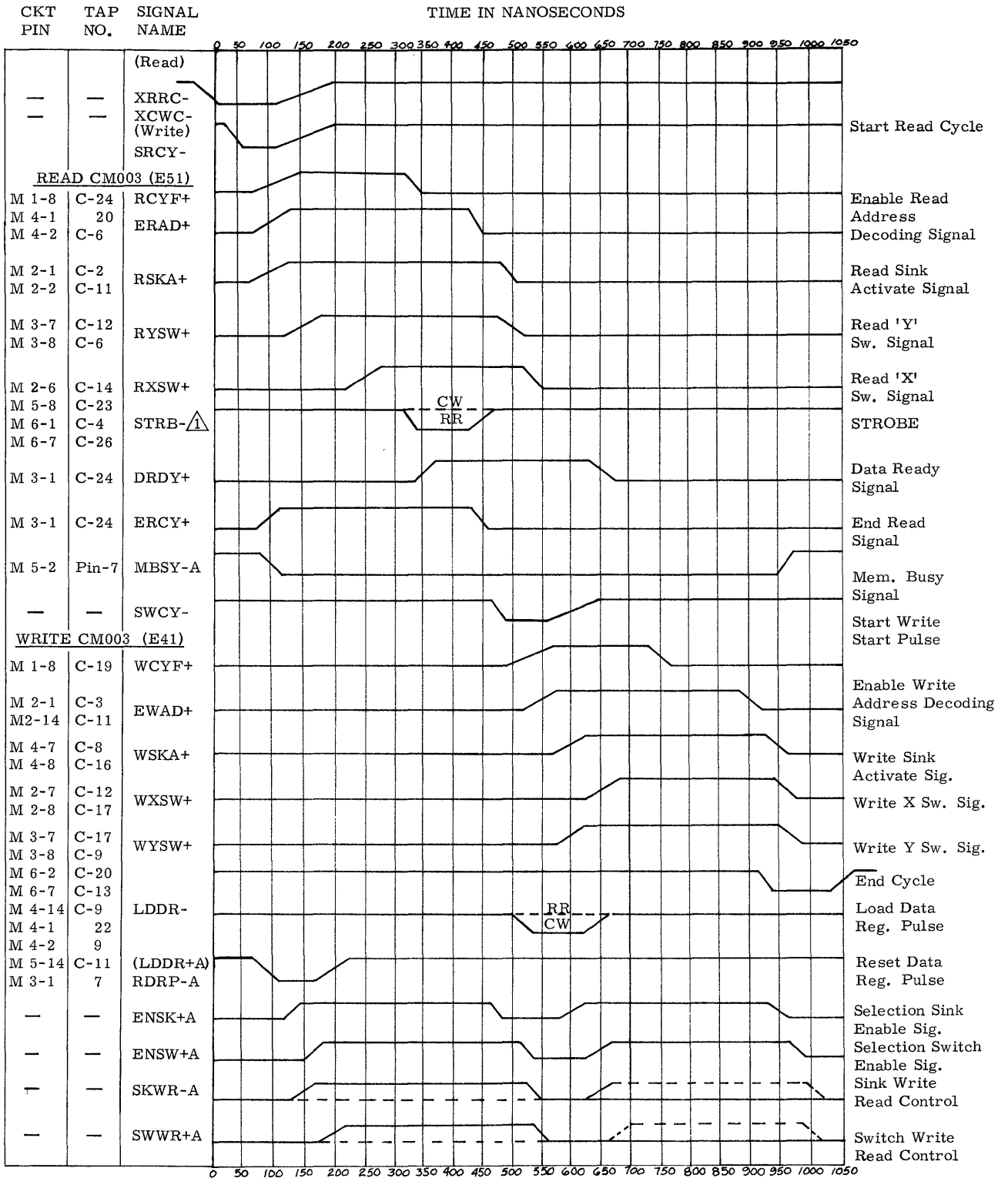


Fig. SYS.14. ICM-40 Timing, USER 1, 2, 3



NOTES: TIMING IS SET FOR EACH MEMORY TO GIVE OPTIMUM OPERATING MARGINS. TAPS AND TIMING ARE NOMINAL AND MAY VARY SLIGHTLY FROM THE VALUES SHOWN.

CW = Clear Write = MRQW  
RR = Read Regenerate = MROW

(CLEAR WRITE/READ REGENERATE TIMING)

Fig. SYS. 15. ICM-40 Timing, USER 1, 2, 3

The SRCY- signal, enabled by XRRC+ and MBSY-, also gates the memory location address from the Memory Control unit into the Address Register of the ICM-40 and resets the Data Register (RDRP).

Setting the Read Control flip-flop RCYF generates a series of pulses to enable the X and Y drive lines. These pulses (ERAD+, RSKA+, RYSW+, and RXSW+) are timed by jumpers from the 300 nanosecond delay line. Fig. SYS.15 illustrates typical timing and delay line connections. Each tap of the delay line provides 12 nanoseconds delay. The precise timing of these signals may vary slightly from one module to another. Each module is stamped with the optimum setting of the Strobe pulse.

The Strobe pulse samples the core signal during this read portion of the cycle. A "one" signal output exceeds the threshold of the sense amplifier providing an output to set the associated Data Register flip-flop (DRXX). A "zero" readback will not exceed the sense amplifier threshold, and the Data Register flip-flop will remain reset. The contents of the Data Register are applied thru the Memory Input Buffer of the Memory Control to the user device.

The trailing edge of End Read Cycle signal (ERCY+), generated from the RCYF signal through the delay line, enables the SWCY- signal to set the Write Control flip-flop (WCYF). The WCYF signal, in conjunction with the 300 nanosecond delay line is then used to enable the addressed X and Y lines. The X and Y lines are also controlled by the contents of the Data Register during this portion of the Read-Regenerate cycle. Enabling these X and Y lines, restores the data back in the addressed memory location to complete the memory cycle.

### Clear-Write Mode

The Clear-Write mode of memory operation is enabled by G0MRQW from the Memory Control logic. This signal is enabled at the start of the memory cycle when a write (store) into memory request from a device is provided access. G0MRQW is applied to the Clear-Write Control input, XCWC-, of the ICM-40.

Operation in the Clear-Write mode is the same as that described for the Read-Regenerate mode except that the Read-Regenerate Control flip-flop (RRCF) is reset by the XCWC+ signal. With RRCF reset, the Strobe pulse (STRB-) is inhibited and the Load Data Register signal (LDDR) is enabled. This results in a loss of data from the addressed location and gates the data to be stored from the user device into the Data Register. This data then controls the Y lines and flips the corresponding cores to the "one" state.

The SRCY- signal used to start the read cycle is enabled by the AND of  $\bar{X}CWC+ \cdot \bar{R}MWC- \cdot \bar{M}BSY-$ .

## PARITY GENERATION

Parity generation, in conjunction with parity checking provides a means of detecting if a stored data word has

gained or lost a bit. The parity generation logic ensures that an odd number of "one" bits are stored in memory during any write operation. The parity checking logic ensures that an odd number of "one" bits are read back for each data word. If an odd number of "one" bits are not read back, a parity error signal is provided to light the Parity Error indicator on the computer console, enable the running program to detect the parity error (JNP), and to optional inhibit further access to memory by the Arithmetic Unit (Stop on Parity switch).

As shown in the AU Write Block Diagram of Fig. SYS.7 and the User 1, 2, 3 Write Block Diagram of Fig. SYS.12, data bits from the multiplexer are applied to the parity generator. The parity generator then enables a "one" to bit 25 of the ICM-40 Data Register if an even number of one bits are contained by the 24 data bits.

Fig. SYS.16 illustrates in block diagram form the logic used in generating a parity bit. This diagram is made from sheets 17 thru 21.1 of the Memory Control logic, GE Drawing 68C972375. As shown in Fig. SYS.16, most of this logic is also used in checking for parity errors during a read operation. Parity Checking is discussed immediately following this discussion of parity generation.

The data bits to be stored in memory are gated from the user device thru the multiplexer by D1MAE1, 2, 3, or 4 when memory access has been granted. These data bits are applied to the ICM-40 Data Register as well as the inputs of the parity generation logic (G0POG1 thru G0POG8).

G0POG1 thru G0POG8 each sample three data bits and if an odd number of "ones" are contained by these three data bits, a "zero" output signal is provided. If an even number of "one" data bits are contained by the three inputs, a "one" output is provided. The outputs of G0POG1 thru G0POG8 are applied to G0POS1, G1POS2, and G0POS3. The output from these gates are applied to G1POFS. The output from G1POFS is then a "one" if an odd number of data bits was detected from the multiplexer or a "zero" if an even number of "one" data bits was detected from the multiplexer.

The output of G1POFS is then ANDed with the Write Request signal (MWRQ) in G1MMPB. A "one" output from G1MMPB is provided only if a "zero" is provided from G1POFS indicating an even number of "one" data bits. This "one" output from G1MMPB is applied to Data Register bit 25 of the ICM-40 Module and stored in the addressed memory location along with the other data bits. In this manner, an odd number of bits are stored in the addressed memory location.

## PARITY CHECKING

During a read operation from memory, the data read from the addressed core cell is checked to determine if it contains an odd number of "one" data bits. Since the parity generation logic ensured that an odd number of "one" bits were stored in the memory location, if an even number of "ones" is read back, a bit has been

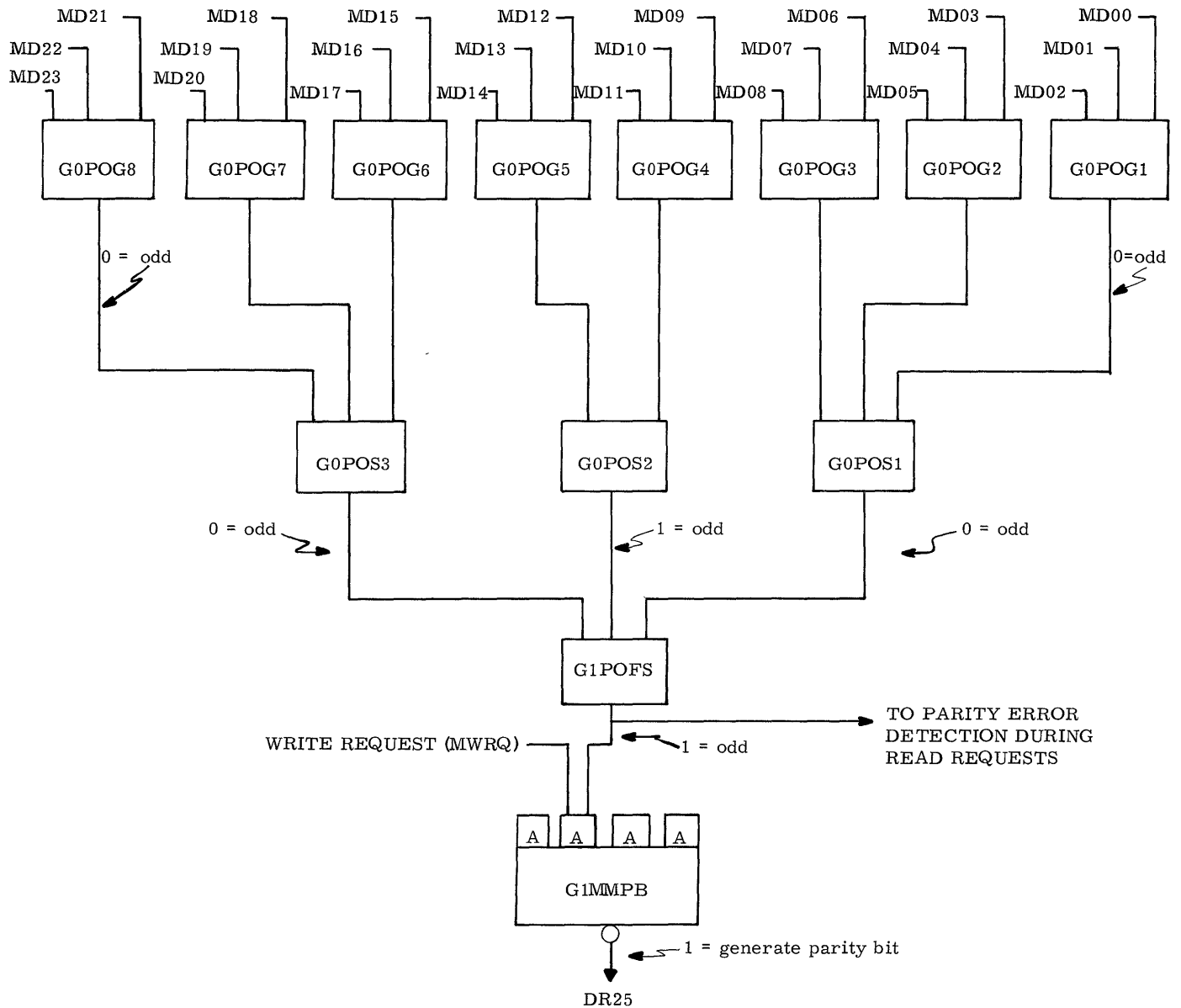


Fig. SYS. 16. Parity Generate

gained or lost and an error condition exists. When an error condition is detected, the Parity Error flip-flop (F1PPER) is set and the data word is restored in the memory location in the same form that it was read.

Setting the Parity Error flip-flop performs the following functions:

- Applies a signal to the Arithmetic Unit (AUGONPER) where the status may be checked by the JNP command.
- Applies a signal to the Stop On Parity switch (CSWSOP) to stop Arithmetic Unit sequencing when a parity error occurs and the Stop On Parity switch is in the Stop position.

- Sets F1PPED to light the Error indicator on the Computer Console.

The Parity Error flip-flop may be cleared by a JNP command, the Clear Alarm switch on the Computer Console, or by Initializing the system. The only method of clearing F1PPED is by pressing the Clear Alarm switch on the Computer Console. Therefore, if a parity error occurs, the Error Indicator on the Computer Console will remain lighted until the Clear Alarm switch is pressed.

As shown in the AU Read Block Diagram of Fig. SYS. 6 and the User 1, 2, 3, Read Block Diagram of Fig. SYS. 11, the data transferred to the user device is gated back through the Multiplexer to the parity check logic

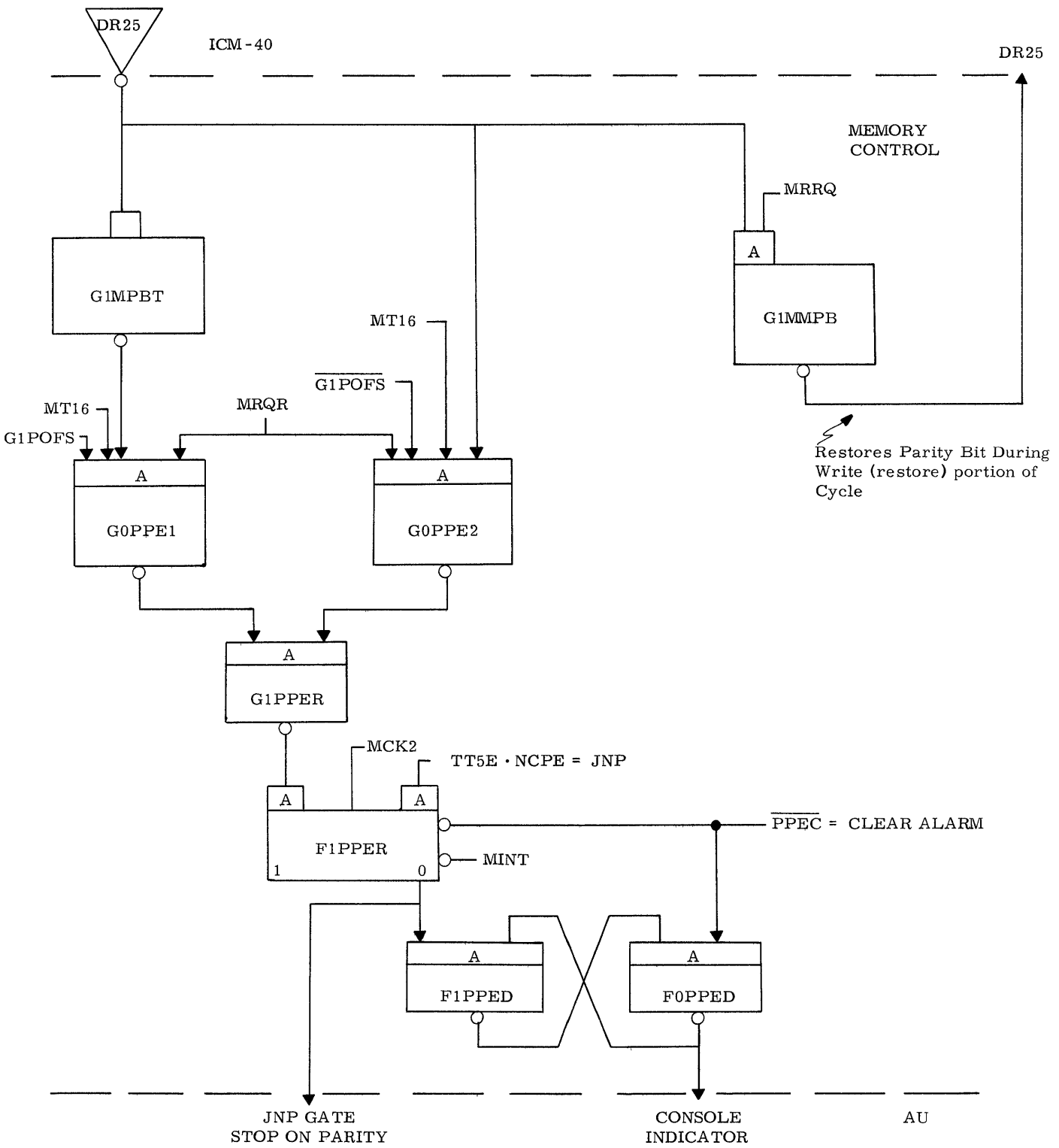


Fig. SYS. 17. Parity Check

along with the parity bit (DR25) read from memory. The data read is applied back through the multiplexer by the MBE1, 2, 3, or 4 signal being true throughout the memory cycle.

Fig. SYS. 17 illustrates the logic associated with parity checking. This logic compares the status of the parity bit read back from memory with the status of G1POFS. As described in the parity generation portion of this discussion, the output of G1POFS is a "one" when the 24 data bits contain an odd number of "ones".

G1POFS is ANDed with the parity bit read back from memory in G0PPE1 and G0PPE2. G0PPE1 is enabled if an odd number of "one" bits was contained by the data word and the parity bit is a "one" indicating that an extra "one" bit was read back from memory. G0PPE2 is enabled if an even number of "one" bits was contained by the data word and a "zero" was read back for the parity bit, indicating that a "one" bit was lost.

$$G0PPE1 = POFS \cdot MPBT \cdot MT16 \cdot MRQR$$

$$G0PPE2 = \overline{POFS} \cdot \overline{DR25} \cdot MT16 \cdot MRQR$$

Enabling either G0PPE1 or G0PPE2 will cause the Parity Error flip-flop (F1PPER) to set at the next clock pulse. As previously described setting F1PPER applies a signal to the Arithmetic Unit for JNP checking and to the Stop On Parity switch to halt sequencing within the Arithmetic Unit. Setting F1PPER also sets F1PPED to light the Console Error indicator. As a system option, the Parity Error flip-flops may be used to light other indicators or enable other system test gates.

$$F1PPER = PPER \cdot MCK2$$

$$G1PPER = PPE1 + PPE2$$

The Parity Error flip-flop may be reset by executing a JNP command, pressing the Clear Alarm switch, or Initializing the system.

$$\overline{F1PPER} = (TT5E \cdot NCPE) + PPEC + MINT$$

F1PPED is cleared by pressing the Clear Alarm switch.

Clearing F1PPED extinguishes the Alarm indicator.

$$\overline{F1PPED} = PPEC$$

## TRANSMISSION PARITY ERROR CHECKING

If a "one" data bit is lost or gained while receiving data from the Model 4511 Peripheral Channel Multiplexer during a write operation, a parity error signal will be applied back to the 4511. The state of the parity bit from the 4511 is compared with the status of G1POFS in G0PERD. G1POFS, as previously described, provides a "one" output when the 24 data bits contain an odd number of "one" bits. G0PERD is enabled, signifying a transmission parity error, when an odd number of "one" bits is contained by the data word and the parity bit is a "one" or when an even number of "one" bits is contained by the data word and the parity bit is a "zero". The output of G0PERD is applied back to the 4511. Refer to the 4511 description for the effect of this parity error signal.

$$G0PERD = POFS \cdot MEDP \cdot \overline{MRRQ} + \overline{POFS} \cdot \overline{MRRQ} \cdot \text{Parity Bit}$$

## ICM-40 DC POWER

The MP-40 Power Supply (PR36) manufactured by Ault Incorporated 3501 48th Ave. North, Minneapolis, Minnesota 55429, is used to provide dc power to the ICM-40 Memory Module. A detailed description of this power supply is provided in a separate instruction manual provided with each system.

A power cable from the power supply connects +6 volts, -6 volts, +24 volts, -24 volts and ground to the memory module as listed in Table SYS. 3. Fig. SYS. 2 illustrates the power connections at the ICM-40.

The dc chassis grounds are isolated from each other. The 24 volt floating supply is referenced to ground by two resistors. The -24 volt and +24 volt outputs are approximately one-half the nameplate value when measured to ground (e.g., the +24 volt output is approximately +11 volts to ground and the -24 volt output is approximately -11 volts to ground when the 24 volt setting is 22 volts).

VOLTAGE	WIRE COLOR CODE	MEMORY TERMINALS	PIN NO.
-6v	White	D6906	32*
+6v	Red	D1006	34
24v+	Yellow	C2902, D6902	2*
24v-	Green	C2904, D6904	1*
dc gnd	Black	D1004, D6906	33
FLT	Violet	NOT USED	
RT	Red & Black	NOT USED	

\*Not used on all circuit cards.

Table SYS. 3. Memory DC Power

## MEMORY RETENTION

The magnetic core array does not require power to provide its static memory capability. A pulse of power is required to switch cores from one state to the other but the pulse is not necessary to hold cores in their respective states. All cores remain in the state to which they have been switched because of the retentivity of the core magnetic material. If power is removed or lost, the magnetic core array retains stored information indefinitely.

The MP-40 power supply turn-on sequencing is designed so that the drive voltage (24v) remains off until the +6v and -6v logic supplies are stabilized. Similarly, the 24v supply is turned off before the logic supplies. This ensures that no currents can flow during the power supply turn-on and turn-off transients, and change data stored in the core array.

### Memory Wrap Around

The Memory Address Multiplexer inputs are wired with only the address bits required to address the implemented core size. That is, if an 8K memory module is implemented, only address bits 0 thru 12 are wired to the address multiplexer, etc. Using this technique, when

the user device addresses a core cell above the implemented size, a corresponding location within the implemented memory will be addressed or zeros will be fetched from core.

The following chart illustrates the 4K memory block addressed and the corresponding 4K block affected for the various implemented memory sizes.

Addressed 4K Block	Implemented Memory			
	4	8	12	16
0-4K	0-4	0-4	0-4	0-4
4-8K	0-4	4-8	4-8	4-8
8-12K	0-4	0-4	8-12	8-12
12-16K	0-4	4-8	*	12-16
16-20K	0-4	0-4	0-4	0-4
20-24K	0-4	4-8	4-8	4-8
24-28K	0-4	0-4	8-12	8-12
28-32K	0-4	4-8	*	12-16

\*Zeros are read with no parity error. A write operation will go undetected unless Memory Protect is used.

# MAINTENANCE

This section contains preventive and corrective maintenance procedures for the 4015A Memory System.

## RECOMMENDED TEST EQUIPMENT

In addition to the test equipment recommended for the GE-PAC\* 4020 System, Table MAINT.1 lists the test equipment that is unique to the 4015A Memory.

Description	Type
Worst Case Core Test	68A976868
Card Extender PAC	3C XP - 330
$\mu$ -PAC Extractor Tool	3C B008428
Resistor	4.7K $\pm 1\%$ , 1/2w

Table MAINT. 1. Recommended Test Equipment

## RECOMMENDED SPARE PARTS

Table MAINT. 2 and 3 lists the recommended spare parts for the ICM-40 Memory Module and the MP-40 Power Supply. Spare parts for the 4015A Memory Control Unit are included in the recommended spare parts for the GE-PAC 4020 System.

### ICM-40 CIRCUIT BOARD HANDLING AND REPAIR

The following paragraphs describe the handling and repair procedures for the ICM-40 circuit boards ( $\mu$ -PAC). Handling of the circuit cards in the 4015A Memory Control Module is described at the beginning of the System Maintenance Manual.

#### Inserting and Removing

The  $\mu$ -PAC connector is polarized to protect against incorrect PAC insertion.  $\mu$ -PAC removal from the memory is accomplished by engaging the two holes in the handle of the PAC with the  $\mu$ -PAC extractor tool. Do not remove or insert printed circuit cards without turning off the dc power to the unit.

#### Troubleshooting

The Extender PAC, Model XP-330, can be used to gain access to points on the  $\mu$ -PACs. Signals on the pins of the  $\mu$ -PACs may be ascertained from the PAC description or logic contained in Appendix A and B.

#### Component Checking

Many  $\mu$ -PACs have identical channels. Components can be checked by resistance comparison with parts on other channels or other  $\mu$ -PACs.

#### Component Replacement

When replacing defective components, use a small soldering iron and rosin core 60/40 solder. Remove excess solder from the printed circuit board. Care should be taken to avoid lifting the etch.

\* Registered Trademark of General Electric Company

Insert the leads of the new component through the drilled hole or eyelet, clip off excess wire, and solder to the printed circuit etch. A Flat Pack should be placed squarely on the etched area, using an insulator between it and the  $\mu$ -PAC. (An insulator should not be used on F-06 Flat Packs.) The leads should then be cut to the proper length and soldered. Examine the PAC carefully for excess solder. Remove rosin deposits with a commercial cleaning solvent and wipe the PAC clean with a dry lint-free cloth.

## OPERATIONAL CHECK

Operation of the 4015A Memory System is checked by running the Worst Case Core Test, General Electric 68A976868. Using this test, errors or failures are typed out to assist the Maintenance Engineer in isolating the probable cause.

## CALIBRATION PROCEDURES

### Drive Voltage Calibration

The drive line currents are determined by the setting of the 24-vdc supply and the precision resistors which are mounted on the resistor boards. The 24-v supply (MP-40) is temperature compensated by a thermistor which is mounted near the core stack. The 24-v supply calibration should be periodically checked according to the following procedure.

- a. Turn off the ac power to the MP-40 Power Supply. Disconnect the thermistor by removing the leads going to the power supply  $R_t$  terminals. Connect a fixed 4.7K  $\pm 1\%$ , 1/2w resistor to the  $R_t$  terminals of the MP-40 using a T and B RB-250 terminal (provided with the power supply) attached to each lead of the resistor.

#### NOTE

A detailed description of the MP-40 Power Supply is provided in a separate book supplied with each system. This book contains detailed theory of operation, adjustment procedures, and schematic diagrams for the MP-40 Power Supply.

- b. Turn on the ac power to the MP-40. Use a voltmeter with at least  $\pm 3\%$  accuracy to set the 24-v supply to the value shown on the core stack nameplate.

The nameplate value is the optimum voltage setting of the power supply at 25° C, using a 4.7K resistor instead of a thermistor to eliminate calibration temperature dependence. It was determined prior to shipment by setting the MP-40 to the voltage which gives the best operating margins, replacing the thermistor with a 4.7K resistor and recording the resultant voltage.



- c. Turn off the MP-40 Power Supply. Remove the 4.7K resistor and replace the thermistor leads. The voltage will change as a function of memory stack temperature (decreases 0.5% for a 1° C rise in temperature) but will be at the proper value for best memory operation. The 24-v supply can vary typically  $\pm 5\%$  without causing memory errors.

### Logic Voltage Calibration

The +6v and -6v MP-40 supplies are not temperature compensated, so they can be calibrated in the conventional manner. The +6v and -6v logic supplies may be monitored at the MP-40 terminals using an accurate voltmeter or the front panel meter, if provided (100% reading corresponds to +6v and -6v). The +6v and -6v supplies should be set to their nominal values (-6.0v and +6.0v). They can vary typically  $\pm 5\%$  from their nominal values without causing memory errors.

### Strobe Timing Calibration

The timing of the sense amplifier strobe pulse is set for each unit to give optimum operating margins. The core stack nameplate shows the time between the leading edge (1.5v point) of the RXSW+ and STRB+A signals. It should not be necessary to adjust the strobe timing. If a change in timing is required to obtain proper memory operation, the associated PACs should be checked (e.g., CM-006, CM-007, and PA-335) before a timing change is made. The CM-003 description in the appendix should be referred to if a timing change is required. Fig. SYS. 10 and SYS. 15 shows the delay line jumpers for a typical system.

## CORRECTIVE MAINTENANCE

Corrective maintenance procedures consist of electrical and mechanical inspection and memory system troubleshooting. Troubleshooting procedures for the MP-40/MP-40E power supply are covered in the power supply instruction manual.

### Corrective Maintenance Inspection

Before beginning troubleshooting procedures, a thorough inspection of the system should be performed. Check that the system is not physically damaged and that no wires have been torn accidentally from the equipment. Make sure electrical connectors and circuit cards fit firmly in their sockets.

### Troubleshooting Procedures

Malfunctions within the memory system are best isolated using the Worst Case Core Test (68A976868). Using the symptoms provided by the Worst Case Core Test, memory failures may generally be isolated into one of the following types.

- Operation failures which are caused by faulty timing and control circuits. Operation failures are indicated when there is no apparent response to commands applied to the memory, or when faulty operation occurs at all addresses. Refer

to Table MAINT. 4 for a list of the possible causes of operation failures.

- Partial data word failures which are caused by a faulty sense amplifier, data register flip-flop, data multiplexer logic elements, or data regeneration circuits. Partial data word failures are indicated when one or more bits at all addresses fail. Refer to Table MAINT. 5 for a list of the possible causes of partial data word failures.
- Address failures which are caused by address register, address multiplexer, or address selection circuits. Address failures are indicated by faulty memory operation at particular addresses only. Refer to Table MAINT. 6 for a list of the possible causes of address failures.
- Parity errors which are caused by faulty parity check and generate circuits. Parity error faults are indicated by parity errors when the data bits read back are the same as the data bits stored.

In many cases, spare circuit cards or identical circuit cards may be interchanged with a suspected faulty board to isolate or correct the problem. All circuit boards may be interchanged with a similar board except the Timing Distribution PAC, CM-003, which has jumper wires for delay line timing. Once a defective circuit card has been found, refer to the circuit card drawing. Many defective components may then be found by resistance comparison with an identical channel or an identical circuit card.

## MAGNETIC CORE STACK MAINTENANCE

Under normal operating conditions, it is unlikely that troubles will occur within the magnetic core stack. However, continuity measurements of the sense and drive windings will enable maintenance personnel to check core stack wiring. Exercise caution in taking these measurements to avoid damaging the matrix windings.

### Sense Windings

- a. Turn off memory power. Remove the Data Register PAC (CM-007, CM-107, or CM-207 depending on word capacity) associated with the sense windings to be checked.
- b. Place the ohmmeter leads across the sense winding inputs (SWXX+ and SWXX-) to the Data Register PAC as determined from the logic diagram of Appendix A. One sense winding links 4,096 cores. For 8,192- and 16,384- word memories, two and four sense windings, respectively, must be checked for continuity.
- c. Resistance readings should be approximately 10 ohms for all sense windings. The resistance readings for all bits should agree within  $\pm 10\%$ .

### Drive Windings

- a. Turn off memory power. Remove the CM-006 and CM-106 Selection Switch PACs associated with the X- and Y-drive line to be checked. This

can be determined from the logic diagrams by relating the bad address to a sink and switch output for both X- and Y-coordinates. The drive winding connections to the core stack are shown in Tables SYS. 1 and 2.

- b. Drive line resistance is a function of the core stack characteristics (number of bits, type and gauge of wire, and core spacing). The actual drive line connections are located on the printed circuit board of the core stack planes. The selection switch outputs are isolated by a diode from each drive line so that the resistance reading of any drive line will include a diode forward drop. The Y-drive line resistance is the same for all size memories (nominally 1.6 ohms) while the X-drive line resistance is a function of word length only (nominally 3 ohms for a 24-bit memory).
- c. Measure continuity by referring to the simplified selection diagram, Fig. SYS. 5. For example, to check the continuity of drive line Y1, put one ohmmeter probe on the corresponding sink output (emitter output of transistor Q3) and the other ohmmeter probe on the proper switch output (collector of transistor Q2). A low resistance (one forward diode drop plus a drive line resistance given in the preceding paragraph) indicates continuity for both diodes and the drive line. It may be necessary to reverse the probes to obtain the correct polarity to forward-bias the selection diodes. The continuity of the current path for the opposite drive polarity should be similarly checked by moving the probe from

the collector of Q2 to the collector of Q1 and reversing the polarity. A high resistance reading in both drive current polarity paths indicates an open drive winding or drive bus. If a drive bus is open, the other drive lines connected to the same bus will also have a high resistance reading. A high resistance reading in only one of the read or write current paths indicates an open F-08 Flat Pack diode.

## POWER DISTRIBUTION

### AC Power

The memory power furnishes ac power to both cooling units. Each cooling unit is individually fused by a 1/2 amp slow-blow fuse. A separate ac power cord is used for the MP-40 Power Supply. The MP-40 Power Supply instruction manual contains information on ac power requirements.

### DC Power

The memory dc power cable connects the memory to the MP-40 Power Supply. The dc power is distributed through the memory via laminated power buses, except for -6v which is wire wrapped to those PACs requiring this voltage. The dc and chassis grounds are isolated from each other. Fig. SYS. 2 shows the memory dc power terminals. The 24v floating supply is referenced to ground by two resistors. The 24v- and 24v+ outputs are approximately one-half the nameplate value when measured to ground (e. g., the 24v+ output is +11v to ground and 24v- is -11v to ground when the 24v setting is 22v).

COMPUTER CONTROL CORP. MEMORY ICM-40

GENERAL ELECTRIC 68A8659P2 (8K) and 68A8659P3 or 68A869P4 (16K)

PART NUMBER	DESCRIPTION	RECOMMENDED SPARES QUANTITY	
		8K	16K
CM-003	Timing Distributor PAC	1	1
CM-005	Address Register PAC	1	1
CM-006	Selection PAC	4	6
CM-106	Selection PAC	2	4
CM-007	Data Register PAC	2	2
CM-207	Data Register PAC	-	2
DI-335	NAND Type 1 PAC	1	1
DL-335	NAND Type 2 PAC	1	1
PA-335	Power Amplifier PAC	1	2
TG-335	Transfer Gate PAC	1	1
XD-335	Line Driver PAC	1	1
F-08	Flat Pack	4	4
930 011 146	Capacitor, M1CA, 510PF $\pm 5\%$ 100v	1	1
68A8519P14	Fuse, 1/2 AMP, Slow Blow	5	5
68A7002P00101	Resistor	1	1
932 215 009	Resistor, 53 ohm $\pm 1\%$ , 5w	5	5
130 217 018	Capacitor, 3.3 $\mu$ f $\pm 20\%$ , 35v	1	1
932 300 004	Thermistor, 4.7K at 25° C $\pm 20\%$ , 0.6w	1	1
68C998339P11	Fan	1	1

Table MAINT. 2. ICM-40 Recommended Spare Parts

AULT INCORPORATED MODELS PR36-A2 & PR36-B1  
 COMPUTER CONTROL CORP. MODELS MP40 & MP40E  
 GENERAL ELECTRIC #68A8659P5 & P6

PART NUMBER	DESCRIPTION	MANUFACTURER	MFG. PART NUMBER	RECOMMENDED SPARES QTY.
080-2409	Circuit Board Asm.	Ault Inc.	080-2409	1
080-2410	Circuit Board Asm.	Ault Inc.	080-2410	1
400-1243	Capacitor, 5000 mfd, 10 Volt DC			2
400-1244	Capacitor, 1200 mfd, 50 Volt DC			1
400-1269	Capacitor, 48,000 mfd, 20 Volt DC			1
400-1272	Capacitor, 22,000 mfd, 50 Volt DC	Sangamo	539-3764-402	2
400-1362	Capacitor, 38,000 mfd, 10 Volt DC	Sangamo	539-3765-02	1
406-1312	Capacitor, 5 mfd, 660 Volt AC	General Electric	49F4344	2
580-4123	Resistor, WW, 5 Watt 250 ohm, 5%			1
580-5359	Resistor, WW, 10 Watt 0.25 ohm, 10%			2
580-5360	Resistor, WW, 10 Watt 0.1 ohm, 10%			1
580-6546	Resistor, WW, 25 Watt 0.5 ohm, 5%			1
580-6601	Resistor, WW, 25 Watt 0.1 ohm, 10%			1
600-1029	Rectifier			1
601-1131	Transistor	RCA	2N3055	3
601-1161	Transistor	RCA	2N3054	1
603-1191	Rectifier	RCA	40209	1
603-1194	Rectifier	RCA	40109	1
603-1198	Rectifier	RCA	40208	2
603-1200	Rectifier	RCA	40211	2
606-1031	Rectifier Asm.	Motorola	SDA-10121	1
626-1022	Switch			1
626-1031	Switch, Toggle			1
12A6275P1	Lamp	General Electric	NE51	5
68A8519P10	Fuse	Bussman	MDL 1/4	5
876B216P1	Fuse	Bussman	AGC-5	5
876B216P4	Fuse	Bussman	AGC-15	5
876B216P5	Fuse	Bussman	AGC-20	5
68C998339P11	FAN	Rotron	Sentinel	1

Table MAINT. 3. Recommended Spare Parts-Power Supply

SYMPTOMS	PROBABLE FAULT
No apparent response to commands	<ol style="list-style-type: none"> <li>1. DC Voltage</li> <li>2. CM-003 PAC</li> <li>3. MBSY-, SRCY-, ERCY+, SWCY-signals in ICM-40</li> <li>4. Priority Access circuits in Control Unit.</li> <li>5. Memory Time Counter in Control Unit.</li> </ol>
Unable to read from any address	<ol style="list-style-type: none"> <li>1. Read and write CM-003 PACs interchanged</li> <li>2. DABL-signal at GND</li> <li>3. 24-v supply</li> <li>4. STRB+ signal</li> <li>5. G0MR0W</li> <li>6. AUD1BMEM</li> </ol>

Table MAINT. 4. Operational Failures

SYMPTOMS	PROBABLE FAULT
Failure of one bit (ZERO or ONE) at all addresses	<ol style="list-style-type: none"> <li>1. Data register PAC</li> <li>2. Y-switch or sink PAC</li> <li>3. TG-335 circuit</li> <li>4. Sense windings</li> <li>5. Data Multiplexer gate</li> </ol>
Failure of one bit at particular addresses	<ol style="list-style-type: none"> <li>1. Data register PAC</li> <li>2. Y-switch or sink PAC</li> <li>3. Sense winding</li> <li>4. Y-drive line</li> <li>5. Y-selection diode</li> </ol>
Failure of one bit at one address	<ol style="list-style-type: none"> <li>1. Marginal data register PAC</li> <li>2. Marginal core</li> </ol>
Failure of four bits at particular addresses	<ol style="list-style-type: none"> <li>1. Y-switch PAC or back plane resistor</li> </ol>

Table MAINT. 5. Partial Word Failures

SYMPTOMS	PROBABLE FAULT
All bits fail as a function of particular address bits	<ol style="list-style-type: none"> <li>1. X-switch or sink PAC</li> <li>2. CM-003 PAC</li> <li>3. X-drive line</li> <li>4. CM-005 PAC</li> <li>5. X-selection diode</li> <li>6. Address Multiplexer gate</li> </ol>

Table MAINT. 6. Address, Decoding, and Selection Failures

## APPENDIX A

DESIGNATION	IDENTIFICATION
AR01+ to AR14+	Address register flip-flop outputs
CBSY-	Signal forming the center of MBSY+
DABL-	Selection disable signal
DRDY-A	Data ready output signal
DR01+A to DR25+A	Data register output signals
EDCY-	End cycle (NOT USED)
ENSK+	Selection sink enable signal
ENSW+	Selection switch enable signal
ERAD+	Enable read address decoding signal
ERCY+	End read cycle signal
EWAD+	Enable write address decoding signal
LDDR+	Load data register pulse
MA01+ to MA14+	Memory address input signals
MBSY+	Memory busy signal
MD01+ to MD28+	Memory data input signals
RCYF+	Read cycle flip-flop signal
RDRP+	Reset data register pulse
RMWC+	Read-modify-write control
RRCF+	Read-regenerate control flip-flop signal
RSKA+	Read sink activate signal
RT01	Thermistor
RXSW+	Read X-switch signal
RYSW+	Read Y-switch signal
SKRW+	Sink read-write control
SKWR+	Sink write-read control
SRCY-	Start read cycle pulse
STRB+	Sense amplifier strobe pulse
SWCY-	Start write cycle pulse
SWRW+	Switch read-write control
SWWR+	Switch write-read control
SW01+A to SW28+A	Sense windings, A half of core stack
SW01+B to SW28+B	Sense windings, B half of core stack
WSKA+	Write sink activate signal
WXSW+	Write X-switch signal
WYSW+	Write Y-switch signal
WD01+ to WD28+	Write data signals
XCWC-	Clear-write command
XRRC-	Read-regenerate command
XD01 to XD32	X-selection switch outputs
XB01 to XB16	X-selection sink outputs
01YD1 to 01YD8	Y-selection switch outputs, bit 1
01YB1 to 01YB4	Y-selection sink outputs, bit 1

Table. 1. ICM-40 Mnemonic Signal List

Pin No.	CONNECTOR			
	E6A	E6B	E6C	E6D
1	MD01+	DR01+A	MA01+	
2	GND	GND	GND	GND
3	MD02+	DR02+A	MA02+	
4	GND	GND	GND	GND
5	MD03+	DR03+A	MA03+	
6	GND	GND	GND	GND
7	MD04+	DR04+A	MA04+	
8	GND	GND	GND	GND
9	MD05+	DR05+A	MA05+	
10	GND	GND	GND	GND
11	MD06+	DR06+A	MA06+	
12	GND	GND	GND	GND
13	MD07+	DR07+A	MA07+	
14	GND	GND	GND	GND
15	MD08+	DR08+A	MA08+	
16	GND	GND	GND	GND
17	MD09+	DR09+A	MA09+	
50	GND	GND	GND	GND
18	MD10+	DR10+A	MA10+	
19	GND	GND	GND	GND
20	MD11+	DR11+A	MA11+	
21	GND	GND	GND	GND
22	MD12+	DR12+A	MA12+	
23	GND	GND	GND	GND
24	MD13+	DR13+A	MA13+	
25	GND	GND	GND	GND
26	MD14+	DR14+A	MA14+	
27	GND	GND	GND	GND
28	MD15+	DR15+A	MD26+*	MBSY+A*
29	GND	GND	GND	GND
30	MD16+	DR16+A	MD27+*	SPARE
31	GND	GND	GND	SPARE
32	MD17+	DR17+A	MD28+*	
33	GND	GND	GND	GND
34	MD18+	DR18+A	PTZ1+**	DRDY-A*
35	GND	GND	GND	GND
36	MD19+	DR19+A	PTZ2+**	EDCY+A*
37	GND	GND	GND	GND
38	MD20+	DR20+A	PTZ3+**	DR26+A*
39	GND	GND	GND	GND
40	MD21+	DR21+A	PTZ4+**	DR27+A No
41	GND	GND	GND	GND
42	MD22+	DR22+A	SPARE	DR28+A
43	GND	GND	SPARE	GND
44	MD23+	DR23+A	XCWC-	DRRC-*
45	GND	GND	GND	GND
46	MD24+	DR24+A	RMWC+	
47	GND	GND	GND	GND
48	MD25+	DR25+A	XRRC-	
49	GND	GND	GND	GND

\* Not Used

\*\* Inputs must be grounded if option is provided but is not being used.

"+" = +6 volts true

"-" = 0 volts true

A = amplified signal

Table. 2. ICM-40 Input/Output Connector List

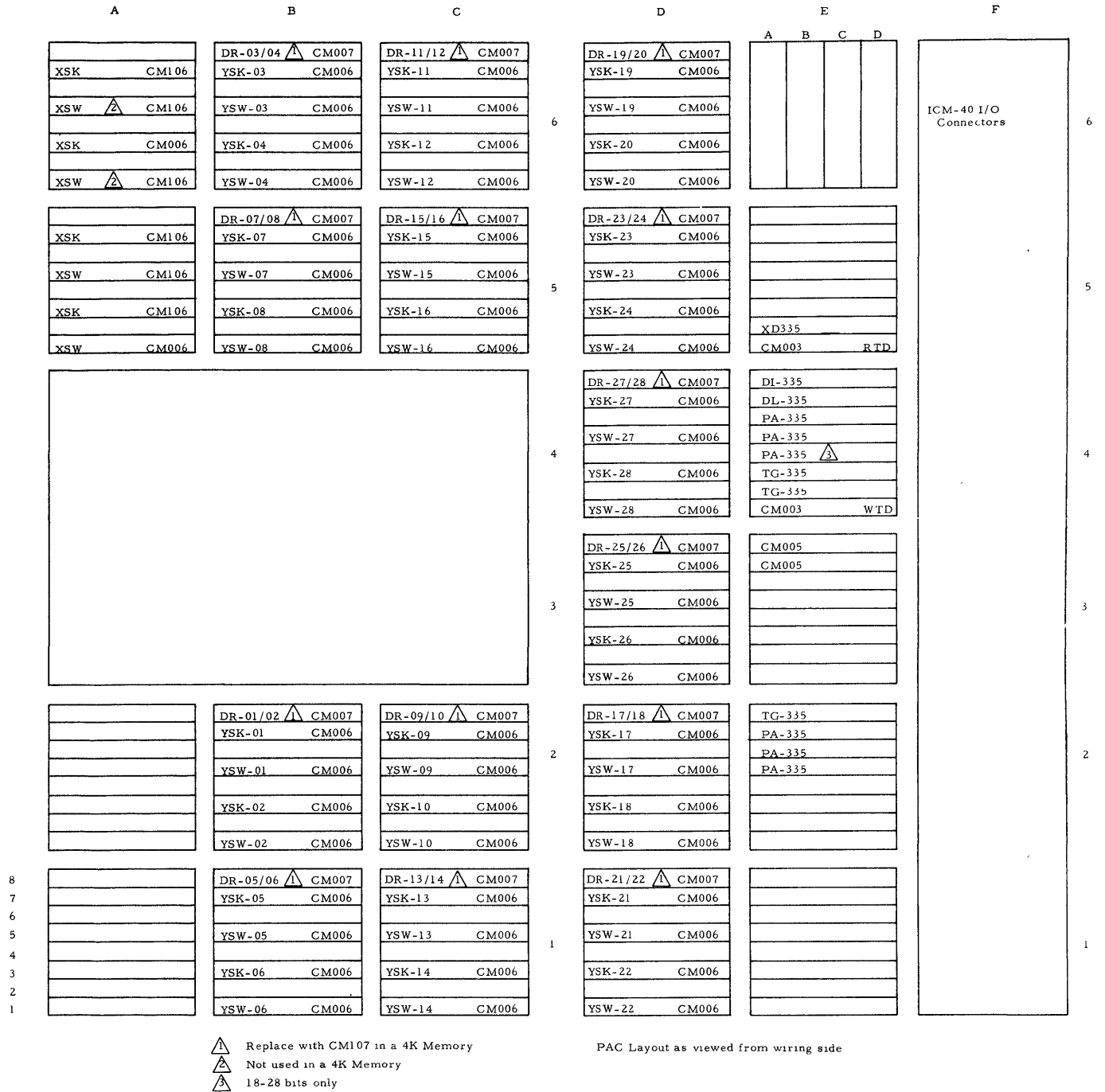


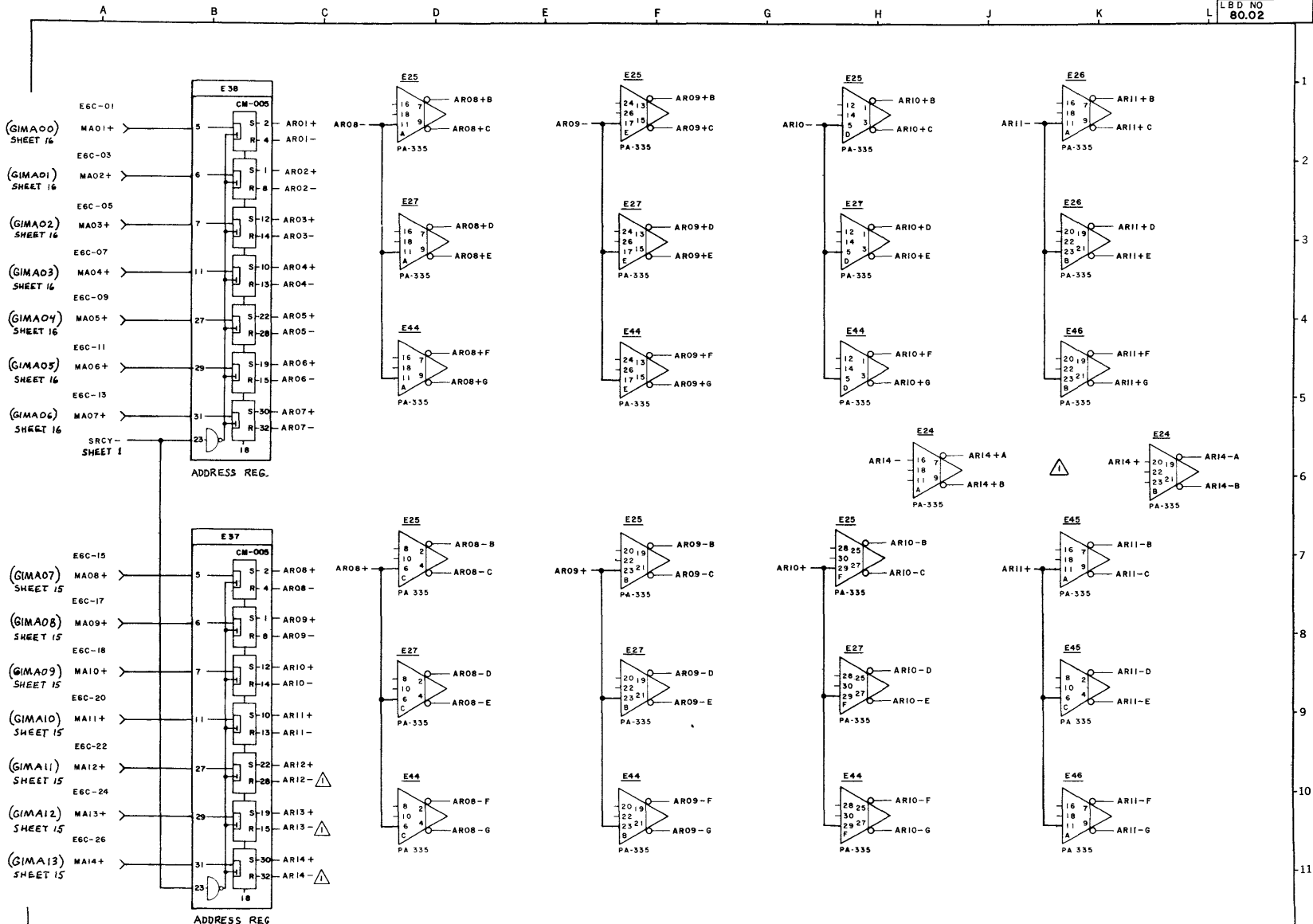
Table 3. Standard ICM-40, PAC Layout Diagram (Sheet 1, 4K/28, 8K/28)





Table 4. Standard ICM-40, PAC Layout Diagram (Sheet 2, 16K/14)





1/ 8K HAS AR01 THRU AR13, 16K HAS AR01 THRU AR14.  
 + = 6 VOLTS = TRUE  
 - = 0 VOLTS = FALSE

REV	1	ECO 3186
REV	2	REWORK WITH NO CHANGES
REV	3	DATE 10/65

		TITLE <b>ICM-40</b> ADDRESS REGISTER 4K, 8K AND 16K	
COMPUTER CONTROL COMPANY, INC. FRAMINGHAM MASS. LOS ANGELES 64 CALIF.		DR S DEAN DATE 10/65	SIZE C
ENG H BODIO DATE 10/65	APP D L ROTHENBERG DATE 12/65	DWG NO 009520	REV C
NEXT ASSY A012324	PROJECT NO 63228		

Fig. 2. ICM-40 Logic (Sheet 2)

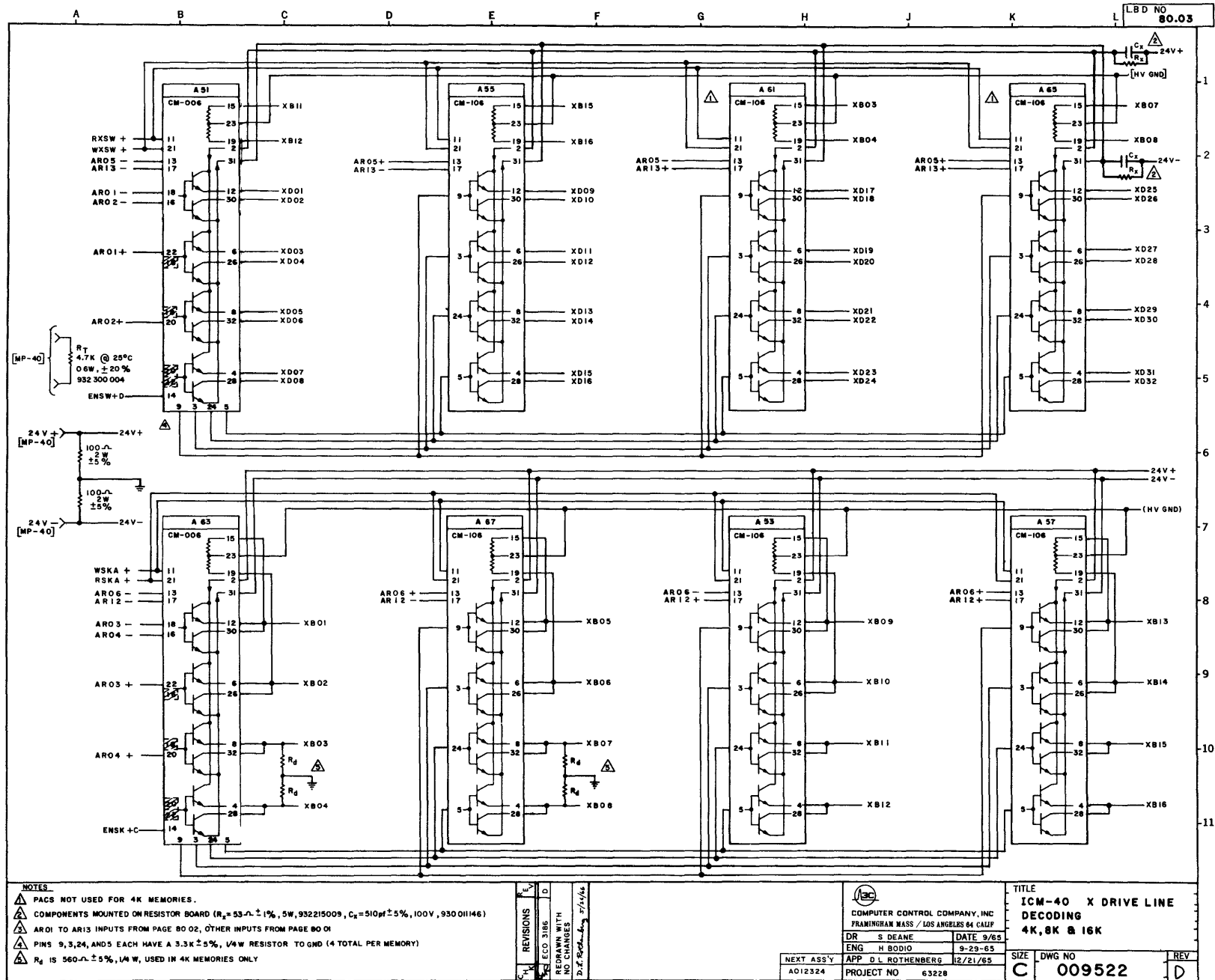


Fig. 3. ICM-40 Logic (Sheet 3)

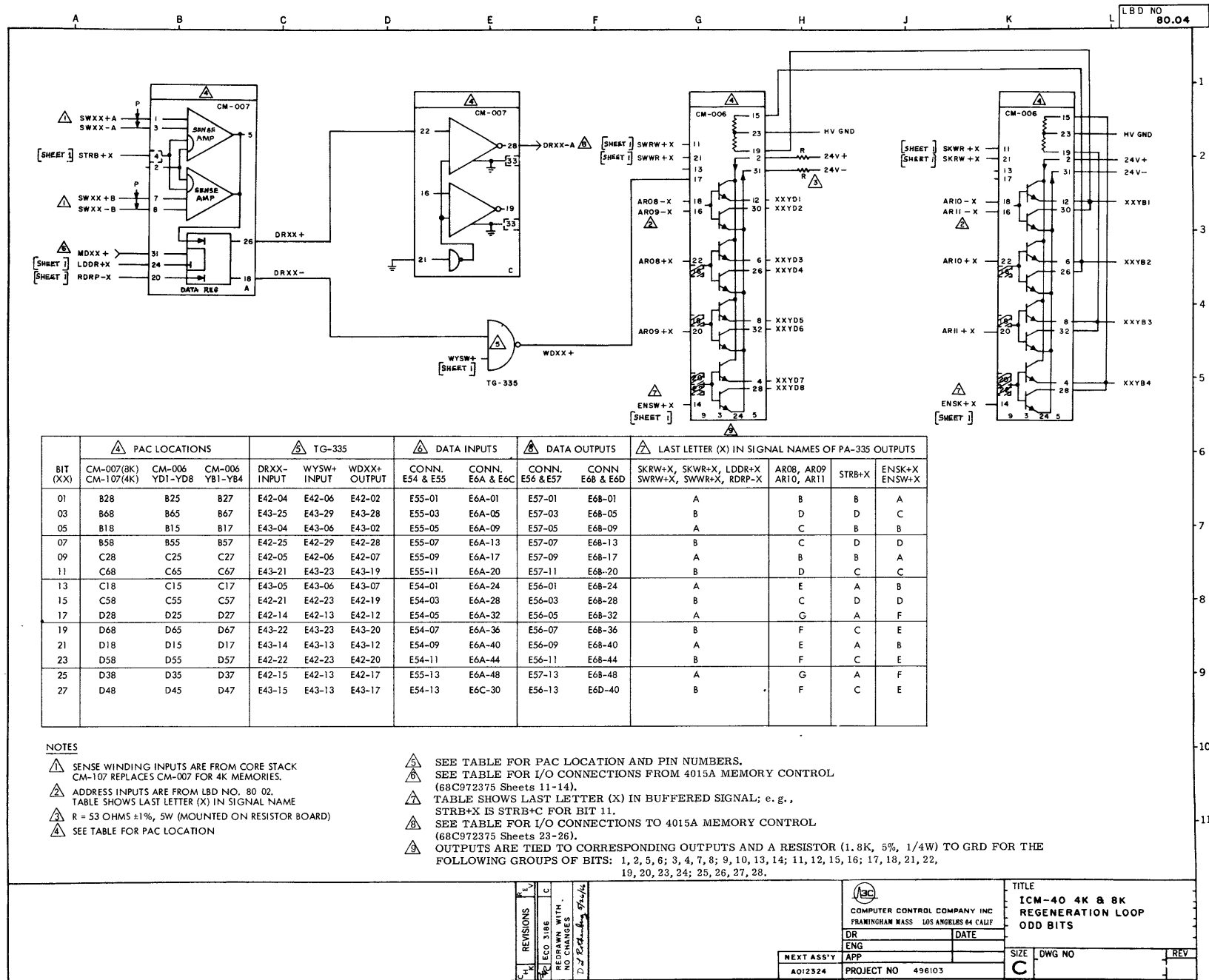


Fig. 4. ICM-40 Logic (Sheet 4)

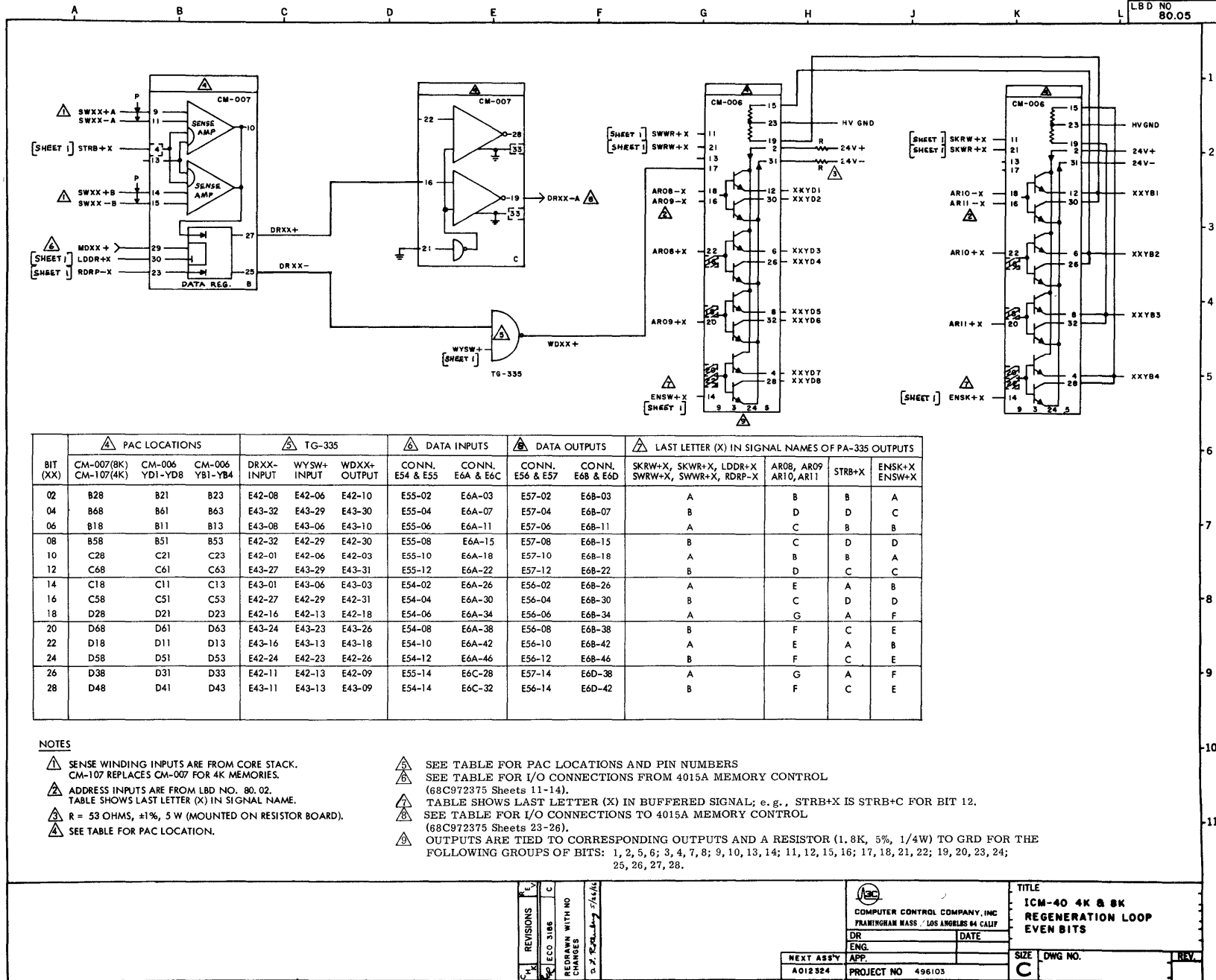


Fig. 5. ICM-40 Logic (Sheet 5)

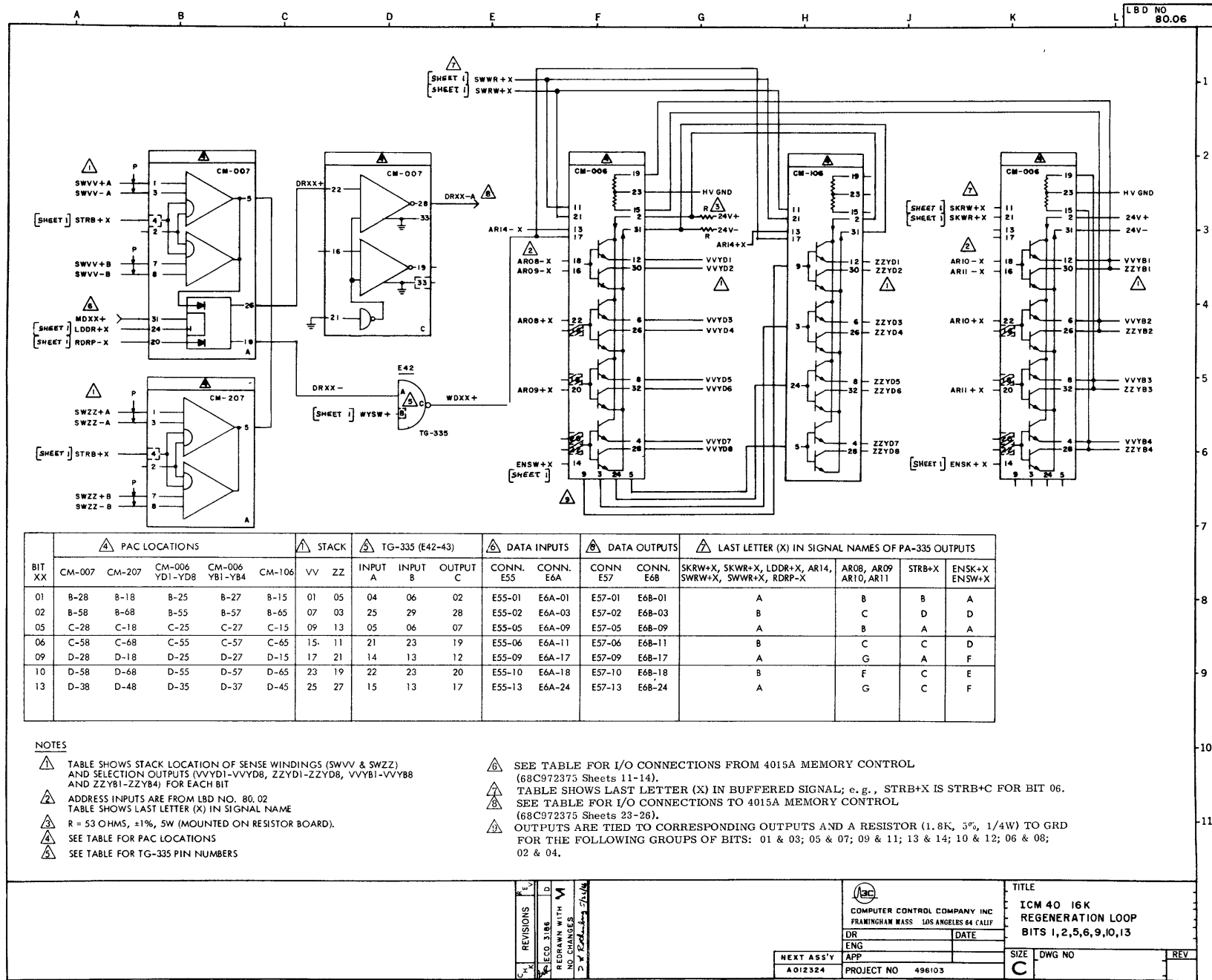


Fig. 6. ICM-40 Logic (Sheet 6)

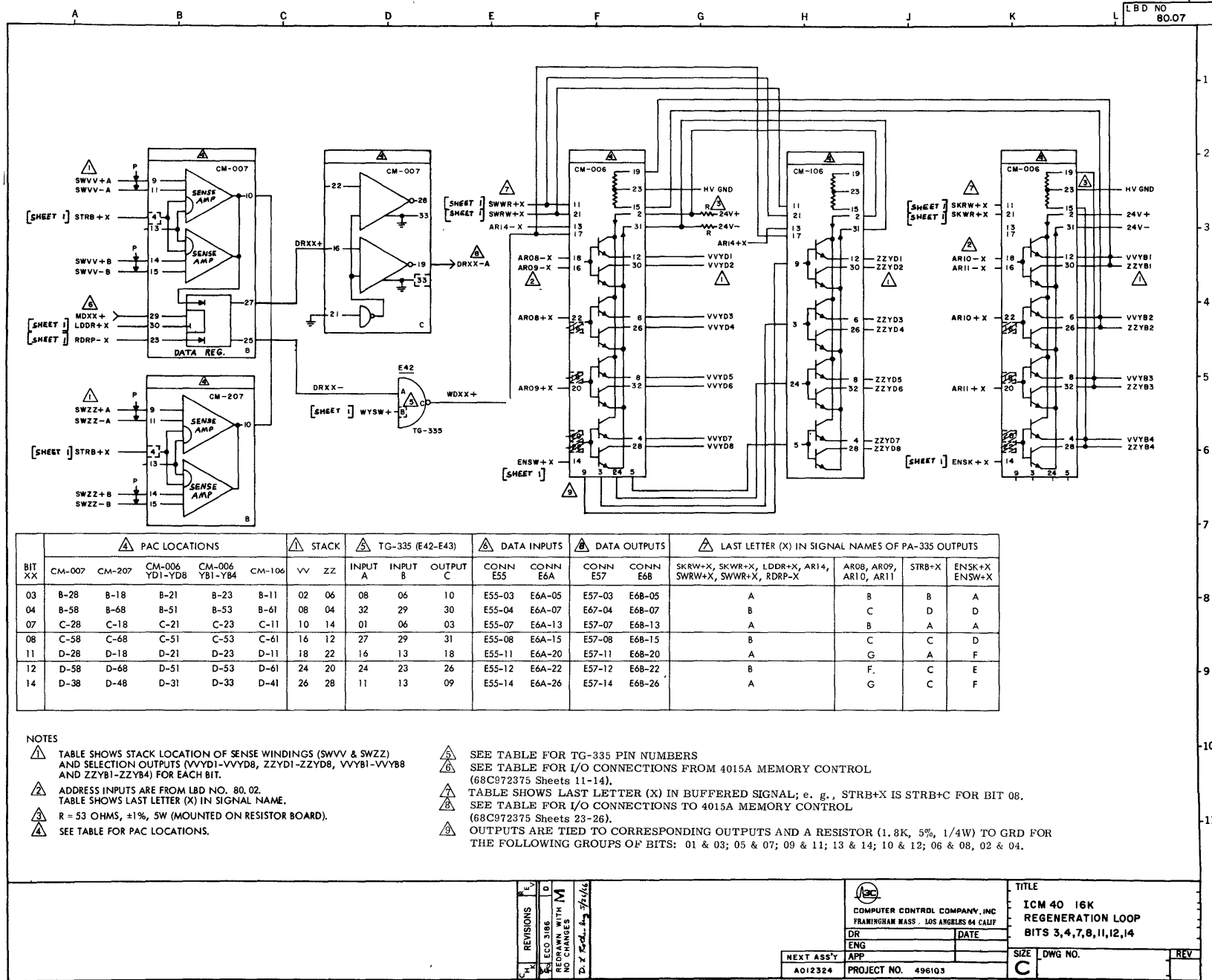


Fig. 7. ICM-40 Logic (Sheet 7)



# APPENDIX B

## TIMING DISTRIBUTOR PAC, MODEL CM-003

### GENERAL DESCRIPTION

The Timing Distributor PAC, Model CM-003 (Figures 1 and 2), provides accurately timed pulse sequences for use in timing and control applications. The CM-003 contains one control flip-flop, a 300-ns long delay line with 12 ns taps, a 50-ns long vernier delay line with 6 ns taps, and nine inverting power amplifier output circuits.

The PAC consists of two double-sided printed circuit boards sandwiched together for ease of mounting in a  $\mu$ -BLOC. Board A, which plugs into the connector, contains the four delay lines (DL1 through DL4) and five F-03 microcircuit power amplifiers. The delay lines are positioned between the two circuit boards to expose the etched side of board A for timing jumper adjustment.

Board B contains an F-04 microcircuit flip-flop, discrete drivers, and termination loads.

### NOTE

The CM-003 PAC occupies two slots in a taper-pin BLOC and three slots in a solderless-wrap BLOC, or the end slot (position 1) in either.

### CIRCUIT FUNCTION

Delay lines DL1 through DL3 can be tapped and jumpered to the output power amplifiers and the vernier delay line, DL4, to provide accurately timed output pulses. Input connection points for each amplifier are located on the PAC to facilitate timing flexibility. Refer to Table 1.

The dc reset of the flip-flop may also be tapped from any point along DL1 through DL3 to allow recirculation of the opposite driving edge, thereby establishing fixed pulse widths. An ac set, a dc reset, and the two outputs of the flip-flop are brought to the PAC connector.

Delay line DL4 and its associated output power amplifiers may be interconnected to provide pulses with a 6-ns delay resolution.

## SPECIFICATIONS

### Input Loading

Flip-flop dc reset: 2/3 unit  
load  
Flip-flop ac set: 1 unit load  
Power amplifiers: 2 unit  
loads each

### Circuit Delay

Flip-flop:  
Set input to set output or  
reset input to reset output  
65 ns (typ); 80 ns (max)  
Set input to reset output or  
reset input to set output  
45 ns (typ); 60 ns (max)  
Power amplifiers:  
24 ns (typ); 30 ns (max) each  
Delay to first tap (C1):  
60 ns (typ); 80 ns (max)

### Output Drive Capability

Flip-flop set: 8 unit loads  
Flip-flop reset: 4 unit loads  
Power amplifiers: 25 unit loads  
each

### Delay Line (DL1 through DL3)

Length:  
300 ns  $\pm 5\%$ , 24 taps, each  
12.5  $\pm 1$  ns  
Minimum pulse width:  
85 ns  
Maximum pulse width:  
330 ns

### Vernier Delay Line (DL4)

Length:  
50 ns  $\pm 5\%$ , 8 taps, each 6  $\pm 1$  ns

### Current Requirements

+6v: 175 ma  
-6v: 4 ma

### Power Dissipation

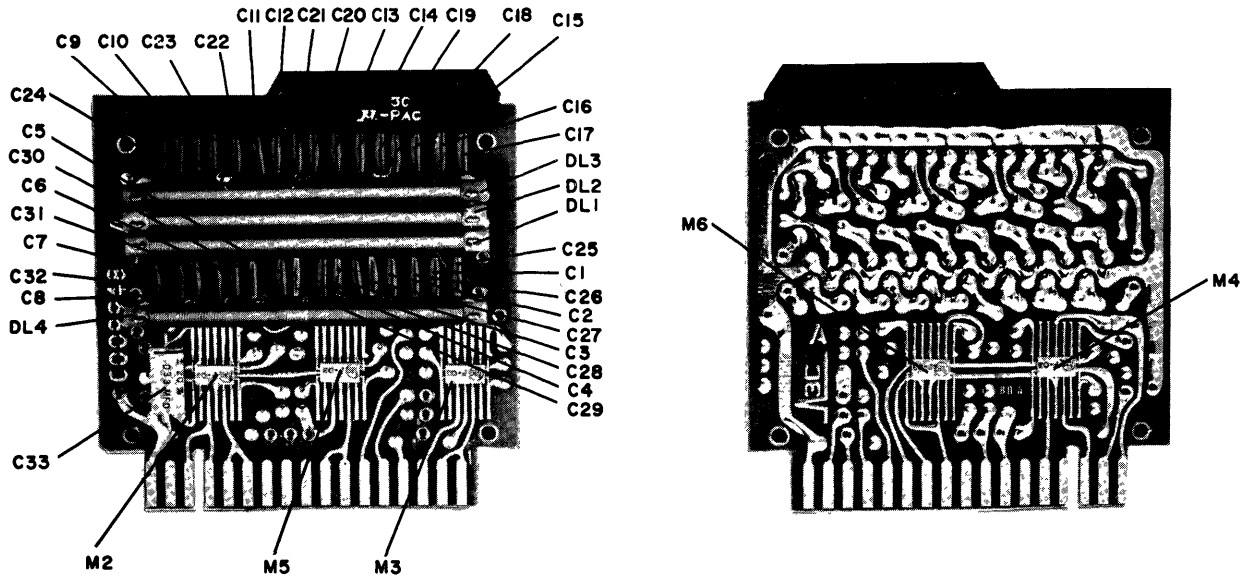
1.10w (max)

## NOTE

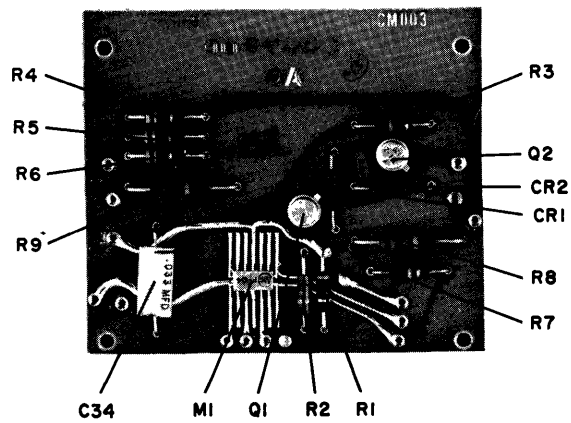
This document contains the information stated in  
Revision A of 3C Document No. B010797.

# Parts Location

## Board A



## Board B



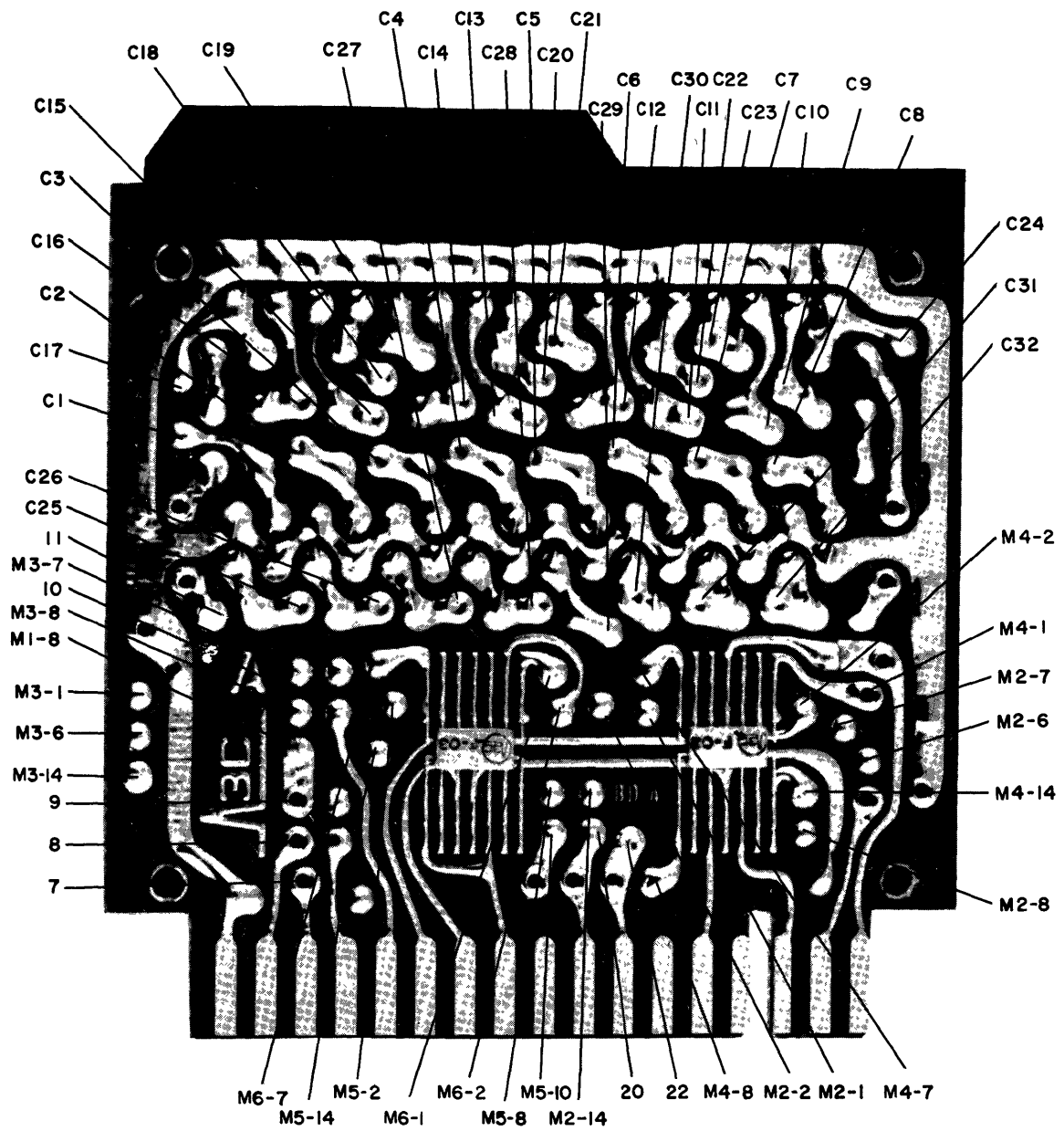
A3321

Figure 1. Timing Distributor PAC, Model CM-003, Parts Location and Identification (Sheet 1 of 2)

Electrical Parts List

Ref. Desig.	Description	3C Part No.
M1	MICROCIRCUIT: F-04, flip-flop integrated circuit	950 100 004
M2-M6	MICROCIRCUIT: F-03, power amplifier integrated circuit	950 100 003
C1-C24	CAPACITOR, FIXED, MICA DIELECTRIC: 120 pf $\pm 2\%$ , 100 vdc	930 004 219
C25-C32	CAPACITOR, FIXED, MICA DIELECTRIC: 29 pf $\pm 2\%$ , 100 vdc	930 004 204
C33, C34	CAPACITOR, FIXED, PLASTIC DIELECTRIC: 0.033 $\mu$ f $\pm 20\%$ , 50 vdc	930 313 016
CR1, CR2	DIODE:	943 088 001
DL1-DL4	COIL, DELAY LINE:	
Q1	TRANSISTOR: Replacement Type 2N3011	943 722 001
Q2	TRANSISTOR: Replacement Type 2N3012	943 721 001
R1	RESISTOR, FIXED, COMPOSITION: 820 ohms $\pm 5\%$ , 1/4w	932 007 047
R2	RESISTOR, FIXED, COMPOSITION: 100 ohms $\pm 5\%$ , 1/4w	932 007 025
R3	RESISTOR, FIXED, COMPOSITION: 1.8K $\pm 5\%$ , 1/4w	932 007 055
R4-R6	RESISTOR, FIXED, COMPOSITION: 330 ohms $\pm 5\%$ , 1/4w	932 007 037
R7	RESISTOR, FIXED, COMPOSITION: 51K $\pm 5\%$ , 1/4w	932 007 090
R8, R9	RESISTOR, FIXED, COMPOSITION: 180 ohms $\pm 5\%$ , 1/2w	932 004 031

Figure 1. Timing Distributor PAC, Model CM-003,  
Parts Location and Identification (Sheet 2 of 2)



A3320

Figure 2A. Timing Distributor PAC,  
Model CM-003, Test Points

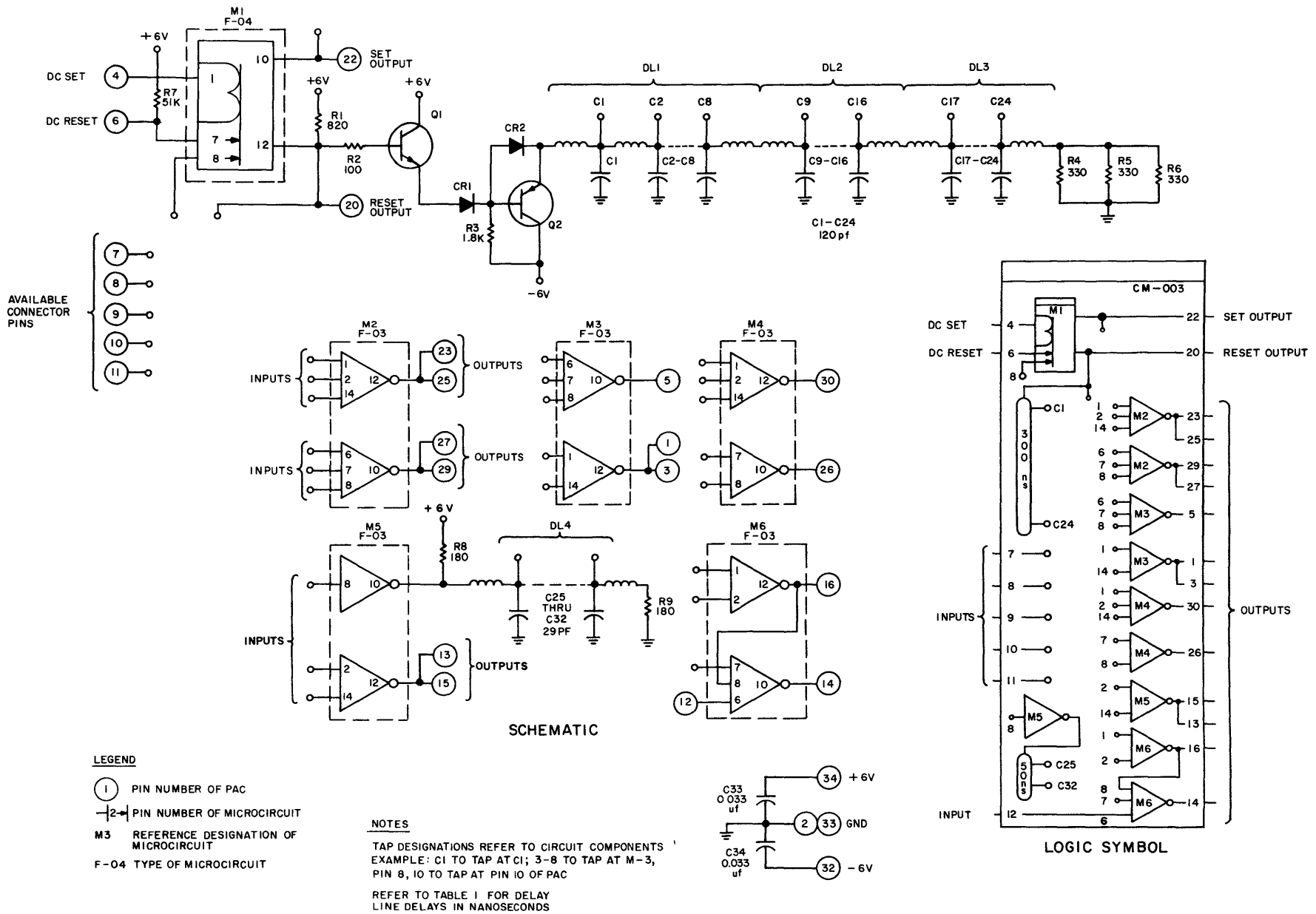


Fig. 2. Timing Distributor PAC, Model CM-003, Jumper Interconnection Diagram, Schematic Diagram, and Logic Symbol

Table 1.  
 Delay Line Tap Points with Corresponding Delay Line Delays  
 (Refer to Figure 2)

Delay Line Jumper Connection	Delay Line Delay (ns)	Delay Line Jumper Connection	Delay Line Delay (ns)
C1	12	C19	228
C2	24	C20	240
C3	36	C21	252
C4	48	C22	264
C5	60	C23	276
C6	72	C24	288
C7	84		
C8	96		
C9	108		
C10	120		
C11	132	C25	6
C12	144	C26	12
C13	156	C27	18
C14	168	C28	24
C15	180	C29	30
C16	192	C30	36
C17	204	C31	42
C18	216	C32	48

## ADDRESS REGISTER PAC, MODEL CM-005

### GENERAL DESCRIPTION

The Address Register PAC, Model CM-005 (Figures 1 and 2), utilizes nine F-02 quad NAND gate microcircuits prewired to form seven bistable circuits, each with a single-to-double rail conversion input. A common drop-in strobe input and a common reset input are also provided.

### CIRCUIT FUNCTION

The basic bistable circuit consists of two cross-coupled NAND gates. Each gate is connected to another NAND gate so that drop-in is accomplished by collector-pulling the cross-coupled pair. Inputs to the collector-pulling gates include the address and inverted address signals, and the common buffered strobe signal. The common reset of the cross-coupled pair does not involve collector pulling.

### SPECIFICATIONS

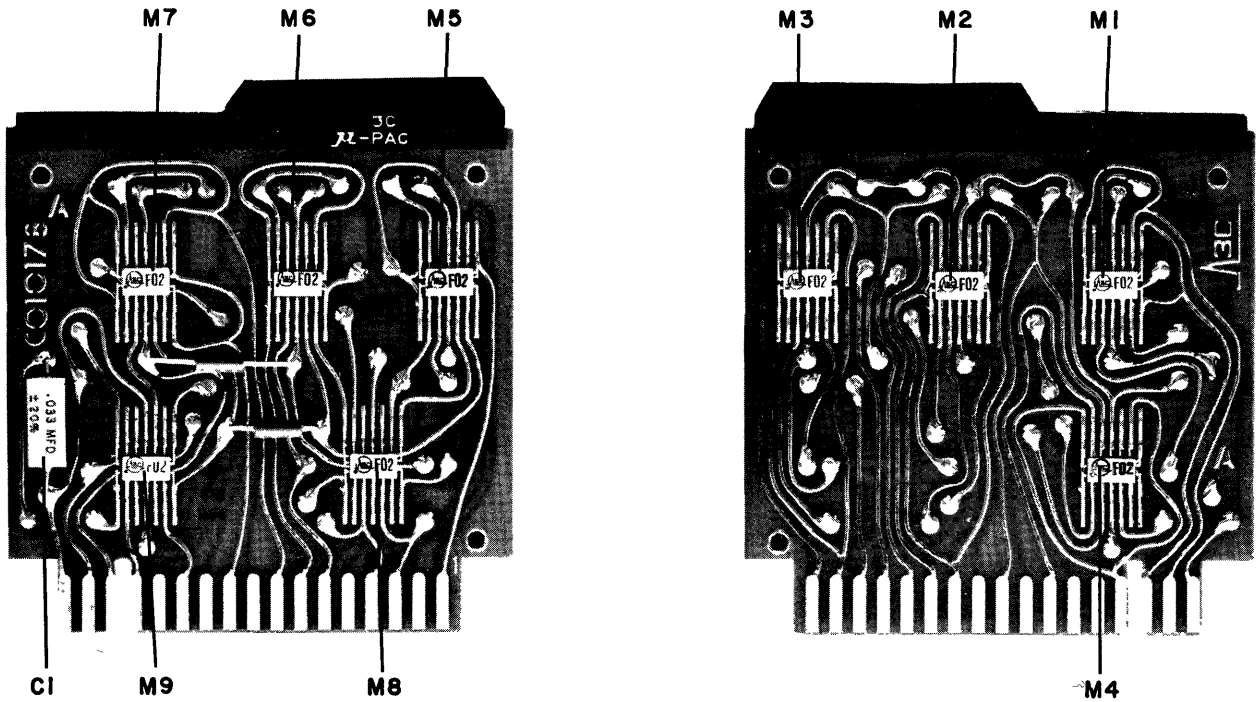
<u>Frequency of Operation</u>	<u>Output Drive Capability</u>
DC to 5 mc	3 unit loads and 30 pf stray capacitance
<u>Input Loading</u>	<u>Current Requirements</u>
DC inputs: 2 unit loads	+6v: 250 ma
Common strobe: 1 unit load	
Common reset: 7 unit loads	<u>Power Dissipation</u>
<u>Circuit Delay</u> (measured at 1.5v)	1.5w (max)
Reset input to set output: 60 ns (max)	
Strobe input to set or reset output: 85 ns (max)	

#### NOTE

This document contains the information stated in Revision A of 3C Document No. B011156.



## Parts Location



### Electrical Parts List

Ref. Desig.	Description	3C Part No.
M1-M9	MICROCIRCUIT: F-02, quad NAND gate integrated circuit	950 100 002
C1	CAPACITOR, FIXED, PLASTIC DIELECTRIC: 0.033 $\mu$ f $\pm$ 20%, 50 vdc	930 313 016

Figure 1. Address Register PAC, Model CM-005,  
Parts Location and Identification

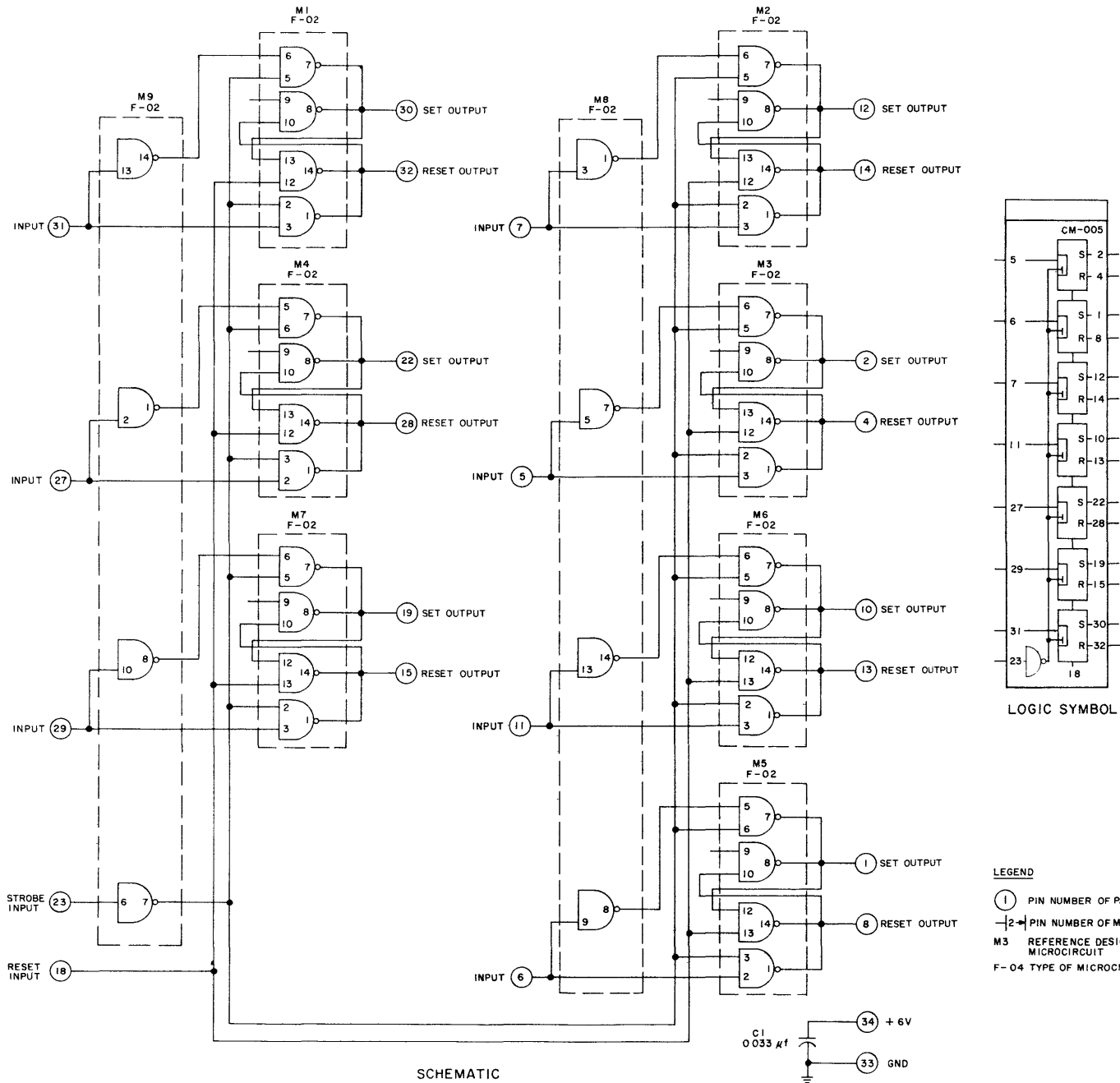


Fig. 2. Address Register PAC, Model CM-005, Schematic Diagram and Logic Symbol

## SELECTION PAC, MODEL CM-006/106

### GENERAL DESCRIPTION

The Selection PAC, Model CM-006/106 (Figures 1 and 2), contains eight transformer driven transistor switches connected as four (read-write) switch pairs. The CM-006 utilizes two F-03 microcircuit power amplifiers with decoding inputs for two address bits and a decoding enable input. An F-06 microcircuit multi-emitter gate provides timing (read-write) and matrix enable inputs. The PAC may be used as an 'X' or 'Y' selection switch or sink.

The CM-106 PAC does not contain the F-03 integrated circuits for address input decoding but is otherwise similar to the CM-006.

### CIRCUIT FUNCTION

Two binary address inputs are decoded to select one of the four F-03 integrated circuits. When the decoding enable input is a ONE (+6v), the output of the selected F-03 circuit will be at 0v. The multi-emitter circuit is turned on when the timing and matrix enable inputs are at +6v. Current flows through the primary of the transformer which is connected to the enabled F-03 and F-06 circuits. The output transistor is turned on by the secondary of the selected transformer.

The basic selection scheme of the CM-006 can be expanded by jumpering pins 3, 5, 9, and 24 to the corresponding pins on the CM-106. Decoding of the two address inputs will then be common to both PACs. Further expansion is possible by using the common F-03 and F-06 circuit inputs.

### SPECIFICATIONS

#### Input Loading

Timing (read-write) inputs:	1 unit load
Matrix enable inputs:	2 unit loads
Address inputs:	3 unit loads
Decoding enable inputs:	4 unit loads

## SPECIFICATIONS (Cont)

### Output Characteristics

Turn-on delay (1.5v of timing input to 10% of output current):	40 ns (max)
Turn-off delay (1.5v of decoding enable input to 90% of output current):	90 ns (max)
Rise time (10%-90%):	80 ns (max)
Fall time (90%-10%):	80 ns (max)
Current:	425 ma (max)
Voltage:	30v (max)
Pulse width:	400 ns (max)
Duty cycle:	45% (max)

### Current Requirements

CM-006:	+6v	100 ma (max)
CM-106:	+6v	65 ma (max)

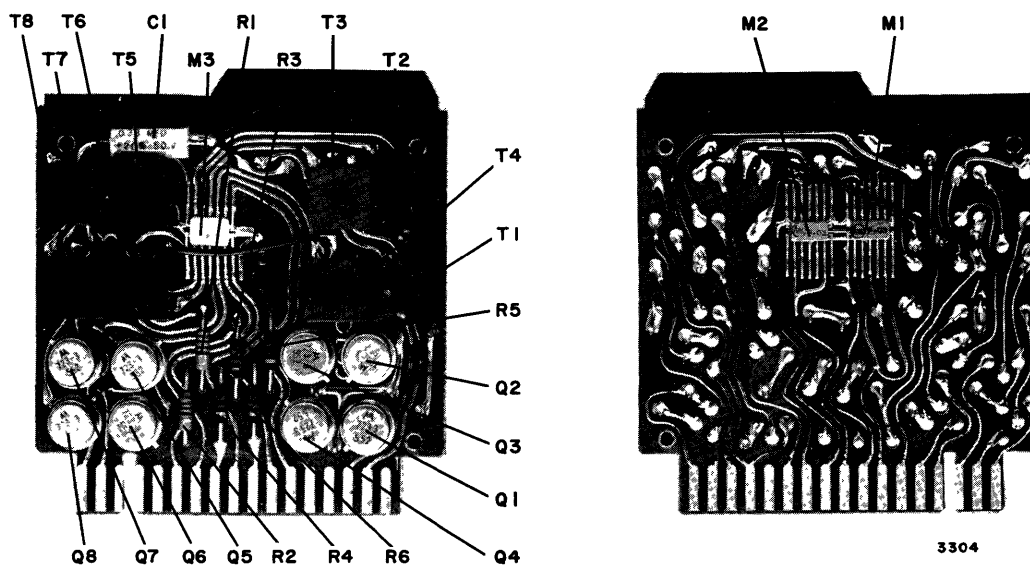
### Power Dissipation (400 ma output current)

CM-006:	1.5w (max)
CM-106:	1.3w (max)

### NOTE

This document contains the information stated in Revision B of 3C Document No. B011157.

## Parts Location

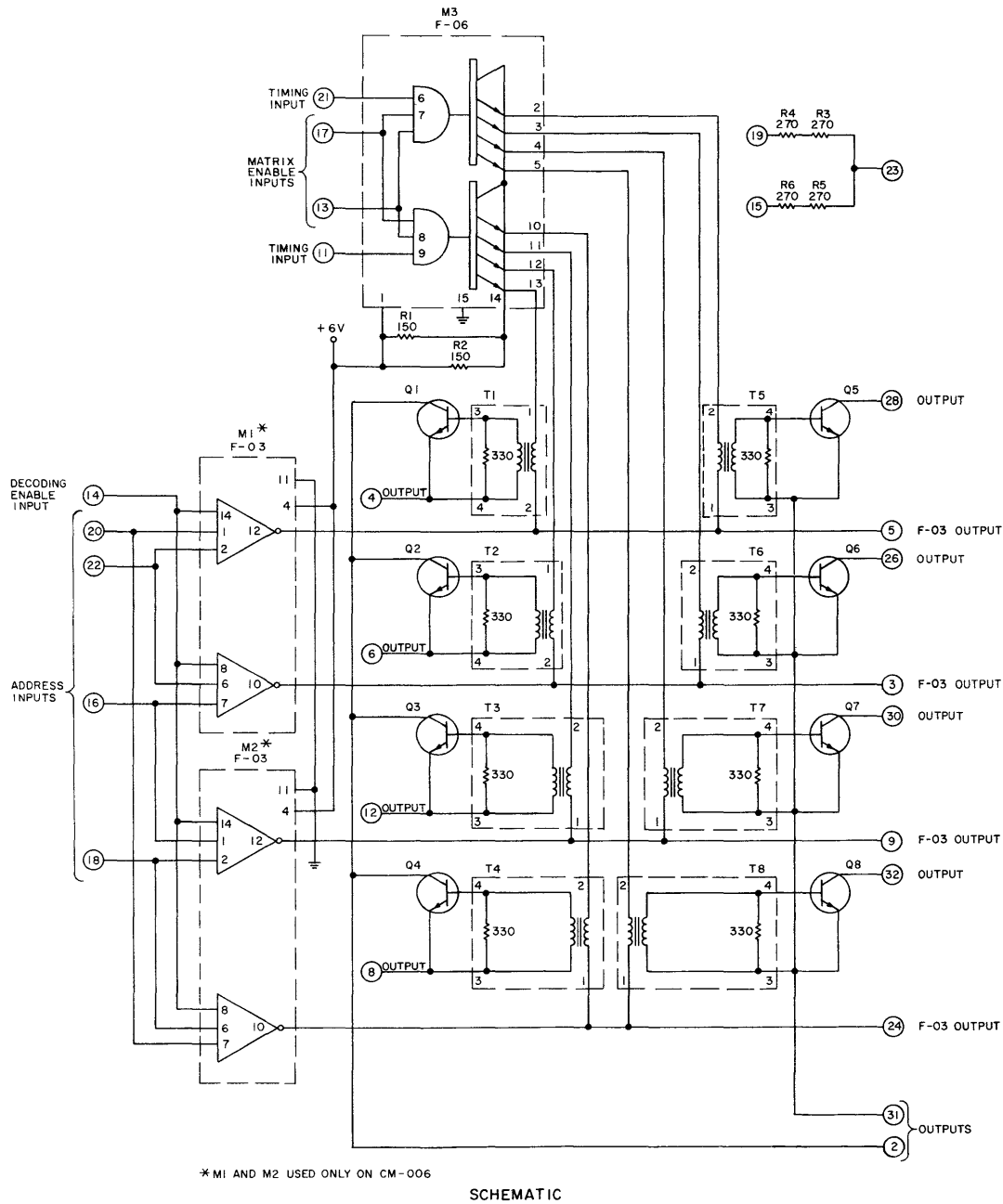


M1 AND M2 ARE USED ON CM-006 ONLY

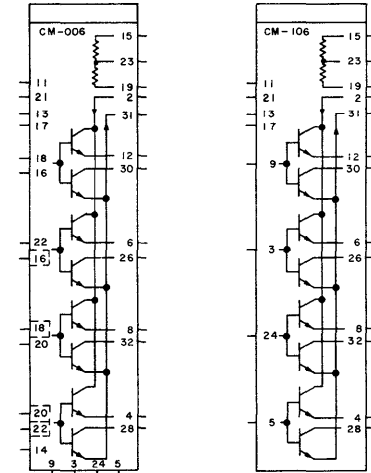
### Electrical Parts List

Ref. Desig.	Description	3C Part No.
M1, M2	MICROCIRCUIT: F-03, power amplifier integrated circuit	950 100 003
M3	MICROCIRCUIT: F-06, multi-emitter integrated circuit	950 100 006
C1	CAPACITOR, FIXED, PLASTIC DIELECTRIC: 0.033 $\mu$ f $\pm$ 20%, 50 vdc	930 313 016
Q1-Q8	TRANSISTOR:	943 723 003
R1, R2	RESISTOR, FIXED, FILM: 150 ohms $\pm$ 2%, 1/4w	932 114 029
R3-R6	RESISTOR, FIXED, COMPOSITION: 270 ohms $\pm$ 5%, 1/4w	932 007 035
T1-T8	TRANSFORMER, PULSE:	938 018 001

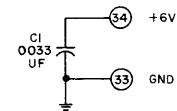
Figure 1. Selection PAC, Model CM-006/106, Parts Location and Identification



SCHEMATIC



LOGIC SYMBOL



LEGEND

- (1) PIN NUMBER OF PAC
- |2| PIN NUMBER OF MICROCIRCUIT
- M3 REFERENCE DESIGNATION OF MICROCIRCUIT
- F-04 TYPE OF MICROCIRCUIT

Fig. 2. Selection PAC, Model CM-006/106, Schematic Diagram and Logic Symbol

## DATA REGISTER PAC, MODEL CM-007/107/207

### GENERAL DESCRIPTION

The Data Register PAC, Model CM-007 (Figures 1 and 2), consists of two channels with associated sense amplifiers, an information register flip-flop, and a line driver. The sense amplifiers, monostable multivibrators, and strobe gates are contained in four F-07 microcircuits. The data registers are contained in three F-02 quad NAND gate microcircuits and the line driver is an F-03 microcircuit power amplifier.

The CM-107 is identical to the CM-007 with the exception that only one sense amplifier is associated with each data register. The CM-207 is similar to the CM-007 except that data registers and a line driver are not provided on the CM-207 PAC.

### CIRCUIT FUNCTION

A differential input signal of sufficient amplitude will trigger the monostable multivibrator if the strobe inputs are a ONE. The multivibrator provides a standard output width with variable input widths. The strobe inputs prevent noise present at times other than read-out time from being mistaken for information signals.

The negation outputs of up to four sense amplifiers can be connected together as an OR circuit to set the data register. Inputs to each flip-flop include data, inverted data, data drop-in and reset signals. One line driver input is normally externally connected to the data register output. An input is provided to disable both line drivers.

### SPECIFICATIONS

<u>Input Loading</u>	<u>CM-007</u>	<u>CM-107</u>	<u>CM-207</u>
Strobe input:	4 unit loads	2 unit loads	4 unit loads
Strobe enable input:	2 unit loads	1 unit load	2 unit loads
Reset input:	1 unit load	1 unit load	
Data input:	2 unit loads	2 unit loads	

SPECIFICATIONS (Cont)

<u>Input Loading</u>	<u>CM-007</u>	<u>CM-107</u>	<u>CM-207</u>
Data drop-in input:	1-1/2 unit loads	1-1/2 unit loads	
Line driver input:	2 unit loads	2 unit loads	
Line driver disable input:	1 unit load	1 unit load	

Sense Input

Input impedance, common mode:	150 ohms
Input impedance, differential mode:	250 ohms
Minimum ONE signal:	45 mv
Maximum ZERO signal:	30 mv
Maximum common mode voltage:	±1.5v

Monostable Output

Pulse width (1.5v): 150 ns (typ)

Circuit Delay (measured at 1.5v)

Sense input to monostable output:	100 ns (max)
Strobe input to monostable output:	80 ns (max)
Set input to set output:	30 ns (max)
Reset input to set output:	60 ns (max)
Data drop-in input to flip-flop output:	60 ns (max)
Line driver input to output:	30 ns (max)

Output Drive Capability

Flip-flop outputs: 3 unit loads  
 Line driver outputs: 93-ohm coaxial or twisted-pair cables up to 10 ft in length terminated with 1 unit load

<u>Current Requirements</u>	<u>CM-007</u>	<u>CM-107</u>	<u>CM-207</u>
+6v:	275 ma	215 ma	120 ma
-6v:	80 ma	40 ma	80 ma

Power Dissipation

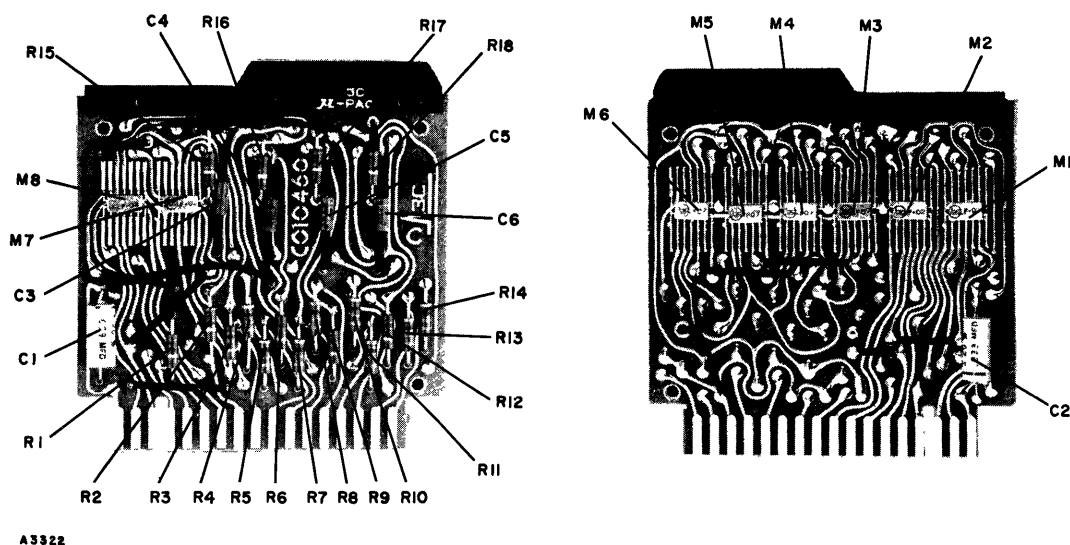
CM-007: 2w  
 CM-107: 1.4w  
 CM-207: 1.2w

NOTE

This document contains the information stated in Revision D of 3C Document No. B011158.



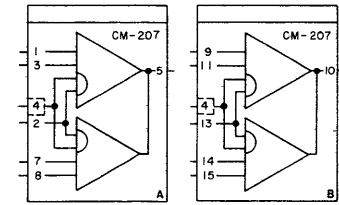
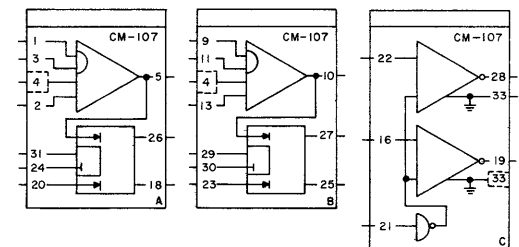
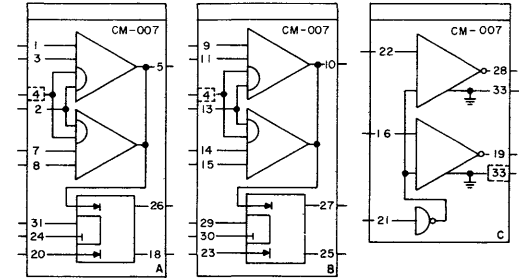
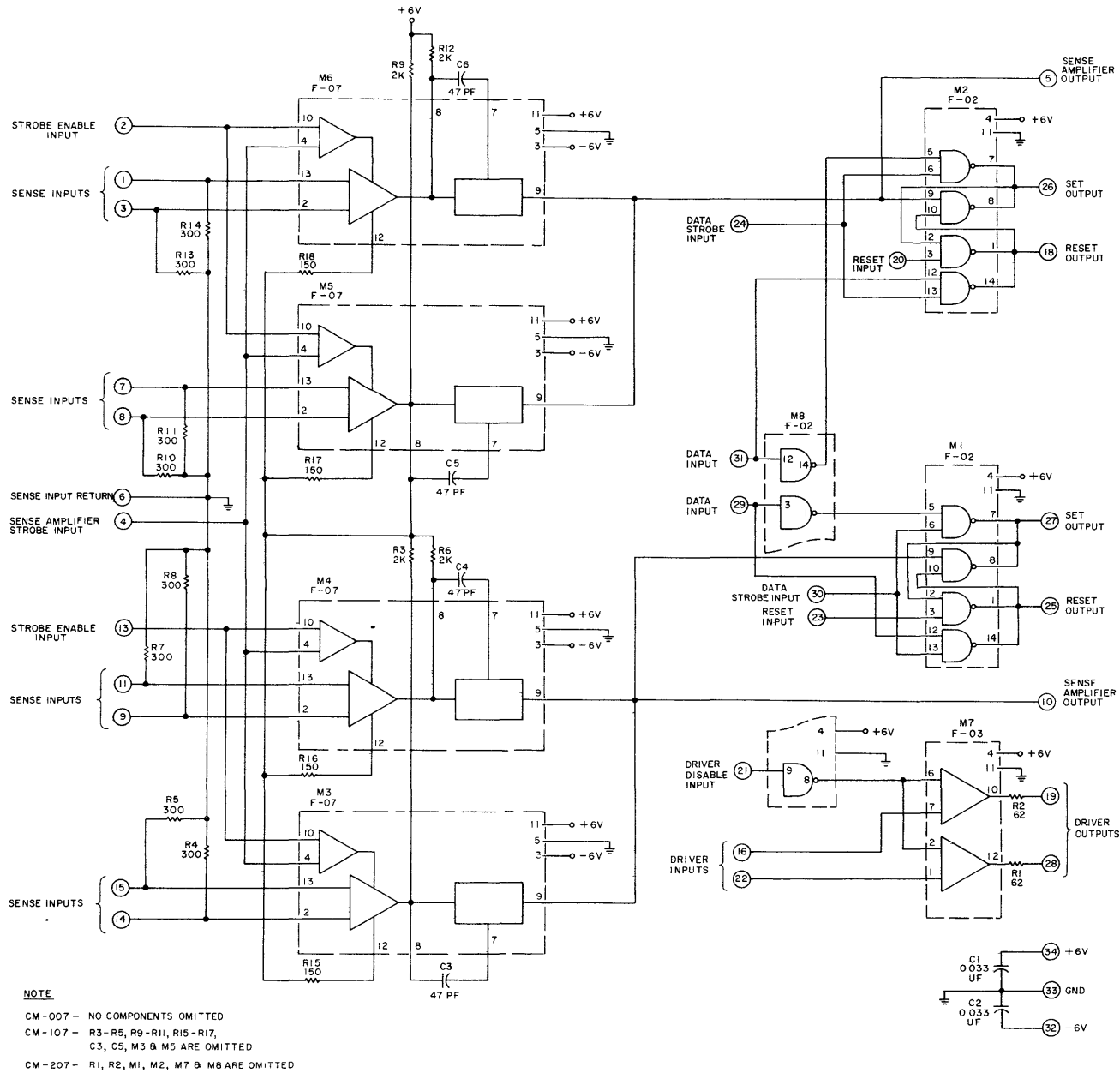
## Parts Location



### Electrical Parts List

Ref. Desig.	Description	3C Part No.
M1, M2	MICROCIRCUIT:	950 100 002
M8	F-02, quad NAND gate integrated circuit	
M3-M6	MICROCIRCUIT:	950 100 007
	F-07	
M7	MICROCIRCUIT:	950 100 003
	F-03, power amplifier integrated circuit	
C1, C2	CAPACITOR, FIXED, PLASTIC DIELECTRIC:	930 313 016
	0.033 $\mu$ f $\pm$ 20%, 50 vdc	
C3-C6	CAPACITOR, FIXED, CERAMIC DIELECTRIC:	930 007 020
	47 pf $\pm$ 5%, 100 vdc	
R1, R2	RESISTOR, FIXED, COMPOSITION:	932 007 020
	62 ohms $\pm$ 5%, 1/4w	
R3, R6, R9, R12	RESISTOR, FIXED, COMPOSITION:	932 007 056
	2K $\pm$ 5%, 1/4w	
R4, R5, R7, R8, R10, R11, R13, R14	RESISTOR, FIXED, COMPOSITION:	932 007 036
	300 ohms $\pm$ 5%, 1/4w	
R15-R18	RESISTOR, FIXED, COMPOSITION:	932 007 029
	150 ohms $\pm$ 5%, 1/4w	

Figure 1. Data Register PAC, Model CM-007/107/207,  
Parts Location and Identification



LOGIC SYMBOLS

Fig. 2. Data Register PAC, Model CM-007/107/207, Schematic Diagram and Logic Symbol

## NAND TYPE 1 PAC, MODEL DI-335

The NAND Type 1 PAC, Model DI-335, Figures 3-10.1 and 3-10.2), contains 10 independent 2-input NAND gates. Each gate performs the NAND function for positive logic (+6v = ONE, 0v = ZERO). For negative logic, it becomes a NOR gate.

Two of the 10 gates have separate load connections available at the PAC terminals. Outputs of these gates can be tied together, using a single load resistor, without loss of output drive capability.

A detailed description of the basic NAND circuit appears in Section II.

### INPUT AND OUTPUT SIGNALS

Inputs. -- When both inputs to a gate are +6v or not connected, the output is at ground. When any input is at ground, the output is +6v.

Load. -- This point is internally connected through a collector load resistor to +6v.

Collector Output. -- The collector output must be connected to at least one load resistor, either internal or external to the module.

Output. -- Each output terminal is internally connected to a collector load resistor. If an output is connected to load points or other outputs, the output drive capability of the structure is reduced.

### SPECIFICATIONS

<u>Frequency of Operation (System)</u>	<u>Circuit Delay</u>
DC to 5 mc	(Measured at +1.5v, averaged over two stages)
<u>Input Loading</u>	30 nsec (max)
1 unit load each	<u>Current Requirements</u>
<u>Fan-In</u>	+6v: 125 ma (max)
Refer to Section II.	<u>Power Dissipation</u>
<u>Output Drive Capability</u>	0.75 w (max)
8 unit loads each	<u>Handle Color Code</u>
<u>Outputs in Parallel</u>	Red
Refer to Section II.	

### APPLICATIONS

The NAND gates operate on levels, pulses, or combinations of both. Two gates can be wired back-to-back to form a dc set-reset flip-flop.

The two gates with separate load outputs form standard NAND gates when the load and collector output terminals are connected. When the collector outputs of gates are connected in parallel as in Figure 3-10.3, the AND-OR-INVERT function is performed. At the point where the outputs are tied together, an AND operation with logic ONES (OR operation with logic ZEROs) takes place.

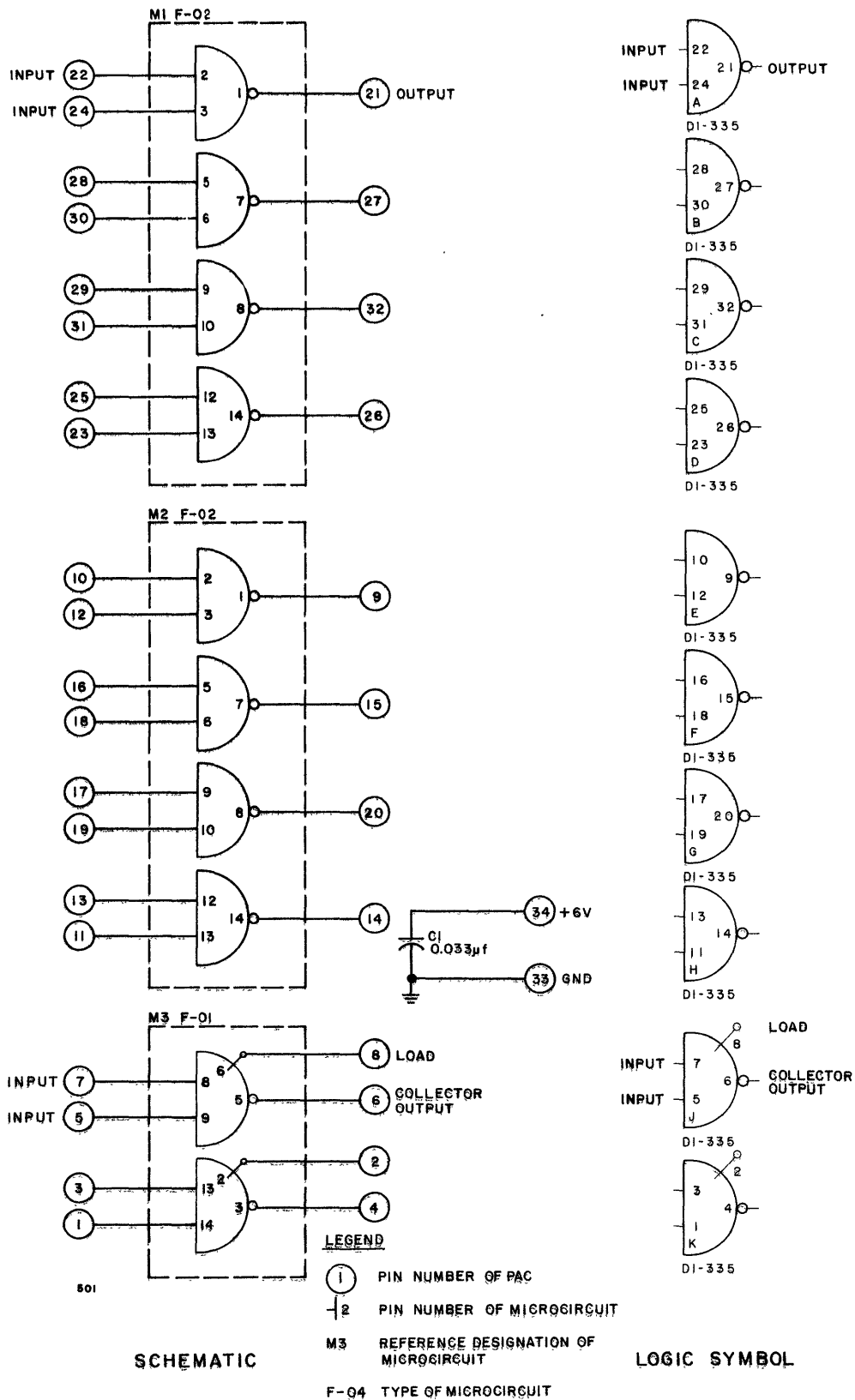
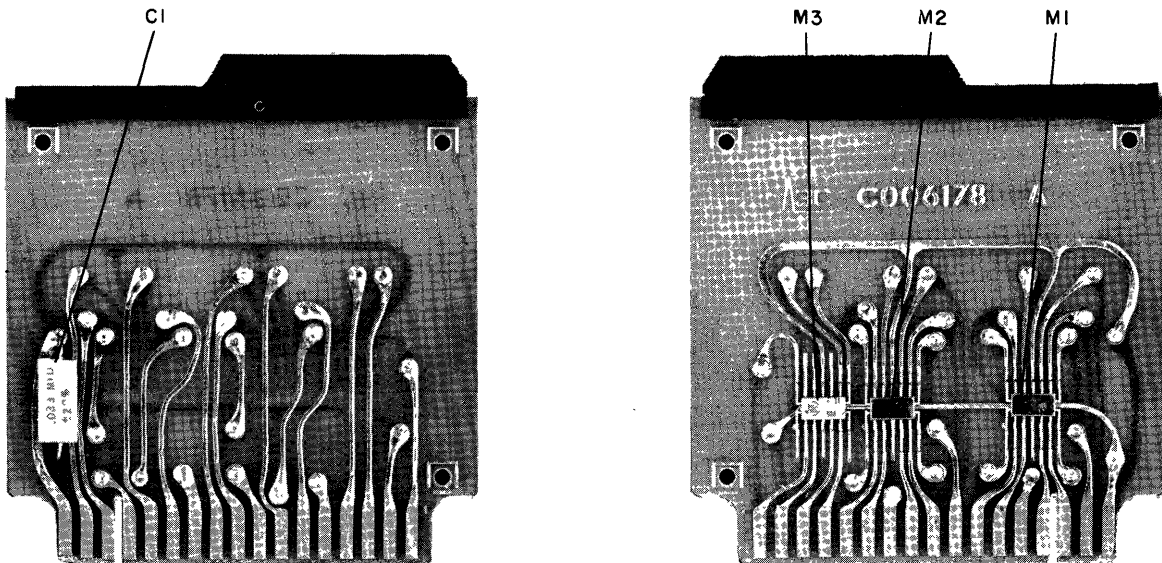


Figure 1. NAND Type 1 PAC, Model DI-355, Schematic Diagram and Logic Symbol

Parts Location



528

Electrical Parts List

Ref. Desig.	Description	3C Part No.
M1, M2	MICROCIRCUIT: F-02, quad NAND gate integrated circuit	950 100 002
M3	MICROCIRCUIT: F-01, dual NAND gate integrated circuit	950 100 001
C1	CAPACITOR, FIXED, PLASTIC DIELECTRIC: 0.033 $\mu$ f $\pm$ 20%, 50 vdc	930 313 016

Figure 2. NAND Type 1 PAC, Model DI-335,  
Parts Location and Identification

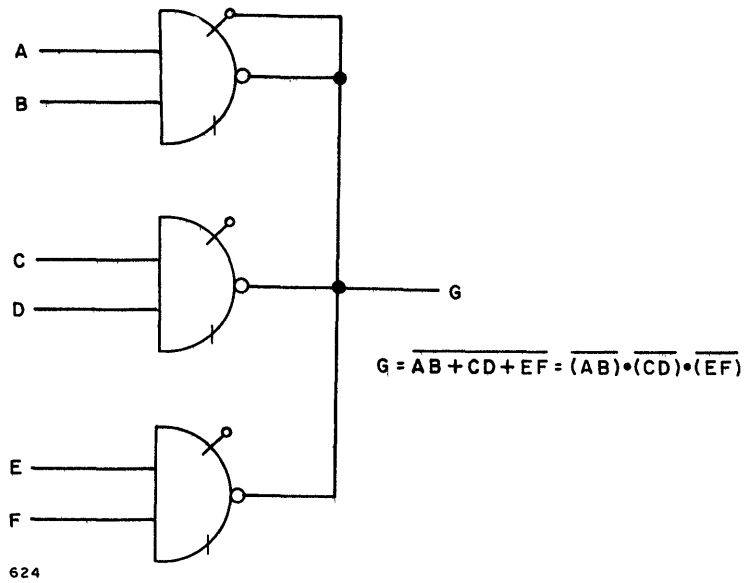


Figure. 3. NAND Type 1 PAC, Model DI-335,  
Gates Used in Parallel

### 3-11 NAND TYPE 2 PAC, MODEL DL-335

The NAND Type 2 PAC, Model DL-335 (Figures 3-11.1 and 3-11.2), contains six 4-input NAND gates. Each gate performs the NAND function for positive logic, (+6v = ONE, 0v = ZERO). For negative logic, it becomes a NOR gate.

Two of the six gates have separate load connections available at the PAC terminals. Outputs of these gates can be tied together, using a single load resistor, without loss of output drive capability.

A detailed description of the basic NAND circuit appears in Section II.

#### INPUT AND OUTPUT SIGNALS

Inputs. -- When all inputs to a gate are at +6v or not connected, the output is at ground. When any input is at ground, the output is at +6v.

Load. -- This point is internally connected through a collector load resistor to +6v.

Collector Output. -- The collector output of any gate must be connected to at least one collector resistor, either internal or external to the module.

Output. -- Each output terminal is internally connected to a collector load resistor. If an output is connected to load points or other outputs, the output drive capability of the structure is reduced.

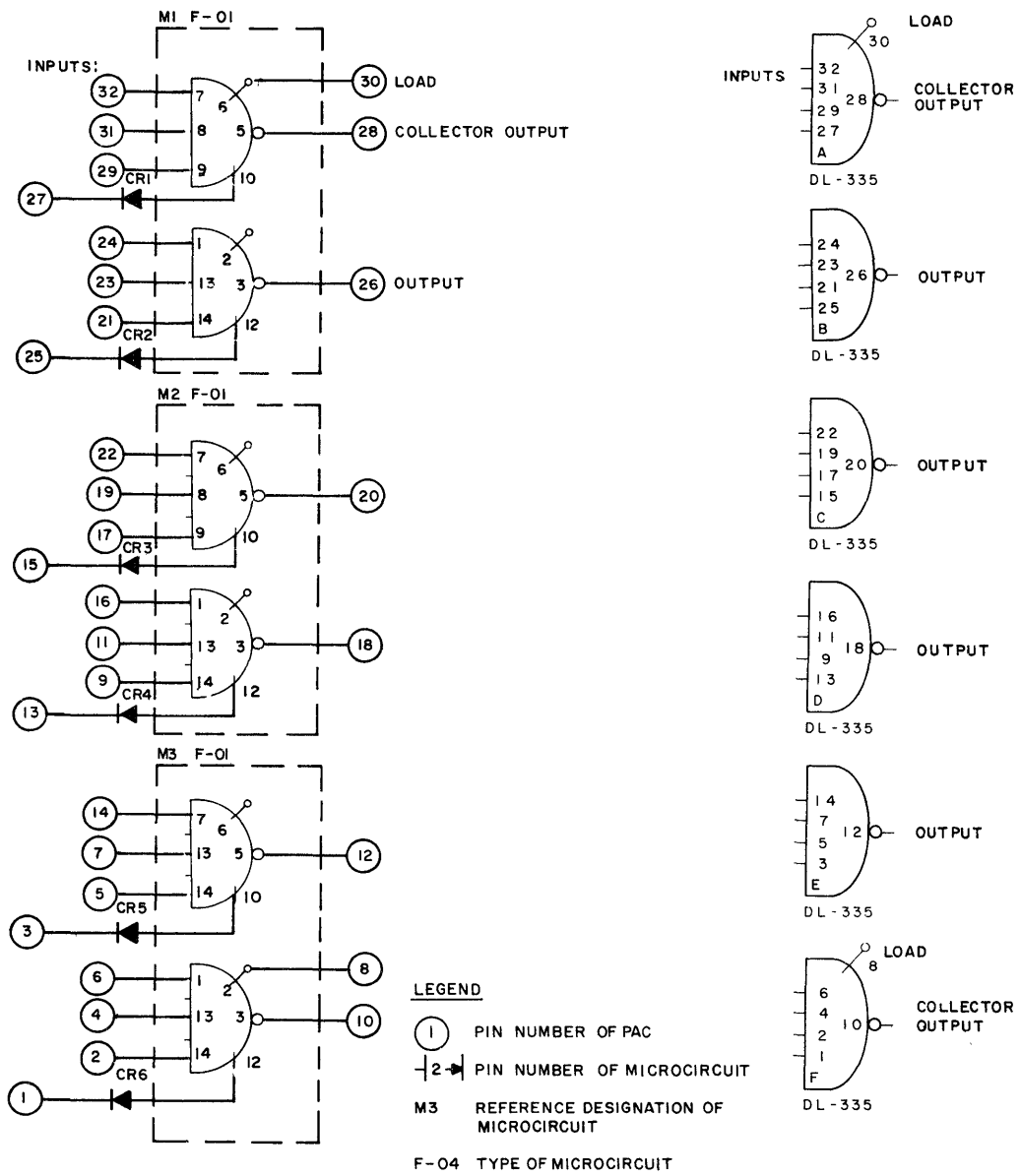
#### SPECIFICATIONS

<u>Frequency of Operation (System)</u>	<u>Circuit Delay</u>
DC to 5 mc	(Measured at +1.5v, averaged over two stages)
<u>Input Loading</u>	30 nsec (max)
1 unit load each	<u>Current Requirements</u>
<u>Fan-In</u>	+6v: 75 ma (max)
Refer to Section II.	<u>Power Dissipation</u>
<u>Output Drive Capability</u>	0.45w (max)
8 unit loads each	<u>Handle Color Code</u>
<u>Outputs in Parallel</u>	Red
Refer to Section II.	

#### APPLICATIONS

The NAND gates operate on levels or pulses, or combinations of both. Two gates can be wired back to back to form a dc set-reset flip-flop.

The two gates with separate load outputs form standard NAND gates when the load and collector output terminals are connected. When the collector outputs of such gates are connected in parallel as in Figure 3-11.3, the AND-OR-INVERT function is performed. At the point where the outputs are tied together, an AND operation with logic ONES (OR operation with logic ZEROs) takes place.



SCHMATIC

LOGIC SYMBOL

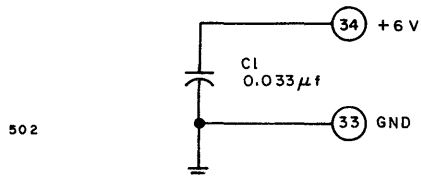
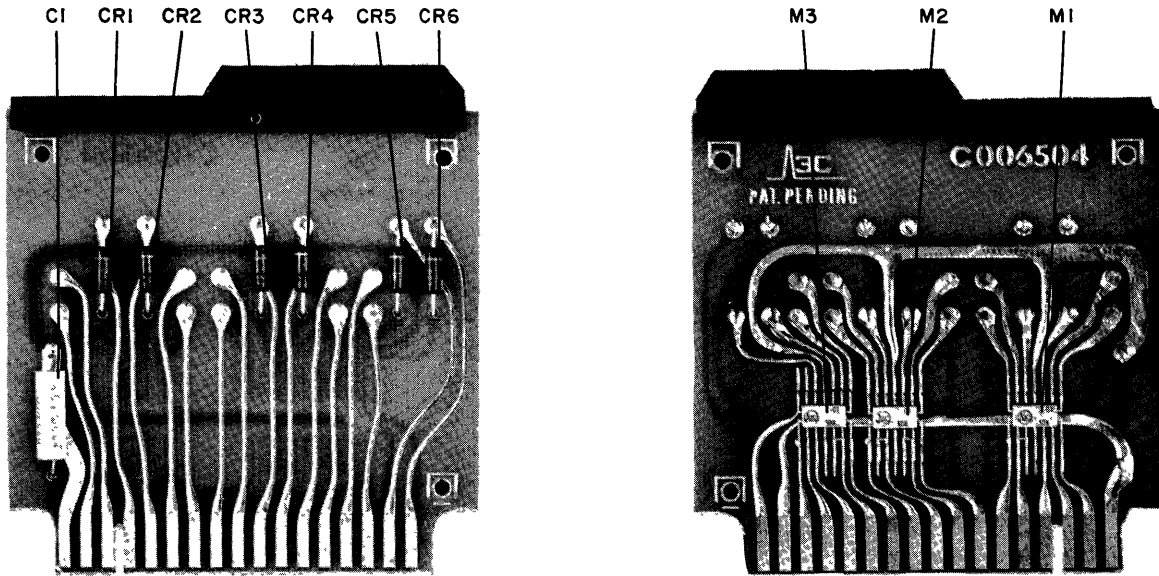


Figure 1. NAND Type 2 PAC, Model DL-335, Schematic Diagram and Logic Symbol



Parts Location



529

Electrical Parts List

Ref. Desig.	Description	3C Part No.
M1-M3	MICROCIRCUIT: F-01, dual NAND gate integrated circuit	950 100 001
C1	CAPACITOR, FIXED, PLASTIC DIELECTRIC: 0.033 $\mu$ f $\pm$ 20%, 50 vdc	930 313 016
CR1-CR6	DIODE: Replacement Type 1N914	943 083 001

Figure. 2. NAND Type 2 PAC, Model DL-335,  
Parts Location and Identification

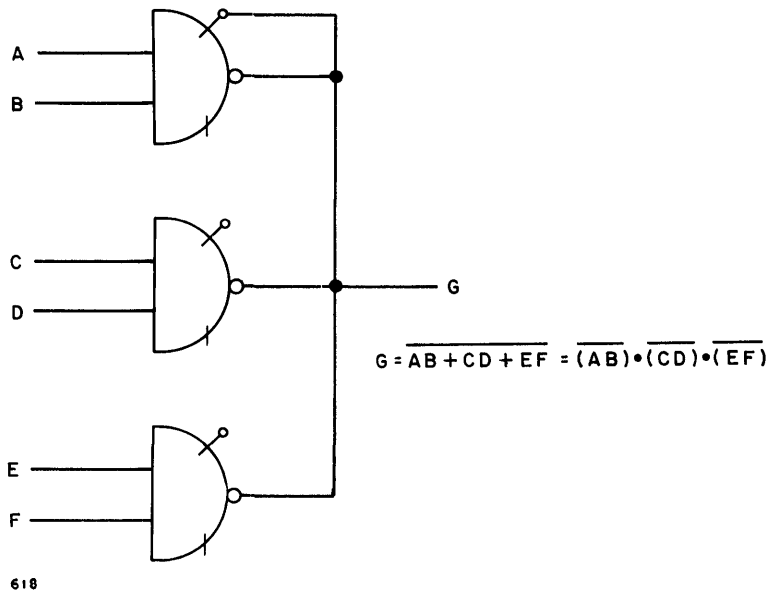


Figure. 3. NAND Type 2 PAC, Model DL-335,  
Gates Used in Parallel

## TRANSFER GATE PAC, MODEL TG-335

The Transfer Gate PAC, Model TG-335 (Figures 3-27.1 and 3-27.2), contains four independent functional gate structures. Two of the structures have four 2-input NAND gates, one input on each gate being common to the four gates. The remaining two structures have three 2-input NAND gates, one input being common to the three gates. (See Figure 3-27.3.)

The PAC can be used for the common transfer control of up to 14 data signals, the data when transferred is inverted in polarity.

### INPUT AND OUTPUT SIGNALS

Common Input. -- This input acts as a control or strobe input to each gate in the structure. The signal is active when at logic ONE.

### SPECIFICATIONS

<u>Frequency of Operation (System)</u>	<u>Circuit Delay</u>
DC to 5 mc	(Measured at +1.5v, averaged over two stages)
<u>Input Loading</u>	30 nsec (max)
Input: 1 unit load each	<u>Current Requirements</u>
Common input: 3 or 4 unit loads (equal to the number of gates in the structure)	+6v: 175 ma (max)
<u>Output Drive Capability</u>	<u>Power Dissipation</u>
8 unit loads	1.05w (max)
	<u>Handle Color Code</u>
	Red

### APPLICATIONS

Each gate structure can be used for the common transfer control of three or four signals (Figure 3-27.4). A separate line is provided for each output signal. The gates may be used separately as inverters when the common inputs are disconnected.

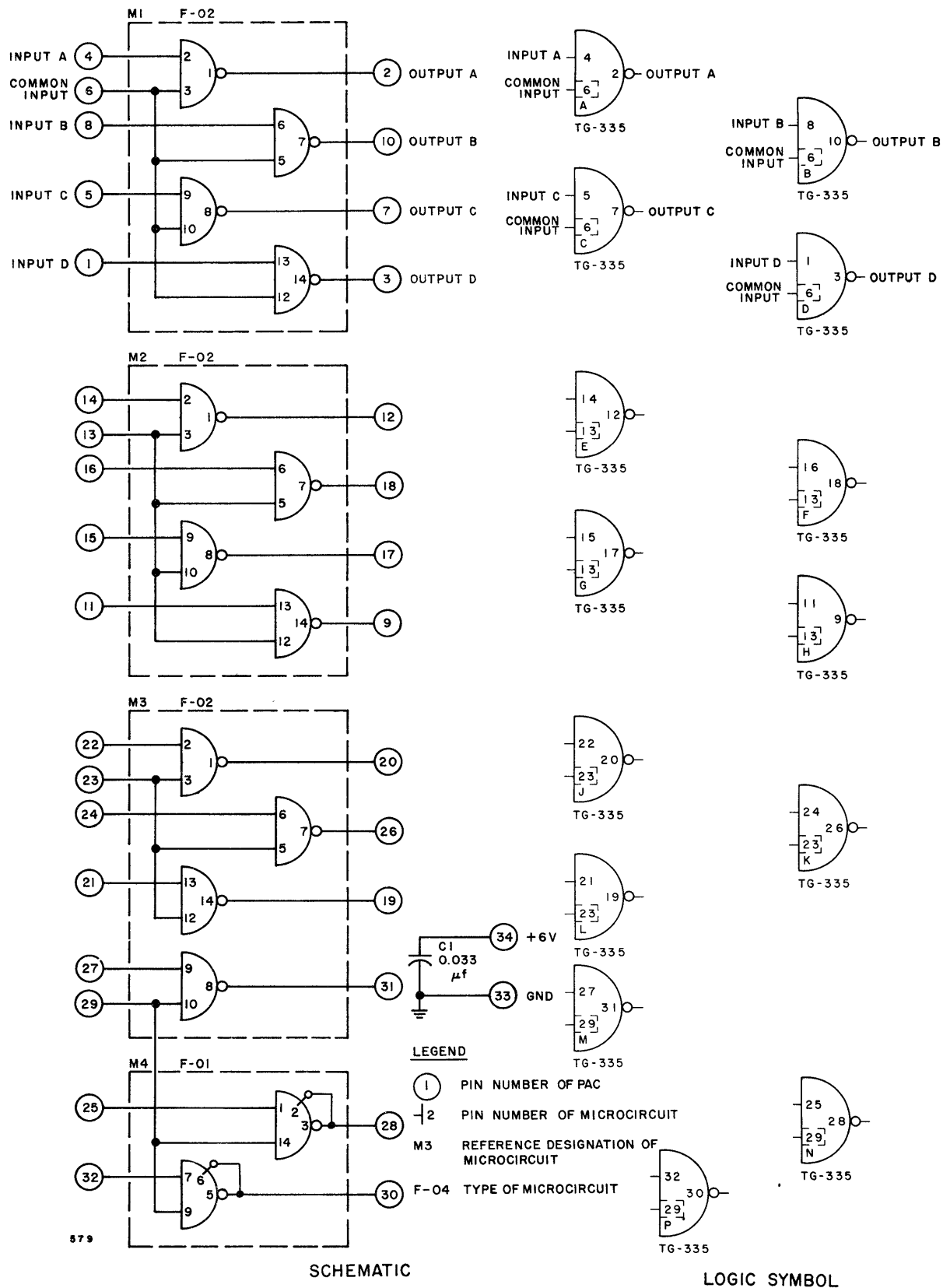
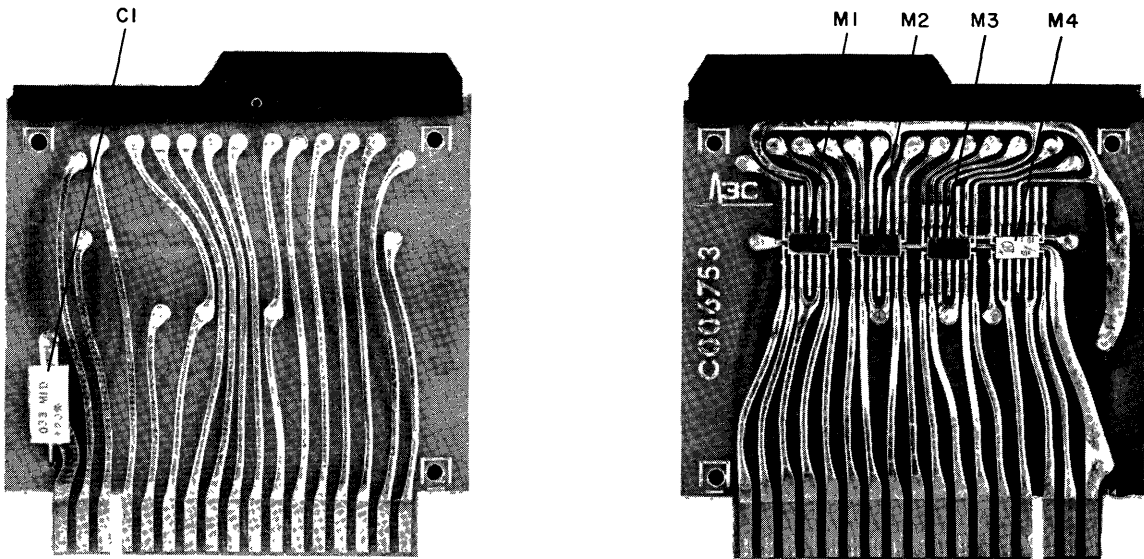


Figure. 1. Transfer Gate PAC, Model TG-335, Schematic Diagram and Logic Symbol

Parts Location



539

Electrical Parts List

Ref. Desig.	Description	3C Part No.
M1-M3	MICROCIRCUIT: F-02, quad NAND gate integrated circuit	950 100 002
M4	MICROCIRCUIT: F-01, dual NAND gate integrated circuit	950 100 001
C1	CAPACITOR, FIXED, PLASTIC DIELECTRIC: 0.033 $\mu$ f $\pm$ 20%, 50 vdc	930 313 016

Figure.2. Transfer Gate PAC, Model TG-335,  
Parts Location and Identification

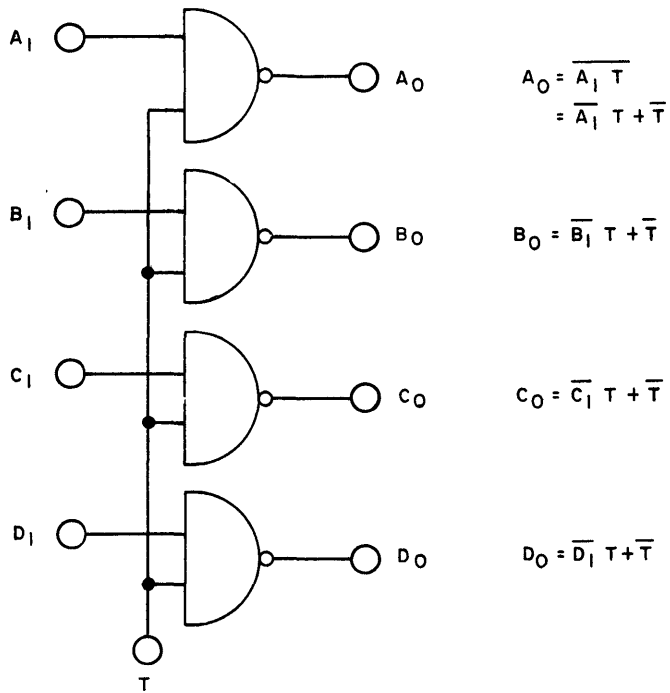


Figure. 3. Transfer Gate PAC, Model TG-335, Gate Structure Logic

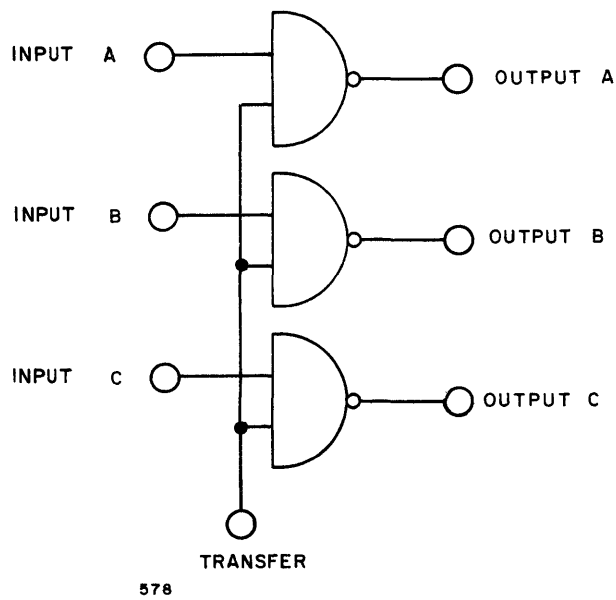


Figure. 4. Transfer Gate PAC, Model TG-335, Controlling Transfer of Three Signals

## POWER AMPLIFIER PAC, MODEL PA-335

The Power Amplifier PAC, Model PA-335 (Figures 3-22.1 and 3-22.2), contains six 3-input NAND gates that can be used for driving heavy loads. Each gate has two electrically common outputs to reduce load distribution current on any single wire. Built-in short-circuit protection limits output current if the output is accidentally grounded.

Each gate performs the NAND function for positive logic (positive voltage is a ONE and 0v is a ZERO). For negative logic, it becomes a NOR gate. When all inputs to a gate are at positive or not connected, the output goes to ground. When any input is at ground, the output goes to a positive voltage.

### SPECIFICATIONS

<u>Frequency of Operation (System)</u>	<u>Output Drive Capability</u>
DC to 5 mc	25 unit loads
<u>Input Loading</u>	<u>Circuit Delay</u>
2 unit loads each	(Measured at +1.5v, averaged over two stages)
<u>Current Requirements</u>	30 nsec (max)
+6v: 80 ma (max)	
<u>Power Dissipation</u>	
0.24w (max) static,	
0.48w (max) at 5 mc and with 250 pf stray capacitance	
<u>Handle Color Code</u>	
Green	

### APPLICATIONS

The power gates operate on levels, pulses, or with combinations of both. Two gates can be wired back to back to form a dc set/reset power flip-flop.

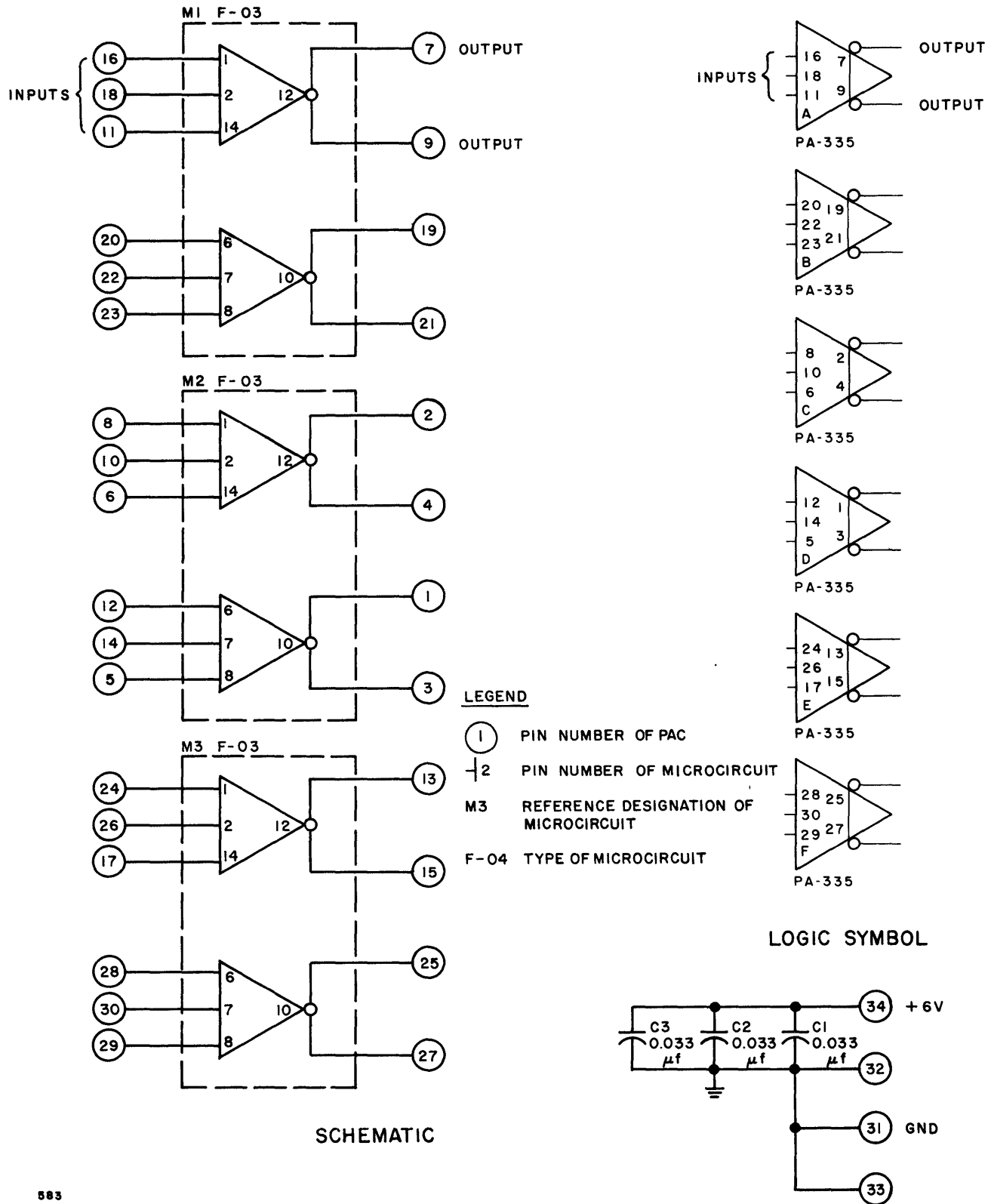
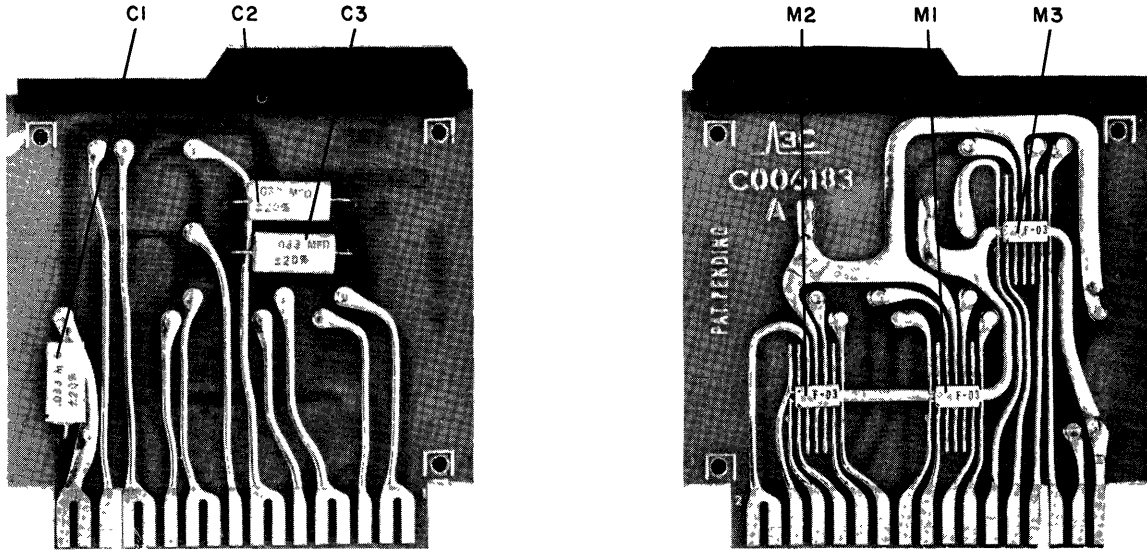


Figure.1. Power Amplifier PAC, Model PA-335, Schematic Diagram and Logic Symbol



## Parts Location



537

### Electrical Parts List

Ref. Desig.	Description	3C Part No.
M1-M3	MICROCIRCUIT: F-03, power amplifier integrated circuit	950 100 003
C1-C3	CAPACITOR, FIXED, PLASTIC DIELECTRIC: 0.033 $\mu$ f $\pm$ 20%, 50 vdc	930 313 016

Figure 2. Power Amplifier PAC, Model PA-335,  
Parts Location and Identification

# TABLE OF CONTENTS

<b>CORE MEMORY</b>	4015B	PRIORITY ACCESS
<b>INTRODUCTION</b>	INT	MEMORY CONTROL TIMING
MEMORY BLOCK DIAGRAM		MEMORY MODULE TIMING
Read/Restore		MEMORY ADDRESS SELECTION
Clear/Write		READ/RESTORE OPERATION
BASIC CORE THEORY		CLEAR/WRITE OPERATION
2 1/2 D Coincident Current Selection		PARITY GENERATION
MEMORY RETENTION		PARITY CHECKING
MEMORY WRAP AROUND		TRANSMISSION PARITY ERROR CHECKING
DEDICATED MEMORY ADDRESSES		CORE HEATER
<b>THEORY OF OPERATION</b>	THEORY	OPERATING POWER
CORE STACK ARRANGEMENT		POWER FAILURE OR SHUTDOWN
		<b>APPENDIX A</b>

# LIST OF ILLUSTRATIONS

Fig. INT. 1	Memory Module Block Diagram	Fig. THEORY. 4	4K Plane Organization and Addressing
Fig. INT. 2	Flux Induction in a Ferrite Core	Fig. THEORY. 5	Sense Winding
Fig. INT. 3	Basic $\beta h$ Curve	Fig. THEORY. 6	Memory Control Timing
Fig. INT. 4	Basic 2 1/2 D Operation	Fig. THEORY. 7	Memory Timing
Fig. INT. 5	$\beta h$ Curve, Coincident Current Effects	Fig. THEORY. 8	Memory Addressing
Fig. INT. 6	Read Coincident Current 2 1/2 D Selection, Basic Operation	Fig. THEORY. 9	Word Matrix Decode
Fig. INT. 7	Write Coincident Current 2 1/2 D Selection, Basic Operation	Fig. THEORY. 10	Address Selection Block Diagram
Fig. THEORY. 1	Memory System Functional Block Diagram	Fig. THEORY. 11	Read Drive Circuit
Fig. THEORY. 2	16K - 32K Word Memory Block Diagram	Fig. THEORY. 12	Write Drive Circuit
Fig. THEORY. 3	4K, 8K, 12K, 16K Stack Arrangement	Fig. THEORY. 13	Read/Restore Block Diagram
		Fig. THEORY. 14	Clear/Write Block Diagram
		Fig. THEORY. 15	Parity Generate
		Fig. THEORY. 16	Parity Check
		Fig. THEORY. 17	Core Heater Alarm Circuitry
		Fig. THEORY. 18	Buss Level Monitor Control

# LIST OF TABLES

Table THEORY. 1	Bit Matrix Decode
Table THEORY. 2	Thermocouple Conversion Chart (°C)

# INTRODUCTION

The 4015B Memory Unit is that part of the GE-PAC\* 4020 Central Processor used to store the program instructions and data. The memory is a random access device operating at a 1.6 micro-second memory cycle time. One memory cycle is the time from any one event in one cycle to the same event in the following cycle. Memory sizes of 4096, 8192, 12,288, 16,384, 20,480, 24,576, 28,672, or 32,768 words are available.

Each memory word consists of 25 bits of which 24 are data and one is used for an odd parity bit. The parity bit is generated prior to storing (writing) a word into memory and then checked when the word is read from memory. The parity bit is used to flag the running program when a word is read from memory that does not contain an odd number of "one" bits. If an odd number of "one" bits is not contained by the word read from memory, it indicates an error has occurred in either parity check and generate circuits, memory operation, or data transfers. If a parity error is detected, a signal is available to either: (1) Stop computer sequencing (via the STOP ON PARITY switch) and to set the Parity Error flip-flop to light the Error indicator and to provide monitoring by the JNP command, or (2) Just set the Parity Error flip-flop to light the Error indicator and to provide monitoring by the JNP command.

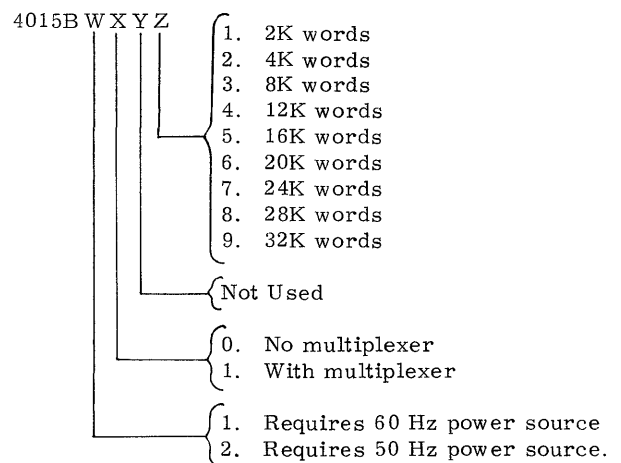
The memory unit has the optional ability to communicate with as many as four devices, one at a time, using the Memory Multiplexer. Using the multiplexer, devices such as drum, disc, etc., in addition to the Arithmetic Unit, may communicate directly with the memory. Access to memory using the multiplexer is on a priority basis. Should a device request access to memory while the memory is operational with another device, the requesting device must wait until the current memory cycle is completed. However, any time lost during the waiting period is minimal in light of the operational speed of the memory.

Features of the Core Memory module are summarized below:

- Type - 2 1/2 D, Coincident Current, Destructive Read.
- Capacity - 4096 (4K), 8192 (8K), 12,288 (12K), 16,384 (16K), 20,480 (20K), 24,576 (24K), and 32,768 (32K) words.
- Word Size - 24 information bits plus 1 parity bit.
- Memory Cycle Time - 1.6 microseconds.
- Access Time - 800 nanoseconds.
- Communication - Parallel.
- Random Access - All memory cells are directly addressable and all require the same time for reading and writing.

- Protection - Word Select currents are disabled during AC power failures and power turn-on so as to protect core cells from arbitrary data destruction.
- Temperature Stability - Temperature controlled core stack allows operation of core at 55°C. This is done to maintain operation throughout ambient temperature of 0° to 55°C.
- Operating Mode - Asynchronous. The Memory enters a dynamic mode only after it has received a request. There may be as many as four channels of communication: The Arithmetic Unit and any combination of from one to three Drum Couplers and/or Disc Couplers, and/or Peripheral Controller Couplers.
- Timing - The speed of the Memory Clock is 10 mega-Hertz.
- Error Checking - Odd parity is employed; all words are stored in core containing an odd number of "1" bits. Bit position 24, the most significant, is the parity bit.
- Memory Wrap Around - In a system that uses less than maximum memory, instructions may be executed with an address greater than implemented memory size.

The model numbers associated with the options of the 4015B Memory Unit are listed below.



## MEMORY BLOCK DIAGRAM

Fig. INT.1 illustrates the basic components of the 4015B Memory System. The 4015B consists of the Memory Control Section (68C972680) and the Memory Module (68C972952).

The Memory Control Module acts as the communication link between the user devices, establishes the priority

\* Registered Trademark of General Electric Company

of memory requests, multiplexes the data and address bits from the user devices to the memory, and provides parity generation and parity error detection.

The memory module contains the core stack to store the data bits, the address register to control the selection of the desired core address, a data register to hold the data read from memory or to be stored in memory, and the circuits required to retrieve data from, or store data into, the addressed memory location.

There are two basic operational modes of the memory module: Read (read/restore) data from memory and store (clear/write) data into memory. Therefore, to access memory the requesting user device must transfer, in addition to a request signal, a signal which

specifies the read or write operation and the bits to specify the desired memory address. Also, if the operation is write, the data bits to be stored in the addressed location must be transferred to the memory module.

Actual operation of the memory is very similar for both a read and write operation. The basic differences are: (1) Parity is checked during a read operation and parity is generated during a write operation, and (2) During a read operation, the data sensed from core is gated to the data register and during write, the data from the user device is gated to the data register. In either mode, the contents of the data register are stored in the addressed memory location. If it is a read operation, this data will be the data previously read from core. If it is a write operation, this data will be the data transferred from the user device.

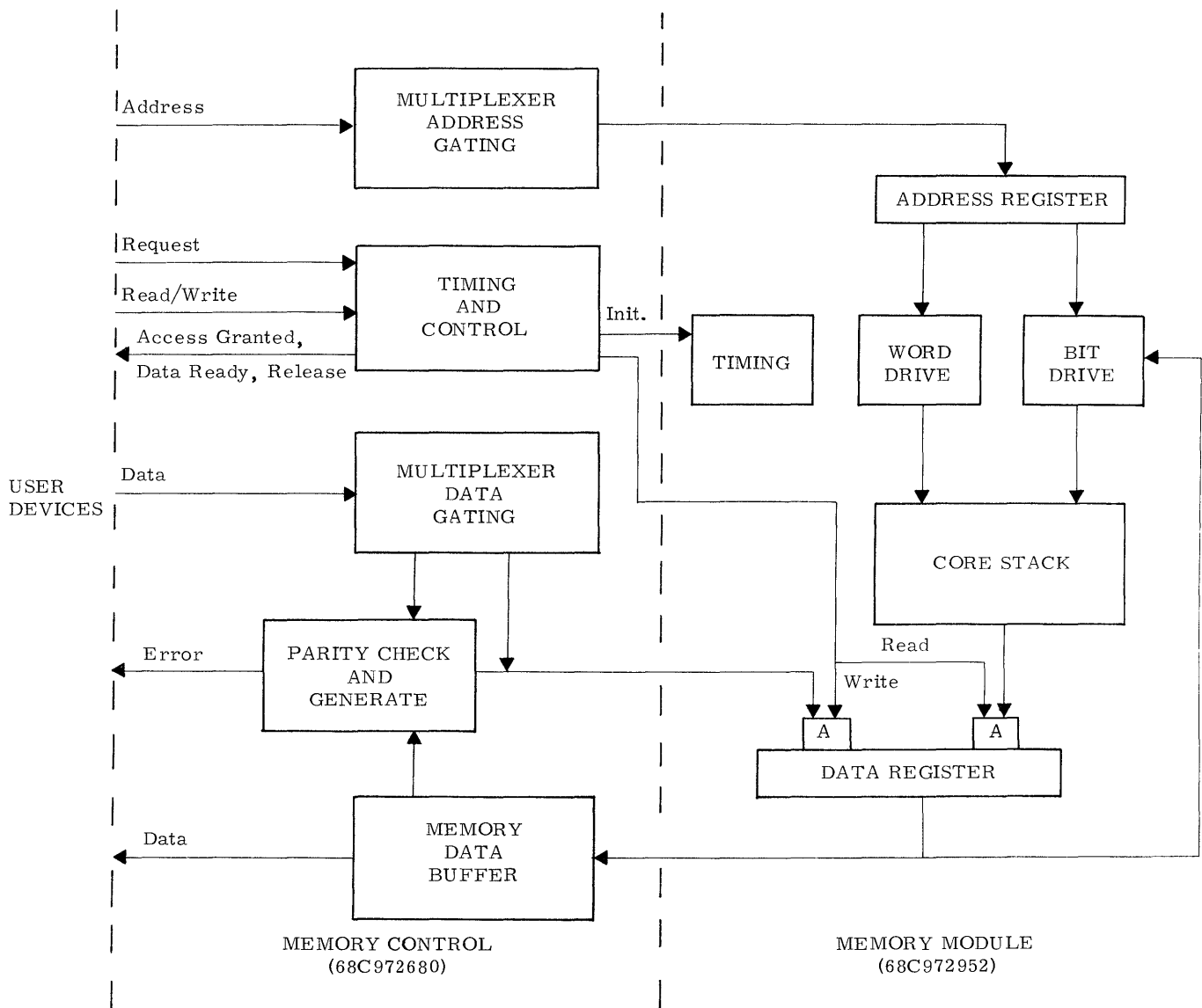


Fig. INT. 1. Memory Module Block Diagram

## Read/Restore

The Read/Restore mode of memory operation is used to obtain the contents of a memory word and to leave the same data in the memory word. The read/restore cycle is initiated by a request from the user device accompanied by a read (store) signal from the device. The memory control unit determines the priority of the request and applies an access granted signal back to the device when access is granted. The memory control section then sends a signal to initiate the operation of the memory module and transfers the address of the desired memory word to the memory address register of the memory module. Decoding of this memory address enables the Word Drive and Bit Drive selection lines to switch all cores of the addressed word that are ones to the zero state. Switching these cores to the zero state induces a voltage in the sense line. This voltage is then amplified and gated to the data register. From the data register, the data read from the addressed memory word is transferred to the user device and parity check circuitry via the memory data buffer. If an even number of one bits is contained by the data word, an error signal is applied to the arithmetic unit. A data ready signal is applied from the memory control logic to the user device to gate the data to the user device. The data contained by the data register is also applied to the Bit Drive Selection circuitry. During the write (restore) portion of the memory cycle, the corresponding Bit Drive Select line is driven only if a one is to be stored in the corresponding bit position. A memory release signal is generated by the memory control logic to indicate the end of the memory cycle.

## Clear/Write

The Clear/Write mode of memory operation is used to store new data in an addressed memory word location. The Clear/Write mode is initiated by a request signal from the user device along with the write (store) signal. The priority of the request signal is determined by the memory control logic and when access is granted, a signal is sent to the user device. The core memory address is then applied to the memory unit along with an initiate signal to start the memory cycle. The address is decoded from the address register and enables the corresponding Word Drive and Bit Drive selection lines. Driving the select lines flips the cores of the addressed word to the zero state. Since this is a write (store) operation, the voltage induced in the sense lines is not used. Instead, the data from the user device to be stored in the addressed memory word is gated through

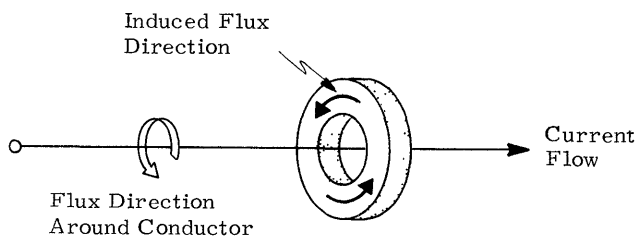


Fig. INT. 2. Flux Induction in a Ferrite Core

the memory control unit where parity is generated. The data word to be stored along with the parity bit is gated to the data register of the memory unit. The write portion of the memory cycle is then entered. The contents of the data register are then used to control the Bit Drive select lines. Each one bit in the data register will enable the addressed Bit Drive line and set the corresponding core to the one position. In this manner, the data bits from the user device are stored in the addressed memory location.

## BASIC CORE THEORY

The magnetic properties of ferrite cores make them ideal high-speed storage elements. If, as shown by Fig. INT. 2, a DC current is applied to a wire threading the core, the magnetic flux around the wire induces a similar flux pattern in the core. According to the left-hand rule, the direction of the induced flux field depends on the direction of current flow in the wire. Thus, the core may be forced to a known magnetic state by driving current in one direction and this state may be reversed by reversing the current. Since the core is capable of retaining an induced magnetic field indefinitely, it possesses all the bi-stable characteristics necessary to store binary information.

A certain amount of energy is required to change a core from one magnetic state to the other. If flux density ( $\beta$ ) within the core is plotted as a function of drive current ( $h$ ), a graph similar to that shown by Fig. INT. 3 is obtained. The area bounded by the curve represents the power expended to overcome hysteresis losses in the core material; for this reason, the  $\beta h$  curve is commonly called a "hysteresis loop".

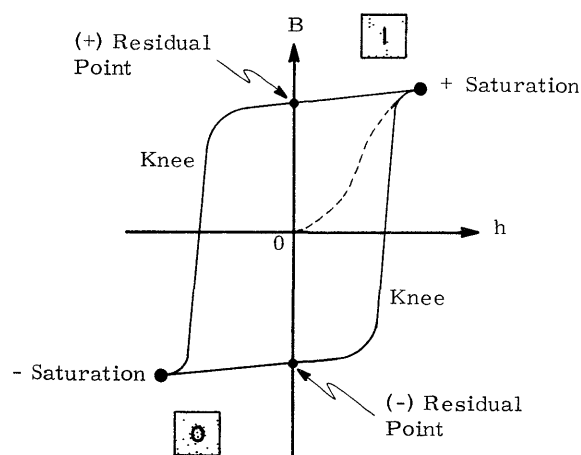


Fig. INT. 3. Basic  $\beta h$  Curve

Assuming the first drive current applied to the core is of sufficient magnitude in the positive  $h$  direction, the core follows the broken line from the origin to the point of positive  $\beta$  saturation. When this drive current is removed, the core retains most of the induced flux as shown by the slight slope of the curve as it returns to the (+) residual point of zero  $h$ .

If current is now increased in the (-h) direction, flux density decreases slightly until the threshold point or "knee" of the hysteresis loop is reached. At this point a slight increase in h is sufficient to cause a rapid reversal of the magnetic field and the core is "flipped" from one state to the other. A similar sequence is followed when the core is moved from the (-) residual point to the (+) saturation point by a (+h) current. Notice that a (+h) current applied to the core has little effect if the core lies at the (+) residual point, since it only forces the core a little farther into saturation; the same is true if a (-h) current is applied to the core which is at the (-) saturation point.

When the core is magnetized to the upper portion of its hysteresis loop it is said to contain a 1; if it lies on the negative portion of the curve it contains a 0. The amount of drive current necessary to flip the core from one magnetic state to the other is called a "full-select" current.

In order to read information from the core, an additional wire, called a sense winding, is threaded through the core. When information is to be read, a negative full select current is passed through the core and the resultant change of flux is sampled by the sense winding. If the core contains a zero, the flux change is negligible; if the core contains a one, however, the large change of flux as the core flips induces a voltage pulse in the sense winding. A high-gain amplifier, called a Sense Amplifier, monitors the output of the sense winding to detect the presence of a one-bit. Since reading sets the core to zero, the stored information is essentially lost; hence, the term "destructive read" is applied to core memories. In order to retain information, a complete memory cycle must consist of both a read and write operation.

The method of writing a one-bit is simply to drive a full-select current through the core. Since the cores are cleared by the read current, writing a zero is accomplished by inhibiting the full select current through the core. Therefore, a zero is written into core by not writing a one.

### 2½ D Coincident Current Selection

In the preceding discussion of Basic Core Theory, selection of a memory cell for reading or writing is shown to be possible through the aid of a full select current applied through a core and a sense wire to detect a change of flux in the core. The complexity of selecting a core with a single drive line is prohibitive, however, in the light of 4096 separate selections necessary for even the smallest memory size.

In order to minimize the address selection logic, each core is threaded by two drive windings, designated Bit Select and Word Select, as shown in Fig. INT. 4.

The core is selected by coincident half-select currents on the Bit and Word Select lines rather than by a full-select current on the single line previously described.

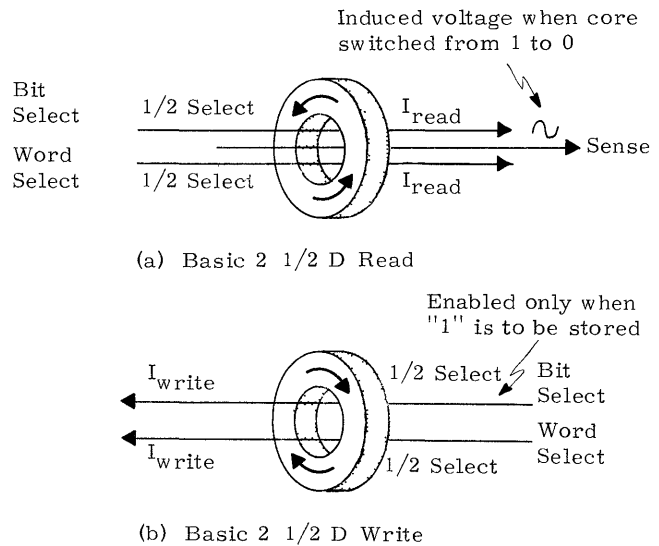


Fig. INT. 4. Basic 2 1/2 D Operation

Fig. INT. 5 illustrates the effect of various combinations of half-select currents on a memory core. Notice the ratio of the sensed output signal,  $V_{sense}$ , for a full-selected current core compared to the sensed output  $V_h$  for a half-selected memory core. This relationship permits one Bit Select line and one Word Select line to be threaded through many cores in a matrix. The only core affected being the one in which the Bit and Word select lines meet with current flowing in the same direction.

During a write cycle, current is reversed in the Bit and Word select lines to induce flux in the one direction if a one is to be stored. Control of whether a one is to be stored or a zero is to be stored is accomplished by enabling the Bit Select current only when a one is to be written. Inhibiting current flow during the write cycle in the Bit Select line leaves the core in the zero state.

Figs. INT. 6 and INT. 7 illustrate two bits of an 8-word memory. Fig. INT. 6 illustrates the direction of current flow in the Bit and Word Select lines to read the contents of two different words. Notice that only one word (bits 0 and 1) is selected by having both the Bit Select and Word Select current in the same direction. The locations,  $00_8$  and  $10_8$ , are provided with arbitrary addresses to illustrate that by changing the direction of current flow through the select lines different addresses may be selected. Therefore, when current is flowing in the direction of the solid arrows, address  $00_8$  will be addressed and the bit 0 and bit 1 cores will be flipped, if previously a one, inducing a signal in the sense lines for interpretation as stored ones in these positions. Addressing cell  $10_8$  will cause the bit 0 and bit 1 cores for this address to flip, as indicated by the unshaded arrows.

Fig. INT. 7 illustrates the same 2-bit, 8 word memory for writing a one back in both bit positions of arbitrary addresses  $00_8$  and  $10_8$ . Again, the solid arrows indicate the current flow for address  $00_8$  and the unshaded

arrows indicate the direction of current flow for address 10<sub>8</sub>. Notice that during the write cycle, the Bit Select lines are controlled by the desired data bit to be stored in each bit position. Current will flow in the Bit Select

line only if a one is to be stored in the addressed bit. If a zero is to be stored current only flows in the Word Select lines leaving the core in the zero state.

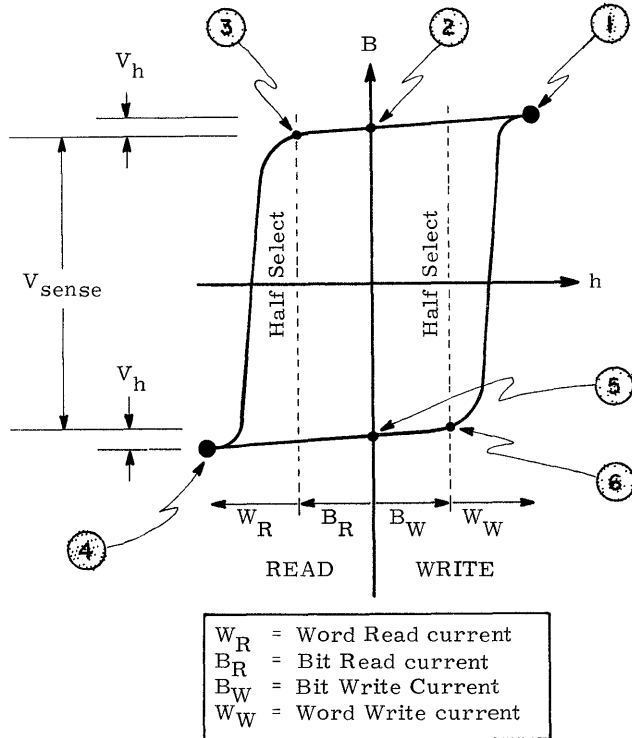
### MEMORY RETENTION

The magnetic core array does not require power to provide its static memory capability. A pulse of power is required to switch cores from one state to the other, but the pulse is not necessary to hold cores in their respective states. All cores remain in the state to which they have been switched because of the retentivity of the core magnetic material. If power is removed or lost, the magnetic core array retains stored information indefinitely.

### MEMORY WRAP AROUND

The Memory Address Multiplexer inputs are wired with only the address bits required to address the implemented core size. That is, if an 8K memory module is implemented, only address bits 0 through 12 are wired to the address multiplexer, etc. Using this technique, when the user device addresses a core cell above the implemented size, a corresponding location within the implemented memory will be addressed or zeros will be fetched from core.

The following chart illustrates the 4K memory block addressed and the corresponding 4K block affected for the various implemented memory sizes.



- ① "1" Saturation Point - Reached from ⑤ during write cycle when B<sub>W</sub> and W<sub>W</sub> provide current in the same direction.
- ② "1" Residual Point - Reached from ① or ③ after current is removed.
- ③ Read Half Select Point - Reached from ② by W<sub>R</sub> or B<sub>R</sub>.
- ④ "0" Saturation Point - Reached from ② or ⑤ during read cycle when B<sub>R</sub> and W<sub>R</sub> provide current in the same direction.
- ⑤ "0" Residual Point - Reached from ④ or ⑥ after current is removed.
- ⑥ Write Half Select Point - Reached from ⑤ by W<sub>W</sub> or B<sub>W</sub>.

V<sub>sense</sub> = Voltage sensed proportional to the change in the B axis.

Fig. INT. 5.  $\delta h$  Curve, Coincident Current Effects

		Implemented Memory							
		4K	8K	12K	16K	20K	24K	28K	32K
4K Block Addressed	0- 4K	0-4	0-4	0- 4	0- 4	0- 4	0- 4	0- 4	0- 4
	4- 8K	0-4	4-8	4- 8	4- 8	4- 8	4- 8	4- 8	4- 8
	8-12K	0-4	0-4	8-12	8-12	8-12	8-12	8-12	8-12
	12-16K	4-8	4-8	*	12-16	12-16	12-16	12-16	12-16
	16-20K	0-4	0-4	0- 4	0- 4	16-20	16-20	16-20	16-20
	20-24K	0-4	4-8	4- 8	4- 8	*	20-24	20-24	20-24
	24-28K	0-4	0-4	8-12	8-12	*	*	24-28	24-28
	28-32K	0-4	4-8	*	12-16	*	*	*	28-32

\* A zero word is read with no parity error. A write operation would go undetected unless Memory Protect were used.

### DEDICATED MEMORY ADDRESSES

Certain memory locations are pre-assigned for specific use. These memory locations should be used only for the purpose intended unless hardware is not included to make use of the specific reserved address. The reserved memory locations and the pre-assigned uses are listed as follows.

Octal Address	Specific Use	Octal Address	Specific Use	
00	Primary Bulk Storage Pointer Word (Channel 1)	27	Not Defined	
01	Index Register Words	30	Not Defined	
07		31	Common Peripheral Pointer Words	
10	Q Register Words	37		
11	Additional Bulk Storage (11 = Channel 2) Pointer Words (12 = Channel 3)	40	Quasi Instruction Branch Vectors (SPB's) or Single Word Quasi	
17		77		
20		100		Memory Protect Status Words
21	Memory Protect, Saved I Register Word	177		
22	Not Defined	8 to 128 Automatic Program Interrupt Control Locations		
23	Not Defined			
24	API Stall Alarm Error Exit Location (SPB)		200	a. Entry Location (SPB's)
25	Not Defined		377	b. Tally Control Instructions (DMT's)
26	Not Defined			c. I/O Control Words (TIM/TOM Control Words)

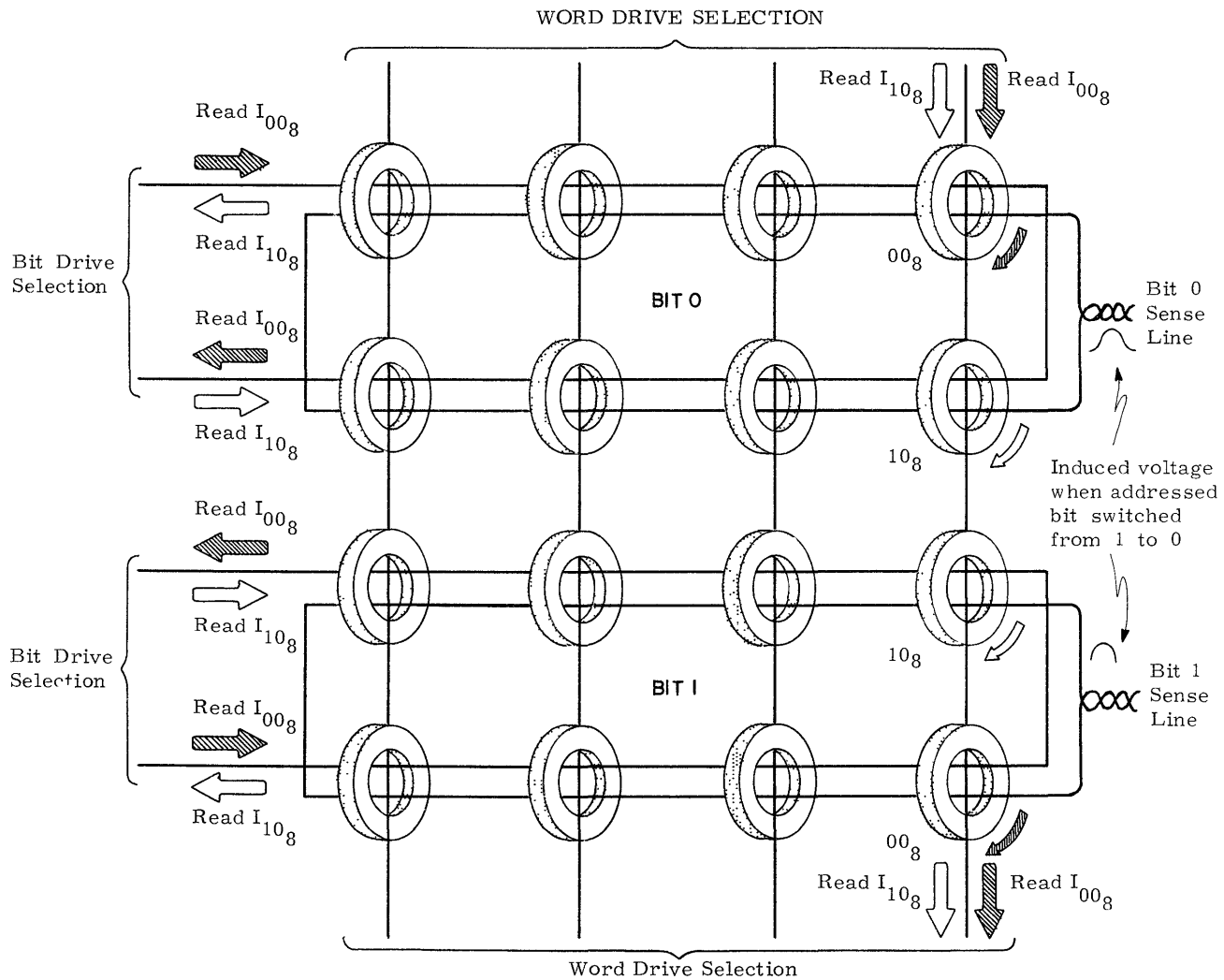


Fig. INT. 6. Read Coincident Current 2 1/2 D Selection, Basic Operation



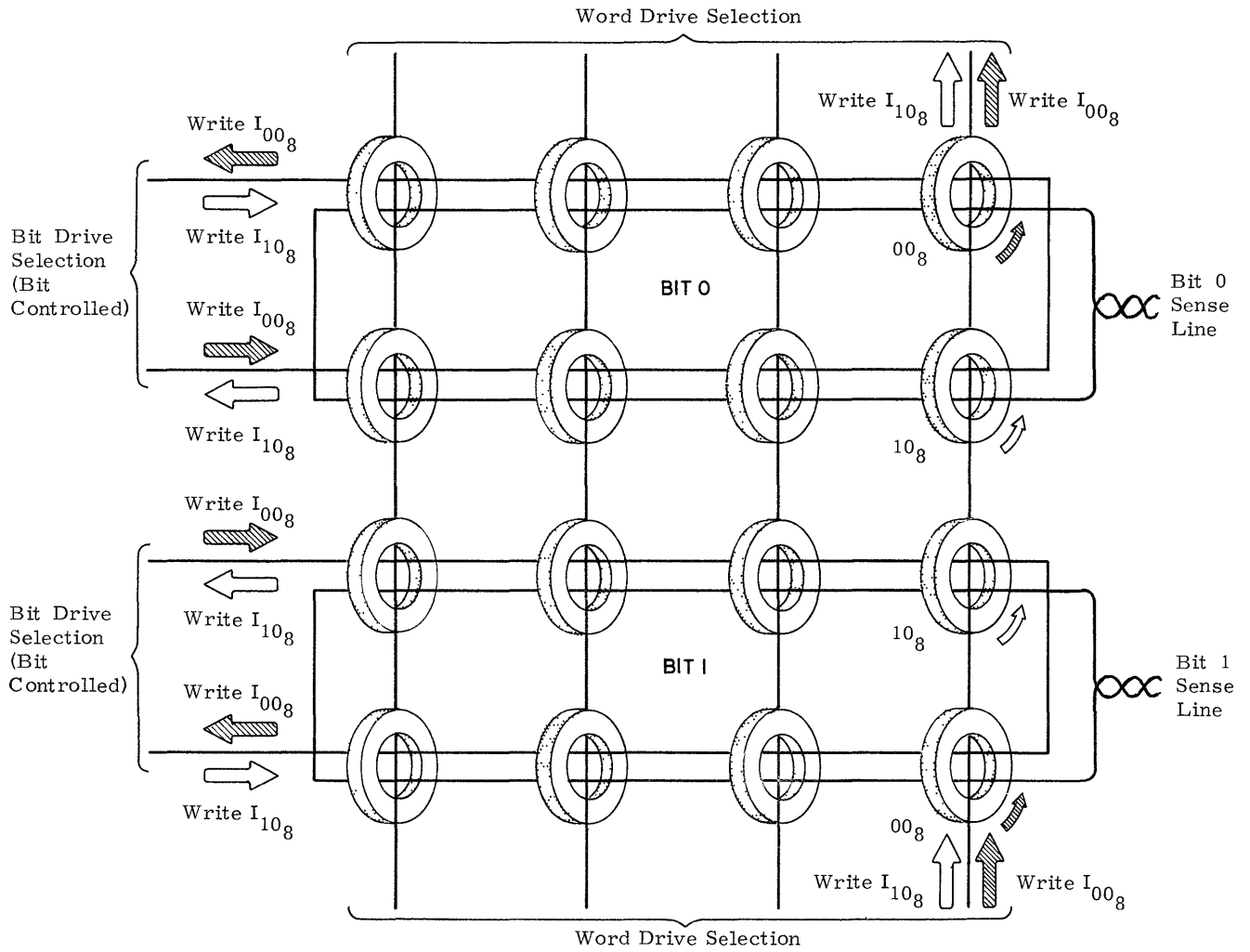


Fig. INT. 7. Write Coincident Current 2 1/2 D Selection, Basic Operation

# THEORY OF OPERATION

A functional block diagram of the memory system is contained in Fig. THEORY. 1. As shown in the diagram as many as three devices, in addition to the Arithmetic Unit, may communicate directly with the memory. These three devices may be any combination of from one to three Drum Couplers, and/or Disc Couplers, and/or Peripheral Controller Couplers.

The user device applies a memory request signal, the operation (read or write) to be performed, and the memory location address to be affected to the Memory Control Unit. If more than one device requests memory at the same time, the Memory Control logic determines which device has priority and grants access to that device only. The device connected to the Channel 1 input has priority over the Arithmetic Unit or devices connected to Channel 2 or Channel 3 inputs. Channel 2 devices have priority over the Arithmetic Unit and the Device connected to Channel 3. Channel 3 requests have priority over Arithmetic Unit requests.

The Arithmetic Unit has the lowest priority since its operation speed is the fastest. The Memory Control Unit applies a device granted signal to the Device that obtains priority.

After the access priority has been established, a signal is applied to the core memory module to indicate the type of operation and to initiate the memory cycle.

If the operation is write (Clear/Write) into core, the data transferred from the user device is gated through the Multiplexer Data Gates. The parity bit, if required, is generated and the data is transferred to the Data Register of the memory module. The address of the core location to be affected is multiplexed from the user device to the Address Register of the memory module. The address is decoded and the corresponding word and bit drive lines are enabled. Enabling the Word and Bit Drive lines will cause the cores in this memory location to clear. Since this is a write into core operation, the data sensed from this location is not used. The "one" data bits contained in the Data Register then control the decoded Bit Address lines to set the corresponding data bits of the addressed location. Setting the core cells of the addressed location to the configuration of "one" bits in the Data Register completes the memory cycle. When the Arithmetic Unit has access to memory, a Memory Release signal is applied from the Memory Control logic to the AU at the end of the cycle.

Read (Read/Restore) core memory operations require the same determination of priority after receipt of the memory request signal. The multiplexed address is again decoded to clear the bits of the corresponding memory location. The data sensed during this clearing operation is applied through the sense amplifiers and strobed into the Data Register. The data contained in the Data Register is then used to control the Bit Drive lines and, thereby, store the same data back in the addressed location. The data in the Data Register is also applied through the Memory Input Buffer for application to the user device. A Data Ready signal is provided by the Memory Control Unit to set this data in the user device. The data read from core is checked for odd parity.

If a parity error exists, a parity error signal is provided. This parity error signal lights the Core Parity and Alarm indicators on the console, provides for program detection of the error (JNP), and may be optionally used to halt further accesses to memory by the AU using the Stop On Parity switch. If the user is the Arithmetic Unit, a memory release signal is applied to the Arithmetic Unit upon completion of the memory cycle.

The Memory Control Unit is capable of controlling up to 32K words of memory storage. A Memory Module contains up to 16K words of memory storage. Therefore, if more than 16K words of memory are required, two Memory Modules are used as shown in Fig. THEORY. 2. These Memory Modules are identical.

The Memory Control Unit will then initiate the proper 16K Memory Module addressed by using memory address bit 14. If memory address bit 14 is a zero, then the lower 16K Memory Module operation is initiated. If memory address bit 14 is a one, then the upper 16K Memory Module operation is initiated.

## CORE STACK ARRANGEMENT

The core stack is a sealed unit which is maintained at a constant temperature - nominally 55°C. The capacity of the stack may vary from 4 to 16 thousand words in 4 thousand word increments. Its physical size is the same regardless of capacity - 12.5" wide x 20" high x 3" thick (see Appendix A).

Within the encased unit are the magnetic cores, a heater element to hold the temperature constant, thermocouples for monitoring the internal temperature, and over/under temperature protection sensors used for alarming.

The magnetic stack is arranged depending on the capacity. It contains from one to four planes, the required diode matrix, and the connections for the Word Drive lines, Bit Drive lines, and Sense lines. Each plane contains the magnetic cores required to store 4K (4096) words of data. Fig. THEORY. 3 illustrates the organization of the stack for the different stack capacities.

Fig. THEORY. 4 illustrates a 4K word plane including the Word and Bit Drive lines. As shown, a single plane contains 4096 cores (.03 inches in diameter) for each of the 25 bits of a word. The cores for each bit position are physically separated.

The 4096 cores for each bit position are organized in a 16 by 256 matrix. To simplify the understanding of address decoding, the 16 by 256 matrix for each bit is further divided into a 16 by 16 group as shown in the figure.

A close study of Fig. THEORY. 4 will illustrate the basic method of addressing one core in each of the 25 bit positions. Notice that although a signal for selecting the Word Line Diode End is common to one of the 16 lines for each of the 16 groups, only one of these lines will be enabled by the Word Line Group End address signal. Therefore, the Word Line enabled passes through 16 cores of each bit position. Individual Bit

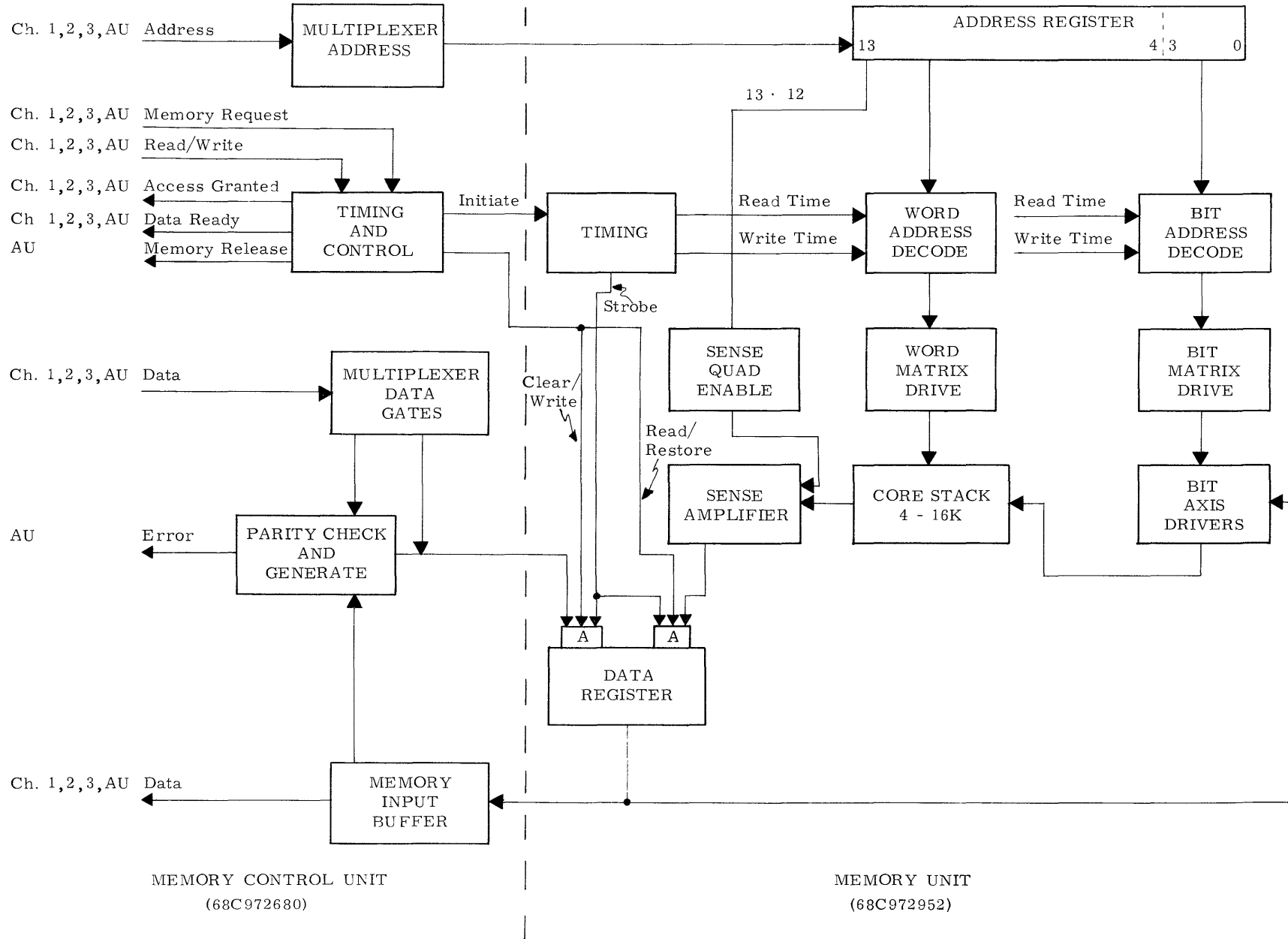


Fig. THEORY. 1. Memory System Functional Block Diagram

Diode Matrix circuits are provided for each bit position. The Bit Diode Matrix circuits enable one of 16 lines to each bit position. The enabled Bit Line selects the addressed core cell from the 16 cores selected by the Word Line. Also, notice that current may flow in both directions in both the Word Line and the Bit Line. Current flows in one direction to read the contents of the addressed cell and current flows in the other direction to set the addressed core cell to the "one" position.

A detailed discussion of address selection is provided later in this section. The previous discussion is provided only as a basic understanding of the core stack organization and interconnections of the drive lines.

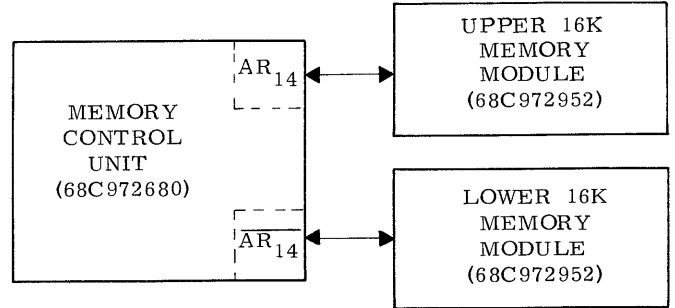
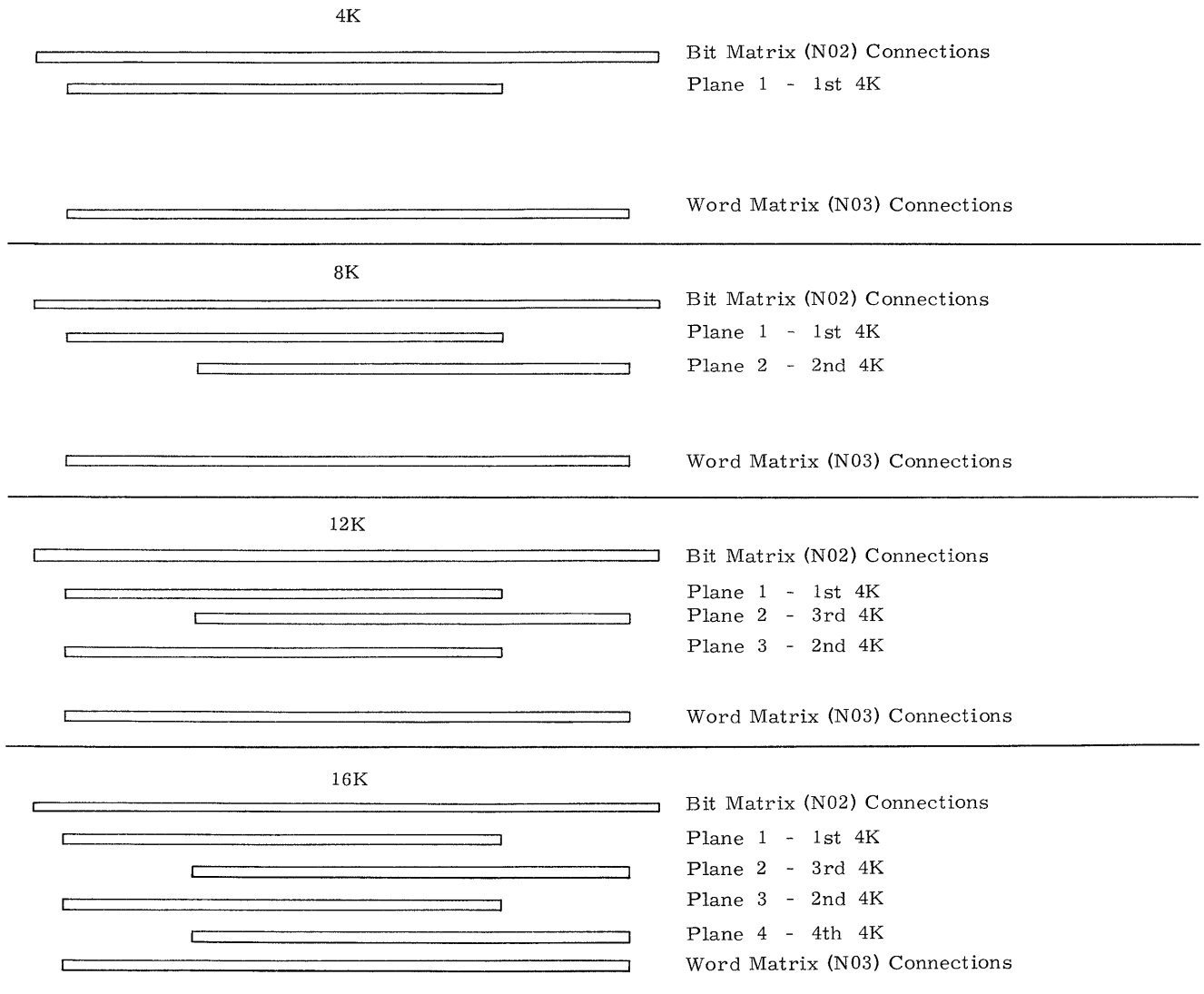


Fig. THEORY. 2. 16K - 32K Word Memory, Block Diagram



NOTE: Appendix A lists the location and pin connections of the Bit (N02) and Word (N03) Matrix connections.

Fig. THEORY. 3. 4K, 8K, 12K, 16K Stack Arrangement

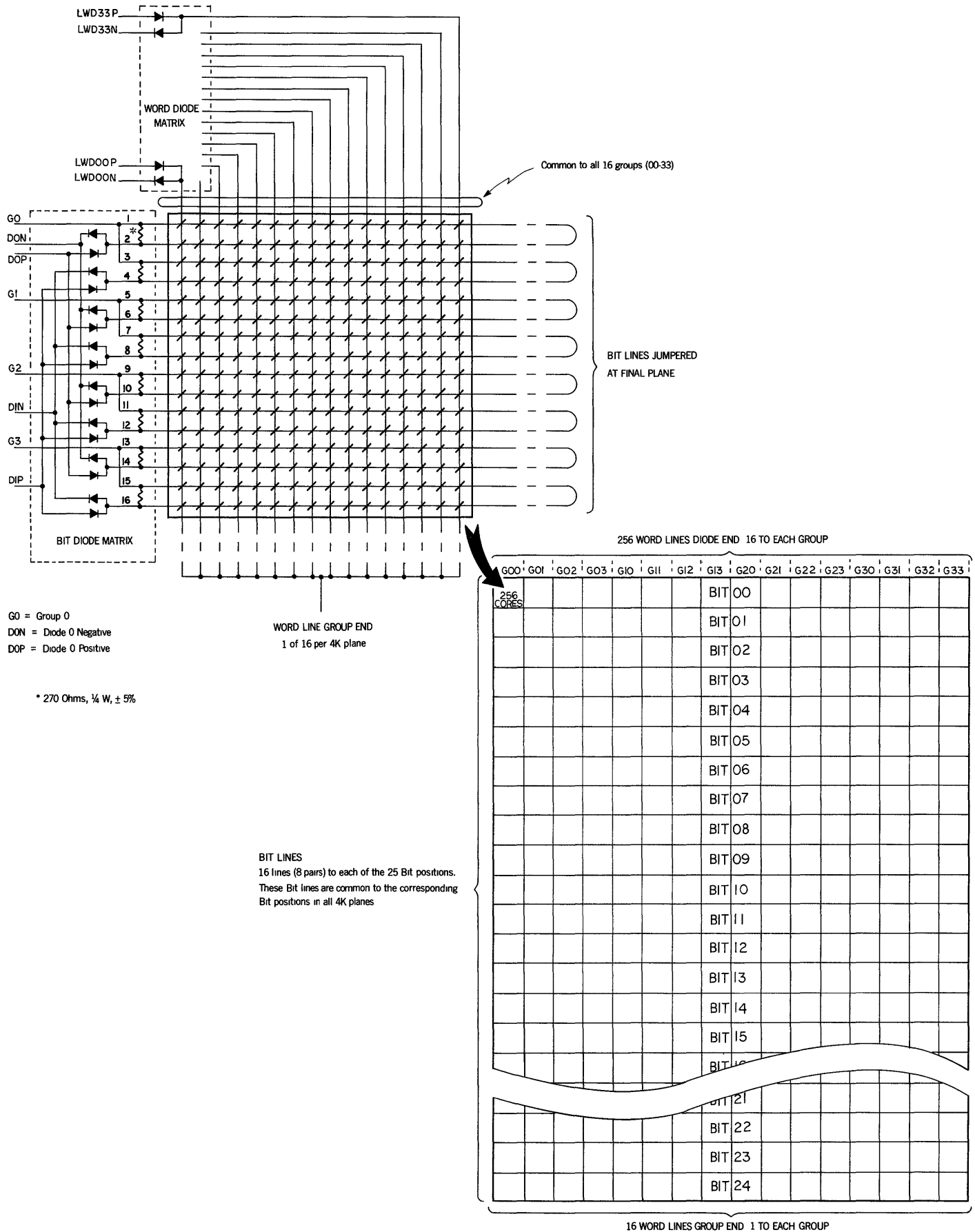


Fig. THEORY. 4. 4K Plane Organization and Addressing

A single sense line is threaded through all cores of each bit position in a 4K plane. Therefore, each plane will contain 25 sense lines. Each sense line will be threaded through 4096 core cells. The sense windings are threaded through the 16 by 256 bit matrix of each bit position in a rectangular fashion with wire transpositions spaced at 128 core intervals as shown in Fig. THEORY. 5.

### PRIORITY ACCESS

The Memory Control logic (68C972680, sheet 7) contains the logic required to enable access to memory of Channel 1 requests over AU, Channel 2, or Channel 3 requests; Channel 2 access to memory over AU or Channel 3 requests; and Channel 3 access to memory over AU requests. The Arithmetic Unit requests have the lowest priority for access to memory due to its operational speed. Channel 1, 2, and 3 requests are assigned as a system option for the devices (Drum, Disc, etc.) used by the system.

Requests from the user devices are applied to the Priority Request flip-flops F1MPR1 through F1MPR4. Request from the Arithmetic Unit (G1SMRQ) are applied to F1MPR4. With no other request present, or at the last clock pulse of a previous memory cycle, the priority flip-flop corresponding to the channel request input is set. Priority of the Priority Request flip-flops is then established in gates GOMBR1 through GOMBR4. Only one of these gates will be enabled according to the priority scheme.

$$GOMBR1 = F1MPR1$$

$$GOMBR2 = F1MPR2 \cdot \overline{F1MPR1}$$

$$GOMBR3 = F1MPR3 \cdot \overline{F1MPR2} \cdot \overline{F1MPR1}$$

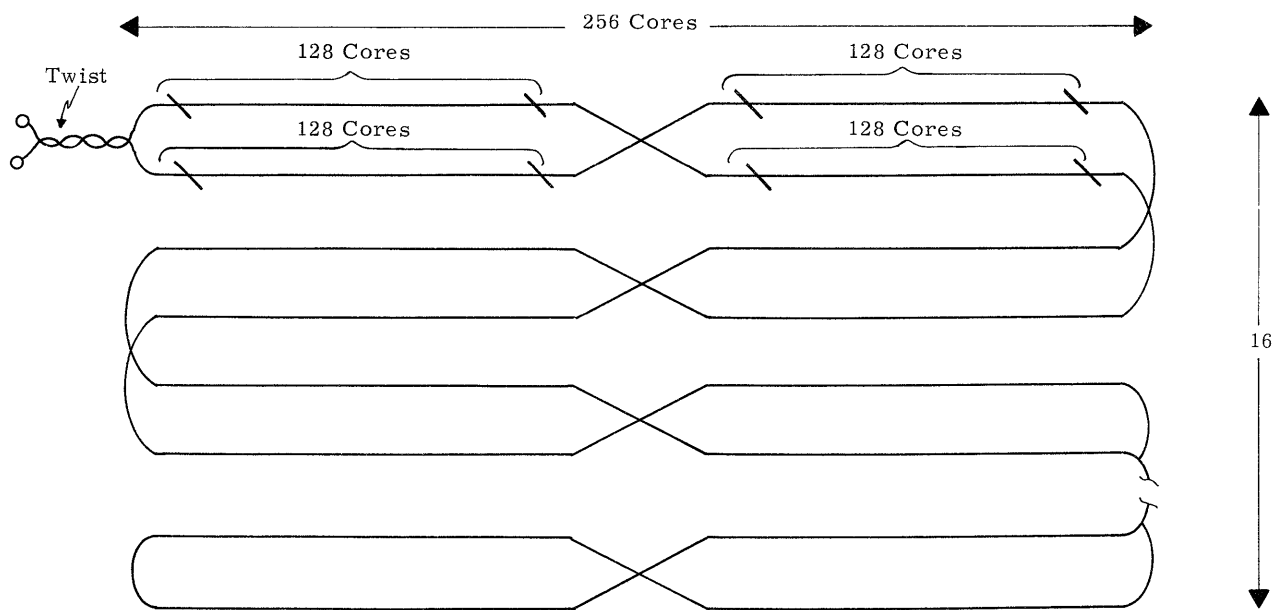
$$GOMBR4 = \frac{F1MPR4 \cdot \overline{F1MPR3} \cdot \overline{F1MPR2}}{F1MPR1}$$

Enabling one of these gates starts the memory cycle and enables access to memory for the corresponding device. Address and data control of the memory is then allocated to the user device granted access.

### MEMORY CONTROL TIMING

Fig. THEORY.6 contains a timing diagram with logic equations of the memory control timing (GE Drawing 68C972680). The memory control unit timing is initiated by a request signal from a user device and then operates independent of the memory unit timing. Fig. THEORY. 6 illustrates a request signal originating from the arithmetic unit (AUG1SMRQ), however, the request signal from other user devices would be similar. The memory control timing applies an initiate signal to the memory module to start memory timing and applies a signal to indicate the memory operation, either read (read/restore) or write (clear/write). The memory control unit then multiplexes the address to the memory module from the requesting device and transfers data from the user device to the memory module (write) or transfers data from the memory module to the user device (read).

The first clock pulse following the request signal is used to determine the priority of memory access. If no other requests are present or if the request is highest priority and the previous memory cycle is complete, the associated priority request flip-flop (F1MPR1, 2, 3, 4) is set and the associated busy return gate (D0MBR1, 2, 3, 4) is enabled. Enabling D0MBR1, 2, 3, 4 applies an access



NOTE: A sense winding threads 4096 cores of a bit position.

Fig. THEORY. 5. Sense Winding

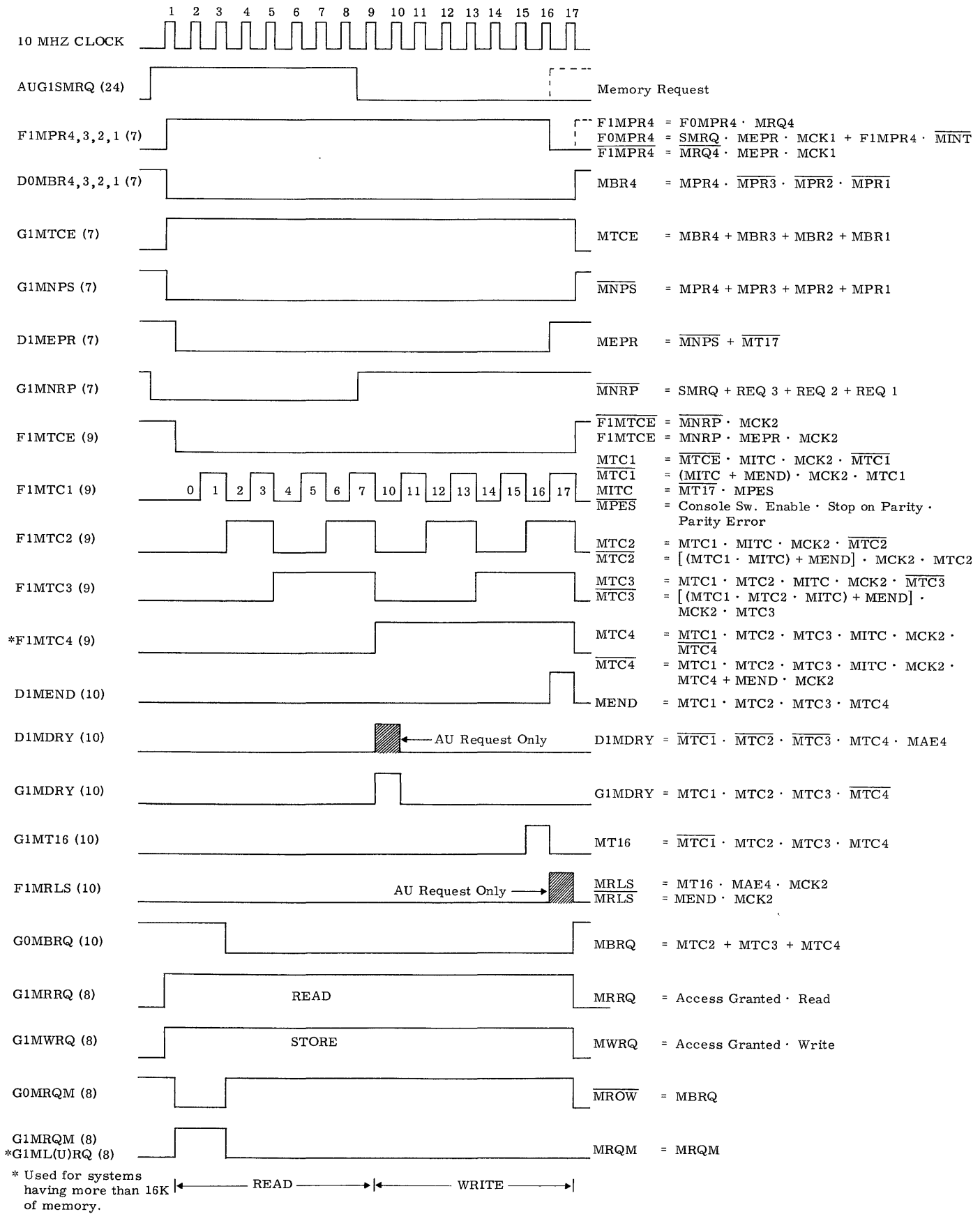


Fig. THEORY. 6. Memory Control Timing

granted signal to the user device, enables the time counter enable gate (G1MTCE) and enables the enable drivers (D1MAE1, 2, 3, 4 and D1MBAE1, 2, 3, 4). The enable drivers D1MA(B)E1, 2, 3, 4 are used to gate the memory address from the device granted access through the address multiplexer (G1MA00-G1MA14) to the address register of the memory unit. These enable drivers are also used to gate the data from the device granted access through the data multiplexer (G1MD00-G1MD23) to the data register of the Memory Module and to select the read/write signal from the device granted access.

The read/write signal from the device granted access will enable either the read request gate (G0MRRQ) or the write request gate (G0MWRQ). The outputs of these gates are applied to the memory module to provide control of the read/restore or clear/write mode of memory operation. The output of G0MRRQ is inverted in G1MRRQ and is used to enable the parity check logic within the memory control unit.

The memory time counter (F1MTC1, 2, 3, 4) is a binary counter incremented by the 10 mega-Hertz clock pulses to provide control throughout the memory cycle. The time counter is enabled following access granted, provided there is not a parity error when the Stop On Parity switch is in the stop position and provided there is not an over/under core temperature alarm.

An initiate signal is applied to the memory module (G1MRQM) for systems having 16K or less of memory (G1ML(U)Q for systems having more than 16K of memory) when access has been granted to memory. This signal is disabled (G0MBRQ) when the memory time counter is incremented to a count of two. Applying this initiate signal to the memory module starts the memory module timing and the read or clear portion of the memory cycle is started. The memory module then operates independent of the memory control timing. The initiate signal is applied only to the 16K stack addressed as controlled by bit 14 of the address (D1MEL(U)2).

After the memory time counter has been incremented to a count of 10g, a data ready signal is applied to the device granted access. This data ready signal indicates to the user device that the data has been read from memory and may be gated to the user or that the addressed core cell has been cleared.

The memory time counter continues to be incremented by each 10 mega-Hertz clock pulse until a count of 17g is reached. At this time the memory module has completed the write or restore portion of the cycle. Therefore, a new memory cycle may be initiated, i. e., another request may be serviced. If the arithmetic unit is the user device, a release signal (F1MRLS) is applied to the arithmetic unit to terminate the current sequence state.

## MEMORY MODULE TIMING

Fig. THEORY.7 illustrates the timing of the memory module as the signals would appear on an oscilloscope. The points at which these signals may be monitored are

indicated. Because of the operational speed of the memory unit, circuit delays become important when considering the memory timing. These delays are taken into account in the timing diagram.

These signals are generated from the initiate signal (G1MRQM, G1ML(U)RQ) generated in the memory control unit when access has been granted to a requesting user device. This initiate signal is ANDed with memory busy, buss level monitor, and the 10 mega-Hertz clock in G0TIMS. If the memory is not busy and the buss level monitor has not detected a power failure, the initiate memory cycle flip-flop (F1TIMC) is set. Setting F1TIMC initiates the memory cycle by applying a pulse to the time delay circuits and by setting the memory busy flip-flop (F1TMBU).

The timing pulse applied to the delay circuits provide precise timing signals for control during the memory cycle. Three delay circuits are provided as shown on sheet 8 of the memory logic (68C972952). Delay circuits 1 and 2 provide taps for obtaining delays of the timing pulse from 25 nanoseconds to 1650 nanoseconds in 25 nanosecond intervals. Delay circuit 3 is used to generate the strobe pulse providing more precise delay increments of 5 nanoseconds from the timing pulse previously delayed in delay circuit 1.

The timing signals obtained from these delay circuits are wired to the delay tap pins to provide optimum operation of the memory module. These delays should not require changing unless component deterioration or component replacement occurs.

Each of the control signals illustrated in Fig. THEORY.7 are described as to the function performed in the following text. A more detailed understanding of these signals will be obtained from the discussion of memory addressing, and the read/restore and clear/write modes of memory operation contained later in this section.

### G1MLRQ - Initiate:

The initiate signal originates in the memory control unit when access has been granted to a requesting device. This signal is applied from G1MRQM if the memory module contains 16K words or less of memory storage. If the system contains more than 16K words of memory, the initiate signal is applied from G1MLRQ for 0K - 16K addresses or from G1MURQ for address between 16K - 32K. In systems with more than 16K words of storage, the initiate signal is applied only to the addressed 16K memory module.

This initiate signal is used to start the memory timing by applying a pulse to the delay circuits from which the remaining timing signals are obtained.

### D0NCK1 - 10 Mega-Hertz Clock:

The 10 mega-Hertz clock is the basic system clock obtained from the Arithmetic Unit (D0NCK1).



SHEET

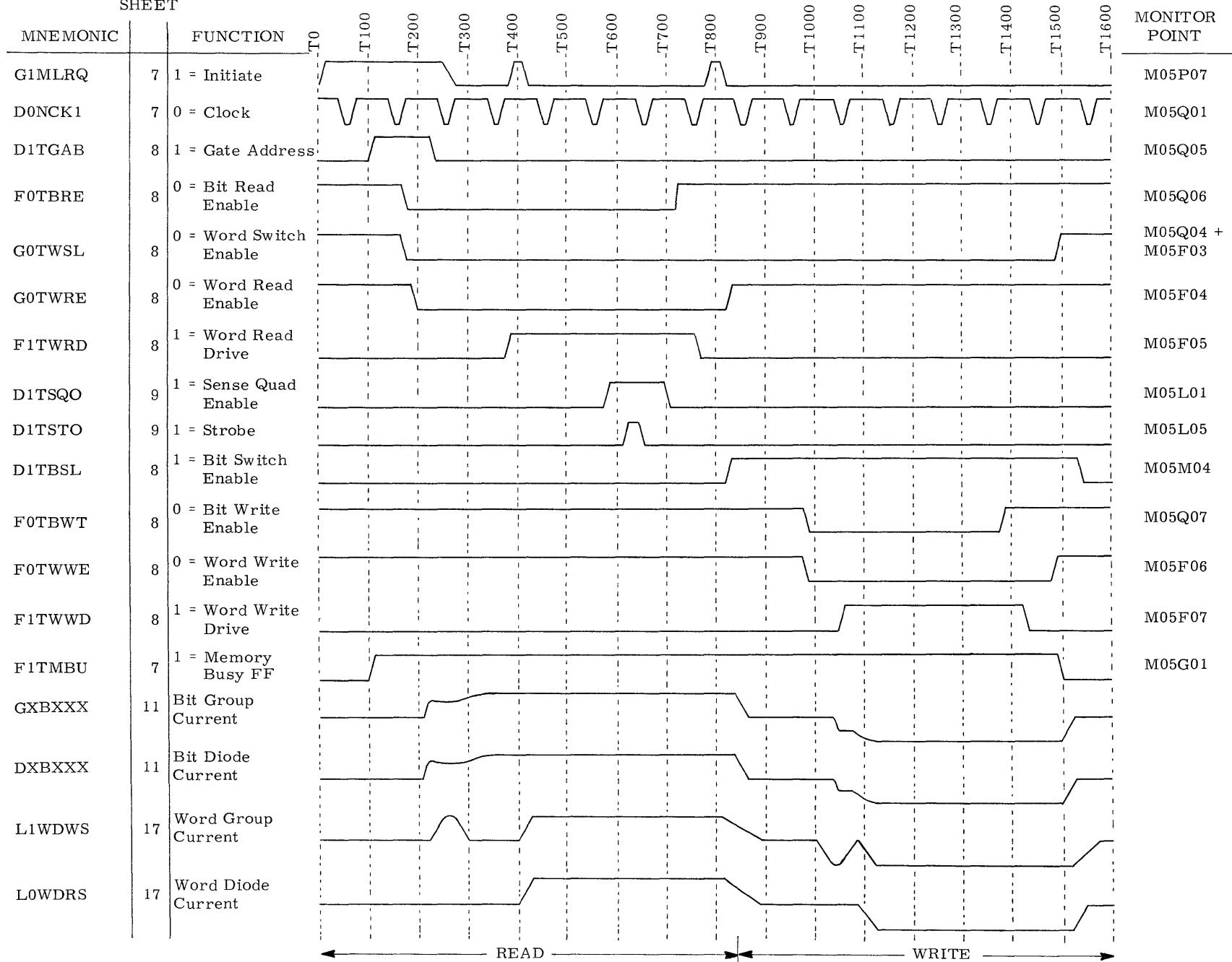


Fig. THEORY. 7. Memory Timing

D1TGAB - Gate Address, Bit:

This signal gates address bits 3 through 0 from memory control to the address register (F1AR03-F1AR00) of the memory module. These bits are decoded to select the bit drive lines.

At the same time that D1TGAB is enabled, D1TGAL and D1TGAU are enabled to gate address bits 4 through 12 to the lower and upper 8K memory address register, respectively, of the memory module. These bits are decoded to select the word drive lines. D1TGAL is also used to gate address bits 12 and 13 to the memory address register from the memory control unit. These bits are decoded to select the sense quad enable signal (D1TSQ0-3) corresponding to the 4K plane addressed.

F0TBRE - Bit Read Enable:

The Bit Read Enable signal enables decoding of address bits 3 through 0 to enable the addressed bit drive lines for the read or clear portion of the memory cycle. F0TBRE controls the time that current is allowed to flow through the addressed bit drive line of the read portion of the memory cycle.

G0TWSL - Word Switch Enable:

G0TWSL supplies one input to enable decoding of address bits 12 through 4 for selection of the addressed word drive lines. The other enable signal required for selecting the word drive lines is the word read enable or word write enable signal described below.

G0TWRE - Word Read Enable:

This signal enables selection of the word drive lines for the read or clear portion of the memory cycle.

F1TWRD - Word Read Drive:

F1TWRD enables the voltage source for the word drive lines. Therefore, current will flow through the addressed word drive lines. In conjunction with the current flow through the bit drive lines, full select current will flow through the addressed word and the core will be flipped to the zero state.

D1TSQ0-3 Sense Quad Enable:

This signal enables the sense amplifier to detect the change in flux for a core cell flipped from the one to zero state. Only the 4K plane addressed by address bits 12 and 13 will be enabled. The output from the enabled sense amplifier will be strobed to the memory data register if the operation is read.

D1TST0-3 Strobe:

This signal will gate data into the memory data register. The data gated will be the data sensed from the addressed core cell during a read/restore operation. During a clear/write operation, the data

gated to the memory data register will be the data from the user device that is multiplexed through the memory control unit. In either case, the data contained in the data register following the strobe will be the data to be written or restored in the memory location.

D1TBSL - Bit Switch Enable:

D1TBSL gates the contents of the memory data register to the bit matrix for controlling the bit drive lines during the write or restore portion of the memory cycle. Only the addressed bit lines corresponding to one bits in the memory data register will be enabled. This will flip the addressed cores corresponding to ones in the memory data register and leave the remaining cores in the zero state.

F0TBWT - Bit Write Enable:

F0TBWT enables the decoding of address bits 3 through 0 for selection of the bit drive lines for the write portion of the memory cycle.

F0TWWE - Word Write Enable:

F0TWWE enables decoding of address bits 12 through 0 for selection of the word drive lines for the write or restore portion of the memory cycle.

F1TWWD - Word Write Drive:

This signal enables the voltage source for the selected word drive lines and, therefore, allows current to flow through these lines. Therefore, with current flowing in the word drive lines and in the bit drive lines for those cells in which a one is to be stored, full select current flows in these cells flipping them to the one state.

F1TMBU - Memory Busy:

This signal inhibits initiation of a memory cycle until the present cycle is completed. In this manner, protection is provided from destroying desired data.

Current Signals:

These signals are provided to illustrate the current waveforms on the drive line.

## MEMORY ADDRESS SELECTION

The following text describes the word drive line and bit drive line selection logic used to enable full select current to flow through the core cells of the addressed memory word. Driving specific word and bit drive lines is required to read previously stored data from a specific memory location or to store new data in a specific memory location.

The word and bit drive lines driven, correspond to the memory address specified by the user device granted access to memory. As shown in Fig. THEORY. 8, the 15 bits of the memory address from the user device is

multiplexed through the memory control unit when access is granted. Bit 14 of the address is used by the memory control unit to determine which 16K stack is being addressed. If bit 14 is a one, D1MEU2 is enabled and the initiate memory sequence signal is applied to the upper 16K stack. If bit 14 is a zero, D1MEL2 is enabled and the initiate memory sequence signal is applied to the lower 16K stack. Since the initiate memory sequence signal is required to start a memory cycle, only the stack initiated will be operative.

Only the address bits required to address the maximum number of implemented memory word storage are wired to the multiplexer address gates of the memory control unit. In this manner, memory wrap around occurs when an address exceeding the implemented memory size is attempted. The Introduction portion of this description

lists the locations addressed when addresses exceeding the memory size are attempted.

From the multiplexer address gates of the memory control unit, address bits 13 through 0 are applied to the memory module for decoding. Fig. THEORY. 8 lists the control provided by each bit.

Refer to the memory block diagram contained on sheet 5 of the memory logic (68C872952) and to the 4K Plane Organization drawing, Fig. THEORY. 4, during the following discussion.

Bit 13 of the memory address is used to determine if the upper or lower 8K portion of a 16K stack is being addressed. Separate, identical decoding of each 8K portion of the stack is provided.

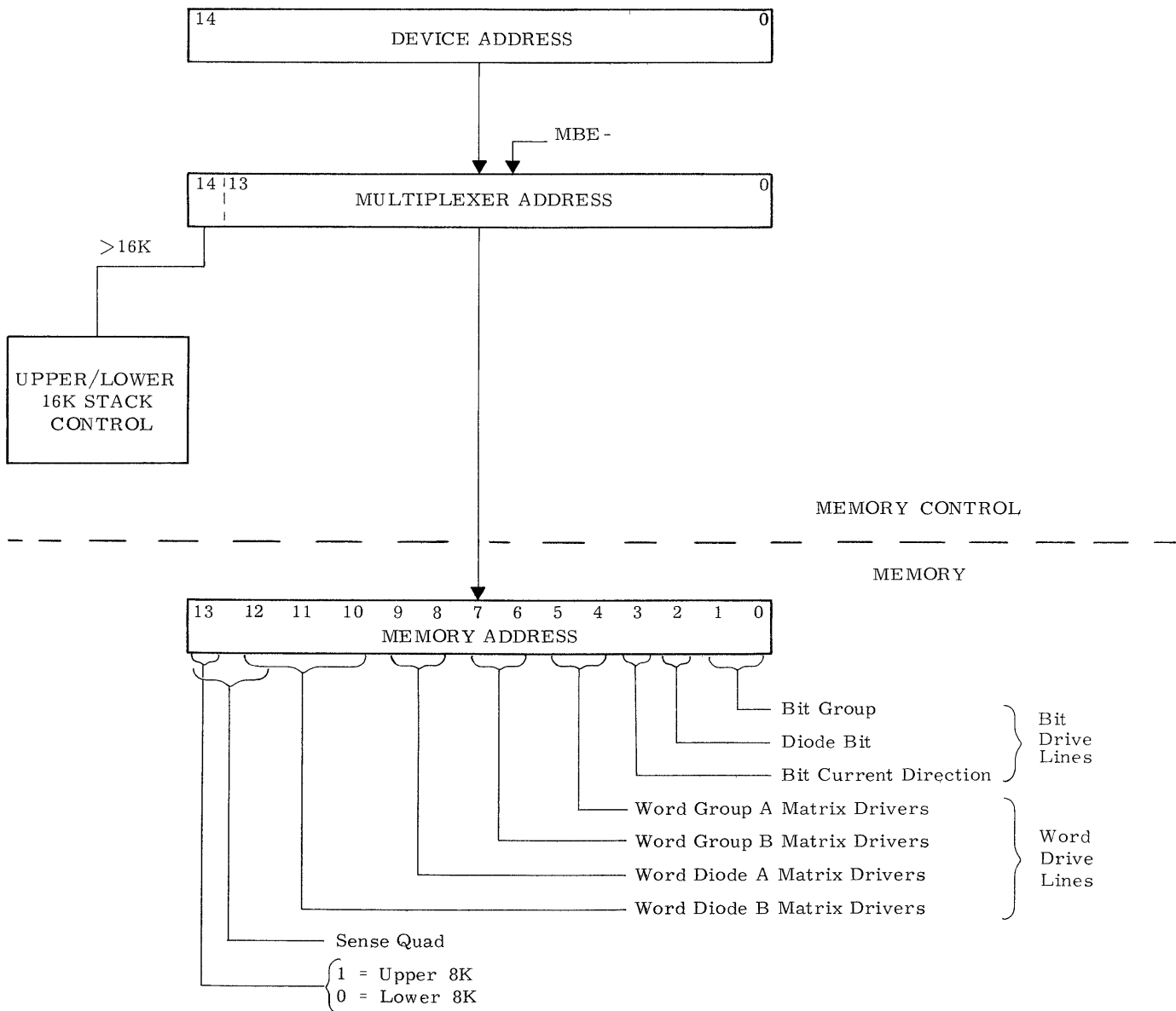


Fig. THEORY. 8. Memory Addressing

Bits 13 and 12 are used to determine which 4K plane is being addressed. From these bits one of four sense quad enable signals will be generated to gate the data sensed from core during the read portion of the cycle through the sense amplifiers. Since each bit position of each 4K plane contains a separate sense line, the sense quad enable signal enables the sense amplifier corresponding to the sense line of the addressed plane.

Bits 12 through 4 are decoded to enable the addressed word drive lines.

Bits 3 through 0 are decoded to enable the addressed bit drive lines. During the write (restore) portion of the memory cycle the addressed bit drive lines are also controlled by the data bits to be stored in core.

Fig. THEORY. 9 lists the word drive line circuits enabled for all combinations of address bits 12 through 4. Both the group end and diode end circuits enabled are shown. Table THEORY. 1 lists the bit drive line circuits enabled for all combinations of address bits 3 through 0. Notice that in addition to the read or write portion of the memory cycle, the bit drive line circuits enabled also depend on whether even or odd numbered bits are being selected. The bit drive circuits enable current to flow in opposite directions for odd and even bits. This arrangement aids in preventing interaction between adjacent bit positions.

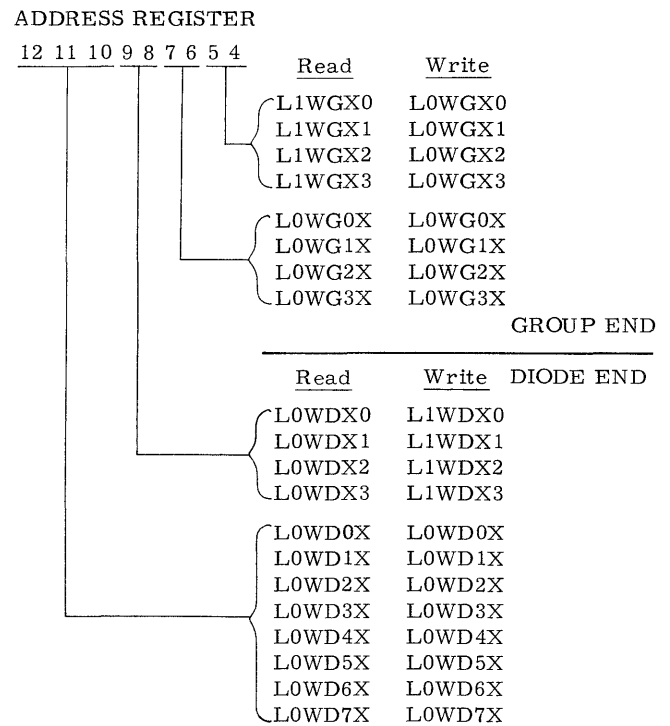


Fig. THEORY. 10 illustrates, in block diagram form, the address selection circuits affected for even and odd

Fig. THEORY. 9. Word Matrix Decode

A Register Bits				READ				WRITE			
				Even Bits		Odd Bits		Even Bits		Odd Bits	
3	2	1	0	Group End	Diode End	Group End	Diode End	Group End	Diode End	Group End	Diode End
0	0	0	0	G1B0XX	D0B0XX	G0B0XX	D1B0XX	G0B0XX	D1B0XX	G1B0XX	D0B0XX
0	0	0	1	G1B1XX	D0B0XX	G0B1XX	D1B0XX	G0B1XX	D1B0XX	G1B1XX	D0B0XX
0	0	1	0	G1B2XX	D0B0XX	G0B2XX	D1B0XX	G0B2XX	D1B0XX	G1B2XX	D0B0XX
0	0	1	1	G1B3XX	D0B0XX	G0B3XX	D1B0XX	G0B3XX	D1B0XX	G1B3XX	D0B0XX
0	1	0	0	G1B0XX	D0B1XX	G0B0XX	D1B1XX	G0B0XX	D1B1XX	G1B0XX	D0B1XX
0	1	0	1	G1B1XX	D0B1XX	G0B1XX	D1B1XX	G0B1XX	D1B1XX	G1B1XX	D0B1XX
0	1	1	0	G1B2XX	D0B1XX	G0B2XX	D1B1XX	G0B2XX	D1B1XX	G1B2XX	D0B1XX
0	1	1	1	G1B3XX	D0B1XX	G0B3XX	D1B1XX	G0B3XX	D1B1XX	G1B3XX	D0B1XX
1	0	0	0	G0B0XX	D1B0XX	G1B0XX	D0B0XX	G1B0XX	D0B0XX	G0B0XX	D1B0XX
1	0	0	1	G0B1XX	D1B0XX	G1B1XX	D0B0XX	G1B1XX	D0B0XX	G0B1XX	D1B0XX
1	0	1	0	G0B2XX	D1B0XX	G1B2XX	D0B0XX	G1B2XX	D0B0XX	G0B2XX	D1B0XX
1	0	1	1	G0B3XX	D1B0XX	G1B3XX	D0B0XX	G1B3XX	D0B0XX	G0B3XX	D1B0XX
1	1	0	0	G0B0XX	D1B1XX	G1B0XX	D0B1XX	G1B0XX	D0B1XX	G0B0XX	D1B1XX
1	1	0	1	G0B1XX	D1B1XX	G1B1XX	D0B1XX	G1B1XX	D0B1XX	G0B1XX	D1B1XX
1	1	1	0	G0B2XX	D1B1XX	G1B2XX	D0B1XX	G1B2XX	D0B1XX	G0B2XX	D1B1XX
1	1	1	1	G0B3XX	D1B1XX	G1B3XX	D0B1XX	G1B3XX	D0B1XX	G0B3XX	D1B1XX

Table THEORY. 1. Bit Matrix Decode

13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADDRESS REGISTER

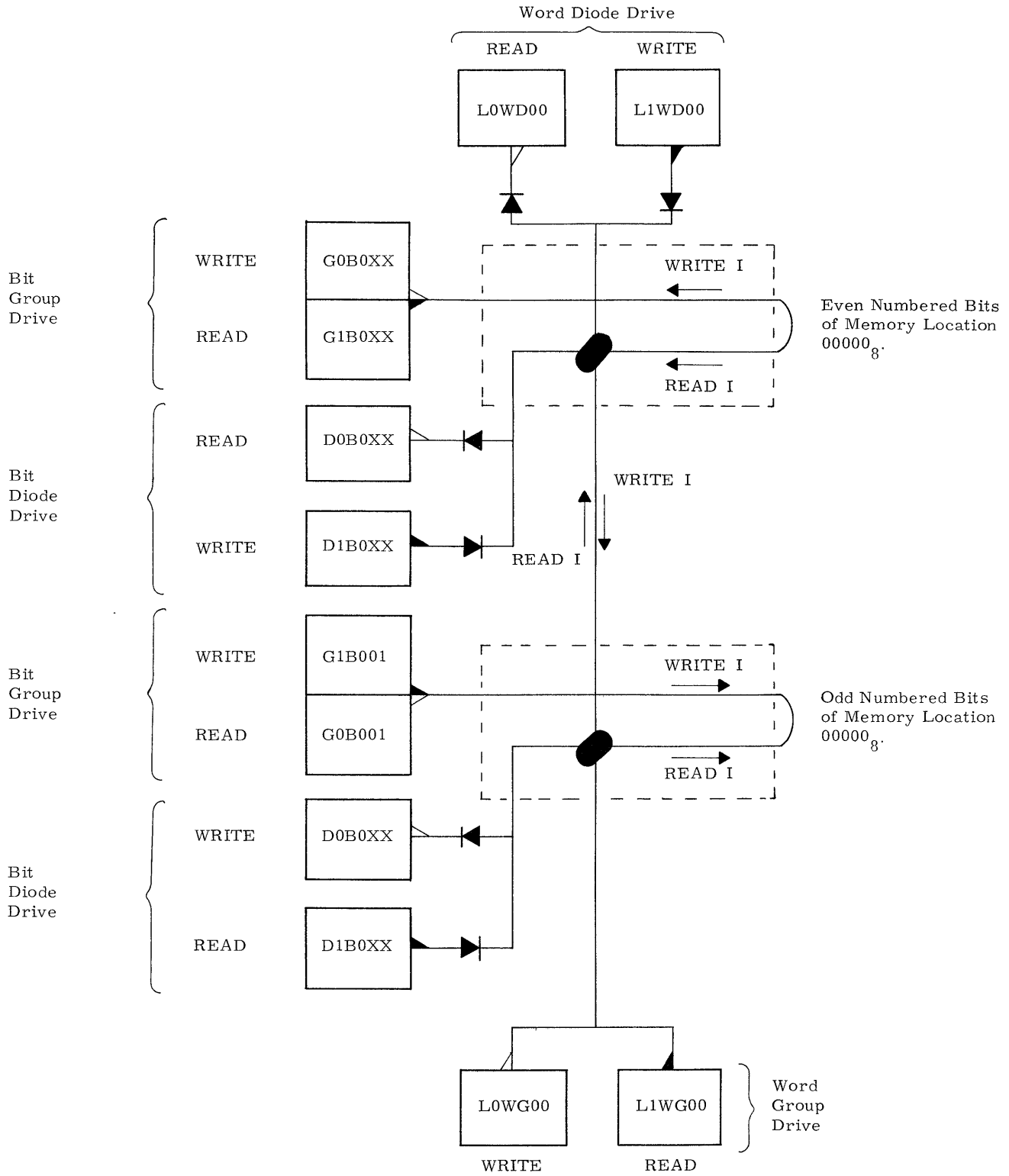


Fig. THEORY. 10. Address Selection Block Diagram

bits of memory location 00000<sub>8</sub>. This diagram illustrates the direction of current flow and the corresponding circuits enabled for both the read and write portions of the memory cycle.

Notice that current flowing in both the bit and word drive lines provide full select current through the cores shown. During the read portion of the memory cycle, current flows in a direction that will flip cores in the one state to the zero state. The resultant flux is induced in the sense line, amplified by the sense amplifier and if the memory operation is read/restore, a one is strobed into the memory data register. If the memory operation is clear/write, this data bit will not be strobed into the memory data register. Instead, the data from the user device is gated to the memory data register. In either case, the read portion of the memory cycle will place all bits of the addressed memory location in the zero state.

Following the read portion of the memory cycle, the write portion of the cycle is entered. Current is enabled in the opposite direction during the write portion of the cycle to flip the cores, corresponding to one bits in the data register, to the one state. Only the bit drive lines corresponding to one bits in the memory data register are enabled.

Figs. THEORY. 11 and THEORY. 12 illustrate the detailed circuit operation of the read and write cycle, respectively, for the bit and word drive lines associated with memory address 00000<sub>8</sub>, bit 0. These circuit drawings illustrate the on or off condition of the various transistors, the relative potential, plus or minus, at the output of the circuit areas, and the direction of current flow. Using these diagrams, the operation of any word and bit drive line circuits may be determined.

Appendix A contains a list of the pin connections at the core stack for both the word and bit drive lines.

## READ/RESTORE OPERATION

The read/restore mode of memory operation retrieves data previously stored in a memory location and restores the same data back in the addressed memory location. The read/restore mode is initiated when a user device has been granted access to memory and the read (store) mode is specified by the device.

The read/write or store signal from the user device is at zero volts for a read operation. This read/write signal is applied to the memory control unit where GOMRRQ is enabled and GOMWRQ is disabled for the read/restore mode. GOMRRQ enables the parity check function of the parity check-generate logic. GOMRRQ applies a "one" to the memory unit enabling the read cycle enable gate D1TRCE. D1TRCE enables the data sensed from core to be gated to the memory data register for transfer to the user device and for control of the bit drive lines during the write portion of the memory cycle.

Fig. THEORY. 13 illustrates the data flow for the read/restore memory operation. During the read portion of the memory cycle, the bit and word drive lines are

enabled according to the memory address specified by the user device. These drive lines provide full select current to flip the addressed core cells that are in the one state to the zero state. Flipping these cores to the zero state induces a voltage into the sense lines. The sense quad enable signal (D1TSQ<sub>1</sub>) is enabled corresponding to the 4K plane addressed. This sense quad enable signal gates the sense winding signal for each bit through the sense amplifiers. The sense amplifiers convert the sense winding signals to logic levels. From the sense amplifiers, the data read from core is strobed into the memory data register since this is a read/restore operation (i. e., D1TRCE is enabled).

The data contained by the memory data register is then used to control the bit drive lines during the write portion of the memory cycle to restore the contents of the memory location addressed. Bits 23 through 0 of the memory data register are also transferred to the user device. Transfer to the user device occurs through the memory input buffer of the memory control unit if the system contains more than 16K words of memory or direct to the user device if less than 16K words of memory are contained by the system. The data ready signal generated within the memory control unit is used by the user device to gate the data to the user device input/output data register. Bit 24 of the memory data register is applied to the parity check circuitry of the memory control unit.

The data gated to the user input/output register is applied back through the multiplexer (or direct from the B Register in systems not requiring the multiplexer) to the parity check logic. In this manner, transmission parity is also checked. A detailed discussion of parity checking is provided later in this section.

As mentioned above, the data in the memory data register is used to control the bit drive lines during the write portion of the memory cycle. If a one is to be stored into a core cell, full select current is passed through the cell in a direction opposite to that of the read portion of the cycle. If the core is to remain in the zero state, no current flows in the bit drive lines, only one-half select current flows through the core from the word drive lines, and the core remains in the zero state.

## CLEAR/WRITE OPERATION

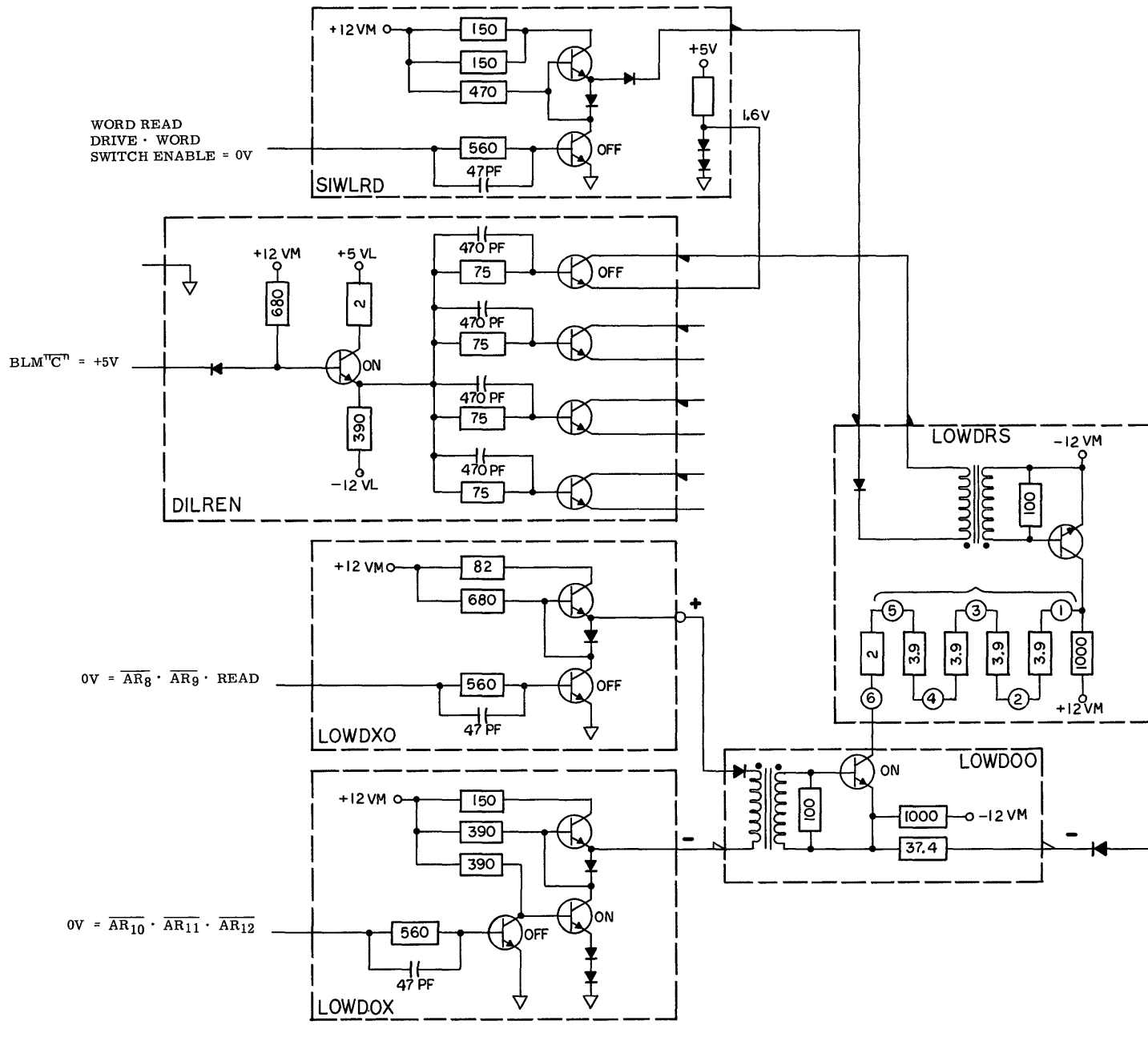
The clear/write mode of memory operation stores new data in a memory location address. The clear/write mode is enabled when the user device granted access to memory specifies the write or store mode of operation.

The clear/write mode of memory operation is very similar to that of the read/restore mode of operation. The basic differences are that the parity generate logic is enabled in the clear/write mode and the data sensed from core during the read portion of the memory cycle is not strobed to the data register in the clear/write mode. Instead, the data to be stored from the user device is strobed to the data register.

The user device specifies the clear/write mode by applying a binary one read/write or store signal to the

ADDRESS REGISTER

13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0



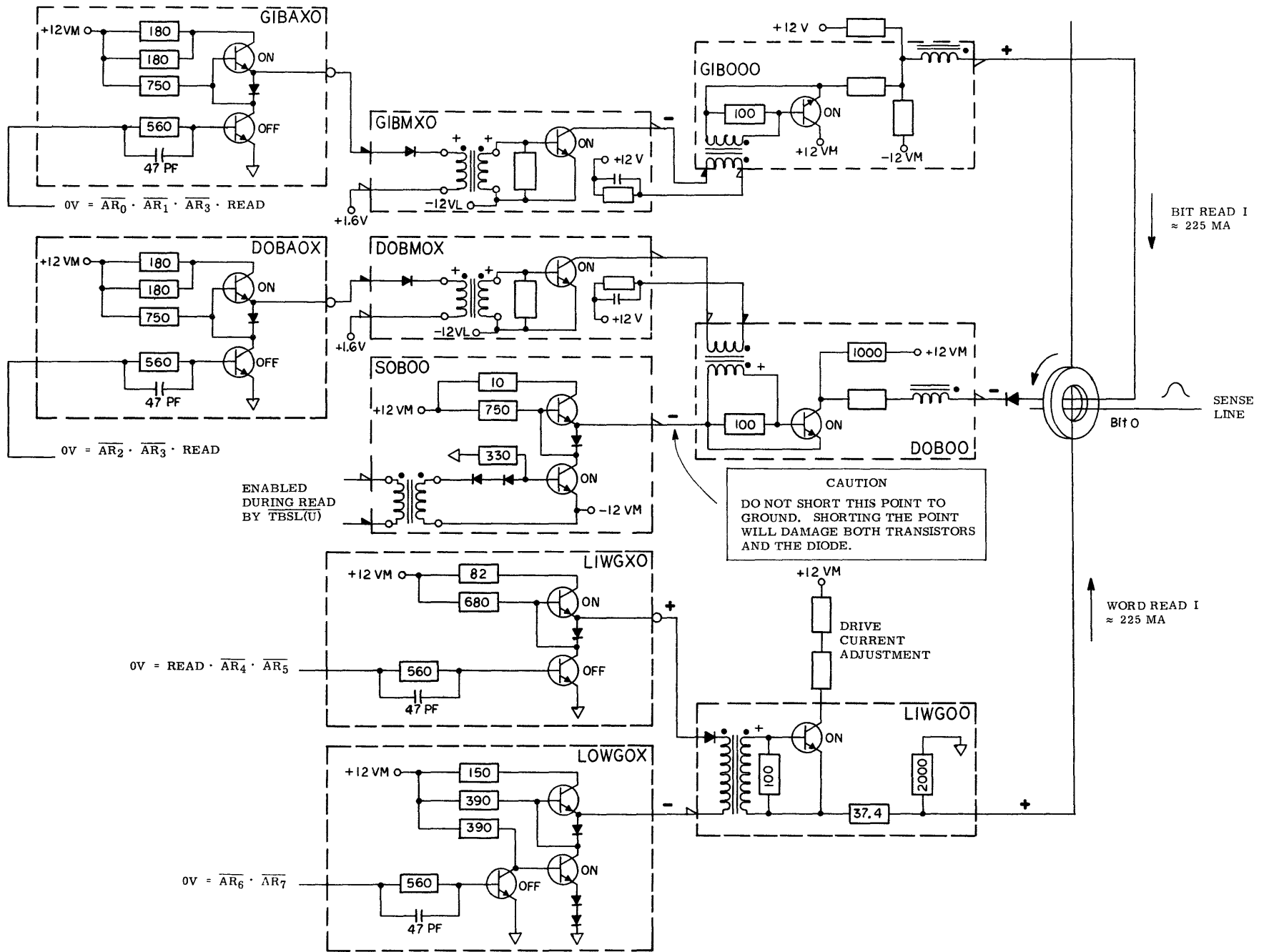
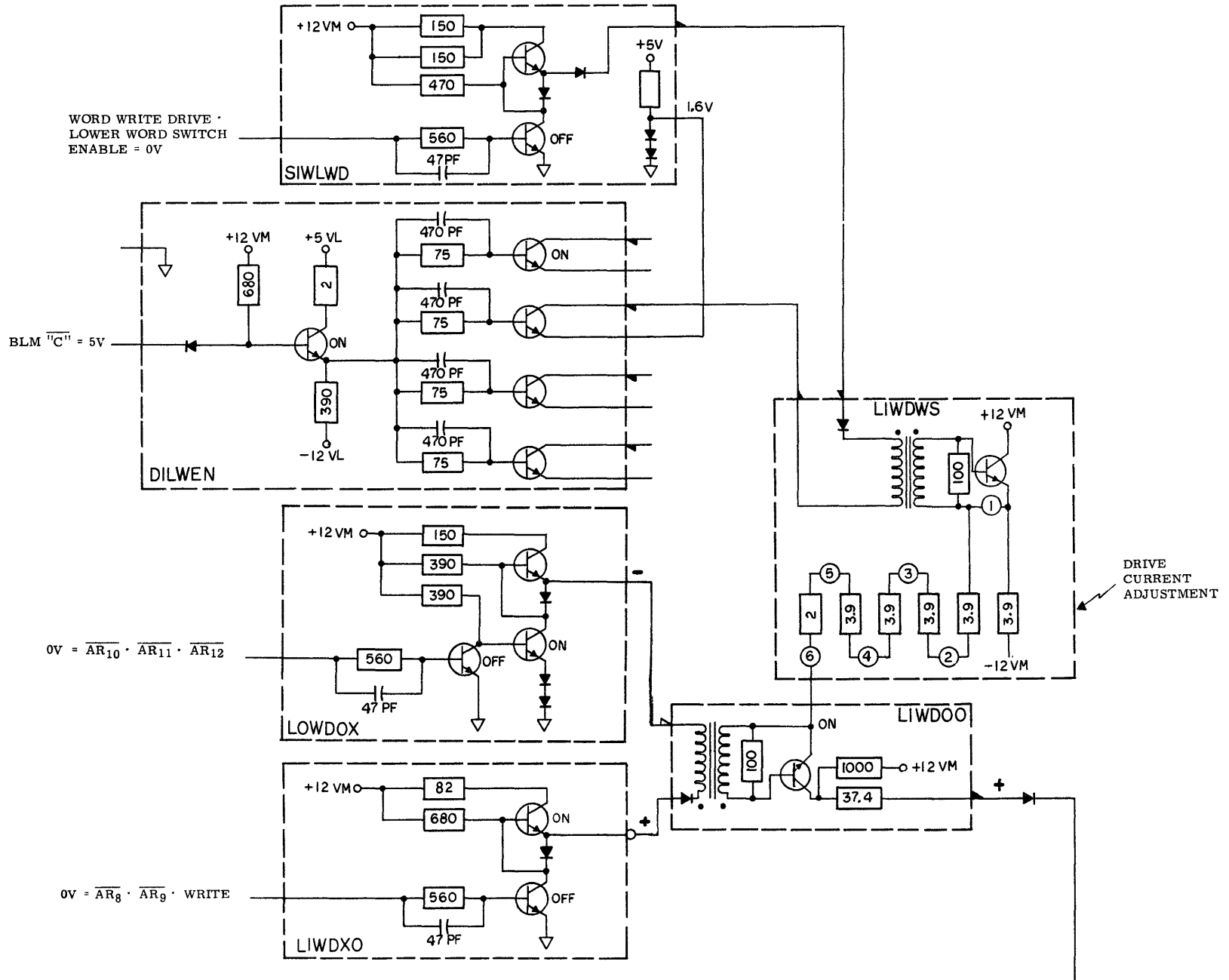


Fig. THEORY. 11. Read Drive Circuit



ADDRESS REGISTER	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0



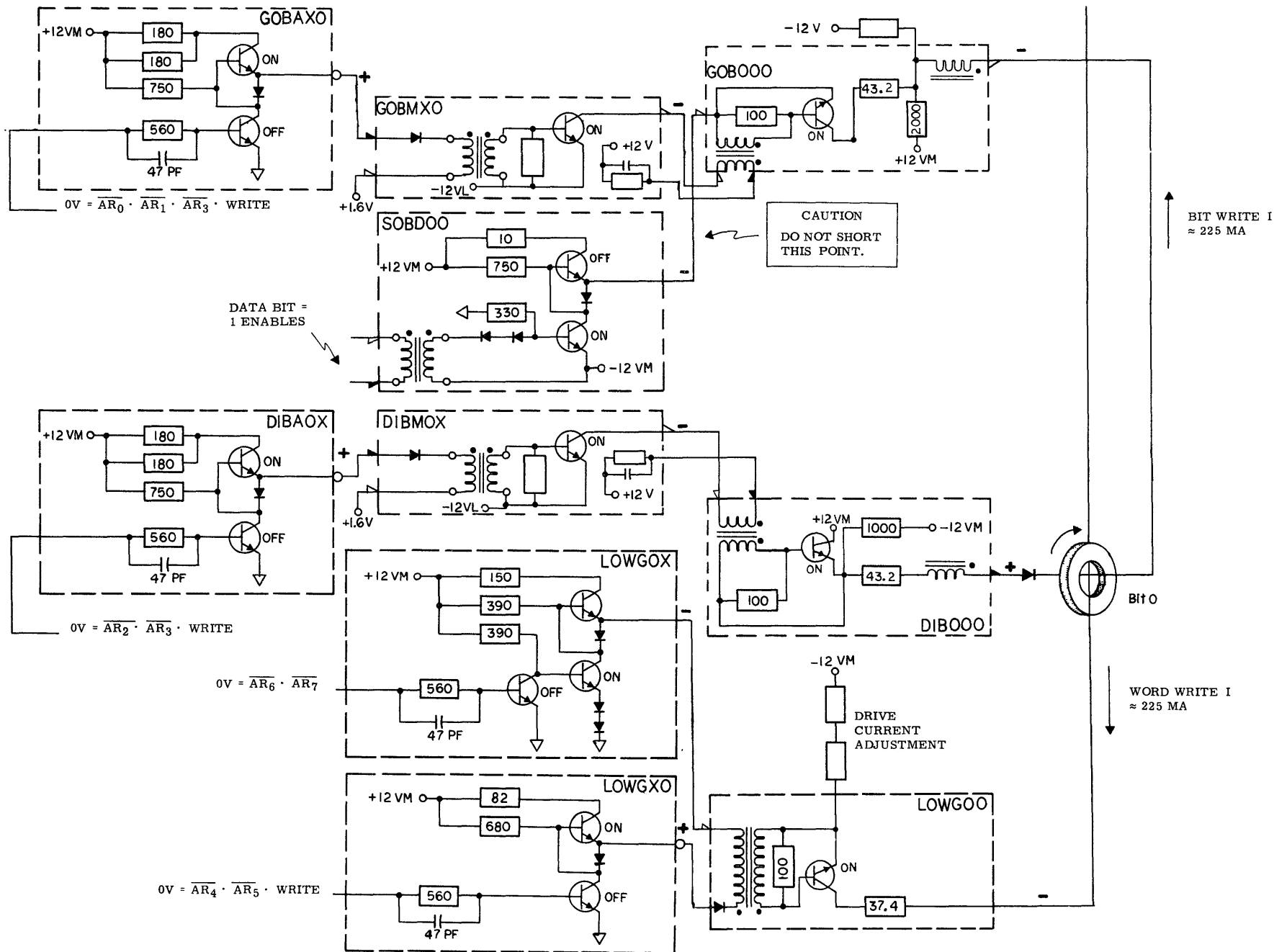


Fig. THEORY. 12. Write Drive Circuit

memory control unit. This signal enables G0MWRQ and inhibits G0MRRQ. G0MRRQ, when inhibited, enables the parity generation portion of the parity check-generate

logic. G0MWRQ, when enabled, enables the write cycle enable gate (D1TWCE) in the memory module. This signal is used to gate the data from the user device to the

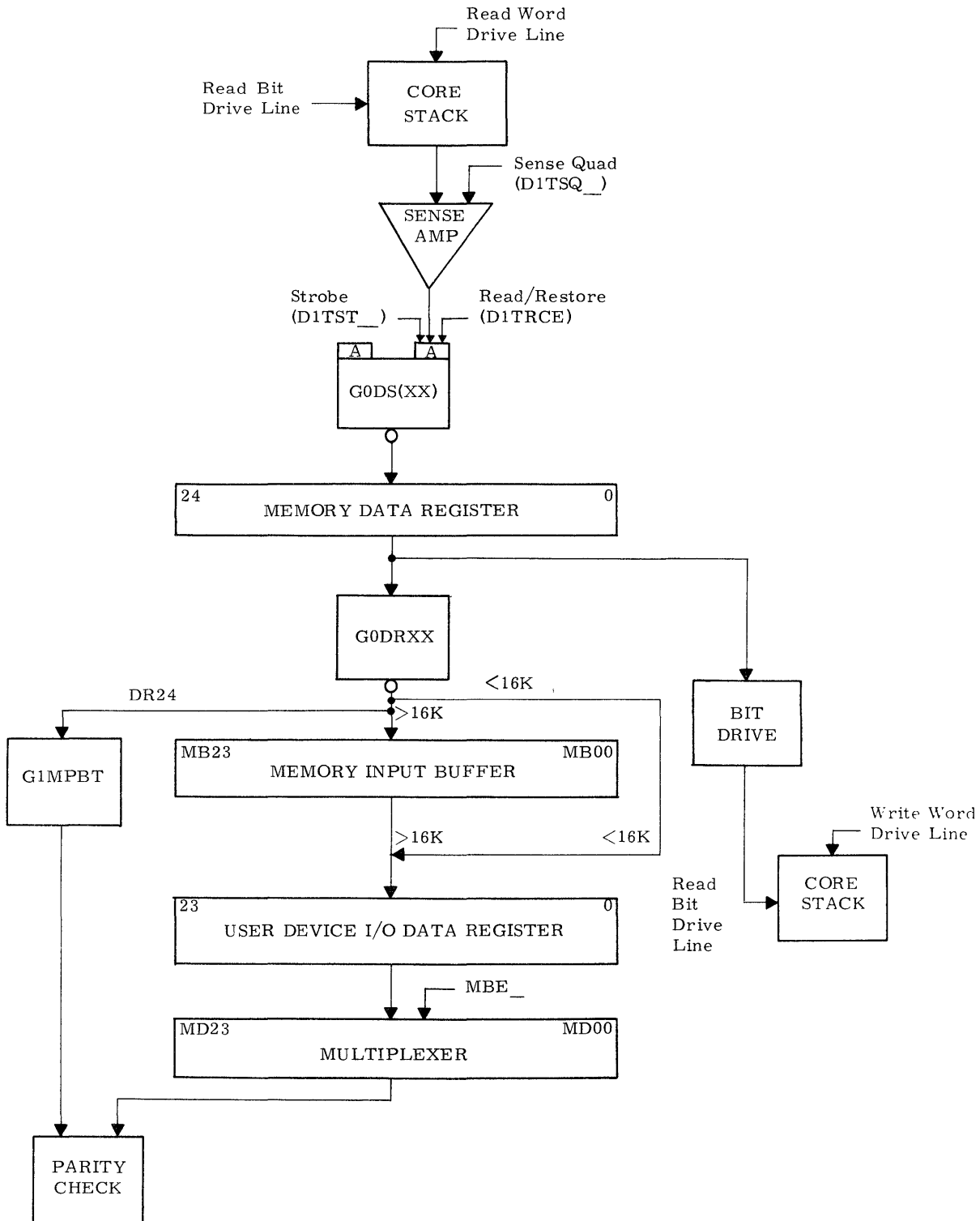


Fig. THEORY. 13. Read/Restore Block Diagram

memory data register for storage in the addressed memory cells during the write portion of the cycle.

(clear) portion of the memory cycle, the bit and word drive lines are enabled according to the memory address from the device. All cores in the one state are flipped to the zero state, however, the resultant signals induced in the sense windings are not gated to the

Fig. THEORY. 14 illustrates the data flow for the clear/write mode of memory operation. During the read

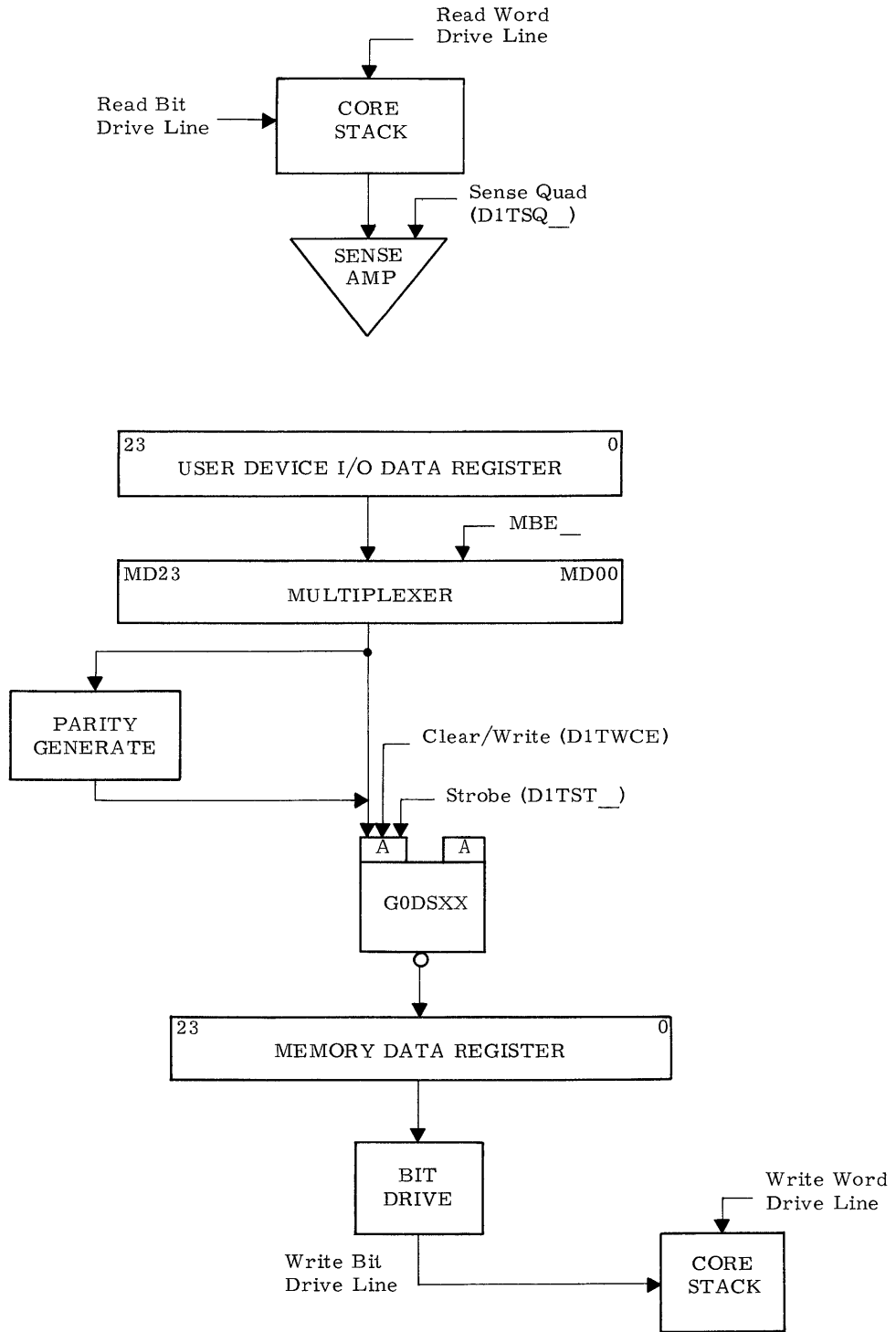


Fig. THEORY. 14. Clear/Write Block Diagram

memory data register. Therefore, the read (clear) portion of the memory cycle is used to place all addressed cores in the zero state.

The data to be stored (written) in the addressed core location is gated through the multiplexer of the memory control unit to the data register and to the parity generation logic. The parity generation logic then generates a one, if required for odd parity, and applies this one to bit position 24 of the data register. The parity and data bits are then strobed in the memory data register. The contents of the memory data register are then used to control the bit drive lines during the write portion of the memory cycle. A one in the data register will enable current through the corresponding bit drive line. This current through the bit drive line, in conjunction with the current through the word drive line, will flip the core to the one state. A zero in the data register will inhibit current flow in the bit drive line. The half-select current through the word drive line will not be sufficient to flip the core and it will remain in the zero state. In this manner the cores of the addressed memory location will be placed in the configuration of the data bits in the memory data register.

## PARITY GENERATION

Parity generation, in conjunction with parity checking, provides a means of detecting if a stored data word has gained or lost a bit. The parity generation logic ensures that an odd number of "one" bits are stored in memory during any clear/write operation. The parity checking logic ensures that an odd number of "one" bits are read back for each data word during read/restore operation. If an odd number of "one" bits are not read back, a parity error signal is provided to light the Parity Error and Alarm indicators on the computer console, enable the running program to detect the parity error (JNP), and to optional inhibit further access to memory by the Arithmetic Unit (Stop on Parity switch). If a parity error occurs when the core temperature is out of limits, memory operation is halted.

As shown in Fig. THEORY. 14, data bits from the multiplexer are applied to the parity generate logic of the memory control unit during the clear/write mode of operation. The parity generate logic then enables a one to bit 24 of the memory data register if an even number of one bits is contained by the 24 data bits.

Fig. THEORY. 15 illustrates, in block diagram form, the logic used to generate the parity bit. The logic associated with parity generation is contained on sheets 17 through 22 of the memory control logic (68C972680). Most of this logic is also used in parity checking as described immediately following this discussion of parity generation.

The data bits to be stored in memory are gated from the user device through the multiplexer by D1MAE1, 2, 3, or 4 when memory access has been granted. These data bits are applied to the memory data register as well as the inputs of the parity generation logic (G0POG1 through G0POG8).

G0POG1 through G0POG8 each sample three data bits and if an odd number of "ones" are contained by these three data bits, a "zero" output signal is provided. If an even number of "one" data bits are contained by the three inputs, a "one" output is provided. The outputs of G0POG1 through G0POG8 are applied to G0POS1, G1POS2, and G0POS3. The output from these gates are applied to G1POFS. The output from G1POFS is then a "one" if an odd number of data bits was detected from the multiplexer or a "zero" if an even number of "one" data bits was detected from the multiplexer.

The output of G1POFS is inverted in G0POFS providing a "one" output to bit 24 of the memory data register if an even number of "one" bits is contained by the data to be stored in memory. This parity bit is gated to the memory data register by the strobe at the same time the remaining data bits from the multiplexer are gated to the memory data register.

## PARITY CHECKING

During a read/restore operation, the data read from the addressed core cell is checked to determine if it contains an odd number of "one" data bits. Since the parity generation logic ensured that an odd number of "one" bits was stored in the memory location, if an even number of "ones" is read back, a bit has been gained or lost and an error condition exists. When an error condition is detected, the Parity Error flip-flop (F1PPER) is set and the data word is restored in the memory location in the same form that it was read.

Setting the Parity Error flip-flop performs the following functions:

- Applies a signal to the Arithmetic Unit (AUGONPER) where the status may be checked by the JNP command.
- Applies a signal to the Stop On Parity switch (CSWSOP) to stop Arithmetic Unit sequencing when a parity error occurs and the Stop On Parity switch is in the Stop position.
- Sets F1PPED to light the Core Parity and Alarm indicators on the Computer Console.
- If the core temperature is out of limits, memory operation is halted.

The Parity Error flip-flop may be cleared by a JNP command, the Clear Alarm switch on the Computer Console, or by Initializing the system. The only method of clearing F1PPED is by pressing the Clear Alarm switch on the Computer Console. Therefore, if a parity error occurs, the indicators on the Computer Console will remain lighted until the Clear Alarm switch is pressed.

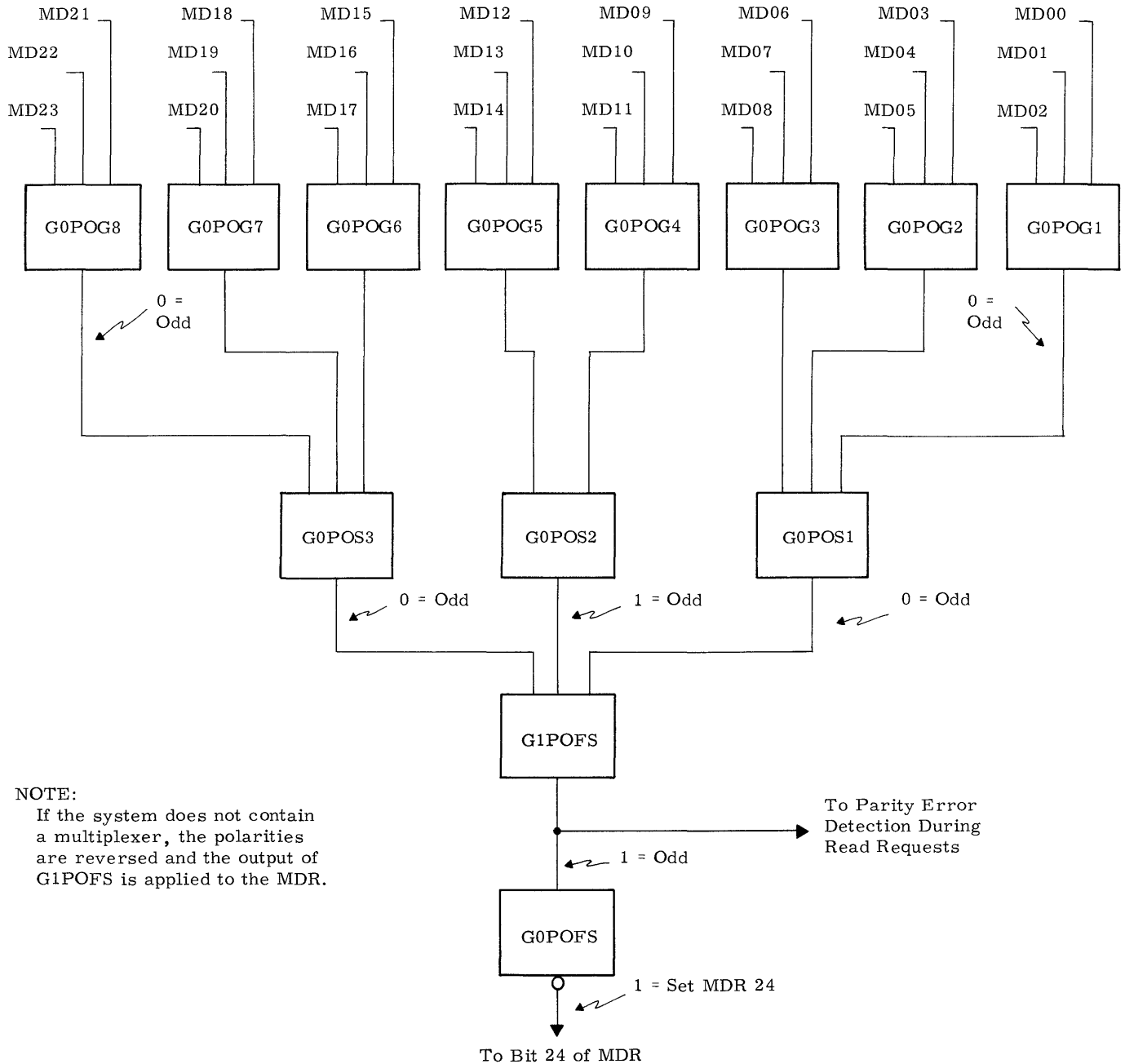
As shown in Fig. THEORY. 13, the data obtained from memory during a read/restore operation is gated to the user device and from the user device back to the multiplexer of the memory control unit. From the multiplexer, this data is applied to the parity check logic.

Fig. THEORY. 16 illustrates the logic associated with parity checking. This logic compares the status of the parity bit read back from memory with the status of G1POFS and G0POFS. As described in the parity generation discussion, G1POFS and G0POFS are enabled when the 24 bit data word contains an odd number of "one" bits.

G1MPBT enables the parity bit read from the addressed memory cell depending upon which stack was addressed. This gate is enabled (G0MIAD) when memory wrap

around causes all zeros to be read from memory as listed in the Introduction. Enabling G1MPBT in this manner simulates a parity bit, preventing a parity error indication.

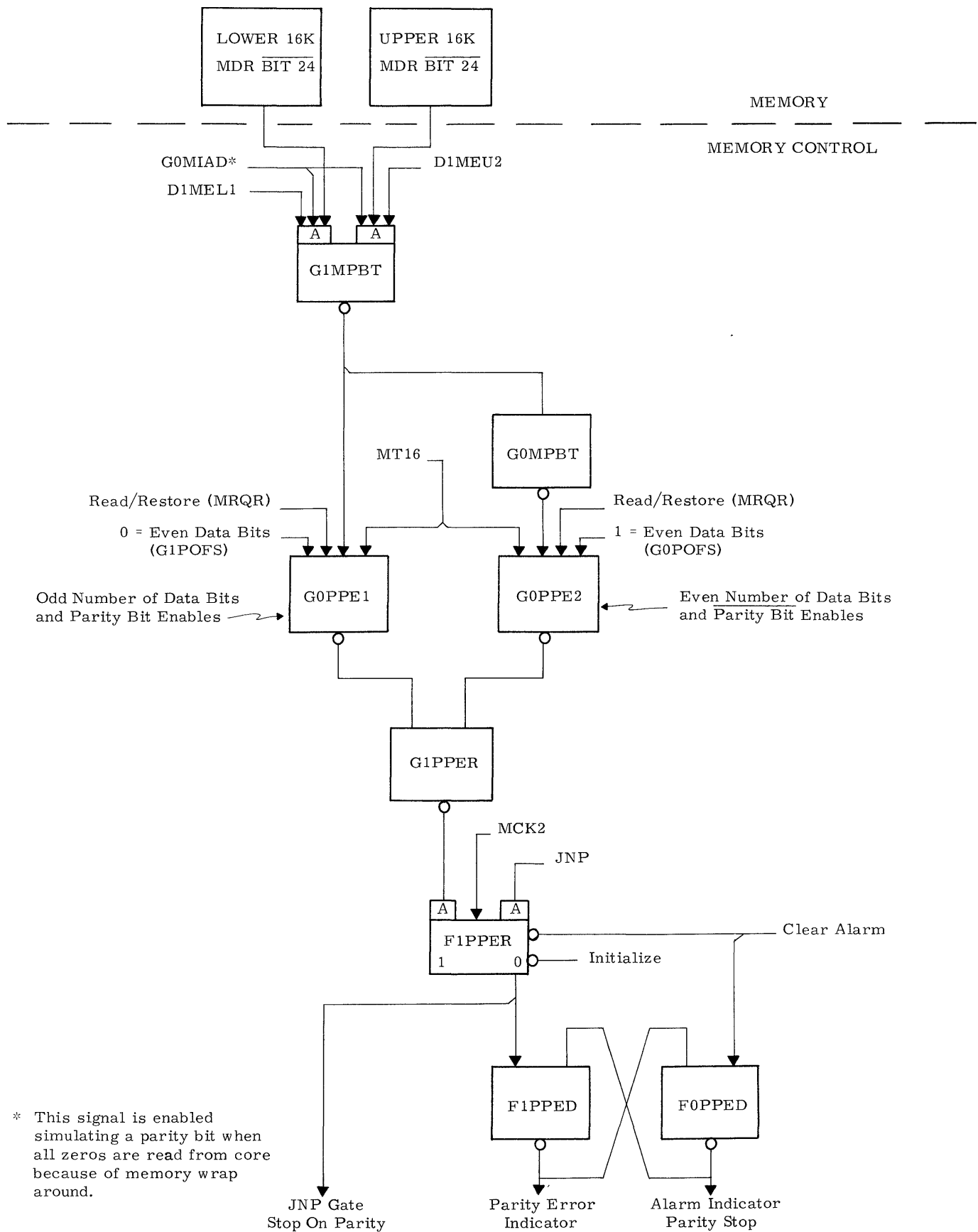
The status of the parity bit is compared with the status of G1POFS and G0POFS in G0PPE1 and G0PPE2 at memory control time  $16_8$  when in the read/restore mode. G0PPE1 is enabled if an odd number of data bits are contained by the word read from memory and the parity bit is a one. G0PPE2 is enabled if an even number of



NOTE:

If the system does not contain a multiplexer, the polarities are reversed and the output of G1POFS is applied to the MDR.

Fig. THEORY. 15. Parity Generate



\* This signal is enabled simulating a parity bit when all zeros are read from core because of memory wrap around.

Fig. THEORY. 16. Parity Check

data bits are contained by the word read from memory and the parity bit is a zero.

Enabling either G0PPE1 or G0PPE2 will cause the Parity Error flip-flop (F1PPER) to set at the next clock pulse. As previously described setting F1PPER applies a signal to the Arithmetic Unit for JNP checking and to the Stop On Parity switch to half sequencing within the Arithmetic Unit. Setting F1PPER also sets F1PPED to light the Console indicators. As a system option, the Parity Error flip-flops may be used to light other indicators or enable other system test gates.

$$F1PPER = PPER \cdot MCK2$$

$$G1PPER = PPE1 + PPE2.$$

The Parity Error flip-flop may be reset by executing a JNP command, pressing the Clear Alarm switch, or Initializing the system.

$$\overline{F1PPER} = (\overline{TT5E} \cdot \overline{NCPE}) + PPEC + MINT.$$

F1PPED is cleared by pressing the Clear Alarm switch. Clearing F1PPED extinguishes the Console indicators.

$$\overline{F1PPED} = PPEC.$$

## TRANSMISSION PARITY ERROR CHECKING

If a "one" data bit is lost or gained while receiving data from the Model 4511 Peripheral Channel Multiplexer during a write operation, a parity error signal will be applied back to the 4511. The state of the parity bit from the 4511 is compared with the status of G1POFS in G0PERD. G1POFS, as previously described, provides a "one" output when the 24 data bits contain an odd number of "one" bits. G0PERD is enabled, signifying a transmission parity error, when an odd number of "one" bits is contained by the data word and the parity bit is a "one" or when an even number of "one" bits is contained by the data word and the parity bit is a "zero". The output of G0PERD is applied back to the 4511. Refer to the 4511 description for the effect of this parity error signal.

$$G0PERD = \frac{POFS \cdot \overline{PEDP} \cdot \overline{MRRQ}}{\overline{MRRQ} \cdot \text{Parity Bit}} + \overline{POFS}$$

## CORE HEATER

The core stack is an enclosed unit maintained at approximately 55°C by heating elements within the stack. Over and under thermorelays are provided within the stack to indicate when the temperature is within a range of 45°C to 60°C. If the temperature of the stack is outside

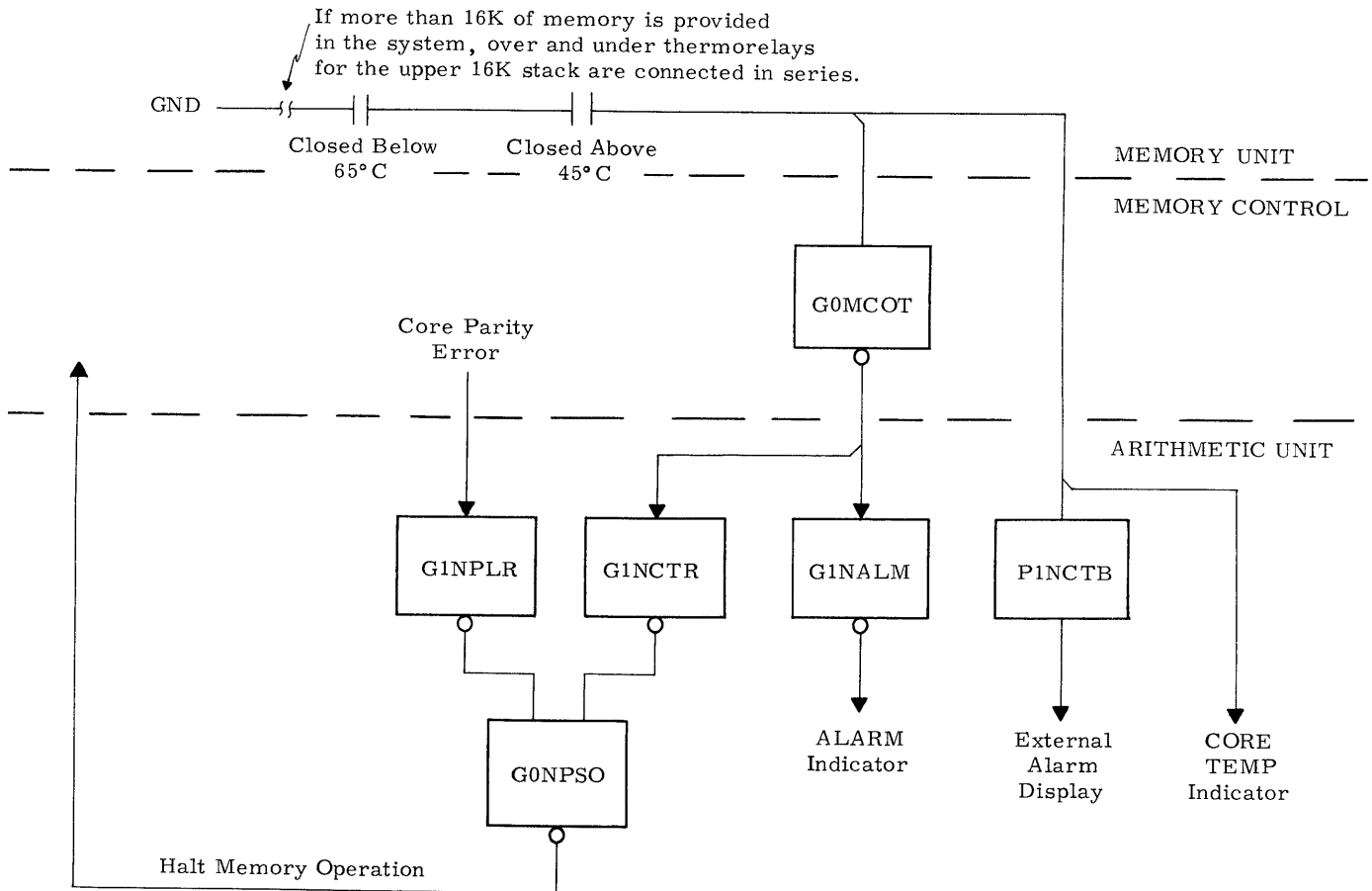


Fig. THEORY. 17. Core Heater Alarm Circuitry



Millivolts	0.00	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
0.00	0.0	3.7	5.2	7.7	10.3	12.8	15.4	17.9	20.1	22.8
1.00	25.3	27.7	30.1	32.6	35.0	37.4	39.8	42.1	44.5	46.8
2.00	49.2	51.5	53.8	56.2	58.5	60.8	63.0	65.3	67.6	69.9
3.00	72.0	74.3	76.5	78.7	81.0	83.1	85.4	87.5	89.7	91.9
4.00	94.1	96.2	98.3	100.5	102.6	104.7	106.9	109.0	111.1	113.2
5.00	115.3	117.4	119.5	121.5	123.6	125.6	127.7	129.8	131.8	133.8

Table THEORY. 2. Thermocouple Conversion Chart (°C)

these limits, the CORE TEMP and ALARM indicators on the computer console are lighted and if a parity error is detected when the temperature is out of limits, memory operation is inhibited. Two copper-constantan thermocouples are provided within each stack for monitoring of the nominal stack temperature. If the stack exceeds approximately 75°C, operating power of 28V AC is removed from the heaters.

The core heater control and associated circuitry is shown on sheet 21 of the memory logic (68C972952). Fig. THEORY. 17 illustrates the core heater alarm circuitry associated with over and under temperature conditions.

The core heaters are designed to maintain the internal temperature of the stack at approximately 55°C over an environmental temperature range of 0° to 55°C. Environmental temperature changes of up to 15°C per hour can be compensated. The 55°C temperature is reached within 1.5 hours from an ambient temperature of 25°C. The core heaters are positioned to maintain all cores at a uniform temperature  $\pm 3^\circ\text{C}$ .

Appendix A illustrates the pin connections and heater circuits contained within the core stack along with a brief adjustment procedure for the heater circuit. Table THEORY. 2 provides a millivolt to temperature conversion chart for the copper-constantan thermocouples. Using this chart the temperature within the stack may be determined using the thermocouples.

### OPERATING POWER

Operating power of  $\pm 12\text{V DC}$ ,  $+5\text{V DC}$ , and  $28\text{V AC}$  is supplied to the memory and memory control unit by the 4780 Power Components (GE Drawing 68C972410). Detailed theory of operation of these components is provided in the Power Distribution section of this book set.

### POWER FAILURE OR SHUTDOWN

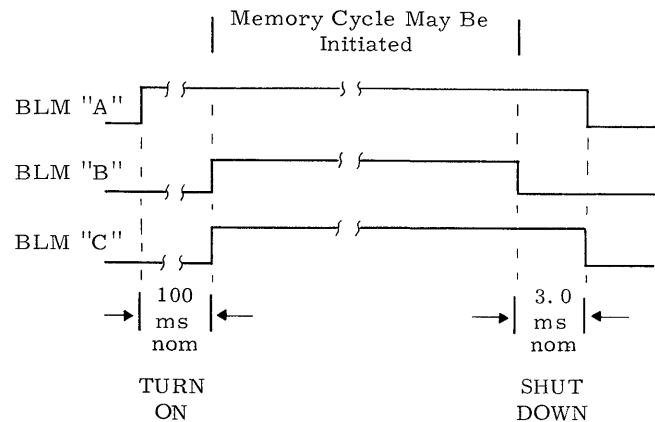
Removal of operating power, either during normal shutdown or by a power failure, is detected by the Buss Level Monitor circuitry contained within the 4780 Power Components (GE Drawing 68C972410). The Buss Level Monitor, described in the Power Distribution section of this book set, provides three control signals to prevent destruction of data stored in core memory during power loss.

The timing of these three signals, BLM "A", BLM "B", and BLM "C" is illustrated in Fig. THEORY. 18. All of these signals must be at the +5 volt level for a memory cycle to be initiated. Once a cycle is initiated, the cycle may be completed. BLM "C" does not go to zero volts until approximately 3 milliseconds after a shutdown has been detected so a previously initiated cycle may be completed.

The BLM "A" signal when at zero volts, clears the initiate memory cycle flip-flop (F1TMTC), clears the Memory Busy flip-flop (F1TMBU), and sets the initialize flip-flop (F1TINT).

The BLM "B" signal when at zero volts, inhibits any initiate signal generated from a request in G0TIMS.

The BLM "C" signal when at zero volts, inhibits the word read and write drive currents by disabling D1UWEN, D1UREN, D1LWEN, and D1LREN.



#### NOTES:

1. BLM "A" at zero initializes memory and inhibits initiating a memory cycle.
2. BLM "B" at zero inhibits initiating a memory cycle.
3. BLM "C" at zero inhibits word read and write drive current.

Fig. THEORY. 18. Buss Level Monitor Control

# **APPENDIX A**

**4015B**

FUNCTION	GO	G1	G2	G3	DON	DOP	DIN	DIP
00	NO2B6	NO2B9	NO2B7	NO2B8	NO2C1	NO2C14	NO2C2	NO2C13
01	NO2C4	NO2C11	NO2C5	NO2C10	NO2C6	NO2C9	NO2C7	NO2C8
02	NO2D1	NO2D14	NO2D2	NO2D13	NO2D3	NO2D12	NO2D4	NO2D11
03	NO2D6	NO2D9	NO2D7	NO2D8	NO2E1	NO2E14	NO2E2	NO2E13
04	NO2E4	NO2E11	NO2E5	NO2E10	NO2E6	NO2E9	NO2E7	NO2E8
05	NO2F1	NO2F14	NO2F2	NO2F13	NO2F3	NO2F12	NO2F4	NO2F11
06	NO2F6	NO2F9	NO2F7	NO2F8	NO2G1	NO2G14	NO2G2	NO2G13
07	NO2G4	NO2G11	NO2G5	NO2G10	NO2G6	NO2G9	NO2G7	NO2G8
08	NO2H1	NO2H14	NO2H2	NO2H13	NO2H3	NO2H12	NO2H4	NO2H11
09	NO2H6	NO2H9	NO2H7	NO2H8	NO2J1	NO2J14	NO2J2	NO2J13
10	NO2J4	NO2J11	NO2J5	NO2J10	NO2J6	NO2J9	NO2J7	NO2J8
11	NO2K1	NO2K14	NO2K2	NO2K13	NO2K3	NO2K12	NO2K4	NO2K11
12	NO2K6	NO2K9	NO2K7	NO2K8	NO2L1	NO2L14	NO2L2	NO2L13
13	NO2L4	NO2L11	NO2L5	NO2L10	NO2L6	NO2L9	NO2L7	NO2L8
14	NO2M1	NO2M14	NO2M2	NO2M13	NO2M3	NO2M12	NO2M4	NO2M11
15	NO2M6	NO2M9	NO2M7	NO2M8	NO2N1	NO2N14	NO2N2	NO2N13
16	NO2N4	NO2N11	NO2N5	NO2N10	NO2N6	NO2N9	NO2N7	NO2N8
17	NO2P1	NO2P14	NO2P2	NO2P13	NO2P3	NO2P12	NO2P4	NO2P11
18	NO2P6	NO2P9	NO2P7	NO2P8	NO2Q1	NO2Q14	NO2Q2	NO2Q13
19	NO2Q4	NO2Q11	NO2Q5	NO2Q10	NO2Q6	NO2Q9	NO2Q7	NO2Q8
20	NO2R1	NO2R14	NO2R2	NO2R13	NO2R3	NO2R12	NO2R4	NO2R11
21	NO2R6	NO2R9	NO2R7	NO2R8	NO2S1	NO2S14	NO2S2	NO2S13
22	NO2S4	NO2S11	NO2S5	NO2S10	NO2S6	NO2S9	NO2S7	NO2S8
23	NO2T1	NO2T14	NO2T2	NO2T13	NO2T3	NO2T12	NO2T4	NO2T11
24	NO2T6	NO2T9	NO2T7	NO2T8	NO2U1	NO2U14	NO2U2	NO2U13

Stack Bit Matrix Pin Designations



FUNCTION	PIN NO	FUNCTION	PIN NO
LWG00	NO3C1	LWDOON	NO3D4
LWG01	NO3C14	LWDOOP	NO3D11
LWG02	NO3C2	LWDO1N	NO3D5
LWG03	NO3C13	LWD01P	NO3D10
LWG10	NO3C3	LWD02N	NO3D6
LWG11	NO3C12	LWD02P	NO3D9
LWG12	NO3C4	LWD03N	NO3D7
LWG13	NO3C11	LWD03P	NO3D8
LWG20	NO3C5	LWD10N	NO3E1
LWG21	NO3C10	LWD10P	NO3E14
LWG22	NO3C6	LWD11N	NO3E2
LWG23	NO3C9	LWD11P	NO3E13
LWG30	NO3C7	LWD12N	NO3E3
LWG31	NO3C8	LWD12P	NO3E12
LWG32	NO3D1	LWD13N	NO3E4
LWG33	NO3D14	LWD13P	NO3E11
		LWD20N	NO3E5
		LWD20P	NO3E10
		LWD21N	NO3E6
		LWD21P	NO3E9
		LWD22N	NO3E7
		LWD22P	NO3E8
		LWD23N	NO3F1
		LWD23P	NO3F14
		LWD30N	NO3F2
		LWD30P	NO3F13
		LWD31N	NO3F3
		LWD31P	NO3F12
		LWD32N	NO3F4
		LWD32P	NO3F11
		LWD33N	NO3F5
		LWD33P	NO3F10

Stack Word Matrix Pin Designations, 1st 4K

FUNCTION	PIN NO	FUNCTION	PIN NO
LWG00	NO3C1	LWD40N	NO3F6
LWG01	NO3C14	LWD40P	NO3F9
LWG02	NO3C2	LWD41N	NO3F7
LWG03	NO3C13	LWD41P	NO3F8
LWG10	NO3C3	LWD42N	NO3G1
LWG11	NO3C12	LWD42P	NO3G14
LWG12	NO3C4	LWD43N	NO3G2
LWG13	NO3C11	LWD43P	NO3G13
LWG20	NO3C5	LWD50N	NO3G3
LWG21	NO3C10	LWD50P	NO3G12
LWG22	NO3C6	LWD51N	NO3G4
LWG23	NO3C9	LWD51P	NO3G11
LWG30	NO3C7	LWD52N	NO3G5
LWG31	NO3C8	LWD52P	NO3G10
LWG32	NO3D1	LWD53N	NO3G6
LWG33	NO3D14	LWD53P	NO3G9
		LWD60N	NO3G7
		LWD60P	NO3G8
		LWD61N	NO3H1
		LWD61P	NO3H14
		LWD62N	NO3H2
		LWD62P	NO3H13
		LWD63N	NO3H3
		LWD63P	NO3H12
		LWD70N	NO3H4
		LWD70P	NO3H11
		LWD71N	NO3H5
		LWD71P	NO3H10
		LWD72N	NO3H6
		LWD72P	NO3H9
		LWD73N	NO3H7
		LWD73P	NO3H8

Stack Word Matrix Pin Designations, 2nd 4K

FUNCTION	PIN NO	FUNCTION	PIN NO
UWG00	NO3N1	UWDOON	NO3P4
UWG01	NO3N14	UWDOOP	NO3P11
UWG02	NO3N2	UWDO1N	NO3P5
UWG03	NO3N13	UWDO1P	NO3P10
UWG10	NO3N3	UWDO2N	NO3P6
UWG11	NO3N12	UWDO2P	NO3P9
UWG12	NO3N4	UWDO3N	NO3P7
UWG13	NO3N11	UWDO3P	NO3P8
UWG20	NO3N5	UWD10N	NO3Q1
UWG21	NO3N10	UWD10P	NO3Q14
UWG22	NO3N6	UWD11N	NO3Q2
UWG23	NO3N9	UWD11P	NO3Q13
UWG30	NO3N7	UWD12N	NO3Q3
UWG31	NO3N8	UWD12P	NO3Q12
UWG32	NO3P1	UWD13N	NO3Q4
UWG33	NO3P14	UWD13P	NO3Q11
		UWD20N	NO3Q5
		UWD20P	NO3Q10
		UWD21N	NO3Q6
		UWD21P	NO3Q9
		UWD22N	NO3Q7
		UWD22P	NO3Q8
		UWD23N	NO3R1
		UWD23P	NO3R14
		UWD30N	NO3R2
		UWD30P	NO3R13
		UWD31N	NO3R3
		UWD31P	NO3R12
		UWD32N	NO3R4
		UWD32P	NO3R11
		UWD33N	NO3R5
		UWD33P	NO3R11

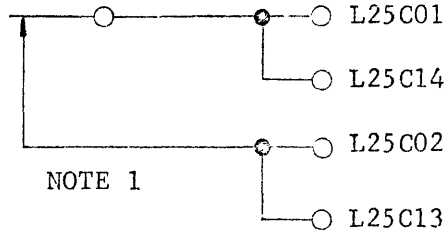
Stack Word Matrix Pin Designations, 3rd 4K

FUNCTION	PIN NO.	FUNCTION	PIN NO
UWG00	NO3N1	UWD40N	NO3R6
UWG01	NO3N14	UWD40P	NO3R9
UWG02	NO3N2	UWD41N	NO3R7
UWG03	NO3N13	UWD41P	NO3R8
UWG10	NO3N3	UWD42N	NO3S1
UWG11	NO3N12	UWD42P	NO3S14
UWG12	NO3N4	UWD43N	NO3S2
UWG13	NO3N11	UWD43P	NO3S13
UWG20	NO3N5	UWD50N	NO3S3
UWG21	NO3N10	UWD50P	NO3S12
UWG22	NO3N6	UWD51N	NO3S4
UWG23	NO3N9	UWD51P	NO3S11
UWG30	NO3N7	UWD52N	NO3S5
UWG31	NO3N8	UWD52P	NO3S10
UWG32	NO3P1	UWD53N	NO3S6
UWG33	NO3P14	UWD53P	NO3S9
		UWD60N	NO3S7
		UWD60P	NO3S8
		UWD61N	NO3T1
		UWD61P	NO3T14
		UWD62N	NO3T2
		UWD62P	NO3T13
		UWD63N	NO3T3
		UWD63P	NO3T12
		UWD70N	NO3T4
		UWD70P	NO3T11
		UWD71N	NO3T5
		UWD71P	NO3T10
		UWD72N	NO3T6
		UWD72P	NO3T9
		UWD73N	NO3T7
		UWD73P	NO3T8

Stack Word Matrix Pin Designations, 4th 4K

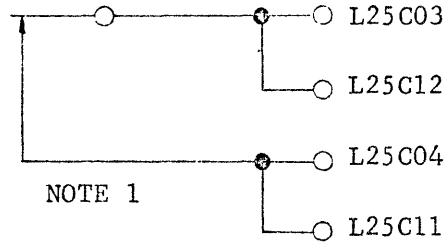


Thermorelay  
open below 45°C  
100 ma



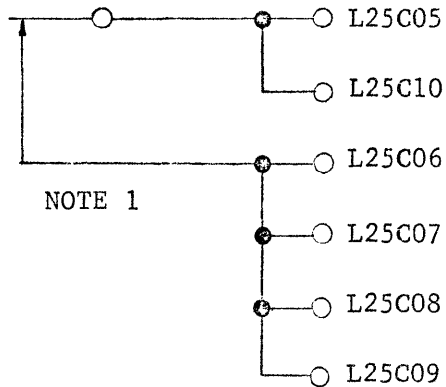
(Thomas A. Edison Ind.  
113°F 292-AA0)

Thermorelay  
closed below 65°C  
100 ma



(Thomas A. Edison Ind.  
149°F 292-AAC)

Thermorelay  
closed below 75°C  
6A

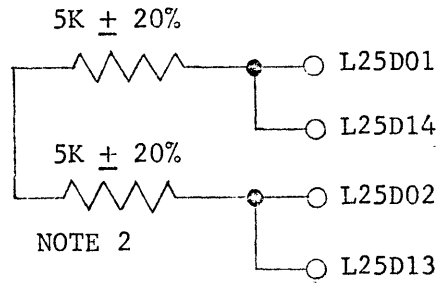


(Thomas A. Edison Ind.  
165°F 292-ABC)

NOTE 1: locate near top of stack on NO2 side.

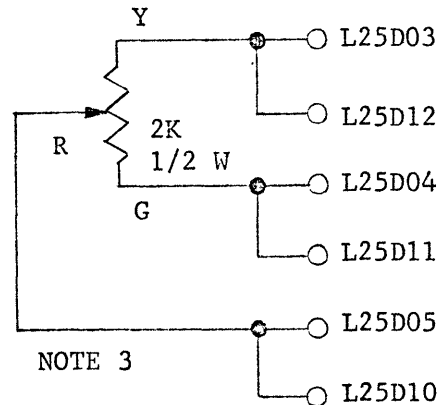
### Temperature Sensors

Thermistors



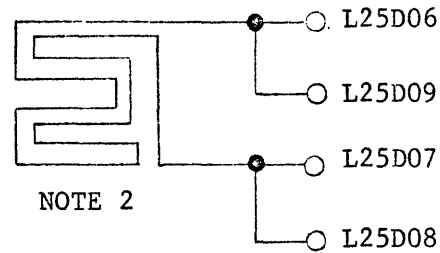
(Fenwall GB35J1)

When pot is turned clockwise resistance between D03 and D05 should increase.

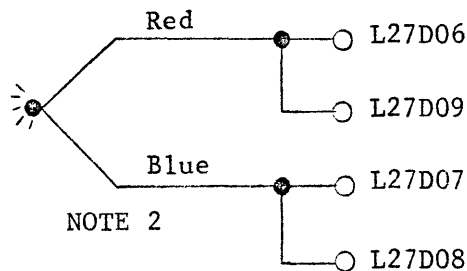


(Bourns 3257L)

Bifilar Heater  
15 ohm ± 10%



Copper Constantan Thermocouple

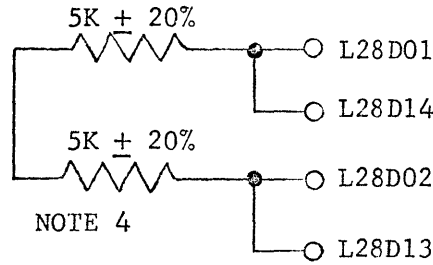


NOTE 2: Locate on N02 side

NOTE 3: Locate near L25D and adjustable from outside of stack L panel side

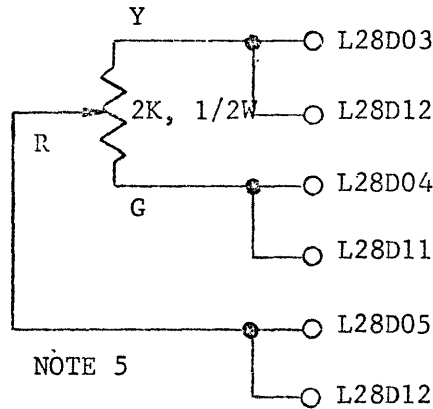
### N02 Side Core Stack Heaters

Thermistors



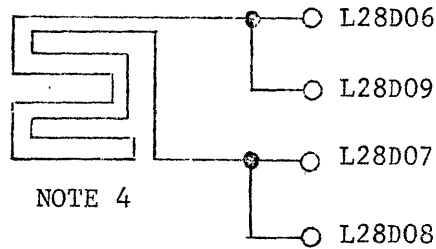
(Fenwall GB35J1)

When pot is turned clockwise resistance between D03 and D05 should increase.

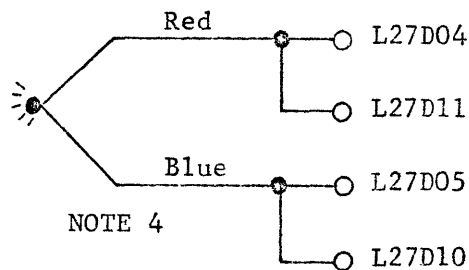


(Bourns 3257L)

Bifilar Heater  
15 ohm ± 10%



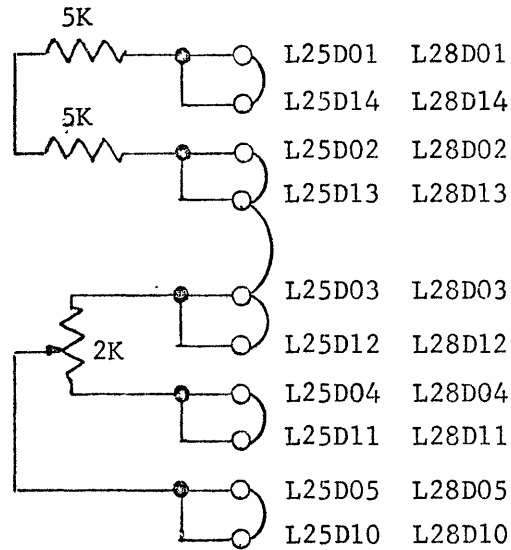
Copper Constantan Thermocouple



NOTE 4: Locate on NO3 side

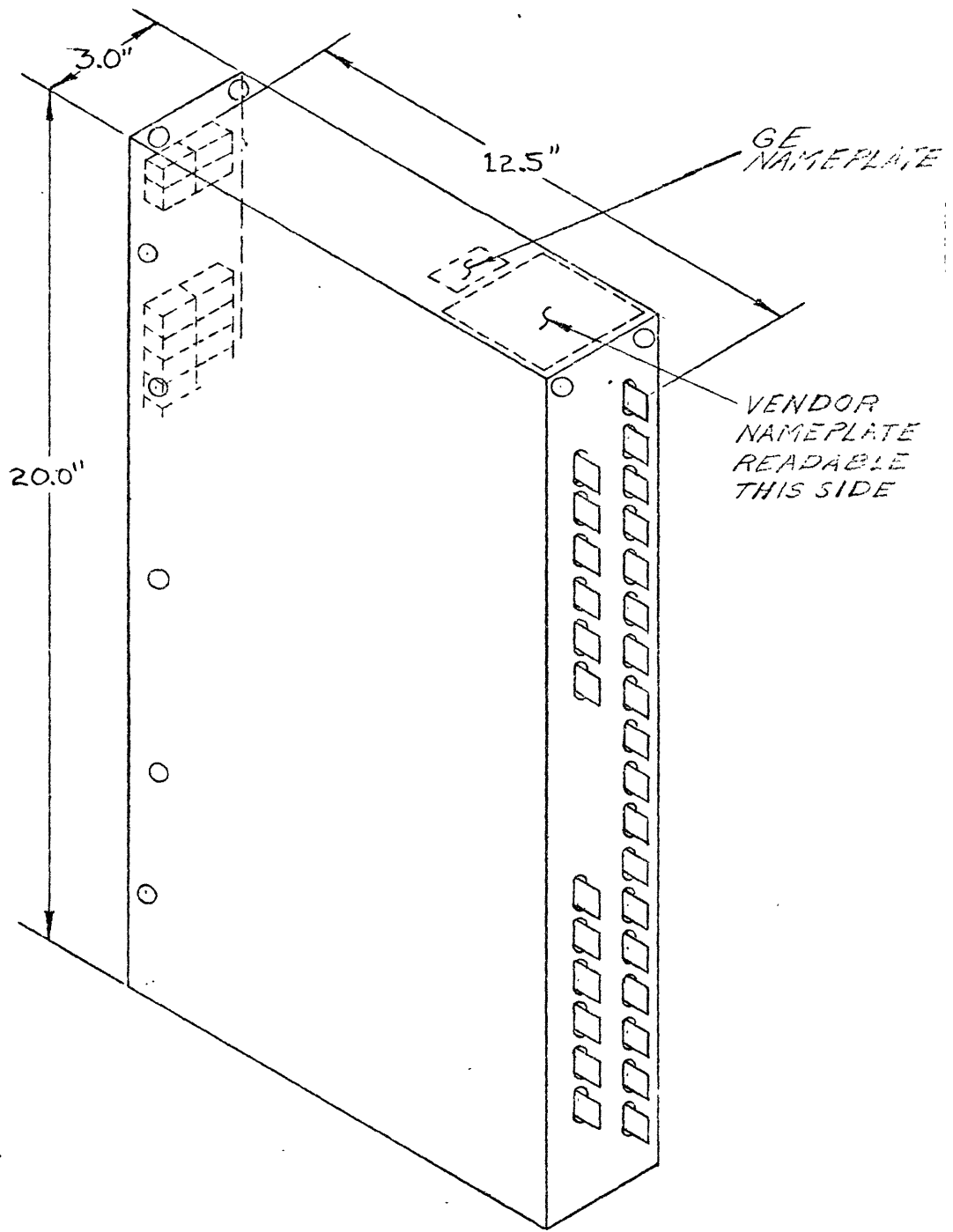
NOTE 5: Locate near L28D and adjustable from outside of stack L panel side.

### N03 Side Core Stack Heaters

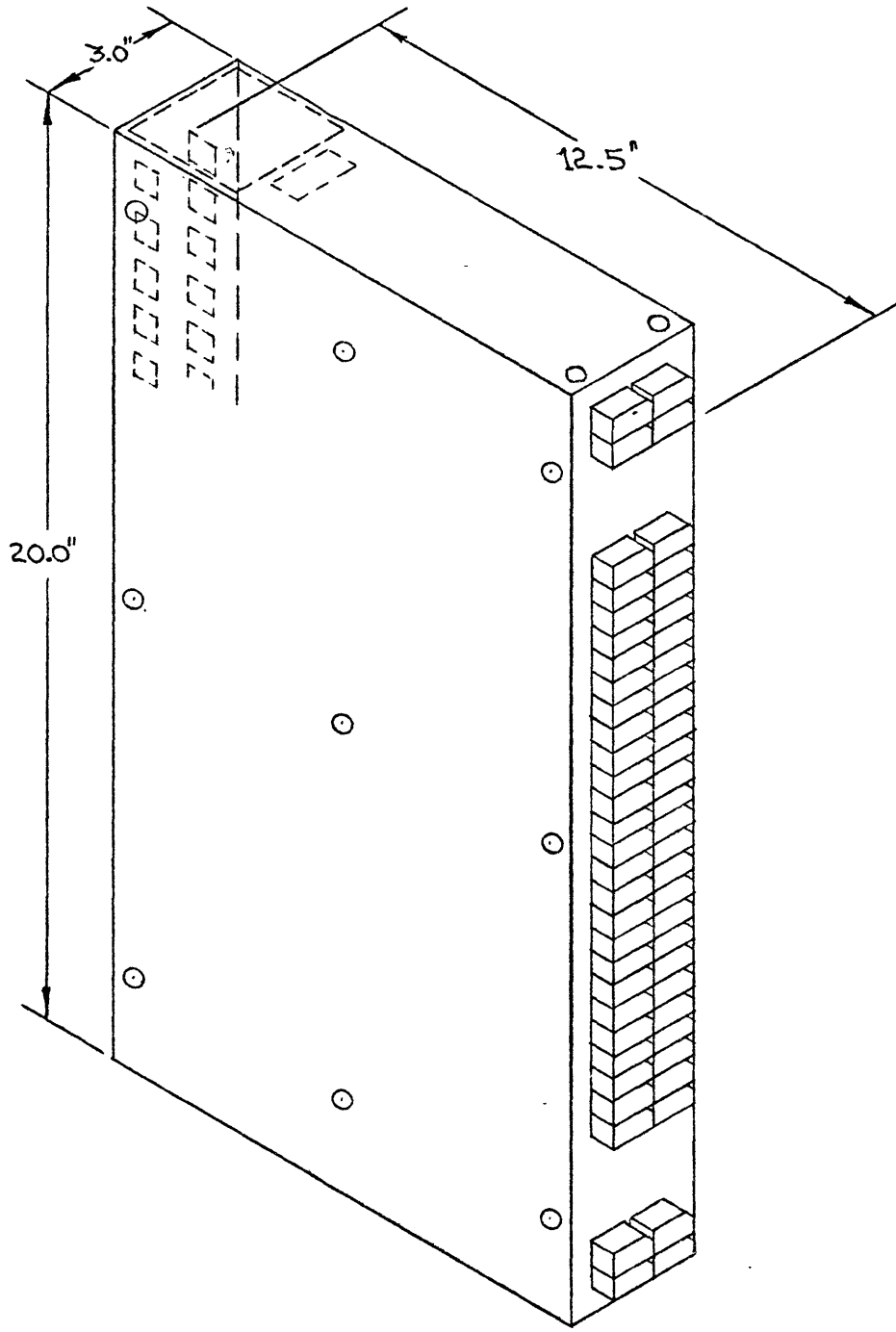


1. Short L25D01 to L25D14.
2. Short L28D01 to L28D14.
3. Short L25D02 to L25D13 to L25D03 to L25D12.
4. Short L28D02 to L28D13 to L28D03 to L28D12.
5. Short L25D04 to L25D11 to L25D05 to L25D10.
6. Short L28D04 to L28D11 to L28D05 to L28D10.
7. Heat stack to 55°C.
8. Set upper pot so that the resistance between L25D01 and L25D05 is 4100ohm  $\pm$  2%.
9. Set lower pot so that the resistance between L28D01 and L28D05 is 4100ohm  $\pm$  2%.

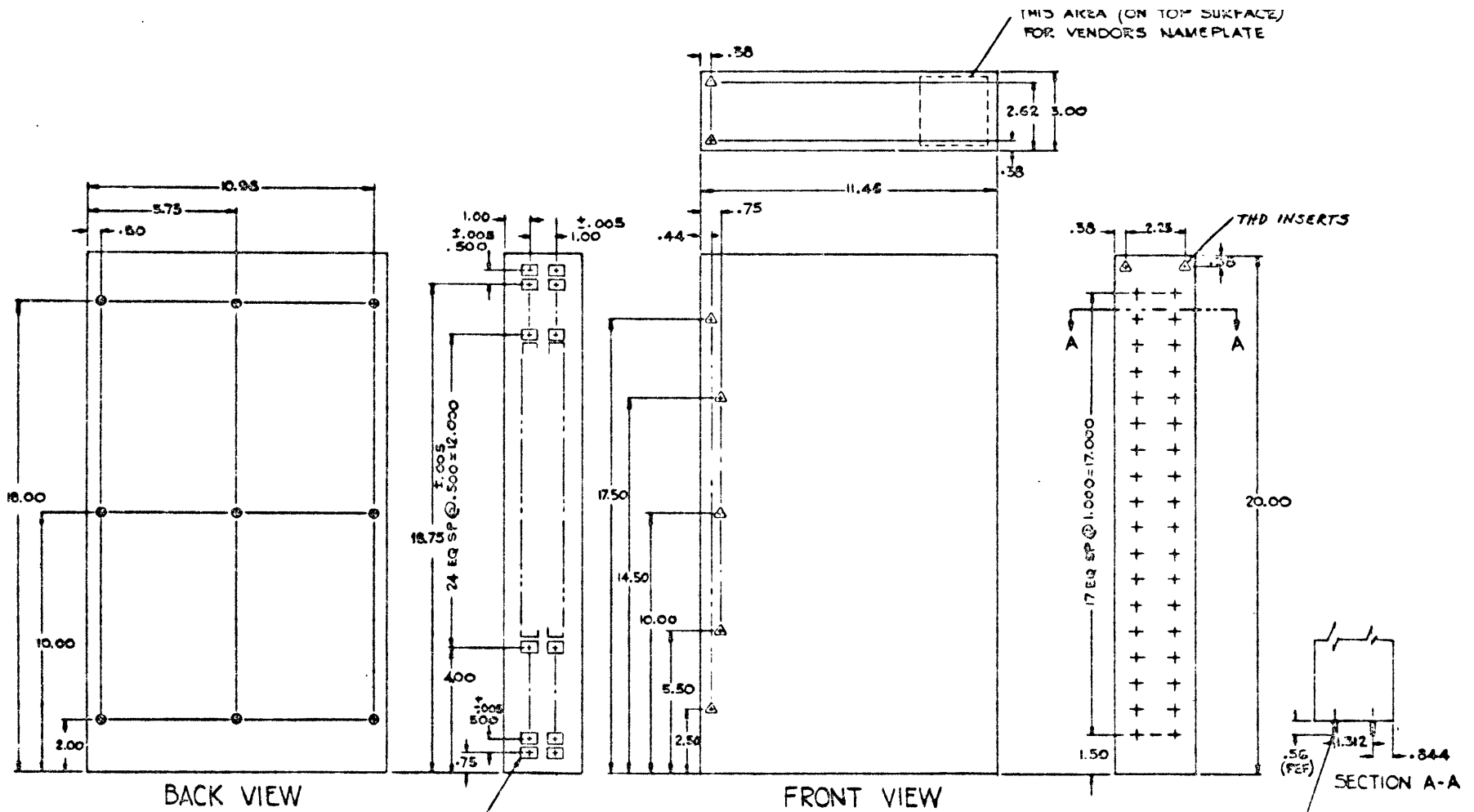
### Temperature Sensor Adjustment



Core Stack, Right Side View



Core Stack, Left Side View



BACK VIEW

FRONT VIEW

SECTION A-A

SYMBOL EXPLANATION				
SYMBOL	HOLE SIZE	QTY	INSERT SIZE	QTY
⊙	.250 <sup>+0.003</sup> <sub>-0.002</sub>	9	#3-32	9
Δ	.281 <sup>+0.003</sup> <sub>-0.000</sub>	11	#8-22	11

PUNCH FOR CONNECTORS  
(58 PL) SEE SHEET 2  
FOR DETAIL OF PUNCH &  
PUNCH PATTERN  
(SEE NOTE #3)

← SEE NOTE #5

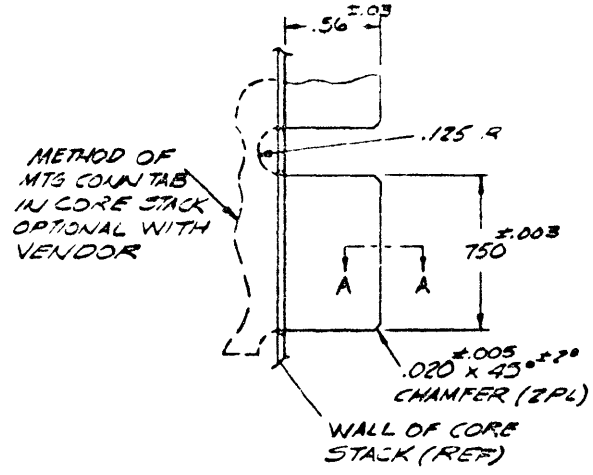
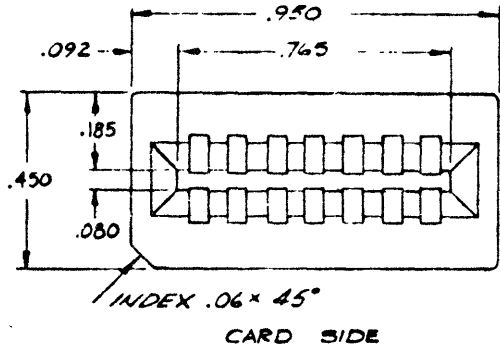
← SEE NOTE #4

LOCATION OF CONNECTOR TABS  
(SEE NEXT PAGE FOR DETAILS)  
(18 REQ PER SIDE 36 TOTAL)

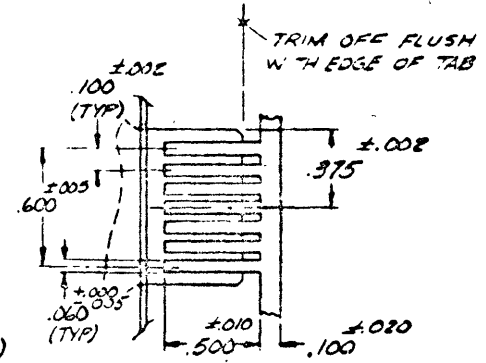
NOTES:

- ENCLOSURE MATERIAL TO BE CLEAR ANODIZED ALUMINUM
- PRECAUTIONS ARE TO BE TAKEN IN DESIGN TO ASSURE THAT DAMAGE CANNOT OCCUR TO INTERIOR ELECTRONICS BY EXCESSIVE PENETRATION OF MOUNTING HARDWARE, I.E. BLIND TAPPED HOLES, ETC.
- CONNECTORS TO BE BONDED TO PANEL 'L' WITH HYSOL EPOXY ADHESIVE (RESIN, HYSOL #AG-4210; HARDNER, HYSOL #H2-3404).
- USE FLOATING, SELF-LOCKING ANCHOR NUTS.
- USE NON-LOCKING INSERT.

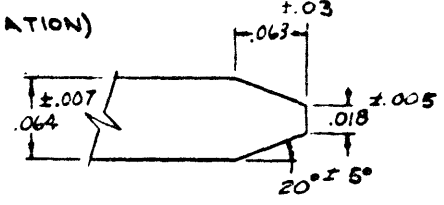
Core Stack, Front/Back View



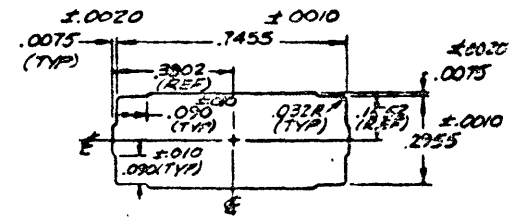
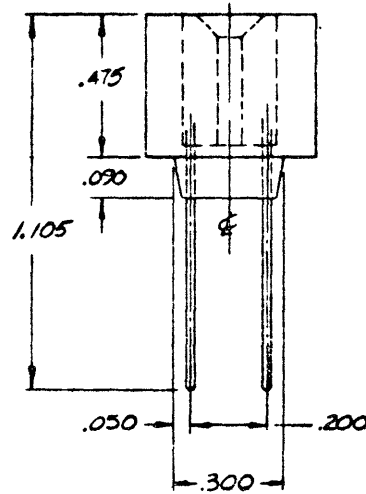
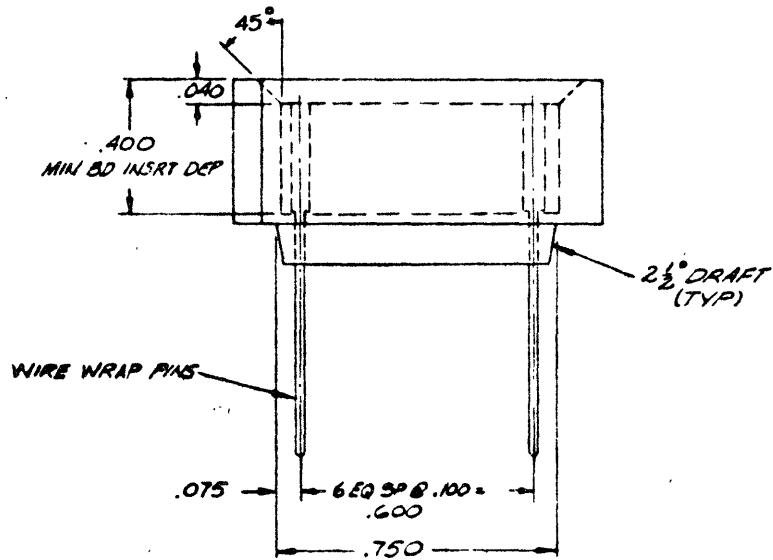
DETAIL OF CONNECTOR TAB (SEE ENCLOSURE DWG. FOR LOCATION)



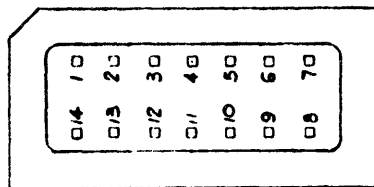
DETAIL OF PLATED FINGER CONTACTS (TYP BOTH SIDES) SEE NOTE #1



SECTION A-A



TYPICAL CONNECTOR PUNCH PATTERN



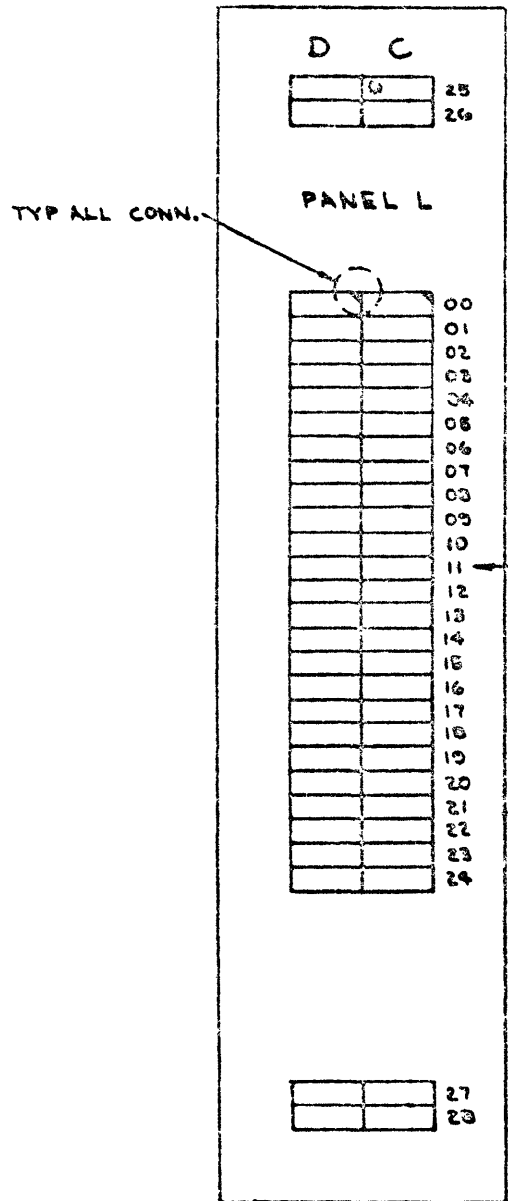
PIN ORIENTATION

CONNECTOR, 14 PIN (DIMENSIONS SHOWN ARE FOR REFERENCE)

NOTES:

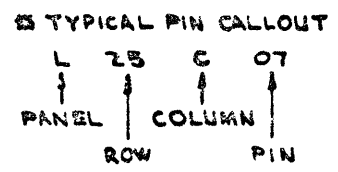
- FINGER MATERIAL IS TO BE PER 68A988E05 P2 OR P3, P3 PREFERRED. CONTACT FINGERS SHALL BE PLATED AS FOLLOWS:
  - .000050" MIN GOLD OVER
  - .000300" MIN. NICKLE





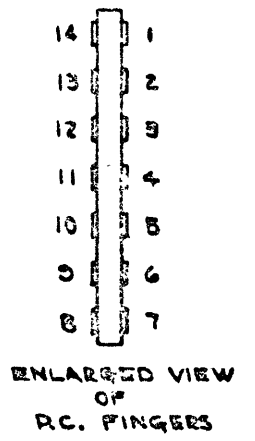
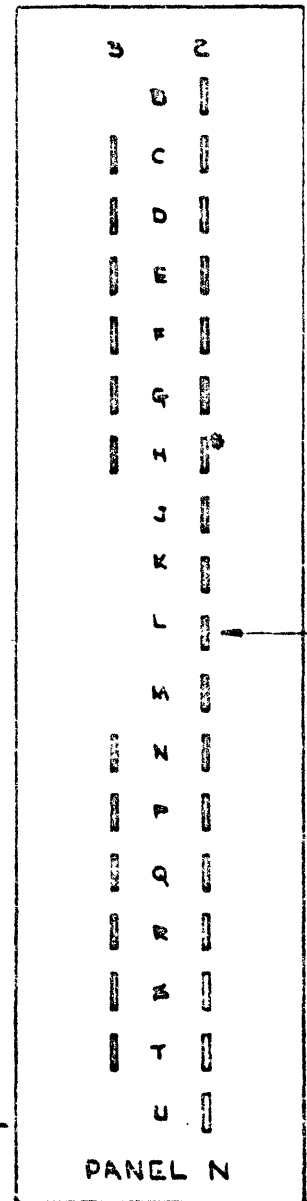
T654321  
091011234

ENLARGED VIEW  
SHOWING POSITION  
OF MOUNTED  
CONNECTORS  
(CARD SIDE)

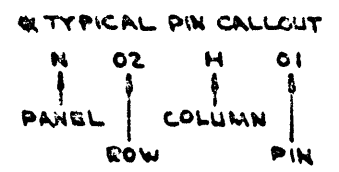


NOTES:

1. PANELS L AND N ARE TO BE MARKED IN ACCORDANCE WITH THIS DRAWING. SILK SCREENING IS SUGGESTED, CHARACTERS ARE TO BE BLACK, .2" HIGH. PIN AND RUN IDENTIFICATION IS NOT REQUIRED.



ILLUSTRATING RUN ORIENTATION



Core Stack Panel Arrangement

# TABLE OF CONTENTS

## API MODULE 4032A

### INTRODUCTION

INTERRUPT CLASSES

RESPONSE ADDRESS AND PRIORITY

ACKNOWLEDGEMENT

ECHO INTERRUPTS

API LOCKOUT

MASK REGISTER

API TEST

API TERMINATION

### OPERATION

Terminology

INTERRUPT DETECTION

PRIORITY AND ADDRESS GENERATION

System With 8 To 16 API Points

System With 17 To 64 API Points

System With 65 to 128 API Points

SEQUENCE CONTROL

ECHO INTERRUPT GENERATION

SPECIAL CONTROL

Initialize Control

API Lockout Control

Inhibitible Interrupt Control

Mask Register Control

Special Instruction IAI<sub>2</sub>

API Test

## LIST OF ILLUSTRATIONS AND TABLES

Fig. INT. 1. Typical Allocations of API Module Inputs

Fig. OPR. 1. API System Block Diagram

Fig. OPR. 2. API Flow Diagram

Fig. OPR. 3. Priority Flow Diagram

Fig. OPR. 4. Typical Timing of Address Generation Relative to Interrupt Initiation

Fig. OPR. 5. API Card Modules

Fig. OPR. 6. API Sequencing Timing Diagram

Fig. OPR. 7. Mask Register Timing Diagram

Fig. OPR. 8. Simplified Schematic of Interrupt Card on Test Extender

Table INT. 1. API Model Numbers

Table OPR. 1. Response Address Generation

Table OPR. 2. Priority Grouping

# INTRODUCTION

The GE/PAC\* 4032A Automatic Program Interrupt Module is designed to operate with the GE/PAC 4022 Arithmetic Unit to provide continuous surveillance of critical inputs without consuming central processor time and, on a priority basis, to execute predetermined instructions when these inputs have interrupted the running program. These inputs can be either internal or external to the central processor. Without this automatic recognition feature, the program would have to contain numerous checking routines to determine whether or not these functions require servicing. After an automatic interrupt has been serviced, control is returned to the running program without loss of information or sequencing.

Every interrupt is associated with a memory core location. An address is generated whenever an interrupt is recognized by the API unit. The interrupt is serviced by executing the instruction at this address. When the core location contains a DMT (decrement memory and test) instruction, the API can serve the functions of pulse counting and accumulation and elapsed time counting. If the instruction at the API address is a TIM (table input to memory) or a TOM (table output from memory) control word, the AU can service external devices independent of the program and of programmable registers. When used with an SPB (save place and branch) instruction, virtually all requirements can be serviced by software programs. Figure INT. 1 shows typical allocations of API module inputs.

The API module will establish an address within 0.5 microsecond from the time an interrupt request is received. The memory request and the actual servicing will start when the execution of any interruptable instruction that may be in progress has been completed. API response is further determined by signal conditioning delays at the input of the API module.

Interrupts can be either inhibitible or non-inhibitible. Inhibitible types are defined as those that are prevented from interrupting the program by the IAI<sub>1</sub> instruction. Both types are inhibited under other conditions.

The API module can be expanded, in eight point increments, from eight to 128 points. The non-inhibitible capability is expandable in four point increments. Echo

generators, described in a subsequent paragraph, are optional and are available in multiples of eight.

## INTERRUPT CLASSES

External interrupts are generated by peripheral devices that request access to the computer. Internal interrupts are generated from such sources as the line frequency timer, the DMT instruction echo, and the echo interrupt from the TIM/TOM function. Both internally and externally generated interrupts can be either inhibitible or non-inhibitible.

Inhibitible interrupts are locked out by the GEN II instruction IAI<sub>1</sub>. Both inhibitible and non-inhibitible interrupts can be locked out by a console switch or by use of optional instruction IAI<sub>2</sub>. The mask register, an optional feature, controls the allowance of all interrupt points in groups of four.

## RESPONSE ADDRESS AND PRIORITY

Each interrupt in the 4020 system is associated with a fixed address known as the interrupt response address. This address automatically provides entry into a subroutine corresponding to the event that caused the interrupt. If the program has allowed interrupts, computer operation can be interrupted at any time after the completion of an instruction, providing the instruction is interruptible, and prior to the accessing of the next instruction from memory.

Of the possible 128 interrupt points, the first group of eight is assigned interrupt response addresses of 200<sub>8</sub> to 207<sub>8</sub>. These are the highest priority interrupts. Additional interrupt points are assigned succeeding response addresses, in the order of decreasing priority, up to address 377<sub>8</sub>.

Each level of priority produces an interrupt request for a specific interrupt. This permits each interrupt level to interrupt the program almost immediately if the only other interrupt is of a lower priority. The servicing of the lower priority request is temporarily halted while the higher order request is recognized. The routine resulting from an interrupt will normally:

4DP4032A0002	Wiring for 8-16 points - with mask register
4DP4032A0003	Wiring for 8-64 points - with mask register
4DP4032A0004	Wiring for 8-128 points - with mask register
4DP4032A1001	Wiring for 1 point - without mask register
4DP4032A1002	Wiring for 8-16 points - without mask register
4DP4032A1003	Wiring for 8-64 points - without mask register
4DP4032A1004	Wiring for 8-128 points - without mask register
4DP4032A0030	Eight-point printed wiring board - without echo
4DP4032A0040	Eight-point printed wiring board - with echo

Table INT. 1. API Model Numbers

\* Registered Trademark of General Electric Company

- Store the contents of the operational registers if these registers are to be used in the performance of the interrupt routine.
- Process the interrupt. For example, it could read an analog input and compare the data with a previous reading.
- Reload the operational registers upon completion of the routine and return control to the main program

## ACKNOWLEDGEMENT

An interrupt request can be recognized by the central processor upon the completion of most instructions. The following instructions inhibit the servicing of an interrupt immediately upon their completion in order to preserve program control:

- BRU Branch unconditionally
- BTS Branch if TSTF is set
- BTR Branch if TSTF is reset
- LDP Load place
- LPR Load place and restore

Some interrupts, such as TIM/TOM input/output device servicing, do not disturb the operational registers. The contents of these registers, therefore, do not require saving and restoration.

### SIGNAL SOURCE

### PROGRAM EXECUTION

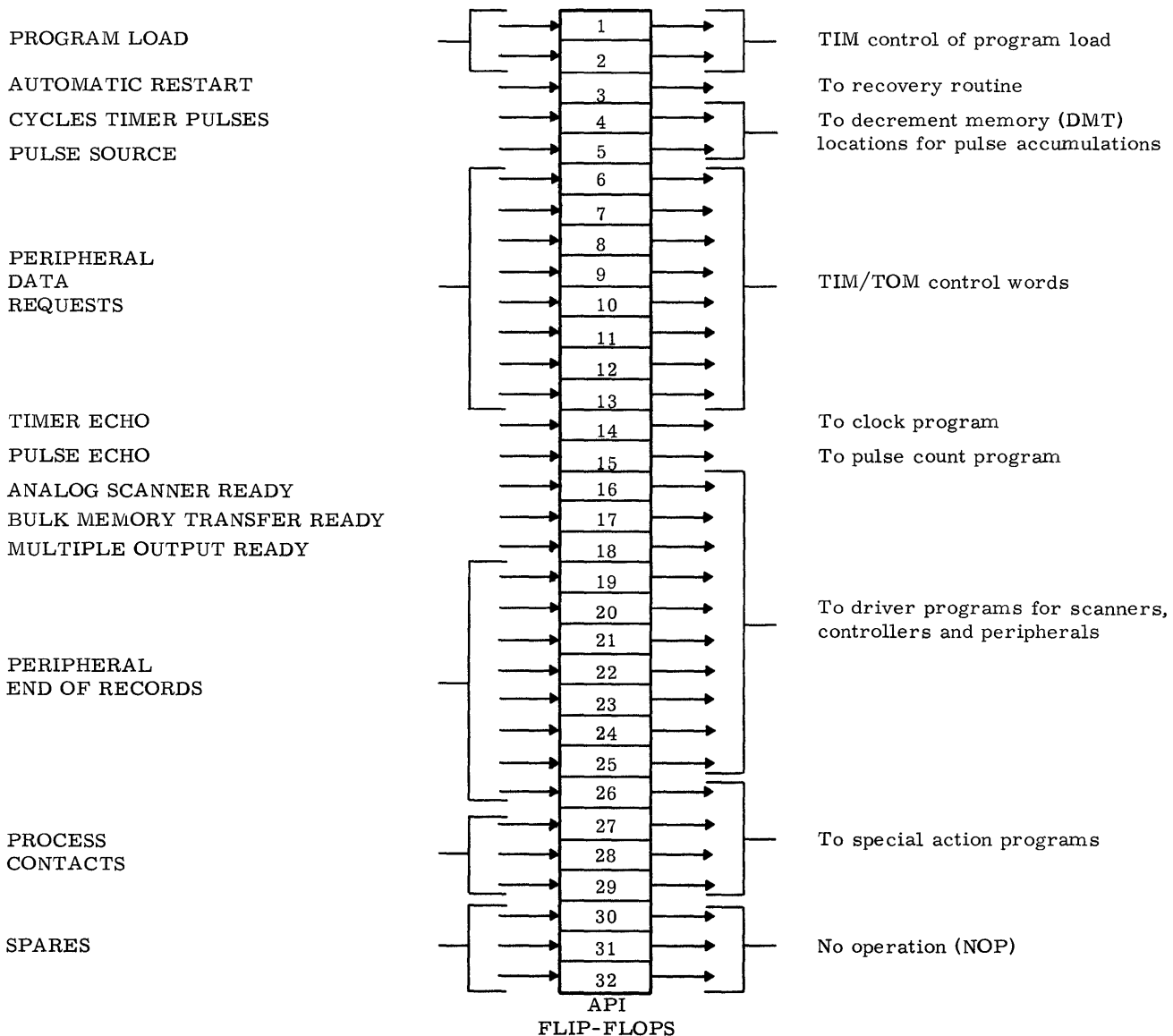


Fig. INT. 1. Typical Allocations of API Module Inputs

- SPB Save place and branch
- TXH Test X high or equal
- XEC Execute
- Quasi entry instructions

The LDX instruction is non-interruptable to facilitate the memory protect feature.

In order to preserve the J-counter, the following instructions prohibit interruption immediately upon their completion:

- CLO Count least significant ones
- CLZ Count least significant zeroes
- CMO Count most significant ones
- CMZ Count most significant zeroes
- CNO Count number of ones
- CNZ Count number of zeroes

The JNR, JCB, and JDR instructions are classified non-interruptable so that when a device is found ready, it is serviced immediately. If an interrupting TIM/TOM operation was allowed following one of these instructions, it would be possible for the program to return to a device no longer ready. The JNO instruction is non-interruptable only because of the logic used in making the other instructions non-interruptable.

Instructions that control the API System are non-interruptable. These are IAI<sub>1</sub>, IAI<sub>2</sub>, LMR, LMR<sub>2</sub>, PAI, and the eight API Test instructions.

The interrupt request is initiated by the interrupt control circuitry which forces execution of the instruction located at the interrupt response address. This is accomplished by allowing the API module to specify the address from which an instruction is to be fetched rather than by the normal CPU function of fetching the instruction. The P-register incrementation is inhibited for this one instruction fetch to ensure that the program count is not lost.

All instructions executed from an interrupt response address allow the next instruction to be executed before another interrupt can be acknowledged.

## ECHO INTERRUPTS

Echo interrupts are special interrupts that are caused by certain internal or external functions. They are normally used to keep track of time counters and to signal the central processor when table empty or table full conditions exist.

The API echo signal may be generated by an interrupting DMT or by a TIM/TOM function. The DMT echo occurs whenever a DMT instruction is executed from an interrupt response address and the count passes from zero

to minus one. The TIM or TOM echo occurs when the word count reaches the table end condition. Echo signals are normally wired into one of the possible 128 interrupt points. Once an echo signal is wired this way it will always generate the same interrupt response address.

## API LOCKOUT

The API lockout switch located on the programming and maintenance console is enabled only when the console key switch is in the AUTO or MANUAL positions. When enabled, depressing the API lockout switch inhibits the servicing of all automatic priority interrupt requests. However, this does not prevent the API module from detecting interrupts and setting the interrupt register. Any one interrupt occurring on a given interrupt line will not be serviced but it will be remembered. If a second interrupt occurs on the same line, however, its incidence will not be known because the flip-flop is already set. Caution in the use of the API lockout switch should be exercised to avoid losing count pulses or data characters.

## MASK REGISTER

The API mask register is an option that permits the program to inhibit or allow interrupts as required during the course of the program. The mask register is programmed from the contents of the A-register by a GEN II command. Each register bit controls four API points. Therefore, for a full complement of 128 points, a 32 bit mask register is required and two GEN II commands (LMR and LMR<sub>2</sub>) are needed to load it. Each command will load the 16 least significant bits of the A-register into the mask register. To permit interrupt requests on four given interrupt points, the program must load a zero into the appropriate bit position of the mask register. For inhibitible interrupts, the Permit flip-flop must also be set.

## API TEST

A method for semi-automatically testing the API system is available with the mask register option. Eight internal GEN II commands are used to simulate interrupts on one eight-point card at a time. In addition to the gating structure for decoding these commands, a special card extender with an attached eight-conductor cable is required.

Special test programs can be written to cause interrupts and to set or reset the mask register and Permit Automatic Interrupt flip-flop as required. The test program would then determine if the priority and addressing logic and all special controls are functioning properly.

## API TERMINATION

The purpose of optional API termination cards is to shape and attenuate interrupt signals that are not otherwise directly suitable for input to the API module. Interrupts that are generated by equipments housed in external cabinets often require this special treatment. One API termination card contains the hardware necessary to reform 18 interrupt signals. Up to seven API termination cards can be included in the central processor, for a total API termination capacity of 126.

# OPERATION

The following theory of operation is referenced to the logic schematic of GE Drawing 68C972382, to the block diagrams of Figures OPR. 1, OPR. 2, and OPR. 3, and to the timing diagram of Figure OPR. 6.

Basically, when an interrupt occurs on an interrupt line, it is sensed by a change detector. The change detector signal sets the associated Interrupt flip-flop. The Interrupt flip-flop signal is ANDed with the No Interrupt Present signal. With both signals true, the appropriate Generate Priority flip-flop is set.

If the interrupt is not to be inhibited by either the IAI<sub>1</sub> instruction or by the mask register, the correct address gates are enabled. Timer action, initiated when the interrupt occurred, sequences events so that memory is addressed from the API area at the proper time. The sequence control circuit also ensures the execution of at least one additional instruction before another interrupt is serviced.

## Terminology

Throughout the following theory of operation, signal

names containing an asterisk (\*) or symbol (#) are used. The asterisk denotes any letter between B and R and refers to areas in the 4032 API module. The symbol (#) can mean any number between 0 and 7. These devices allow the discussion to be general in nature and imply that signals in one portion of the overall circuit are treated identically to corresponding signals in another part.

Some signal names end with the letter L, meaning lower. If the corresponding signal on the upper half of a card is acted upon in the same way or at the same time, the signal appears ending with L(U). This means either L or U.

Signal names with only six characters originate in the API module. With the exception of well known signals like SC01 and SCA, all signals originating in the arithmetic unit are listed here preceded by the letters AU.

## INTERRUPT DETECTION

Interrupt signals are detected by differentiating elements that sense when the signals have momentarily changed

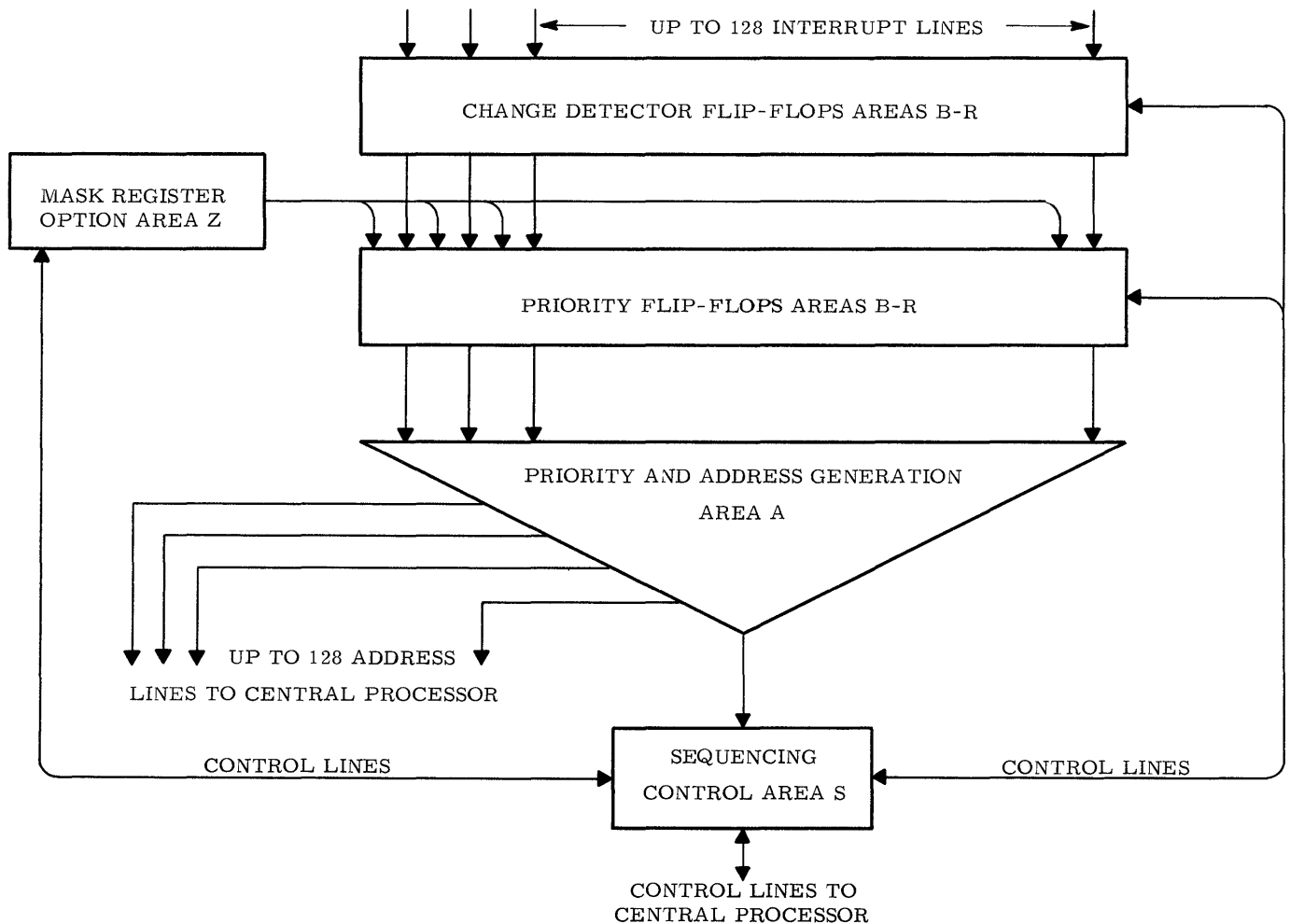


Fig. OPR. 1. API System Block Diagram

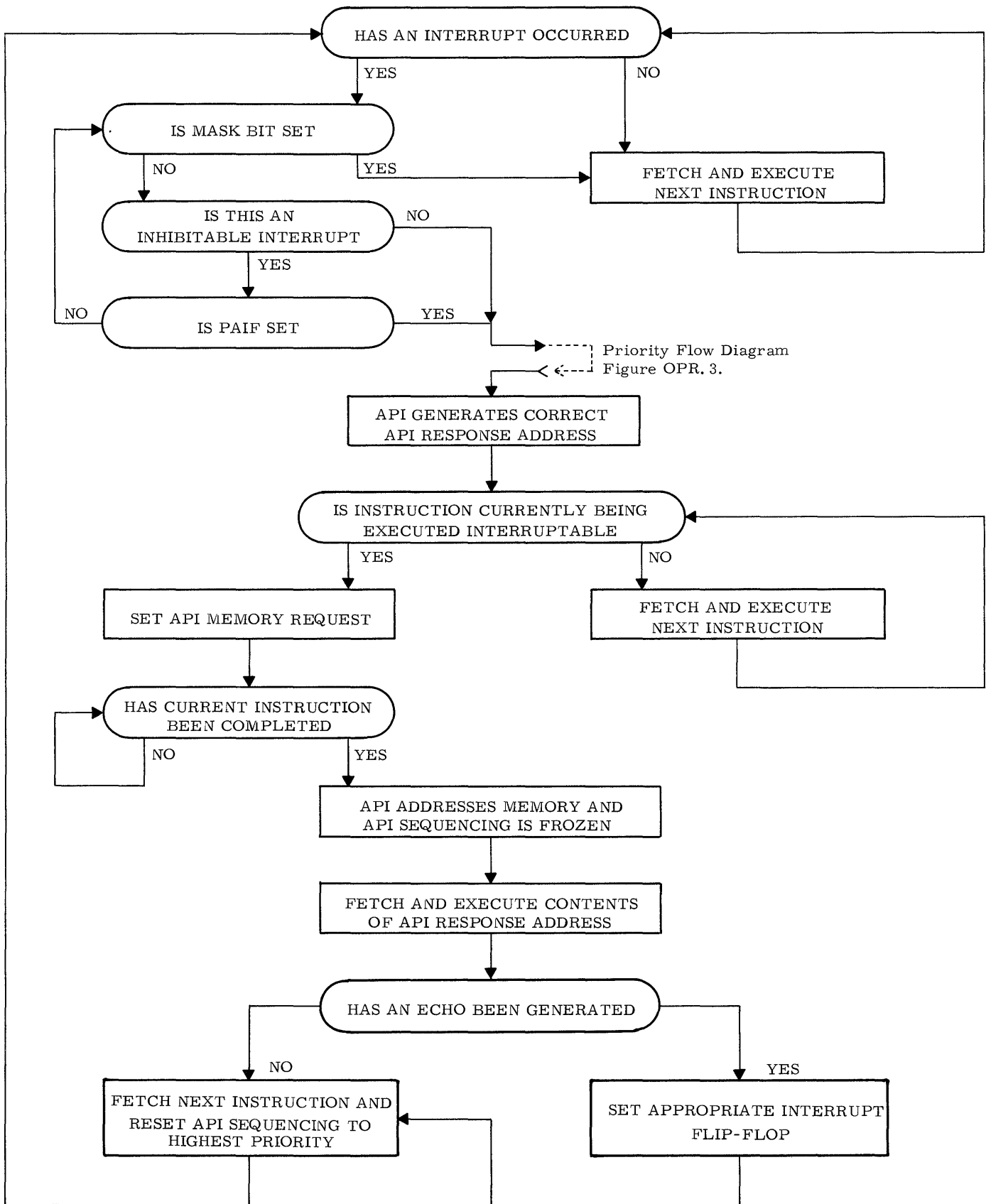


Fig. OPR. 2. API Flow Diagram

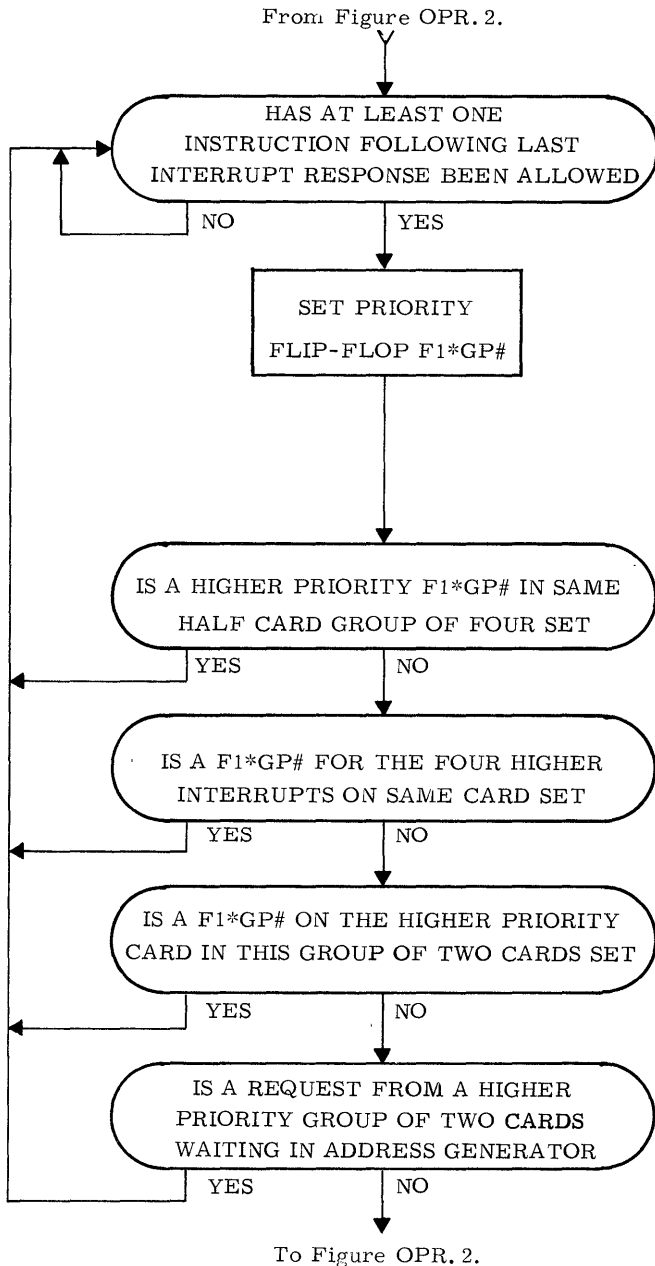


Fig. OPR. 3. Priority Flow Diagram

from 3.6 volts to zero volts. The negative going spikes from the elements cause associated Interrupt flip-flops  $F1*GI\#$  to set. If no previous interrupt is requesting address generation, the appropriate Priority flip-flops  $F1*GP\#$  are set. Several Priority flip-flops may set simultaneously if their associated Interrupt flip-flops are set due to several interrupts occurring at about the same time.

$$F1*GP\# = F1*GI\# \cdot D0SNIP$$

The signal from the set side of each Priority flip-flop is ANDed with the clear sides of all preceding Priority flip-flops within its own group of four. If no flip-flop of

a higher priority in the group of four is set, the signal is gated to bit 0 and 1 identification gates  $G1*GL0$  and  $G1*GL1$  ( $G1*GU0$  and  $G1*GU1$  for the higher numbered side on a card of eight). The decoded bit 0 and 1 signals are ANDed with  $D1*GB0$  ( $D1*GB1$  for the higher numbered side on a card of eight).

$D1*GB0$  or  $D1*GB1$  will be true if no other  $F1*GP\#$  of a higher priority within its group of 16 interrupts is set and if the particular group of four interrupts is not inhibited. The status of the outputs of gates  $G0*GL0$  and  $G0*GL1$  ( $G0*GU0$  and  $G0*GU1$  for the higher numbered side of a card) identifies which of the four interrupts is requesting service and determines the state of bits 0 and 1 in the response address.

### PRIORITY AND ADDRESS GENERATION

The highest priority interrupt occurring within a two card group is recognized by enabling the appropriate driver  $D0*GB0$  or  $D0*GB1$  for the 16 interrupts involved and by enabling bit 0 and 1 gates as required. This is accomplished on the Change Detector and Priority Logic cards.

$D0*GB0$  or  $D0*GB1$  cause the generation of bits 2 through 6 as required for the particular API response address. See Table OPR. 1.

Bit 7 is generated whenever memory is addressed from the API area (area W). Derivation of bit 7 is explained in the theory of operation for the Sequence Control circuit.

### System With 8 To 16 API Points

In a system with less than 17 API points, all of the necessary priority logic is accomplished on the Change Detector and Priority Logic cards. Address generation is performed on a portion of the Sequence Control card.

Any interrupt will enable Lower Request gate  $G0ALRQ$ :

$$G0ALRQ = D1ABCI$$

$$D1ABCI = D0BGB0 + D0BGB1 + D0CGB0 + D0CGB1$$

The  $G0ALRQ$  signal enables the time counter in the sequence control area and causes the No Interrupt Present signal to go untrue:

$$\overline{D0SNIP} = G0ALRQ$$

$D0SNIP$  prevents a higher priority interrupt that may have just occurred from interfering at this time, allowing the interrupt that enabled the timing action to generate its response address. Figure OPR. 4 shows typical timing of address generation relative to interrupt initiation.

Memory address lines for bits 0, 1, 2, and 3 are connected to the outputs of gates  $G1AB00$ ,  $G1AB01$ ,  $G1AB02$ , and  $G1AB03$ , respectively.

Bits 0 and 1 are generated as required for the particular address:

$$G1AB00 = G0BGL0 + G0BGU0 + G0CGL0 + G0CGU0$$



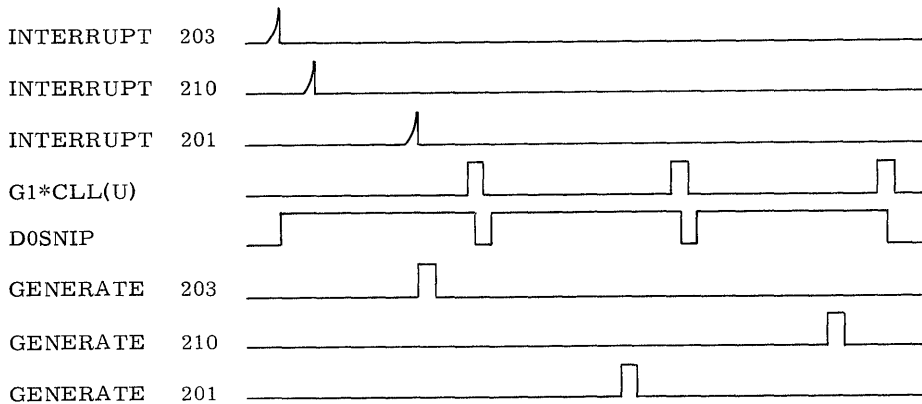


Fig. OPR. 4. Typical Timing of Address Generation Relative to Interrupt Initiation

$$G1AB01 = G0BGL1 + G0BGU1 + G0CGL1 + G0CGU1$$

Generation of bit 2 is required for any interrupt that occurs in the upper half of either area B or C:

$$G1AB02 = D0BGB1 + D0CGB1$$

Bit 3 is generated for any interrupt occurring in area C:

$$G1AB03 = D0CGB0 + D0CGB1$$

### System With 17 To 64 API Points

Granting service to the highest priority interrupting group is accomplished on the Priority and Address Generator cards. There is only one D0\*GB0 or D0\*GB1 signal, from a two card group of 16 interrupts, true at a time. Any one will enable its group Interrupt gate or driver. See Table OPR. 1 or OPR. 2 for the two card priority grouping.

$$D1ABCI = D0BGB0 + D0BGB1 + D0CGB0 + D0CGB1$$

$$G1ADEI = D0DGB0 + D0DGB1 + D0EGB0 + D0EGB1$$

$$G1AFGI = D0FGB0 + D0FGB1 + D0GGB0 + D0GGB1$$

$$G1AHII = D0HGB0 + D0HGB1 + D0IGB0 + D0IGB1$$

Any of these group interrupt signals will enable Lower Request driver D1ALRQ.

$$D1ALRQ = G0ABCI + G0ADEI + G0AFGI + G0AHIP$$

The Lower Request signal enables the time counter in the sequence control area and causes the No Interrupt Present signal to go untrue. D0SNIP prevents a higher priority interrupt that may have just occurred from interfering at this time, allowing the interrupt that enabled the timing action to generate its response address. Figure OPR. 4 shows typical timing of address generation relative to interrupt initiation.

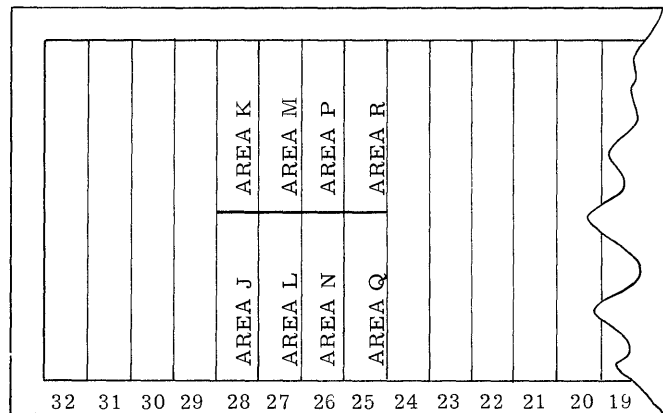
A group Priority gate is enabled if its associated group Interrupt signal is true and there is no higher priority group interrupt:

$$G0A DEP = G1ADEI \cdot \overline{D1ABCI}$$

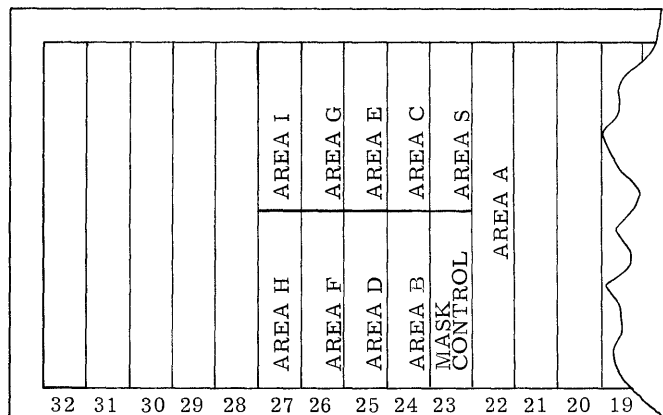
$$G0AFGP = G1AFGI \cdot \overline{D1ABCI} \cdot \overline{G1ADEI}$$

$$G0AHIP = G1AHII \cdot \overline{D1ABCI} \cdot \overline{G1ADEI} \cdot \overline{G1AFGI}$$

There is no group Priority gate for areas B and C because it is the highest priority group and an address is



PANEL K



PANEL A

Fig. OPR. 5. API Card Modules

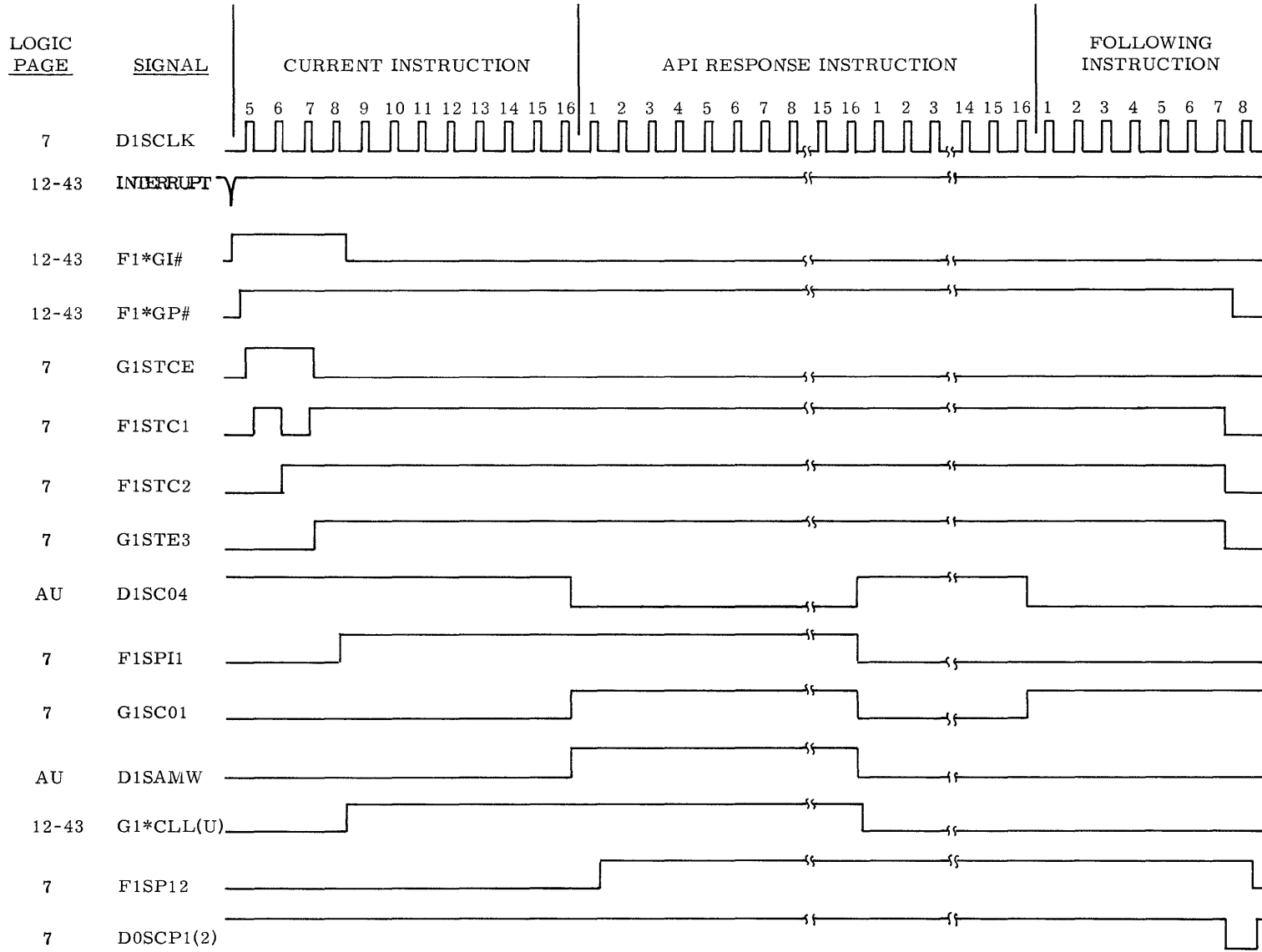
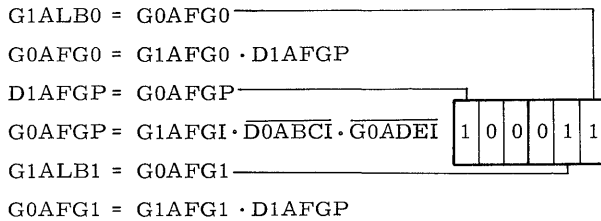


Fig. OPR.6. API Sequencing Timing Diagram

generated from this group first, if an interrupt has occurred within it.

Bits 0 and 1 of the response address are determined by which group Interrupt gate is enabled. The condition of the selected bit 0 and 1 identification gates (G0\*GL0 and G0\*GL1 or G0\*GU0 and G0\*GU1) on the Change Detector cards is gated to Lower Bit 0 gate G1ALB0 and to Lower Bit 1 gate G1ALB1 and then to the memory addressing control circuitry of the AU. For example, bits 0 and 1 must both be set for response address 243<sub>8</sub>:



Bits 4 and 5 of a response address are generated when Lower Bit 4 gate G1ALB4 and Lower Bit 5 gate G1ALB5, respectively, are enabled. These gates are enabled directly from the group Priority gates. Refer to Table OPR.1. It is seen that bit 5 is required for all response addresses originating in areas F, G, H, and I:

$$G1ALB5 = G0AFGP + G0AHIP$$

Bit 4 is required for all response addresses originating in areas D, E, H, and I:

$$G1ALB4 = G0ADEP + G0AHIP$$

Bit 3 is required for all response addresses originating in areas C, E, G, and I:

Area C  $G1ALB3 = D0CGB0 + D0CGB1$

Area E  $G1ALB3 = G0AEB3$   
 $G0AEB3 = D1ADEP \cdot G1AEB3$   
 $G1AEB3 = D0EGB0 + D0EGB1$

Area G  $G1ALB3 = G0AGB3$   
 $G0AGB3 = D1AFGP \cdot G1AGB3$   
 $G1AGB3 = D0GGB0 + D0GGB1$

Area I  $G1ALB3 = G0AIB3$   
 $G0AIB3 = D1AHIP \cdot G1AGB3$   
 $G1AIB3 = D0IGB0 + D0IGB1$

Bit 2 is required for all response addresses originating in the lower priority (higher numbered) half of each card or area. Bit 2 is generated as follows:

Area B  $G1ALB2 = D0EGB1$

Area C  $G1ALB2 = D0CGB1$

Area D  $G1ALB2 = G0AEB2$   
 or E  $G0AEB2 = D1ADEP \cdot G1AEB2$   
 $G1AEB2 = D0DGB1 + D0EGB1$

Area F  $G1ALB2 = G0AGB2$   
 or G  $G0AGB2 = D1AFGP \cdot G1AGB2$   
 $G1AGB2 = D0FGB1 + D0GGB1$

Area H  $G1ALB2 = G0AIB2$   
 or I  $G0AIB2 = D1AHIP \cdot G1AIB2$   
 $G1AIB2 = D0HGB1 + D0IGB1$

## System With 65 To 128 API Points

The upper 64 API points (lower priority) are also divided into two area groups. Each group consists of two cards or 16 interrupts. Any upper group Interrupt gate will enable the Upper Request gate:

$$G1AURQ = G0AJKI + G0ALMI + G0ANPI + G0AQRQ$$

Bit 6 is generated if there is an upper group request and no request from the lower 64 interrupts is present. This makes the starting address 300<sub>8</sub> rather than 200<sub>8</sub>.

$$D1AB06 = G0AB06$$

$$G0AB06 = G1AURQ \cdot \overline{G0ALRQ}$$

The G0AB06 signal, signifying an upper request, also enables the time counter in the sequence control area.

A group Priority gate is enabled if its associated group Interrupt signal is true and there is no higher priority upper group interrupt. The gating structure for determining the highest priority interrupting upper group is identical to that for determining the highest priority interrupting lower group.

Derivation of bits 0 through 5 for the upper API response addresses is accomplished in the same way that bits 0 through 5 were derived for the lower addresses. However, Upper Bit gates, rather than Lower Bit gates, are enabled. These are G1AUB0, G1AUB1, G1AUB2, G1AUB3, G1AUB4, and G1AUB5.

In systems with more than 64 API points, the memory address lines for bits 0 through 5 are not connected directly to the outputs of the Lower Bit gates. They are connected instead to bit gates G1AB00 through G1AB05. These gates can be enabled by either Lower Bit gates or Upper Bit gates. For example, bit 4 is generated as follows:

$$G1AB04 = G0ALB4 + G0AUB4$$

$$G0ALB4 = G1ALB4$$

$$G0AUB4 = G1AUB4 \cdot D1AB06$$

## SEQUENCE CONTROL

An interrupt request from the address generator area enables Time Counter Enable gate G1STCE and causes the No Interrupt Present signals to go untrue:

$$D0SNIP = G0ALRQ + G0AB06$$

$$D0SNI1 = G0ALRQ + G0AB06$$

$$G1STCE = (G0ALRQ + G0AB06) \cdot (\overline{F1STC1} + \overline{F1STC2})$$

Drivers D0SNIP and D0SNI1 serve the lower and upper 64 interrupts, respectively. Both signals, untrue, prevent

a higher priority interrupt that occurs after the time counter has been enabled from being serviced before the one that enabled the time counter. This is accomplished by prohibiting the setting of additional Priority flip-flops on the Change Detector cards for the duration of  $\overline{D0SNIP}$  and  $\overline{D0SN11}$ .

flip-flop F1STC1 to set. F1STC1 sets on the next clock, arms F1STC2 to set, and clears on the following clock, at which time F1STC2 sets. On the third clock, F1STC1 sets again. With both F1STC1 and F1STC2 set, G1STCE goes untrue, preventing further timer action. The delay of three clock times is necessary to ensure that the address of the highest priority interrupt has been firmly established in the address generator. As interrupts

Time Counter Enable gate G1STCE arms Time Counter

AREA	Bits Generated Due To D0*GB0 OR D0*GB1								OCTAL ADDRESS
	7	6	5	4	3	2	1	0	
B lower	1						x	x	200-203
B upper	1					1	x	x	204-207
C lower	1				1		x	x	210-213
C upper	1				1	1	x	x	214-217
D lower	1			1			x	x	220-223
D upper	1			1		1	x	x	224-227
E lower	1			1	1				230-233
E upper	1			1	1	1			234-237
F lower	1		1						240-243
F upper	1		1			1			244-247
G lower	1		1		1				250-253
G upper	1		1		1	1			254-257
H lower	1		1	1					260-263
H upper	1		1	1		1			264-267
I lower	1		1	1	1				270-273
I upper	1		1	1	1	1			274-277
J lower	1	1							300-303
J upper	1	1				1			304-307
K lower	1	1			1				310-313
K upper	1	1			1	1			314-317
L lower	1	1		1					320-323
L upper	1	1		1		1			324-327
M lower	1	1		1	1				330-333
M upper	1	1		1	1	1			334-337
N lower	1	1	1						340-343
N upper	1	1	1			1			344-347
P lower	1	1	1		1		x	x	350-353
P upper	1	1	1		1	1	x	x	354-357
Q lower	1	1	1	1			x	x	360-363
Q upper	1	1	1	1		1	x	x	364-367
R lower	1	1	1	1	1		x	x	370-373
R upper	1	1	1	1	1	1	x	x	374-377

Table OPR. 1. Response Address Generation

occur, the address to be generated continues to change to that of the highest priority interrupt until D0SNIP and D0SNI1 become untrue.

$$\begin{aligned} F1STC1 &= G1STCE \cdot CLK \\ \overline{F1STC1} &= G1STCE \cdot CLK \\ F1STC2 &= G1STCE \cdot F1STC1 \cdot CLK \end{aligned}$$

With F1STC1 and F1STC2 both true, gate G0STE3 is enabled which, in turn, enables G1STE3:

$$\begin{aligned} G0STE3 &= F1STC1 \cdot F1STC2 \\ G1STE3 &= G0STE3 \end{aligned}$$

One output from G1STE3 provides the optional API Watchdog Timer in the AU with a monitor point. If F1STC1 and F1STC2 are not cleared within a reasonable length of time, it indicates that a lengthy or continuous loop of non-interruptable instructions has prevented acknowledgement of all interrupts for too long a period.

Priority Interrupt flip-flop F1SPI1 is armed to set at T3 of SC04 if the current instruction is interruptable at its completion, interrupts are not locked out by console switch, and the delay time counter has completed its count:

$$F1SPI1 = G1STE3 \cdot AUG1WENA \cdot \overline{F1SPI2} \cdot CLK$$

F1SPI1, set, enables drivers D0SPI1, D1SCD1, and D1SCD2. On the Change Detector Cards, the D1SCD1 or D1SCD2 signal is ANDed with the group Priority

driver signal from the Address Generator. The Interrupt flip-flop for the interrupt presently being serviced is cleared. No others are affected.

Signal D0SPI1 is directed to the P-register control and to the sequence control circuits of the AU so that, with the setting of F1SPI2, the P-register does not increment and memory is addressed from the API area.

$$F1SPI2 = G1STE3 \cdot F1SPI1 \cdot SC01 \cdot CLK$$

Bit 7, to form the API starting address of 200<sub>g</sub>, is forced when memory is addressed from the API area (area W):

$$\begin{aligned} AUD1MA07 &= AUG0SAMW \\ AUG0SAMW &= AUG1SWPI \cdot SC01 + TTF \cdot SC03 \\ AUG1SWPI &= F1SPI1 \end{aligned}$$

Memory addressing from W is enabled by:

$$AUD1SAMW = AUG0SAMW$$

The instruction located at the API response address is fetched. At the clock following last pulse of this SC01 time, F1SPI1 clears and drivers D1SCD1 and D1SCD2 go untrue. This allows the Interrupt flip-flop for the interrupt that was just serviced to set again if another interrupt on the same line occurs. Driver D0SPI1 also goes untrue, disabling the addressing of memory from W.

$$\begin{aligned} \overline{F1SPI1} &= LPE \cdot SC01 \cdot CLK \\ \overline{D1SCD1} &= \overline{F1SPI1} \\ \overline{D1SCD2} &= \overline{F1SPI1} \\ \overline{D0SPI1} &= \overline{F1SPI1} \end{aligned}$$

At least one instruction of the normal program is allowed to follow an interrupt response instruction before another interrupt can be serviced. This is accomplished by holding F1STC1, F1STC2, and F1SPI2 set during the execution of the API response instruction and during the fetch of the next instruction.

Clear Priority signals D0SCP1 and D0SCP2 become true at SCA of SC01 for the instruction that follows the API response instruction. These signals "DC clear" F1STC1 and F1STC2. All Priority flip-flops on the Change Detector and Priority Logic cards are also cleared, allowing the No Interrupt Present signal to go true.

$$\begin{aligned} D0SCP1 &= F1SPI2 \cdot \overline{F1SPI1} \cdot SC01 \cdot SCA \\ D0SCP2 &= F1SPI2 \cdot \overline{F1SPI1} \cdot SC01 \cdot SCA \\ \overline{F1STC1} &= D0SCP1 \\ \overline{F1STC2} &= D0SCP1 \\ \overline{F1*GP\#} &= F1*GP\# \cdot \overline{G1*PEL(U)} \\ \overline{G1*PEL(U)} &= D0SCP1(2) + \overline{D1SPA1(2)} + AUD0NIN2 + F0ZM_{xx} \end{aligned}$$

The interrupt cycle is completed when Priority Interrupt

LOGIC PAGE	AREA	PRIORITY	RESPONSE ADDRESS
12, 13	B	1-8	200-207
14, 15	C	9-16	210-217
16, 17	D	17-24	220-227
18, 19	E	25-32	230-237
20, 21	F	33-40	240-247
22-23	G	41-48	250-257
24, 25	H	49-56	260-267
26, 27	I	57-64	270-277
28, 29	J	65-72	300-307
30, 31	K	73-80	310-317
32, 33	L	81-88	320-327
34, 35	M	89-96	330-337
36, 37	N	97-104	340-347
38, 39	P	105-112	350-357
40, 41	Q	113-120	360-367
42, 43	R	121-128	370-377

Table OPR. 2. Priority Grouping

flip-flop F1SP12 clears at the following clock:

$$\overline{F1SP12} = \overline{F1SPI1} \cdot SCA \cdot SC01 \cdot CLK$$

## ECHO INTERRUPT GENERATION

Any interrupt that occurs on a eight point card equipped with optional echo generator gates will cause an echo interrupt to occur when all of the following conditions are met.

- The output of the echo generator is wired to the input of another change detector.
- The first interrupt has been recognized as the one being processed by the API sequencer.
- The Generator Echo Interrupt flip-flop, F1SGEI, is set.

For example, the Generate Echo gate for interrupt 227, designated G0DGE7, is enabled when:

$$G0DGE7 = G1DGP7 \cdot D1DGB1 \cdot D1ADEP \cdot G1DGEU$$

The first three signals identify 227 as the interrupt being processed by the API sequencer in the same way the address generator recognized it as the interrupt for which to generate a response address. That is, G1DGP7 shows that it is the highest priority request in its half-card group of four. D1DGB1 indicates there is no interrupt from the four higher priority interrupts on the same card. And finally, the Group Priority signal, D1ADEP, from the address generator, has identified it as coming from the highest priority interrupting group of two cards and that it is the group that enabled the time counter in the sequence control area.

Signal G1DGEU is true when the Generate Echo Interrupt flip-flop is set:

$$G1*GEL(U) = D0SGEI$$

$$D0SGEI = F1SGEI$$

Generate Echo Interrupt flip-flop F1SGEI sets if the current API response instruction is a DMT and the count passes from a zero to a minus one or when the API response location contains a TIM/TOM control word and the word count reaches a table end condition:

$$F1SGEI = F1SPI2 \cdot AUG1WEKO \cdot CLK$$

$$AUG1WEKO = \overline{A1U23C} \cdot \overline{SCB} \cdot SCA \cdot (AUG0WEK1 + AUG0WEK2)$$

$$AUG0WEK1 = SC02 \cdot DMT$$

$$AUG0WEK2 = SC01 \cdot AUG1HECO (TIM/TOM)$$

## SPECIAL CONTROL

Special controls are important factors in the operation of the API system. These include initialize control, API lockout control, inhibitible interrupt control, mask register control, and control to inhibit all interrupts by special instruction. The mask register and the special

inhibit instruction (IAI<sub>2</sub>) are optional features.

## Initialize Control

When the computer is initialized, the setting of Interrupt flip-flops due to erratic signals on API input lines is prevented. Initialize signals also clear all Priority flip-flops on the Change Detector cards and "DC clear" the Time Counter flip-flops, Priority Interrupt flip-flops, and the Generate Echo flip-flop in the sequence control area.

On the Change Detector cards:

$$\overline{F1*GI\#} = G0*INL(U)$$

$$G0*INL(U) = AUD1NIN2(3)$$

$$\overline{F1*GP\#} = \overline{G1*PEL(U)}$$

$$\overline{G1*PEL(U)} = \overline{G0*PEL(U)}$$

$$\overline{G0*PEL(U)} = AUD0NIN1(2)$$

In the sequence control area, signal D0SINT clears the Time Counter flip-flops, Priority Interrupt flip-flops, and the Generate Echo flip-flop:

$$D0SINT = G1SINI$$

$$G1SINI = G0SPMT$$

$$G0SPMT = INITIALIZE$$

The INITIALIZE signal is from gate G1ZINI if the mask register is provided or from AUD1NIN2 if the mask register option is not specified.

## API Lockout Control

The API Lockout switch located on the programming and maintenance console is enabled only when the console key switch is not OFF. The API Lockout switch is normally used when performing maintenance procedures. The switch is a latching type and when locked in the ON position, all interrupts are inhibited.

Ground from the API Lockout switch causes Enable Interrupt gate signal AU G0WENE to go untrue. The Enable API signal G1WENA is disabled:

$$\overline{AUG0WENE} = AUSWCAPI$$

$$\overline{AUG1WENA} = \overline{AUG0WENE}$$

Priority Interrupt flip-flop F1SPI1, in the sequence control area, is prevented from setting, thus prohibiting the addressing of memory from the API system.

## Inhibitible Interrupt Control

The Inhibit Automatic Interrupt instruction IAI<sub>1</sub> is used to prevent critical program routines from interruption by low priority interrupts. Only those interrupts classified as inhibitible are affected by the IAI<sub>1</sub> instruction. Any half-card group of four interrupts is rendered inhibitible when the appropriate input of the applicable Priority Enable gate G0\*PEL or G0\*PEU is wired to

the output of Permit Automatic Interrupt driver D1SPAI or D1SPA2.

Inhibitible interrupts are inhibited when the IA<sub>1</sub> instruction is decoded which clears the Permit flip-flop and causes D1SPAI and D1SPA2 to go untrue:

$$\overline{D1SPAI} = \overline{AUF1WPMT}$$

$$\overline{D1SPA2} = \overline{AUF1WPMT}$$

Inhibitible interrupts are allowed when the Permit flip-flop is set by the PAI (permit automatic interrupt) instruction.

### Mask Register Control

Half-card groups of four interrupts, whether or not wired as inhibitible, can be inhibited as required during the course of the running program by using the mask register. One register bit, or flip-flop, is needed for each group of four interrupts selected for mask register control. These flip-flops are located on the Change Detector cards. There is one flip-flop for the lower half of the card and another for the upper half of the card.

Each mask register flip-flop is controlled by one of the least significant 16 bits of the A-register. The mask register is loaded by Load Mask Register instruction LMR. API systems with more than 64 interrupts, for which masking is desired, must also use the LMR<sub>2</sub> instruction which loads the upper 16 mask register bits.

A "one" in the appropriate bit position of the mask register will inhibit the particular group of four interrupts even if not wired as inhibitible by the IA<sub>1</sub> instruction. To permit interrupt requests from a given group of four interrupt points, the program must load a zero into the appropriate bit position of the mask register. For normally inhibitible interrupts, the Permit flip-flop must also be set.

The following logic description of the loading and controlling of the mask register by the LMR command (25000 30<sub>2</sub>) is referenced to the timing diagram of Fig. OPR.7. Operation of loading and controlling the

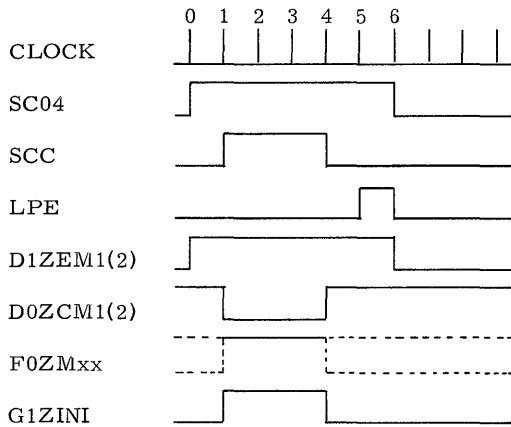
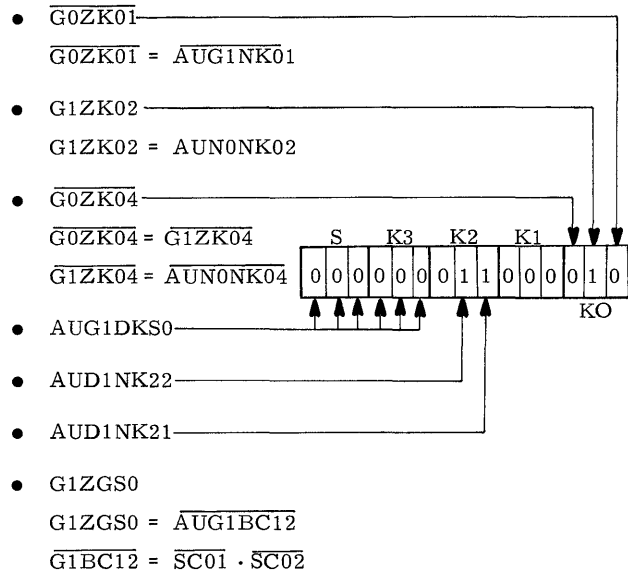


Fig. OPR.7. Mask Register Timing Diagram

upper 16 bits of the mask register by the LMR<sub>2</sub> command is identical, with the exceptions that 25000 30<sub>2</sub> is decoded and corresponding signal names end with the numeral two (2).

The Enable Mask gate, G0ZEM1, on the mask register control card is enabled during sequence state 4 to recognize the GEN II command LMR (2500030<sub>2</sub>) and to cause the execution of that command. G0ZEM1 is true when all of the following signals are true:



Enable Mask driver signal D1ZEM1 becomes true:

$$D1ZEM1 = G0ZEM1$$

During SCC time of sequence state 4, the Clear Mask driver D0ZCM1 goes true and clears all mask register flip-flops for the lower 64 interrupts and halts API sequencing.

$$D0ZCM1 = G1ZCLM$$

$$G1ZCLM = G0ZCLM$$

$$G0ZCLM = (D1ZEM1 \cdot AUD1TSCC) + (D1ZEM2 \cdot AUD1TSCC)$$

A given mask register flip-flop will remain cleared if the corresponding bit of the previously loaded A-register is a zero. However, if a particular group of four interrupts is to be inhibited, a "one" will have been loaded into the corresponding bit position of the A-register. The A-register bit is ANDed with Enable Mask signal D1ZEM1. With both signals true, the mask register flip-flop sets:

$$F1ZM_{xx} = G0ZM_{xx}$$

$$G0ZM_{xx} = D1ZEM1 \cdot AUF1AR_{xx}$$

The clear side output of each mask register flip-flop is wired to an input of its associated Priority Enable gate G0\*PEL or G0\*PEU. If a mask flip-flop is set, the zero signal will cause G0\*PEL(U) and, in turn, G1\*PEL(U) to go untrue, preventing the setting of Priority flip-flops

in the group of four interrupts:

$$\overline{G0*PEL}(U) = F0ZM_{xx}$$

$$\overline{G1*PEL}(U) = \overline{G0*PEL}(U)$$

API sequencing was halted during the setting of the mask register to allow the priority and addressing logic to adjust to the highest priority of the interrupts to be allowed by the mask register. This is accomplished by "DC clearing" the Time Counter flip-flops, Priority Interrupt flip-flops, and the Generate Echo flip-flop with signal D0SINT.

$$D0SINT = G1SINI$$

$$G1SINI = G0SPMT$$

$$G0SPMT = G1ZINI$$

$$G1ZINI = D0ZCM1 + D0ZCM2$$

### Special Instruction IAI<sub>2</sub>

The IAI<sub>2</sub> instruction inhibits all interrupts, both those that are inhibitible and those that are non-inhibitible by the IAI<sub>1</sub> instruction, provided the Permit flip-flop is clear. The decoding and control structure for the IAI<sub>2</sub> instruction, a GEN II internal command, comes as part of the control circuit for the optional mask register.

The Set Inhibit Flip-flop gate, G0ZSIF, is enabled during sequence state 4 to recognize the IAI<sub>2</sub> command (25000304g) and to cause the execution of that command. G0ZSIF is true when all of the following signals are true:

- $\overline{G0ZK01}$   
 $\overline{G0ZK01} = \overline{AUG1NK01}$
- $\overline{G0ZK02}$   
 $\overline{G0ZK02} = \overline{G1ZK02}$   
 $\overline{G1ZK02} = \overline{AUN0NK02}$
- $G1ZK04$   
 $G1ZK04 = AUN0NK04$
- $AUG1DKS0$
- $AUD1NK22$
- $AUD1NK21$
- $G1ZGS0$   
 $G1ZGS0 = \overline{AUG1BC12}$   
 $\overline{G1BC12} = \overline{SC01} \cdot \overline{SC02}$

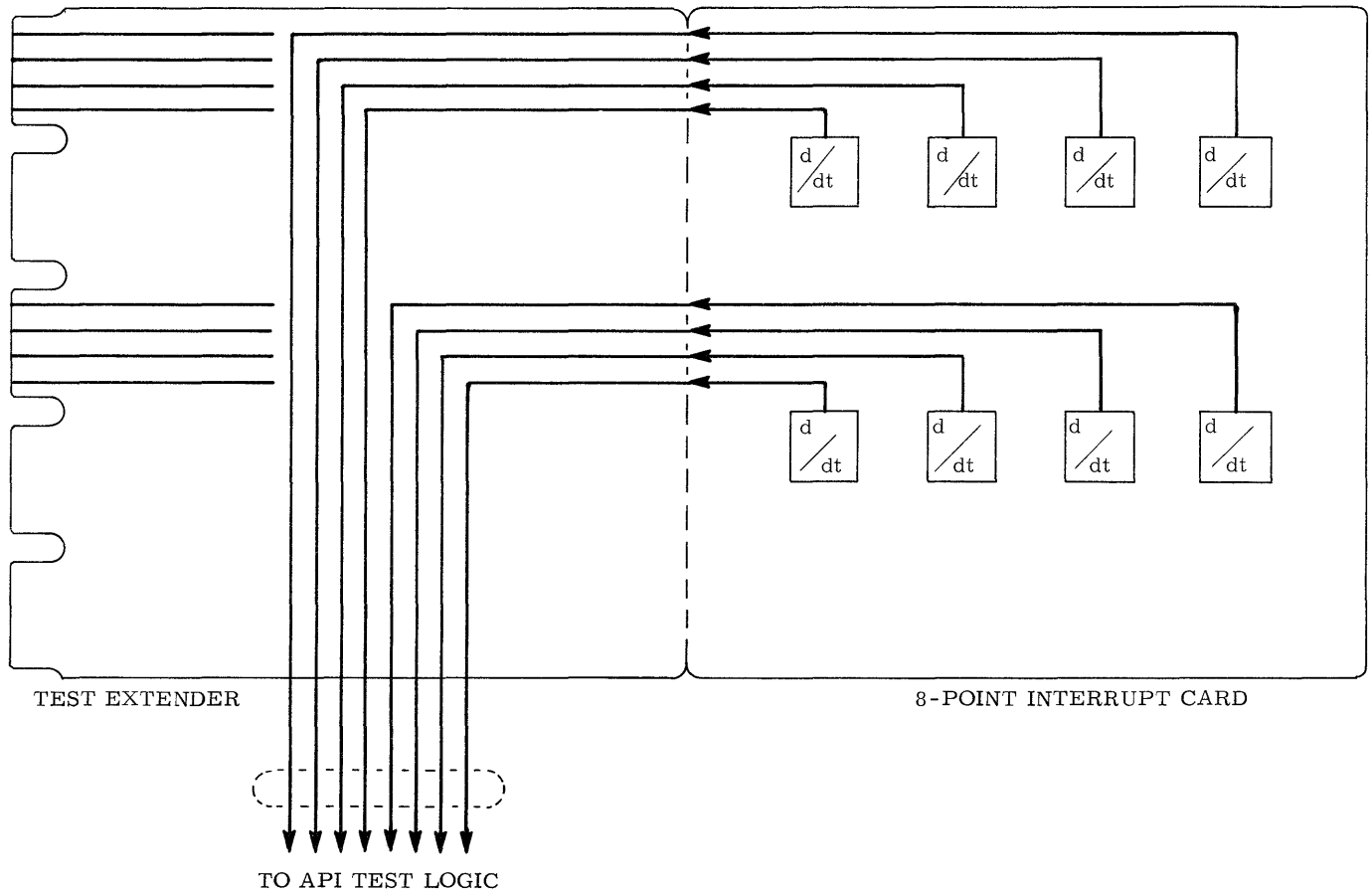
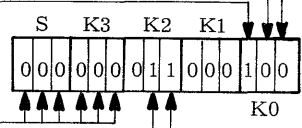


Fig. OPR. 8. Simplified Schematic of Interrupt Card on Test Extender



The Inhibit flip-flop sets with G0ZSIF if the Permit flip-flop is clear:

$$F1ZINH = G0ZSIF$$

$$F0ZINH = F1ZINH \cdot \overline{AUF0WPMT} \cdot \overline{AUD0NINI}$$

All interrupts are prevented from interrupting the program because API sequencing is halted. This is accomplished by "DC clearing" the Time Counter flip-flops, the Priority Interrupt flip-flops, and the Generate Echo flip-flop with signal D0SINT.

$$D0SINT = G1SINI$$

$$G1SINI = G0SPMT$$

$$G0SPMT = G1ZINI$$

$$G1ZINI = F0ZINH + AUD0NINI$$

The Inhibit flip-flop is cleared either by the PAI instruction which sets the Permit flip-flop or when the computer is initialized:

$$\overline{F1ZINH} = AUF0WPMT + AUD0NINI$$

### API Test

To use the API test feature, the Change Detector card is pulled from its slot and plugged into one end of the special test extender. The other end of the extender is plugged into the vacated card slot. The nine-pin plug on the eight-conductor cable is mated with the connector on the rear of the mask register control card. This procedure opens the normal API input path and, instead, connects the inputs of the change detectors to outputs of gates in the API Test logic via the eight-conductor

cable. All other leads pass directly through the test extender. See Figure OPR. 8.

Each of eight GEN II instructions is used to set one Interrupt flip-flop on the eight-point card under test. A decoded API test instruction enables an associated test gate. An interrupt is simulated when the gate's output goes from 3.6 volts to zero volts during the execution of the instruction. The list below shows which test gate is enabled and which Interrupt flip-flop sets for a given instruction.

Octal Code	Test Gate	Interrupt Flip-flop
25000210	G0T200	F1*GI0
25000220	G0T201	F1*GI1
25000242	G0T202	F1*GI2
25000212	G0T203	F1*GI3
25000222	G0T204	F1*GI4
25000244	G0T205	F1*GI5
25000214	G0T206	F1*GI6
25000224	G0T207	F1*GI7

In addition to the decoding logic for the S and K bits of the instruction, all test gates are enabled with signal  $\overline{AUG1BC12}$ . This inhibits decoding during sequence states 1 and 2. Sequence state 2, however, will not normally be used because API test instructions are not usually indexed.

# TABLE OF CONTENTS

## DISK SUB-SYSTEM 4515B/4547B

### INTRODUCTION

DISK SPECIFICATIONS

POWER REQUIREMENTS

DISK SUB-SYSTEMS OPTIONS

PRINT TREE

### FUNCTIONAL DESCRIPTION

BASIC READ/WRITE THEORY

PROGRAM CONTROL

GEN II Command

Control Word I

Control Word II

Control Word III

COUPLER FUNCTIONS AND COMPONENTS

Functions

Components

CONTROLLER FUNCTIONS AND COMPONENTS

FUNCTIONS

COMPONENTS

Registers

Timing

Checking

### COMMAND DESCRIPTION

**OUT COMMAND SS0**

**OUT COMMAND SS1**

**OUT COMMAND SS3**

**OUT COMMAND SS4**

**OUT COMMAND SS5**

BLOCK/WORD COUNT = 0

BLOCK/WORD COUNT  $\neq$  0

**OUT COMMAND SS6**

INITIAL SEQUENCING (SS5-SS6)

DESIRED CYLINDER INCREMENT (SS7-SS6)

### OUT COMMAND SS7

WRITE OPERATION

READ OPERATION

### WRITE HEADER OPERATION

### READ STATUS SS0

### READ STATUS SS1

### READ STATUS SS2

### NON-SEQUENCING COMMANDS

RETURN TO ZERO SEEK

ACTIVATE

ABORT COMMAND

JNE COMMAND

Bit 9 Unit Select Error

Bit 8 Timing Error

Bit 7 Selected Seek Error

Bit 6 Pack Not On-Line

Bit 5 Write Protect Error

Bit 4 Address Check-Sum Error

Bit 3 Address Compare Error

Bit 2 Limit Error

Bit 1 File Unsafe Error

Bit 0 Data Check-Sum Error

JNR COMMAND

### MAINTENANCE CONSOLE DESCRIPTION

SELECTOR SWITCH (CSW10)

Position 1

Position 2

Position 3

Position 4

Position 5

NORMAL/HEADER SWITCH (CSW05)

MEMORY WRITE PROTECT SWITCHES

## LIST OF ILLUSTRATIONS

- Fig. INT. 1. Basic Block Diagram-Disk Sub-Unit
- Fig. FNL. 1. Magnetic Lines of Force
- Fig. FNL. 2. "Horseshoe" Magnet Showing Concentration of Magnetic Field
- Fig. FNL. 3. Lines of Force Inducing Magnetism
- Fig. FNL. 4. Inducing Current in a Coil
- Fig. FNL. 5. Write Current Path
- Fig. FNL. 6. Basic Block Diagram-4515B Coupler
- Fig. FNL. 7. Basic Block Diagram-4547B Controller
- Fig. OUT 0. 1. Timing Diagram-SS0
- Fig. OUT 0. 2. Block Diagram, OUT Command-SS0
- Fig. OUT 1. 1. Timing Diagram, OUT Command-SS1
- Fig. OUT 1. 2. Block Diagram, OUT Command-SS1
- Fig. OUT 3. 1. Timing Diagram, OUT Command-SS3
- Fig. OUT 3. 2. Block Diagram, OUT Command-SS3
- Fig. OUT 4. 1. Cylinder Difference Calculation
- Fig. OUT 4. 2. Block Diagram, OUT Command-SS4
- Fig. OUT 4. 3. Timing Diagram, OUT Command-SS4
- Fig. OUT 5. 1. Block Diagram, OUT Command-SS5
- Fig. OUT 5. 2. Timing Diagram, OUT Command-SS5  
Block/Word Count  $\neq 0$
- Fig. OUT 5. 3. Timing Diagram, OUT Command-SS5  
Block/Word Count = 0
- Fig. OUT 6. 1. Timing Diagram, OUT Command-SS6
- Fig. OUT 7. 1. Sector Format
- Fig. OUT 7. 2. Data Transfer Path-WRITE
- Fig. OUT 7. 3. Timing Diagram-WRITE
- Fig. OUT 7. 4. Timing Diagram-WRITE
- Fig. OUT 7. 5. Data Transfer Path-READ
- Fig. OUT 7. 6. Timing Diagram-READ
- Fig. OUT 7. 7. Timing Diagram-READ
- Fig. WHD 7. 1. Sector Format After Write Header
- Fig. WHD 7. 2. Block Diagram, Write Header-SS7
- Fig. WHD 7. 3. Timing Diagram, Write Header-SS7
- Fig. WHD 7. 4. Flow Chart-OUT Command (Read/Write, Seek and Write Header Operation).
- Fig. RDS 0. 1. Flow Chart, Read Status Command
- Fig. RDS 0. 2. Block Diagram, Read Status-SS0
- Fig. RDS 0. 3. Timing Diagram, Read Status-SS0
- Fig. RDS 1. 1. Timing Diagram, Read Status-SS1
- Fig. RDS 1. 2. Block Diagram, Read Status-SS1
- Fig. RDS 2. 1. Timing Diagram, Read Status-SS2
- Fig. RDS 2. 2. Block Diagram, Read Status-SS2
- Fig. NSC. 1. Block Diagram, Return-To-Zero Seek
- Fig. NSC. 2. Timing Diagram, Return-To-Zero Seek
- Fig. NSC. 3. Block Diagram, Abort
- Fig. NSC. 4. Timing Diagram, Abort
- Fig. NSC. 5. PNL Error Detection
- Fig. NSC. 6. Write/Erase-Select Error Circuit
- Fig. MCS. 1. Maintenance Console

## LIST OF TABLES

- Table FNL. 1. Pointer Word Addresses

# INTRODUCTION

The Model 4515B/4547B Disk Sub-System connects to the GE-PAC\* 4020 system and consists of three major components: the Model 4DP4515B Disk Coupler; the Model 4DP4547B Disk Controller; and either the Model 4DP4548A100 (100 tracks) or the Model 4DP4548A200 (200 tracks) Disk Unit.

The Disk Unit functions as a high capacity, rapid access memory device. Utilizing GEN 2 encoded instructions, it is possible to select a starting disk address and transfer from 1 to 256 blocks of information between Core Memory and the Disk Memory with a single instruction. Each block contains 64 24-bit words of information. The transfer of information from Core Memory to the Disk Memory is referred to as a Write operation, while Disk to Core transfer is defined as a Read operation. After the initial instruction is issued by the AU, the Disk Sub-System communicates directly with Core Memory for transfer of control words and data.

The initial instruction, which determines the type of operation to be performed on the Disk Unit, is decoded in the Coupler. The Coupler module, located in the same cabinet as the Core Memory, receives data from the Core Memory during the Write operation and transmits data to the Core Memory during a Read operation. It also addresses control words, during the Out command, which are transferred to the Controller as unit select, disk address, starting core address for data transfers, and word count information.

Notification of the type of operation decoded within the Coupler is sent to the Controller, located in another cabinet, to record the information by setting a flip-flop corresponding to that type of operation. (Refer to "Program Control" within the Functional Description section for a description of the types of operations.) The Controller will then govern the operation, selecting the desired Disk Unit and address, using the control words provided by the Coupler. A "seek" will be

initiated, by the Controller, to locate the desired address on the disk, and upon locating it, the Controller will request data from Core for a Write operation or transfer data from Disk to Core during a Read operation.

An indication of errors is kept in the Controller and made available to a pre-determined Core Address via the Read Status command. Upon detecting an error, any active Controller operation will be terminated.

The Disk Unit contains an interchangeable "Disk Pack", upon which the information is recorded, consisting of ten recording surfaces located on six disks. Magnetic recording is used to store the information on the disks and is described under the heading "Basic Read/Write Theory" within this section.

The information is recorded on a disk in concentric circles (tracks), located at evenly spaced intervals from the outer track to the inner track. The outer track is referred to as track 00 and the inner track is either track 99 or 199, dependent upon whether the Disk Unit is a 100 or 200 track model.

Each recording surface has a corresponding read/write head and all ten heads are connected to a common arm, which can be moved radially across the disks to position the heads over any specific track. The combination of the ten recording surfaces, located on six disks with a common hub so that they are located one above another, will make available ten tracks of information whenever the heads are positioned over any given track. This arrangement is referred to as a "cylinder" and the information is organized on the disks in such a manner, e.g., information will be written on, or read from, cylinder zero (track zero, heads 0-9) before advancing to cylinder one (track one, heads 0-9).

The information is further subdivided into 16 sectors within a track of information. The sector will contain

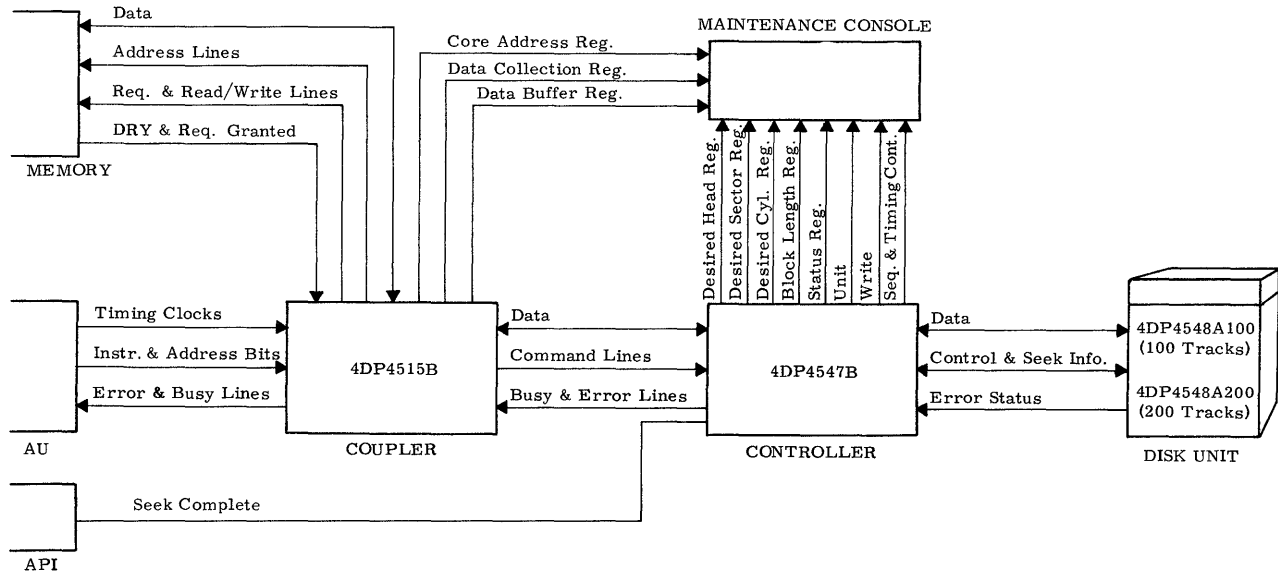


Fig. INT. 1. Basic Block Diagram - Disk Sub-System

\* Registered Trademark of General Electric Company

the address of that block of information, i.e., cylinder, sector, and head, as well as the block of information (64 24-bit words). This represents the smallest amount of information that can be recorded on the disk by a single command.

## DISK SPECIFICATIONS

- Storage Capacity:
  - Model 4548A100 - 1,024,000, 24-bit words (100 track)
  - Model 4548A200 - 2,048,000, 24-bit words. (200 track)
- Transfer Rate:
  - 52,200, 24-bit words per second.
- Access Times:
  - Rotational delay - 25 ms max.
  - Track-to-track, min. (1 track seek) - 30 ms
  - Track-to-track, max. (full seek) - 165 ms.
- Sub-System Configuration:
  - No. of Disk Units/Controller - 4 max.
  - No. of Controllers/System - 3 max.
- Disk Record Format:
  - Cylinders/Disk Pack (4548A100) - 100
  - Cylinders/Disk Pack (4548A200) - 200
  - Tracks/Cylinder - 10
  - Heads/Cylinder - 10
  - Sectors/Track - 16
  - Sectors/Cylinder - 160
  - Words/Track - 1,024
  - Words/Cylinder - 10,240
  - Words/Sector - 64
- Disk Pack Specifications (Packs Interchangeable):
  - No. of Disks - 6
  - Recording surfaces - 10
  - Disk diameter - 14"
  - Disk surface - Magnetic Oxide.
- Validity Checks:
  - Address check-sum for address bits.
  - Data check-sum for data bits.

- Disk Unit Size:
  - 41"H x 36"D x 24"W.
  - Weight - 500 lbs.
- Environment:
  - Temperature - 65 - 85°F.
  - Relative Humidity - 20 - 80%.

## POWER REQUIREMENTS

208V AC  $\pm 10\%$ , 3-phase @ 3 amps/phase, 60 or 50 Hz. +1%, -2% or 230V AC  $\pm 10\%$ , 1-phase @ 10 amps, 60 or 50 Hz. +1%, -2%.

## DISK SUB-SYSTEM OPTIONS

The following description is that of the various options provided for the Disk Sub-Systems, along with the respective model numbers relating to these options. The model number of the 4515B Disk Coupler will be concluded with four numbers representing the options of that module. These numbers are referred to as WXYZ, in a general sense, in whose place a number representing a specific option will be substituted. The options are as listed below.

- 4515B Disk Coupler (Option 4DP4515BWXYZ):

W = 0

XYZ - Memory Multiplex Channel Options

X - Channel 1 Option  
 Channel 1 = 1  
 Channel 1 = 0

Y - Channel 2 Option  
 Channel 2 = 1  
 Channel 2 = 0

Z - Channel 3 Option  
 Channel 3 = 1  
 Channel 3 = 0

By utilizing the above combination of XYZ bits, any combination of up to three couplers may be assigned to a system, with channel 1 receiving highest priority and channel 3 receiving the lowest. When assigning the couplers to channels 1 through 3, the following addresses will be assigned to the couplers.

Channel 1:

Pointer word address = 00g  
 GEN 2 K<sub>3</sub>K<sub>2</sub>K<sub>1</sub>K<sub>0</sub> address = 1000g

Channel 2:

Pointer word address = 11g  
 GEN 2 K<sub>3</sub>K<sub>2</sub>K<sub>1</sub>K<sub>0</sub> address = 2000g

Channel 3:

Pointer word address = 12<sub>8</sub>

GEN 2 K<sub>3</sub>K<sub>2</sub>K<sub>1</sub>K<sub>0</sub> address = 4000<sub>8</sub>

Disk Couplers will be assigned the highest channel priority available, after the assignment of a device such as a drum, which would ordinarily receive a higher priority assignment. For this reason a Disk Coupler may be assigned to any of the above channels. For examples of typical priority assignments involving Drum and Disk Sub-Systems, refer to the "Out Command" description, under "Program Control" in the Functional Description section.

The Disk Controller and Disk Unit model numbers will be concluded with the use of three numbers representing the options of the Controller and Disk Unit. The numbers are referred to as XYZ, in a general sense, with the numbers, listing the options, representing these bits as shown in the following.

● 4547B Disk Controller (Option 4DP4547AXYZ):

X = 1

Y - Disk Drive Capability

Y = 0 One Disk Unit

Y = 1 Two Disk Units

Y = 2 Three Disk Units

Y = 3 Four Disk Units

Z - AC Frequency Options

Z = 0 60 Hz.

Z = 1 50 Hz.

● 4548A Disk Unit (Option 4DP4548AXYZ):

X - Disk Storage Size

X = 1 One million words

X = 2 Two million words

Y = 0

Z - AC Power Options

Z = 0 208V AC, 60 Hz., 3∅

Z = 1 208V AC, 50 Hz., 3∅

Z = 2 230V AC, 60 Hz., 1∅

Z = 3 230V AC, 50 Hz., 1∅

### PRINT TREE

● 4515B Disk Coupler (4020 System)

Model List 4DP4515B

Logic 68C972617

● 4547B Disk Controller

Model List 4DP4547B

Logic 68C972186

● 4548A Disk Unit

Model List 4DP4548A

Assembly Modification Drawing  
(100 track models) 68A975864G1

Assembly Modification Drawing  
(200 track models) 68A975864G2

# FUNCTIONAL DESCRIPTION

## BASIC READ/WRITE THEORY

Disk recording utilizes the principal of inducing concentrated flux fields on the surface of the disk and retaining these fields for later pickup. The process of inducing the fields upon the disk is referred to as writing, while the pickup of these fields is referred to as reading. The recording medium is a magnetic oxide coated disk and the writing and reading is accomplished using a time sharing transducer, referred to as the read/write head.

In order to understand the recording principles involved, it is necessary to examine the theory of magnetism. All magnets have at least two poles, a north pole and a south pole. There is a magnetic field around the magnet consisting of imaginary lines of force. The lines of force emanate from the north pole and enter the south pole, returning to the north pole through the magnet itself, thus forming a closed loop. (See Fig. FNL. 1.) The entire quantity of magnetic lines surrounding a magnet is called magnetic flux, while the number of lines per unit area is referred to as flux density.

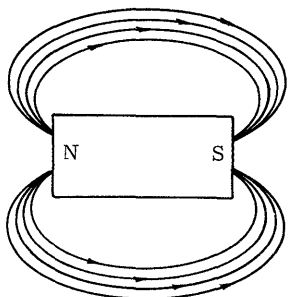


Fig. FNL. 1. Magnetic Lines of Force

When a magnet is bent to form a loop without the ends touching, as shown in Fig. FNL. 2, there is still a north and south pole, but the magnetic field is of shorter length and greater concentration than the bar magnet.

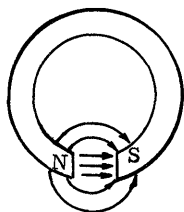


Fig. FNL. 2. "Horseshoe" Magnet Showing Concentration of Magnetic Field

One characteristic of the imaginary lines of force in a magnetic field is that they tend to take the path of least resistance, or in other words flow through the material that has the greater permeability (conductivity for magnetic flux). Air offers more resistance to the lines of force than a piece of iron or steel.

If a piece of iron is brought in close proximity to the gap of the horseshoe magnet as shown in Fig. FNL. 3, the lines of force will tend to bend and flow through the iron. This is the principle of recording utilized, with an electromagnet being used as a magnet.

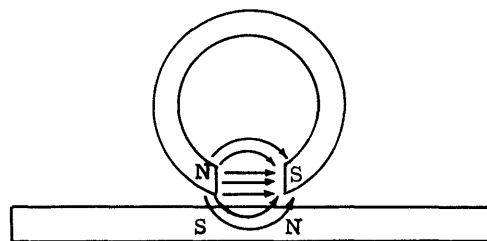


Fig. FNL. 3. Lines of Force Inducing Magnetism

The electromagnet sets up a north and south pole in relation to the direction of electron flow through its coil. With electron flow in the direction shown in Fig. FNL. 4, the magnet will set up a corresponding north/south pole relationship as shown. Reversing the direction of electron flow will, in turn, reverse the polarity of the magnet. As also indicated by the figure, the induced field will be of opposite polarity of the electromagnet. Thereby, information may be recorded by varying the direction of current flow through the electromagnet, or in this case, write head, as the recording medium passes beneath the head.

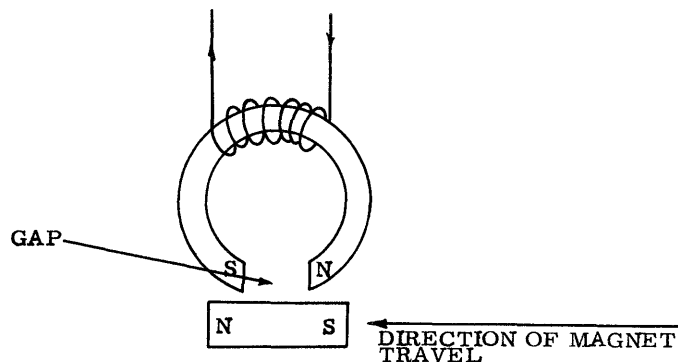


Fig. FNL. 4. Inducing Current in a Coil

Once recorded, the information remains until written on again. By removing current from the electromagnet, and allowing current to be induced into the coil by rotating the disk beneath it and having the recorded information induce current into the coil in relation to the manner of which it was recorded, we may read the information back (see Fig. FNL. 4). The signal is then amplified and shaped into pulses to be detected as a binary one or zero.

The write circuitry employs a non-return to zero method of recording, during which there is a constant flow of current through the write head. The write current is alternately supplied from the two legs of the write amplifier, through the write head to the grounded center tap, as shown in Fig. FNL. 5. An erase head, part of the head mechanism, precedes the read/write head as the disk spins beneath it. The erase head is enabled during a Write operation and erases previously recorded information by having a DC current passed through the erase head, thereby aligning the disk track flux in a constant direction. The write head, positioned behind the erase head physically, as well as in time, then

ALTERNATELY SWITCHED ON - FROM UKA/UKB GATED TOGGLE FLIP-FLOP

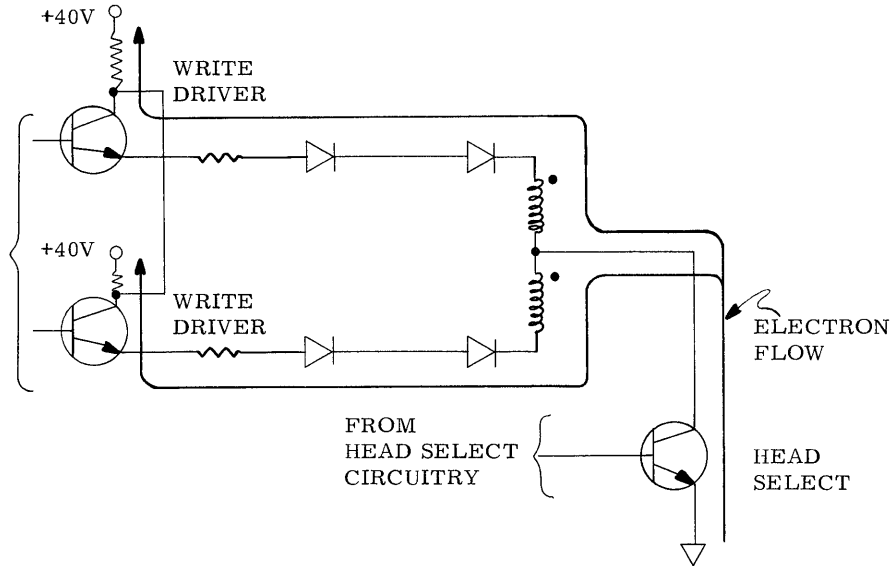


Fig. FNL. 5. Write Current Path

writes data provided by the controller, over the erased surface.

During the Write operation, pulses are sent to the write circuitry at regular intervals. These intervals are known as clock A's (clock 1's). At a period halfway between these clocks, another clock is also generated. This clock is referred to as clock B (clock 2). During this time, the data to be written on the disk is sampled to determine its value (binary "one" or "zero"). If the data to be recorded is a "one", the pulse is allowed to be recorded. If the data is a "zero", the pulse is not recorded. Whenever the data is read back, it is examined at clock B (clock 2) time to determine its value. A sample waveform of data is illustrated on Fig. OUT 7. 6.

### PROGRAM CONTROL

Program control of the Disk Unit is accomplished by use of the GEN II Command. A breakdown of the command structure is shown in the following:

#### GEN II Command

23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

OP CODE	X	(S)	(K3)	(K2)	NOT USED
2	5	X	1 - 7	OPTIONAL	

In the above illustration, bits 23 - 18 equal an octal twenty-five, which identifies the command as a GEN II. Bits 17 - 15 are not normally used (see manual for AU, under "GEN II Commands" for restrictions on using these bits) and, therefore, remain an octal zero. The next group of bits, 14 - 12, identify the type of operation to be performed by this command and are further discussed in the following description. Bits 11-6 constitute the K3/K2 address assigned to the Disk Sub-System and these bits vary with individual systems. (This address may be determined by referring to the individual system's I/O Summary). The remaining bits, 5-0, are not used

for the Disk operation and are ignored by the hardware.

The GEN II command, upon execution by the AU, informs the Disk Coupler of the type of operation to be performed, via the "S" bits. The Coupler, in turn, will transmit this information to the Disk Controller, which synchronizes operations within the Disk Coupler and Disk Unit.

The Controller remains in a passive condition until it receives a command. It will then remain active until its operation is successfully completed, it receives an abort command, or an error condition is detected. The following is a description of acceptable commands, whose operation is determined by the "S" bits of the GEN II Command.

- OUT COMMAND (S = 4) - The Out command is used to initiate a "seek" operation or to perform a read/write operation. There are two types of write operations which may be performed: a "normal" write; and a "header" write. The command instructions are similar for both writes, but the Controller differentiates between the two by the status of the "NORMAL/HEADER" switch, located on the maintenance console. A normal write may be executed with up to 256 blocks of information being transferred, but the header write should be restricted to one track of information at a time, i.e. 32 words, to be used to establish address and other non-data information on the disk.

The execution of an Out command results in three control words being transferred to the Disk Coupler from the Core Memory. The contents of these control words are illustrated by the following:



### Control Word I

23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

R/W*	IGNORED	UNIT	CYLINDER	HEAD	SECTOR
------	---------	------	----------	------	--------

\*0 = READ                      1 = WRITE

### Control Word II

23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

IGNORED	BLOCK COUNT	IBH
---------	-------------	-----

IBH = INHIBITED BY HARDWARE

### Control Word III

23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

STARTING CORE ADDRESS
-----------------------

The address of the first control word is located in a core cell address as specified by the Disk pointer word. The pointer word is located in a specific core address, determined by individual system organization. Examples of typical priority assignments are shown in the following table:

DRUM UNIT	DISK COUPLERS		
	No. 1	No. 2	No. 3
00 <sub>8</sub>	X	X	X
00 <sub>8</sub>	11 <sub>8</sub>	X	X
00 <sub>8</sub>	11 <sub>8</sub>	12 <sub>8</sub>	X
00 <sub>8</sub>	11 <sub>8</sub>	12 <sub>8</sub>	13 <sub>8</sub>
00 <sub>8</sub>	11 <sub>8</sub>	12 <sub>8</sub>	13 <sub>8</sub>
X	00 <sub>8</sub>	X	X
X	00 <sub>8</sub>	11 <sub>8</sub>	X
X	00 <sub>8</sub>	11 <sub>8</sub>	12 <sub>8</sub>
X	00 <sub>8</sub>	11 <sub>8</sub>	12 <sub>8</sub>

Table FNL. 1. Pointer Word Addresses (X = NO UNIT)

Initially, the Out command presets the Core Address Register (CAR), within the Coupler, to the address of the pointer word. The pointer word is then addressed and the contents transferred to the CAR. The contents of the CAR are then used to address the first control word from Memory.

The operation to be performed is determined by bit 23 of the first control word. Bit 23 equal to a "one" specifies a Write operation, while bit 23 would be a "zero" for a Read operation. Also contained in the first control word is the information necessary to determine which of four Disk Units is to be accessed (bits 16 and 17), the starting cylinder address (bits 8 - 15), the starting head address (bits 4 - 7), and the starting sector address (bits 0 - 3). Upon receipt of control word

one, the contents of the CAR are incremented by one to provide the effective address of control word two. Control word two is then addressed and brought from Memory.

Control word two contains a count (bits 0 - 13) representing the number of words to be transferred between the Disk Memory and Core Memory. The address contained in the CAR is again incremented to obtain the address of the third control word, which is then addressed and brought from Memory.

Control word three is brought to the Coupler and stored into the CAR. The information is the starting address of the first word of data to be read/written. During the read/write operation, the CAR is incremented, with each data transfer, through the required number of transfers specified by the block/word length of control word two.

Upon receipt of all three control words, the Controller directs the Disk Unit to position the heads to the desired cylinder, selects a head, and when the correct sector arrives under the head, the read/write operation commences. The transfer will then continue until the Block/Word Count Register reaches zero (successful completion), an abort command is executed, or an error condition occurs.

Initiating an Out command with a block count equal to zero will result in a head movement to the address specified in control word one. The read/write operation will be ignored as a result of the absence of a block count, but the control words will be transferred to indicate the desired disk address. The Controller will then issue a "seek" signal to the Disk Unit, at which time the Controller will go "not busy", while the Disk controls the seek. This feature, known as the "seek overlap", may be used to advantage by directing a Disk Unit to perform a seek operation while another Unit is being accessed by the Controller.

After the heads arrive at the desired address, which may take 190 ms after initiating the seek signal, the Controller, upon determining address comparison, will begin the read/write operation. If the operation continues until the last sector of a cylinder, and the word count contained in the Block/Word Count Register in the Controller has not reached zero, the operation will automatically wait until the heads position to the next higher cylinder, and address comparison takes place, before continuing the operation. Cylinder incrementing may continue until cylinder 99<sub>10</sub> on the 4548A100 model Disk, or cylinder number 199<sub>10</sub> on the 4548A200 model Disk. Attempts to address beyond the limits of the Disk will result in either a "selected seek" or "limit" error, depending upon the Disk model. (See the JNE Test description, under the respective error conditions).

Execution of an Out command will clear the Status Register. An error check should, therefore, be made before executing an Out command, since the error may remain in the Sub-System, which, upon detection, would terminate the operation.

- **READ STATUS (S = 5, GEN II IN COMMAND)**  
Execution of a Read Status instruction will result in the Controller Status Register's 10 bits being transferred to the least significant 10 bits (9 - 0) of the pointer word, located in Core Memory. The Status Register indicates which, if any, errors have occurred within the Disk Sub-System. A "one" bit in any of the bit positions indicates an error, for which that respective bit pertains, has occurred. All of the errors are also ORed to the error line to notify the AU and API of an error condition. The errors to which the bits pertain, along with a brief description of their significance, are described under the heading "JNE COMMAND".
- **RETURN-TO-ZERO SEEK (S = 2)**  
Returns the head mechanism to position over cylinder zero. Primarily used to return the heads to a known position (cylinder zero) if a Disk "seek" error has occurred. This command will also clear the seek error flip-flop in the Disk.
- **ABORT (S = 3)**  
The Abort command will terminate any active (Out command) Controller operation as soon as possible. The abort flip-flop, in the Controller, will record the fact that an Abort command has been executed and upon reaching a point in the operation where it is possible to terminate, the sequence state counter will return to zero and the Controller busy line will go "not busy".  
  
If the Abort command is received during a "seek" operation, the seek will be completed, however, the busy line will go "not busy" upon receiving the Abort command. If the Abort command is executed during a Write operation, the operation will terminate at the end of the current sector. Execution of an Abort command, whenever the Controller is not busy, will have no effect.
- **ACTIVATE (S = 1)**  
The execution of the Activate command results in the Controller's busy line being toggled, if the Controller is in a not busy condition. If the Controller is busy at the time of the Activate, no change will be reflected on the busy line.
- **JUMP IF NO ERROR (S = 7)**  
The execution of the JNE command takes place within the AU. If the error line from the Disk Sub-System indicates an error condition, the program sequence (P Register) will be incremented by one. If there are no errors present at the time of the test, the program sequence will increment by two.
- **JUMP IF NOT READY (S = 6)**  
The execution of the JNR command also takes place within the AU. It samples the condition of the busy line from the Disk Sub-System. If the Controller is busy during the test, the program sequence (P Register) is incremented by two. If the Controller is found not busy, the program sequence is incremented by one.

## COUPLER FUNCTIONS AND COMPONENTS

The following is intended as a general description of the Model 4DP4515B Disk Coupler; to define its basic functions and contents. Further references to the Coupler will be included within the command descriptions.

### Functions

- Instruction decoding
- Memory addressing of control words and read/write information
- Data Buffering
  - a. Memory to Controller (control words/write information)
  - b. Controller to Memory (read information/read status bits)

### Components

- Data Buffer Register (DBR) - A 24 bit register which buffers control words (from memory to DCR/CAR), read data (from DCR to Memory), write data (from Memory to DCR), and Read Status information (from DCR to Memory). Information to/from this register is transferred parallel.
- Data Collection Register (DCR) - A 24 bit register capable of receiving and transferring information parallel (from DBR), or transferring and receiving information serially (to/from Controller).
- Core Address Register (CAR) - A 15 bit register used for addressing control words, read/write information, and the pointer word. There is also control circuitry within the Coupler, to direct information to the Controller or Memory. Parity checking in the Coupler is not used. Information validity is accomplished in the Controller by means of a check-sum comparison.

## CONTROLLER FUNCTIONS AND COMPONENTS

The model 4DP4547B Controller is intended to be used to control the operation of the Models 4DP4548A100 and 4DP4548A200 Disk Units. It receives its control information and data from the 4DP4515B Disk Coupler. The Controller may be located at distances up to 50 feet from the Coupler. Control information and data exchange between the Controller and Coupler is serial. Read/write data between the Controller and Disk is also serially transmitted.

The main functions of the Controller, along with a general description of its main components, are given below. Further references to the Controller will be included within the command descriptions.

### FUNCTIONS

- Selects Disk Unit
- Provides desired address and compares against current address

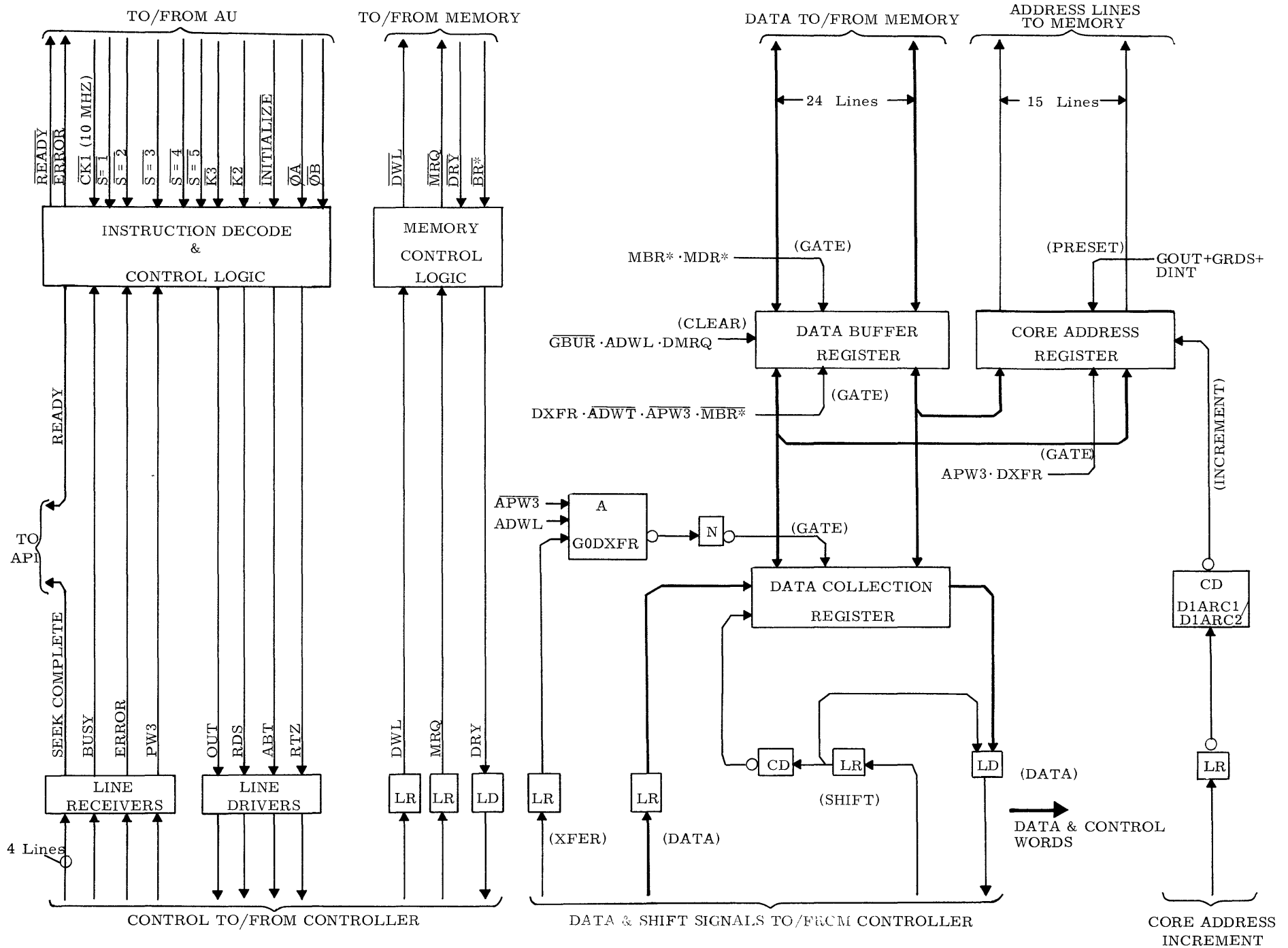


Fig. FNL. 6. Basic Block Diagram-4515B Disk Coupler

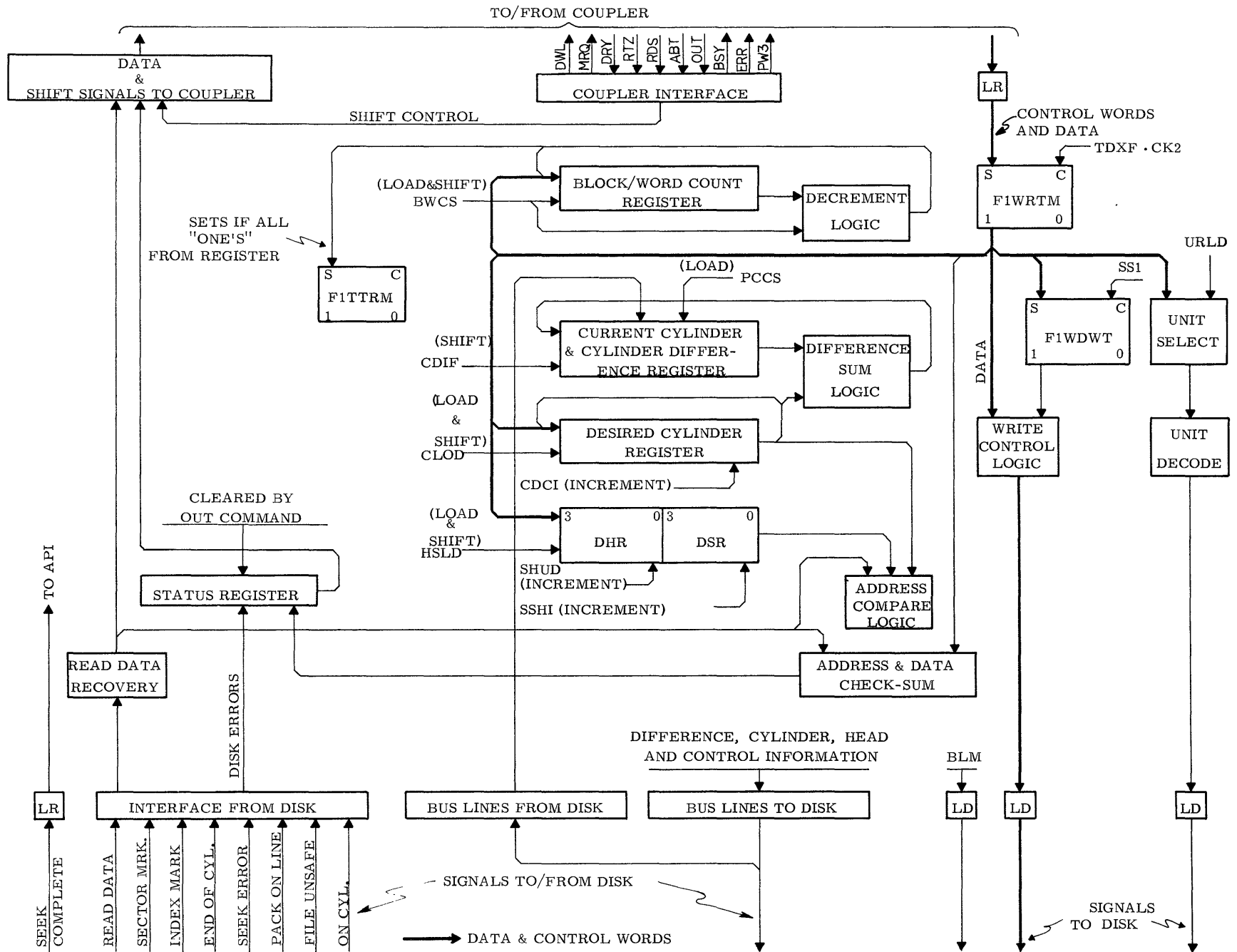


Fig. FNL. 7. Basic Block Diagram-4547B Controller

- Generates number of cylinders head must be moved and direction (obtained from difference sum between desired and current cylinders)
- Buffers data from Coupler and transmits it to Disk after retiming it to Disk timing (Write operation)
- Buffers data between Disk and Coupler (Read operation)
- Provides timing to control transfers to/from Disk/Coupler
- Checks validity of data
  - a. Data from Disk during Read operation
  - b. Address from Coupler during Write Header operation
- Monitors error status of Disk Sub-System

## COMPONENTS

### Registers

- Unit Select - Receives and holds desired unit information from Coupler.
- Block-Word Count - Holds the count of the desired number of words to be read/written. Initially loaded during control word two time and decremented with each word of data read or written on disk. When register is decremented from all zeroes, the data transfer is recognized as complete.
- Desired Cylinder - Initially loaded during control word one time. The desired cylinder count is incremented by one each time the end of the current cylinder is detected during a read/write operation and dictates the cylinder from which data transfers will be made.
- Current Cylinder - Initially loaded during control word two time of Out command or Write Header operation. The current cylinder address is held in a register within the Disk Unit. It is read into the Controller and compared with the address in the Controller's Desired Cylinder Register. The difference sum is then generated and placed into the Current Cylinder Register, from which it is sent to the Disk to determine the amount of head movement to take place. Whenever head movement is required, due to end of cylinder detection, the current cylinder address is again read into the Controller and compared against the Desired Cylinder Register to determine direction and amount of head movement needed.
- Desired Head Address - Initially loaded during control word one time. The register is updated under the control of the head update flip-flop, F1HHUD. The register is cleared at the end of cylinder, if the operation is not complete. Thus, the next cylinder starts at head 00.
- Desired Sector Address - Loaded during control word one of an Out command, or during non-data

portion of a Write Header operation. The register is incremented once, for each sector, as the disk spins. The register holds binary counts 00 through 15. The actual sector address is achieved by combining the head and sector bits; e.g. head 00, sector 00 would be sector address 00, while head 01, sector 00 would be sector address 20<sub>8</sub>, or 16<sub>10</sub>.

- Read Status - A 10 bit register used to hold a record of the occurrence of an error condition within the Disk Sub-System. By use of the Read Status command, it is possible to transfer the contents of the register to the 10 lsb's of the Disk pointer word. The relevant significance of the respective bits may be found by referring to "JNE COMMAND" under the Command Descriptions. The contents of the Status Register may be displayed on the maintenance console in the Controller.

### Timing

- Sequence State Counter - The primary counter, located in the Controller, which governs the over-all sequence of operations taking place during the operational commands. There are eight possible sequence states. The sequence, and operations performed during the sequence states of the commands, are described under the command descriptions.
- Timing Counter (L pulses) - Counts from 0 through 23 and designates bit times in relation to words on the disk. The L pulses are timed from the CK2 (CKB) pulses, during the Write operation, and from the data itself, during the Read operation. Although the timing counter counts through the full 24 pulses, only the required pulses are decoded for use. Also decoded from the timing counter are timing envelopes which cover four individual time periods.
- Position Timing - The four timing periods generated from the gray-code counter are used to control the transfer of data between the Disk and Controller regarding the positioning of the heads. They are used when first initiating a read/write as well as during a cylinder carry-over.
- 72 Bit (3 Word) Delay - Used by the Controller to time the number of bits written for a gap, e.g. tolerance gap before and after data.

### Checking

- Check-Sum Generate and Compare - In lieu of parity checking, a check-sum of data is utilized. Each "one" bit is counted during the read (generation) and compared with the check-sum read from the disk (compare). An error indicates the check-sum differed from the count obtained during the read. There is also a check-sum performed on the address portion transmitted from Core during a Write Header operation. This is to verify the information received from Core.

## COMMAND DESCRIPTION

The commands are divided into two categories within this section: those which require the Controller to step through sequence states and those that do not. The commands which require sequencing are the Out and Read Status commands. The remaining commands are grouped under the heading "Non-sequencing commands".

The Write Header operation, although basically an Out command, is listed at the end of the Out command under the separate heading: Write Header Operation. The Write Header is a "special case" Out command, which is differentiated from an Out command, specifying a write operation, by the position of the "Normal/Header" switch located on the Maintenance Console. Although the execution of a "normal" write and a "header" write are basically the same, the pattern written upon the disk differs between the two operations. The write portion (SS7) is, therefore, described separately for the Write Header operation

The organization of this manual, is intended to serve both trained and untrained personnel dealing with the Disk

Sub-System. The descriptions of the commands are presented in levels of detail. The general descriptions of the command functions are included within the Functional Description, under the heading "Program Control". The next level of detail (Out and Read Status only) is the flow charts, illustrating the sequence of events to take place during the commands. The flow chart of the Out command, Fig. WHD.7.4, is folded so that it may be extended to the side while referring to the individual sequence states of the Out and Write Header descriptions.

The next level is that of the block and timing diagrams, which may be used while troubleshooting. The pages of the logic and the test points of the individual signals have been included to aid this purpose. The illustrations depict the functions which take place during the individual sequence states and the sequence of these functions. At the beginning of each sequence state is a brief summation of the events which take place during that period. The greatest detail, then, is contained in the text for the individual commands and sequence states.

# OUT COMMAND SS0

During sequence state zero, upon receipt of an Out command, the Coupler presets the Core Address Register (CAR) to the address of the pointer word and clears the Data Collection Register (DCR). The Out flip-flop in the Controller is set to record the type of operation to be performed and the busy line goes "busy". The Status Register of the Controller is also cleared. The following text refers to Figs. OUT 0.1 and 0.2, as well as the Coupler and Controller logic.

Essentially, sequence state zero is a "ready state" in which the Disk Controller is passively awaiting a command. The L counter, however, continues to count from L00 through L23 under the control of time counter increment, D1TTCI, logic. The timing of TTCI is controlled by the 1.25 MC clock, D1TCK2.

Upon execution of an Out command (S = 4), with the K3-K2 bits of the addressed Coupler (K2 = 0), the  $\phi A$  pulse from the AU enables the G0GOUT gate. G0GOUT is then inverted and sent to clock driver, D0APZT, which presets the CAR to the address of the pointer word. (The correct address of the pointer word may be determined by referring to the Program Control description, under the heading "Out Command"). Since transfers between the Data Buffer Register (DBR) and the DCR are "single-ended" transfers, clock drivers D0DPZX and D0DPZT will also be enabled at this time, to clear the DCR in preparation of incoming data.

The decoded Out command, G0GOUT, is also inverted and coupled to the Disk Controller to set the Out flip-flop, F1IOUT. The setting of the Out flip-flop enables line driver, X0IBSY, to establish that the Controller is now in a "busy" state. X0IBSY is sent to the Coupler to enable the busy lines to the AU and API and to disable G0GOUT, blocking succeeding Out commands from being acknowledged while the Controller is in a "busy" state.

Upon receipt of the busy signal in the Coupler, G0GOUT was disabled. The Out flip-flop in the Controller, however, set during the period that this gate was enabled. As a result of F1IOUT being set, the first L23 pulse to occur during SS0 steps the sequence state counter to SS1, via D1QSCI.

The existence of any of the ten error conditions (see "JNE COMMAND", under the Functional Description section) present during SS0 would have armed the error test line within the AU. Once the Out flip-flop is set, the Status Register is cleared on the following CK1. For this reason, a check should be made of error conditions before the execution of the Out command. If the error is still present in the Disk Sub-System, the corresponding error will again be recognized at its specified time of sampling and the operation will terminate.

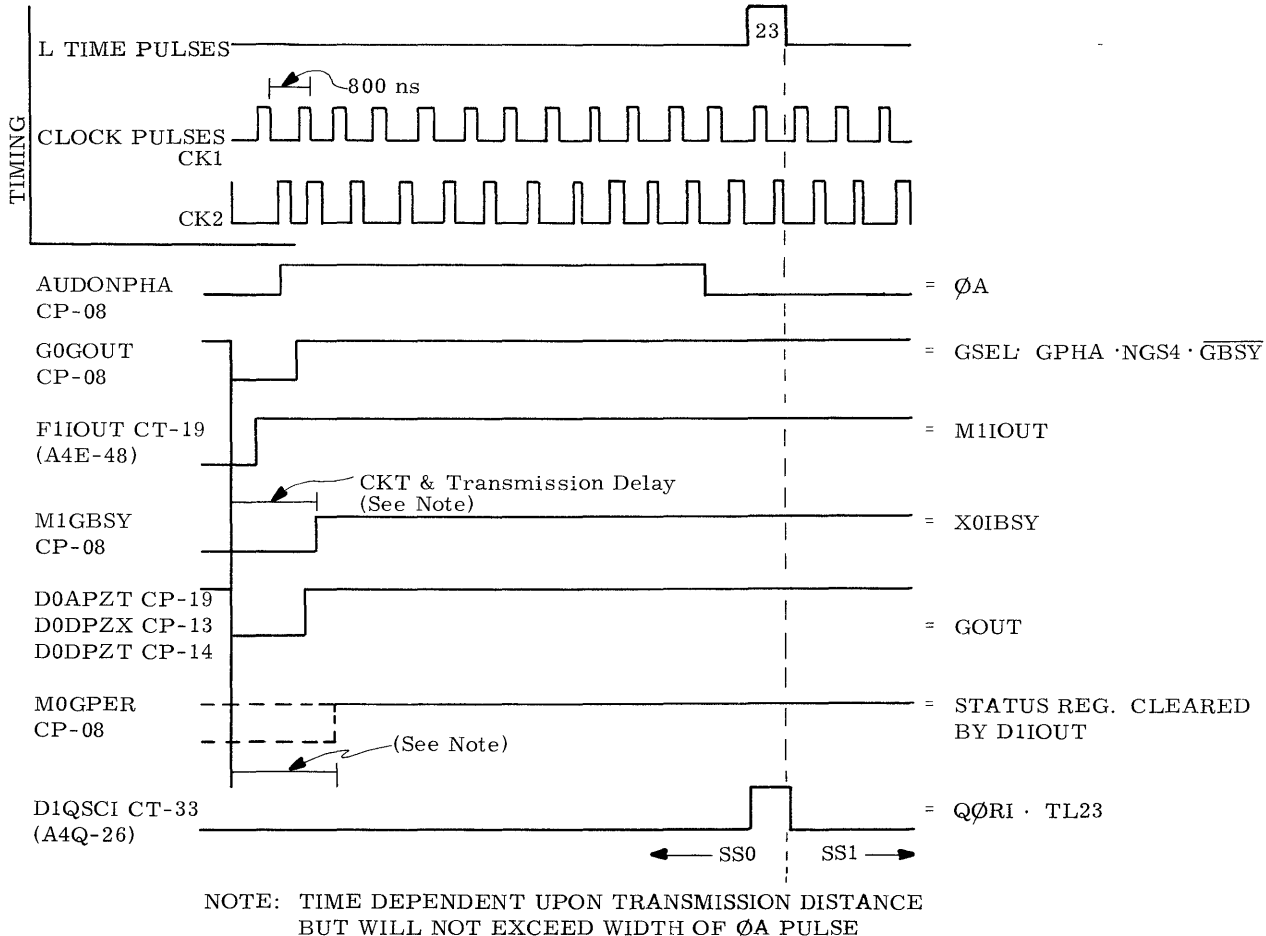


Fig. OUT 0.1. Timing Diagram-SS0

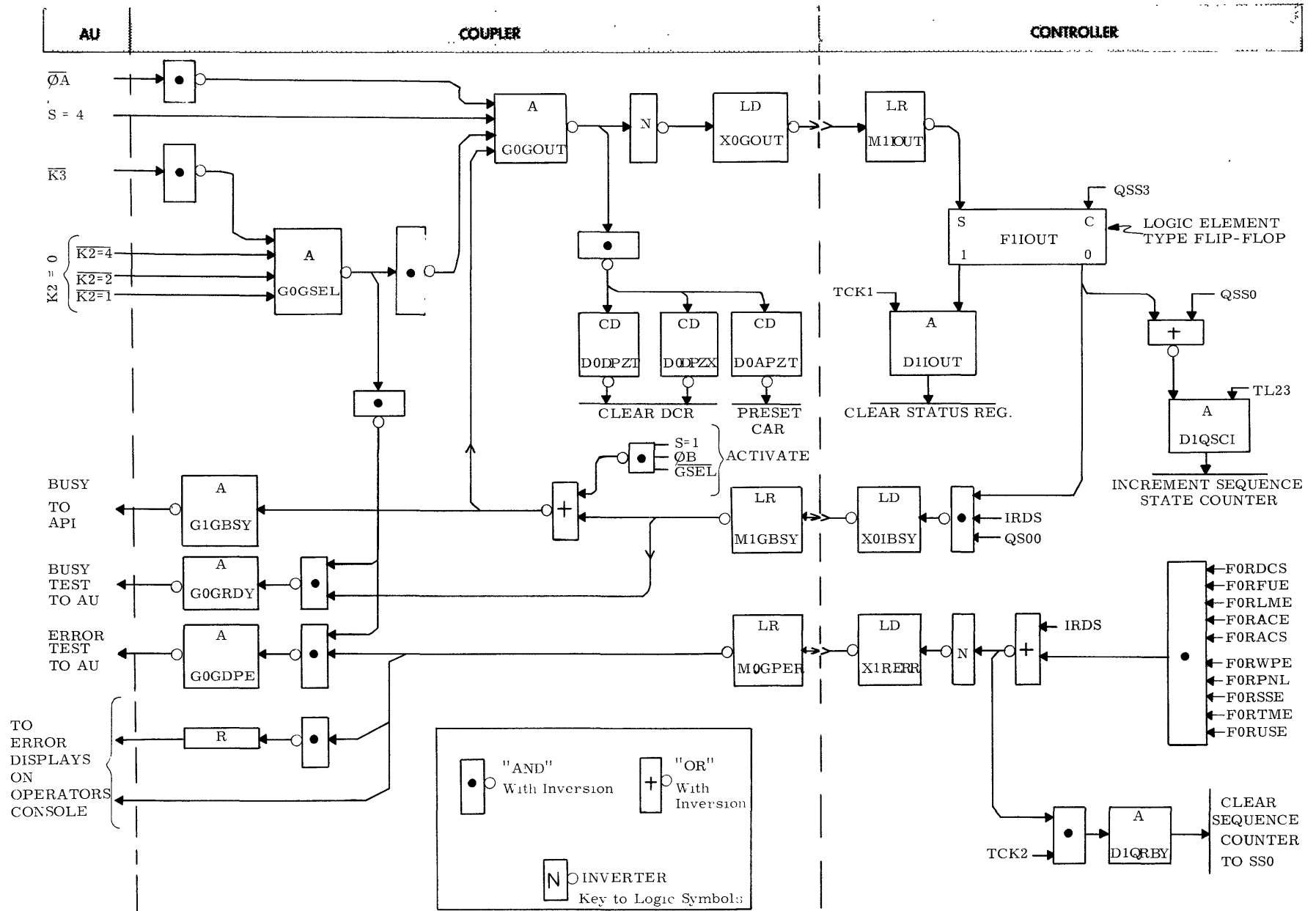


Fig. OUT 0.2. Block Diagram, OUT Command-SS0



## OUT COMMAND SS1

Sequence state one remains true for at least two word times. During this period the pointer word is addressed, gated from core to the Coupler Data Buffer Register (DBR) and finally to the Core Address Register (CAR). If, during the memory requests of SS1 - SS5, the memory is unavailable before the end of the word time in which the request originated, the sequence state will be extended until the Coupler gains access from Memory. The contents of the pointer word, brought from Memory during this sequence state, represent the address of the first control word, which will be addressed and brought from Memory during sequence state three. During the following description, Figs. OUT 1.1 and 1.2 as well as the Coupler and Controller logic should be referenced.

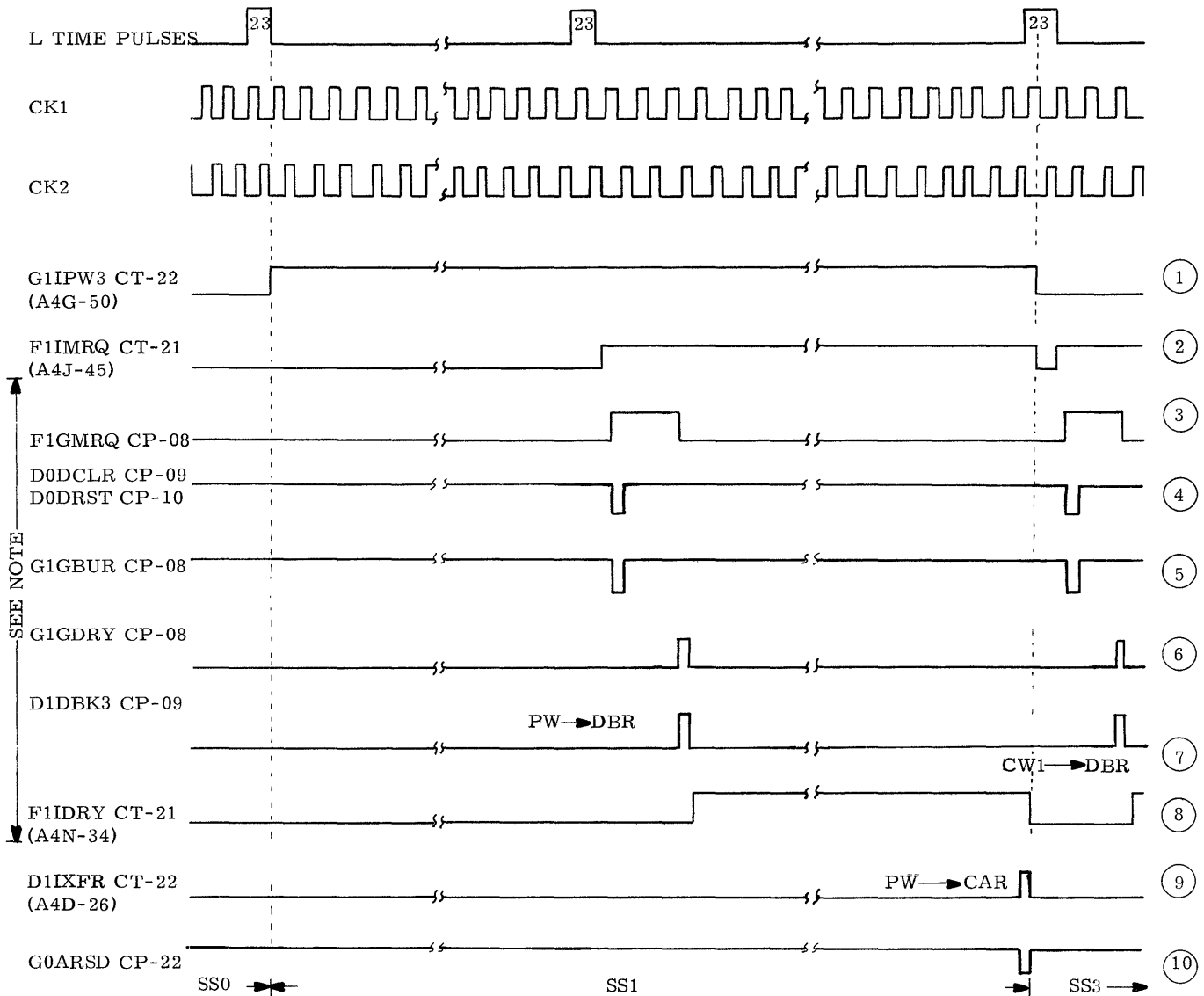
This first memory request originates within the Controller, with the setting of the F11MRQ flip-flop. The Coupler is subsequently informed of the request through the line driver/line receiver combination of X11MRQ and M1DMRQ, by setting F1GMRQ. The "one" side of this flip-flop then notifies Memory of the request.

The core address of the information requested is designated by the CAR, which had been preset to the address of the pointer word during SS0. This address is sent to Memory to select the core address of the pointer word. Line receiver, M1ADWL, which is always enabled until SS7 of an Out command, combines with F1GMRQ and N1GBUR to generate the clear pulses, D0DCLR and D0DRST, to clear the DBR in preparation of the

incoming data from Memory.

The period of elapsed time before data is available from Memory will vary with Memory speed and priority availability. Once the priority has been established, the inputs to the DBR are armed and when data from Memory becomes available (DRY), the data is clocked into the DBR. The receipt of the data ready signal from Memory, inverted through G1GDRY, arms F1GMRQ in the Coupler to clear on the next CK1, thereby indicating that the transfer of information from Memory has been accomplished. G1GDRY is also inverted and coupled to the Controller to indicate, by setting F1IDRY, that the Coupler has received data from Memory. With F1IDRY set, the data transfer pulse, D1IXFR, is enabled at CK1 of the following L23. This pulse clears the data ready and memory request flip-flops in the Controller and is also sent to the Coupler to AND with M1APW3 and M1ADWL, which are both enabled throughout this sequence state, to enable G0ARSD. G0ARSD is inverted and used to clock the pointer word from the DBR into the CAR.

With the transfer of the pointer word into the CAR, the sequence state counter is advanced to sequence state three. The sequence state counter is not incremented in the normal manner since sequence state two is used only during a Read Status command. F1QSB0 is already set to decode SS1 and it only requires the setting of F1QSB1 to decode SS3. At the trailing edge of the IXFR pulse, QSB1 is set advancing the Controller to SS3.



NOTE: THE PULSE WIDTH AND TIMING RELATIONSHIP OF THESE SIGNALS WILL VARY ACCORDING TO MEMORY SPEED AND AVAILABILITY

- ①  $G1IPW3 = QSS1$
- ②  $F1IMRQ = \overline{IRDS} \cdot IXF3 \cdot TL23 \cdot WDWL \cdot TCK2$ ,  
 $\overline{F1IMRQ} = IXFR$
- ③  $F1GMRQ = M1DMRQ \cdot GCK7$   
 $\overline{F1GMRQ} = GDRY \cdot GCK7$
- ④  $D0DCLR / D0DRST = ADWL \cdot \overline{GBUR}$
- ⑤  $G1GBUR = \overline{GMRQ} + MBR^*$
- ⑥  $G1GDRY = MDR^*$
- ⑦  $D1DBK3 = MDR^*$
- ⑧  $F1IDRY = M1IDRY$ ,  $\overline{F1IDRY} = IXFR$
- ⑨  $D1IXFR = IXF3 \cdot TCK1 \cdot WDWL \cdot IDRY \cdot TL23$
- ⑩  $G0ARSD = APW3 \cdot DXFR \cdot ADWL$

Fig. OUT 1.1. Timing Diagram, OUT Command - SS1

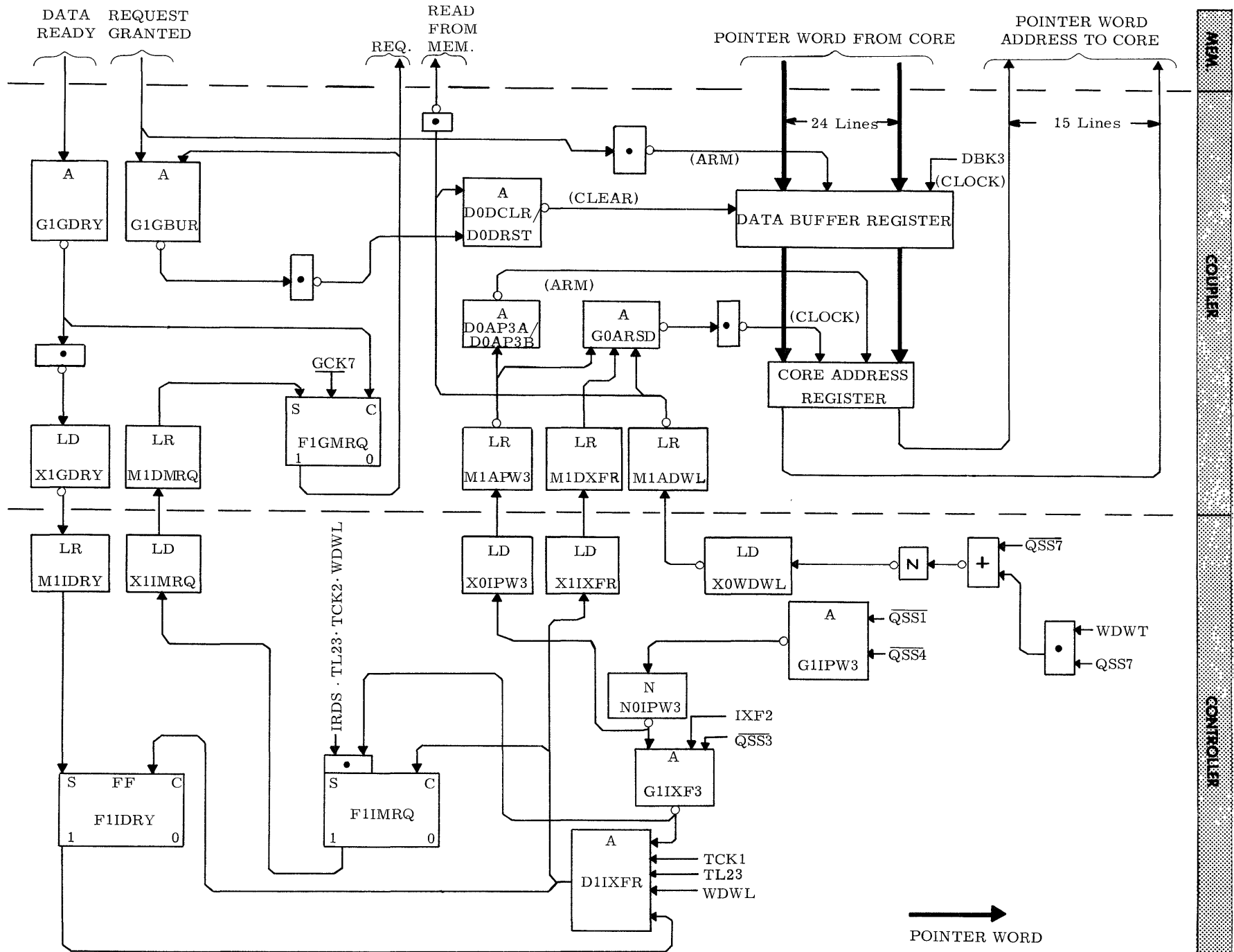


Fig. OUT 1.2. Block Diagram, OUT Command-SS1

## OUT COMMAND - SS3

Sequence state three, as did sequence state one, remains true for at least two word times. During this period, control word one and control word two are addressed and brought from Memory. Control word one, which is the first information loaded into the Data Buffer Register (DBR) during this state, is transferred parallel to the Data Collection Register (DCR) and then shifted serially to the Controller. In the Controller, the bits are directed to their corresponding registers and flip-flop. While control word one is shifted to the Controller, control word two is requested from Memory and upon receipt of the data ready signal from Memory, is gated into the DBR. Control word two is transferred to the DCR at the L23 following its transfer from Memory. Bits 16 and 17 of control word one, which were directed to the unit select circuitry in the Controller, are decoded to select a Disk Unit. Upon selection of the unit, a signal is sent to that unit to gate the data from its Cylinder Address Register to the Controller. The data is the address of the current cylinder over which that Disk Unit's heads are positioned. The data will be gated into the Current Cylinder Address Register in the Controller. The sequence counter is then incremented to SS4. The figures relating to this sequence state, Figs. OUT 3.1 and 3.2, along with the Coupler, Controller and Disk logic should be referenced during this description.

As in the previous sequence state, the memory requests originate in the Controller with the setting of F1IMRQ. The first request fetches control word one from the address specified by the contents of the Core Address Register (CAR), which was loaded during SS1. Control word one is gated into the DBR in the same manner that data was gated into the DBR during SS1, with the DBP1/DBP2 signals arming the inputs and the DBK1/DBK2/DBK3 signals clocking control word one into the register. The DBR is cleared prior to its loading, as in sequence state one, by the DRST/DCLR drivers. At the CK1 of the L23 following the loading of the DBR, control word one is transferred to the DCR. The transfer is accomplished by arming the inputs to the DCR with the DXF1/DXF2 drivers, enabled by the data transfer pulse from the Controller (M1DXFR), and clocking the information into the DCR with the DCK4/DCK5/DCK6 drivers, also generated as a result of the data transfer pulse from the Controller. The data transfer pulse, D1IXFR, generated in the Controller, sets the data transfer flip-flop, F1TDXF, at this time. The data transfer flip-flop now controls the serial shifting of data from the Coupler DCR to the Controller. With the arrival of the first CK2, following the setting of TDXF, control word one begins shifting serially to the Controller with each D1ICDS shift pulse generated in the Controller. The core address increment line driver, X1ICAI, is also enabled as a result of the IXFR pulse, to be sent to the Coupler to increment the Core Address register to the address of control word two. Control word two is then requested from Memory, in the same manner as control word one and, upon becoming ready, is gated into the Coupler DBR.

The serial shifting of control word one from the Coupler DCR to the Controller is timed by shift pulses originating in the Controller. These pulses are received at the

Coupler through line receiver, M1DCSH. The "true" output from this line receiver is inverted and enables clock drivers DCS1/DCS2/DCS3 to arm the shifting of "zeroes" into the DCR, while control word one is shifted from it. This effectively clears the register for receipt of the next word, which in this case is control word two. The "true" output of DCSH is also used to enable the output of the register, while the "false" output of DCSH is inverted and used to clock control word one from the register. Control word one is coupled through a line driver/line receiver combination to the Controller and emerges through the retime flip-flop at "L" times relative to its bit positions. Bits 00 - 07 are shifted from the retime flip-flop during L00 - L07 and are directed to the Desired Head and Sector Registers by D1HSLD. These registers are treated as one register during this shift. The bits enter the most significant end of the Desired Head Register and are shifted through to the Desired Sector Register. The Sector Register ends up with bits 00 - 03 and the Head Register receives bits 04 - 07. During L08 - L15, bits 08 - 15 are shifted into the Desired Cylinder Address Register under the control of D1CLOD. The unit select flip-flops, F1UUB0 and F1UUB1, receive bits 16 and 17, respectively, under the control of D1URLD, which is enabled during L16 and L17.

The data transfer flip-flop, F1TDXF, set at the time the data transfer pulse gated control word one to the Coupler, remains set and during the remainder of sequence state three, D1TCW1 is enabled. With the arrival of the first L00 pulse following the decoding of TCW1, the gray-code positioning counter, consisting of F1PT00 and F1PT01, commences counting. The counter increments through three counts, which are decoded and used for the following purposes:

- D1PT01 - Read the current cylinder address over which the selected unit's heads are positioned (during SS3).
- D1PT02 - Transmit the cylinder difference, calculated by the Controller, to the selected Disk Unit (during SS4).
- D1PT03 - Transmit desired cylinder, head and control select information to the selected Disk Unit (during SS5).

Through this sequence, the Controller reads the current cylinder address from the Disk Unit, compares it against the desired cylinder, and sends the difference sum to the Decrement Counter in the Disk and the desired cylinder address to the Cylinder Address Register in the Disk. It is then possible, by decrementing the Decrement Counter each time the heads move past a cylinder, to determine when the heads have reached the desired cylinder.

The first of these counts, D1PT01, ANDs with timing envelope five, G1TEV5, to enable the select unit driver, D1USLU, during L18 - L23 of the remainder of SS3. At the end of SS3, D1USLU is held true by F1QSB2, which will be true for the remainder of the Out command. D1USLU is used to enable the unit select line decoded by F1UUS1 and F1UUS0, which were loaded during L16

and L17. The decoded select line is used to select the desired Disk Unit on which the operation is to take place. When the unit select signal is received in the Disk Unit it acknowledges receipt by sending a return signal to the Controller. This signal is compared with the original unit select signal to determine if the correct unit has been selected. Compare circuitry is also utilized to determine if more than one Disk Unit is selected at the same time. If either the wrong unit or more than one unit is selected, the unit select error flip-flop, F1RUSE, is set. Upon return of the select signal from the Disk, if no unit select error is detected, the signal D1PT01 enables the read cylinder select signal, X1PRCS, to gate the contents of the Cylinder Address Register, in the Disk, to the Controller. The lines from the Disk are applied to the set inputs of the Current Cylinder

Address Register in the Controller, which was cleared at L16 by D1CCCR. These inputs are gated into the register at the end of L22 by D1PCCS.

Since they are not used, bits 18 - 22 of control word one are ignored upon entering the Controller from the Coupler. During L23, bit 23 of control word one is gated to the write flip-flop, F1WDWT. At this point, however, the sequence state counter will have incremented to sequence state four. Also, at CK1 of this L23, the data transfer pulse, D1IXFR, is generated to gate control word into the DCR. D1IXFR also results in the generation of X1ICAI to increment the contents of the CAR, in preparation of addressing control word three. The sequence counter is incremented to SS4 at this time by D1QSCI.

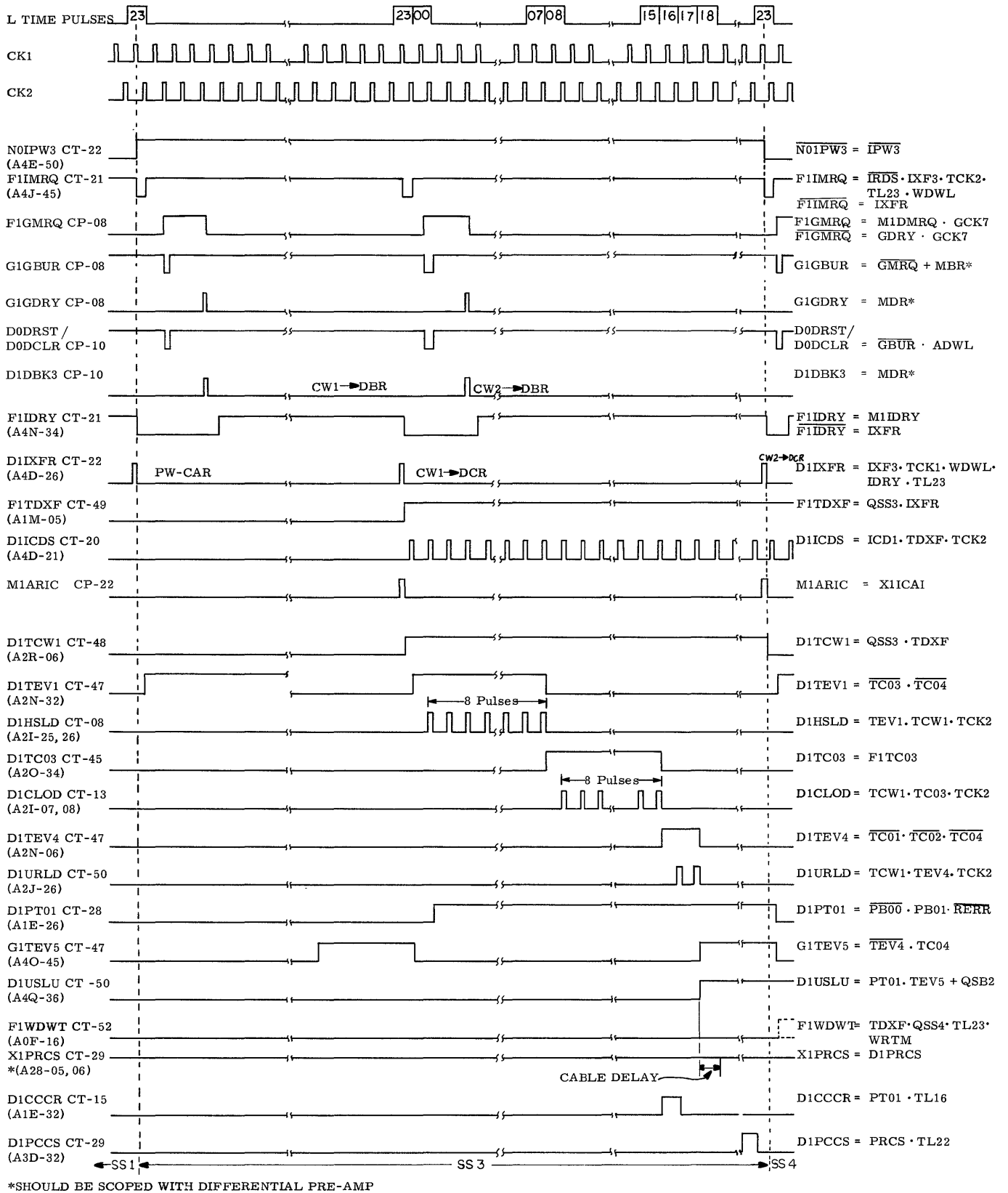


Fig. OUT 3.1. Timing Diagram, OUT Command-SS3

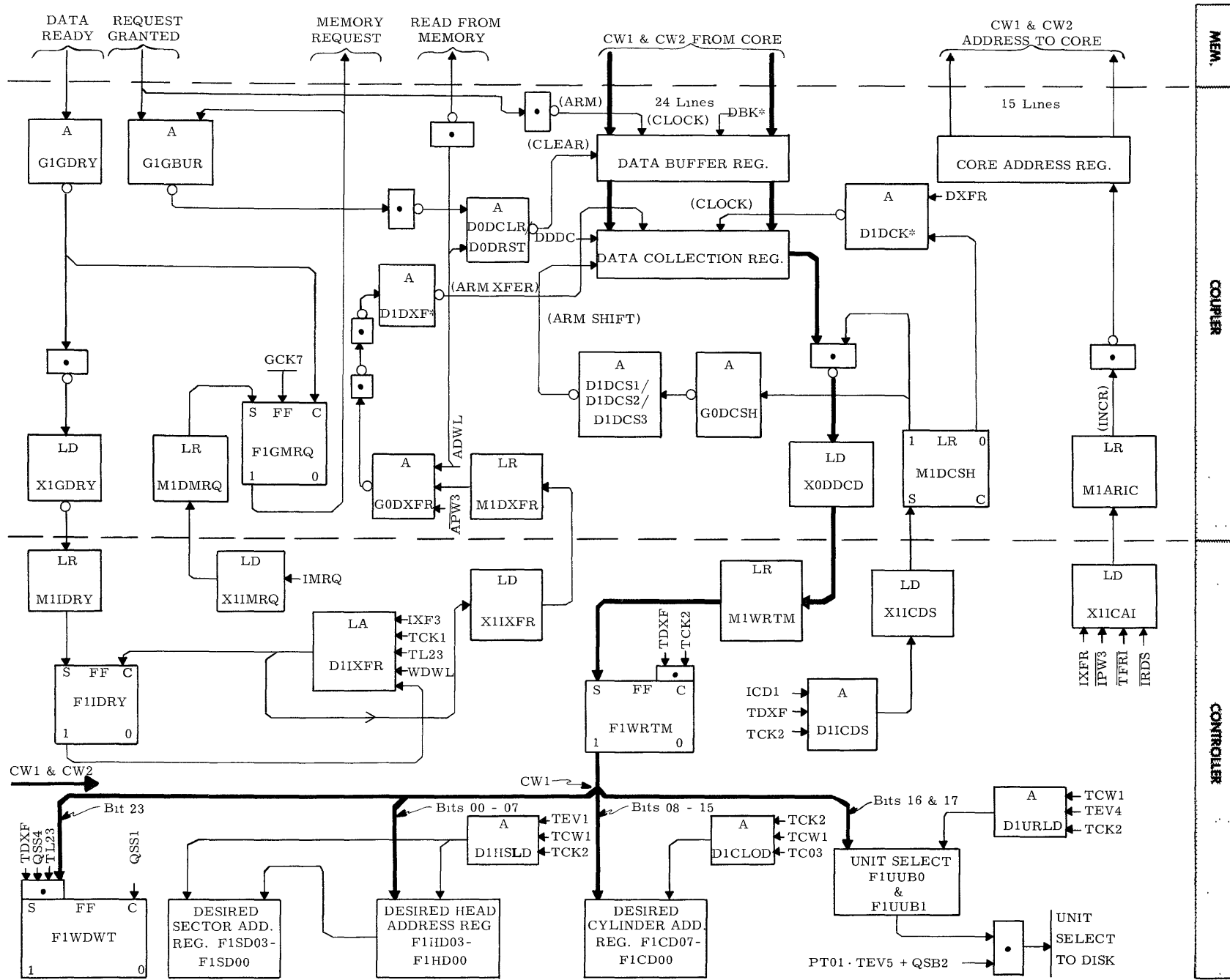
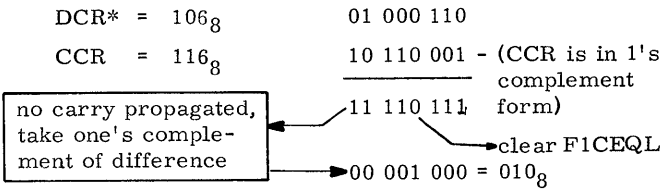


Fig OUT 3.2. Block Diagram, OUT Command-SS3





**Example 3 CCR > DCR\***



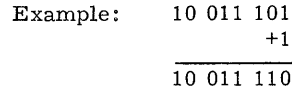
The output of the DCR\* and the inverted output of the CCR are ANDed, as in the previous examples, during L00 - L07. During L03, in this example, a false output is detected from the difference control logic and the equal flip-flop is cleared. During the final addition of the first shift (L00 - L07), the output of the current stage carry gate, G1CCSC, is false and the complement flip-flop, which was cleared by PT01, is set. This results in the contents of the CCR being complemented during the second ring shift (L08 - L15). This is accomplished as in example one.

The direction of head movement necessary is determined by the results of the first ring shift of the registers. As may be seen in the examples, if the CCR was less than the DCR\*, the add-one flip-flop would be set, indicating a forward seek is necessary. If the contents of the CCR were greater than the DCR\*, the complement flip-flop would be set, indicating a seek backward operation is necessary. However, if the registers were equal, both drivers, D1LSBW and D1LSFW, would be disabled and no seek would take place. The transmittal of head movement information to the Disk is described in SS5.

The concept of the complementing cycle is relatively simple, since the inverted contents of the CCR are merely gated through the difference control logic. However, the add-one cycle is a little more complex, so an example is given in the following text, along with the outputs of relevant gates and flip-flops involved during the shift.

As may be seen in the example, the bit being shifted from the CCR is inverted when there is a carry indicated by either the current stage carry gate, G1CCSC, or the previous stage carry flip-flop, F1CPSC. Assuming a carry from the most significant bit of the first ring shift, which places the Controller in an add-one cycle,

CCSC is true at the first bit time of the shift. This causes the previous stage carry flip-flop, F1CPSC, to set at L09 time. This results in the inversion of the first two bits shifted from the CCR. The remaining bits, since no more carries are generated, are gated through the logic in their true form. Therefore, the result gated back to the CCR is one greater than what was gated from it.



L time	CCR	CCCA	CCSC	CPSC	CSUM
L08	1	0	1	1	0
L09	0	1	0	1	1
L10	1	0	0	0	1
L11	1	0	0	0	1
L12	1	0	0	0	1
L13	0	1	0	0	0
L14	0	1	0	0	0
L15	1	0	0	0	1

If the results of the difference sum indicate that the CCR was not equal to the DCR\*, the difference select line driver, X1PDFS, is enabled during L18 and L19, to gate the contents of the Cylinder Difference Register to the Disk's Decrement Counter. The gray-code counter, incremented on the trailing edge of the L23 at the beginning of this sequence state, is decoded at a count of D1PT02, which dictates the period during which the difference sum is sent to the Disk. If the difference sum indicates that the CCR and DCR\* had been equal, F1CEQL would disable G1PTC1, which would prevent the generation of the difference select strobe, X1PDFS.

On the trailing edge of the IXFR pulse, which is sent to the Coupler and enables G0ARSD, control word three is clocked into the CAR. At the same time, the sequence state counter is incremented to SS5.

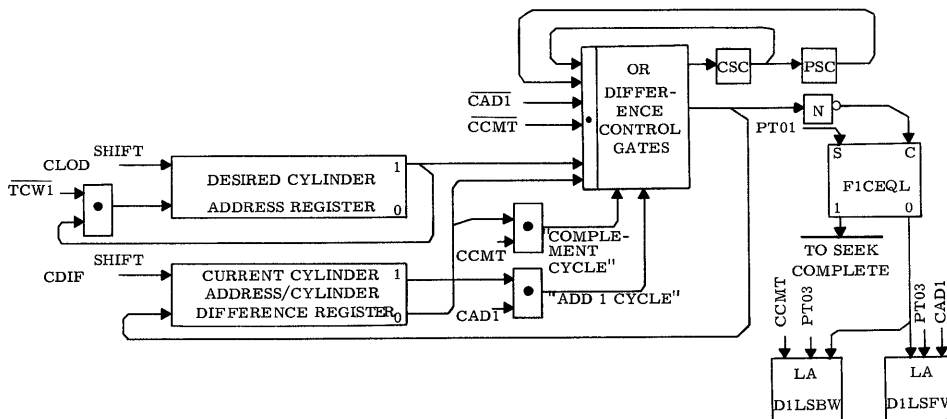


Fig. OUT 4.1. Cylinder Difference Calculation

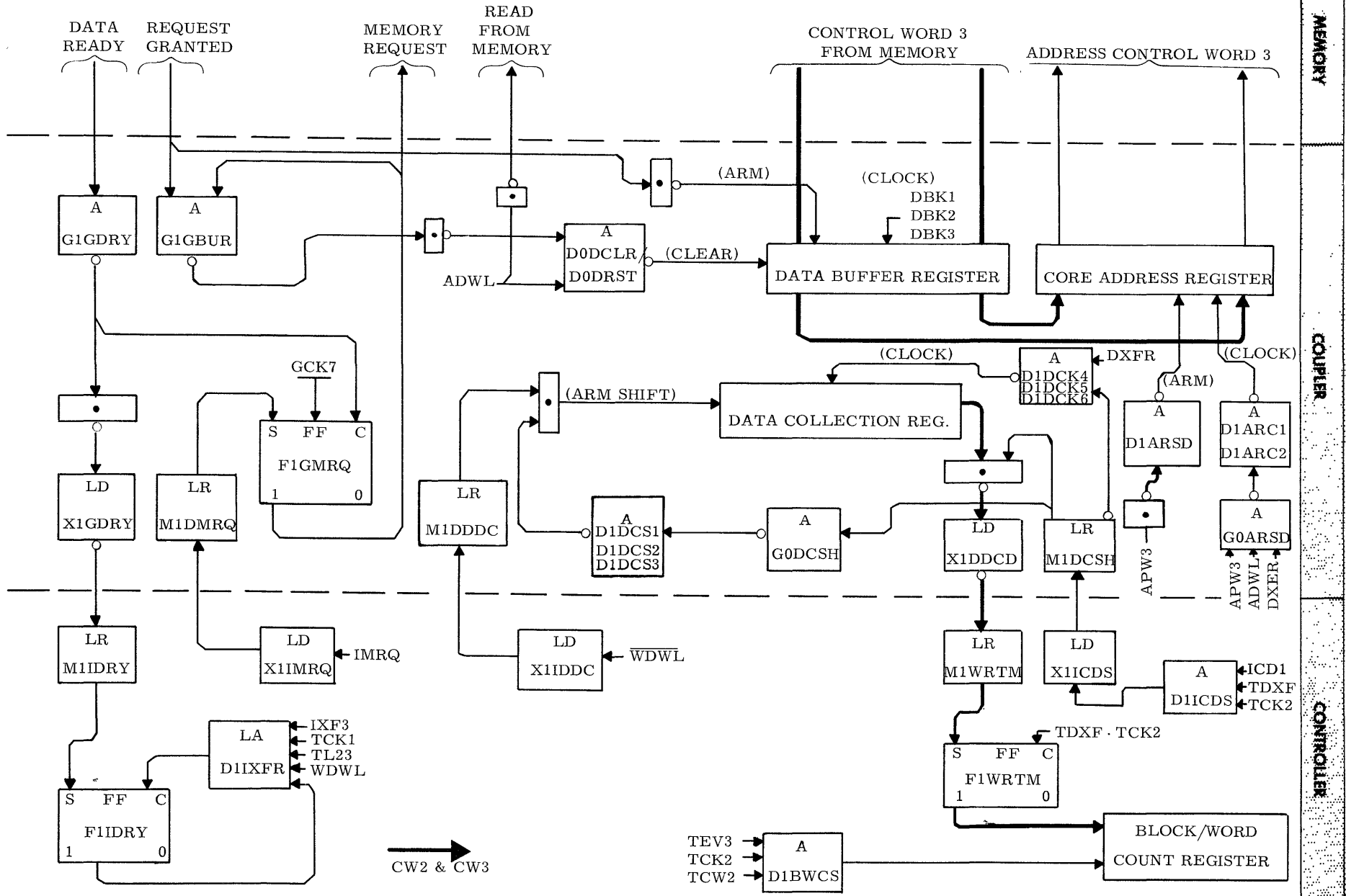
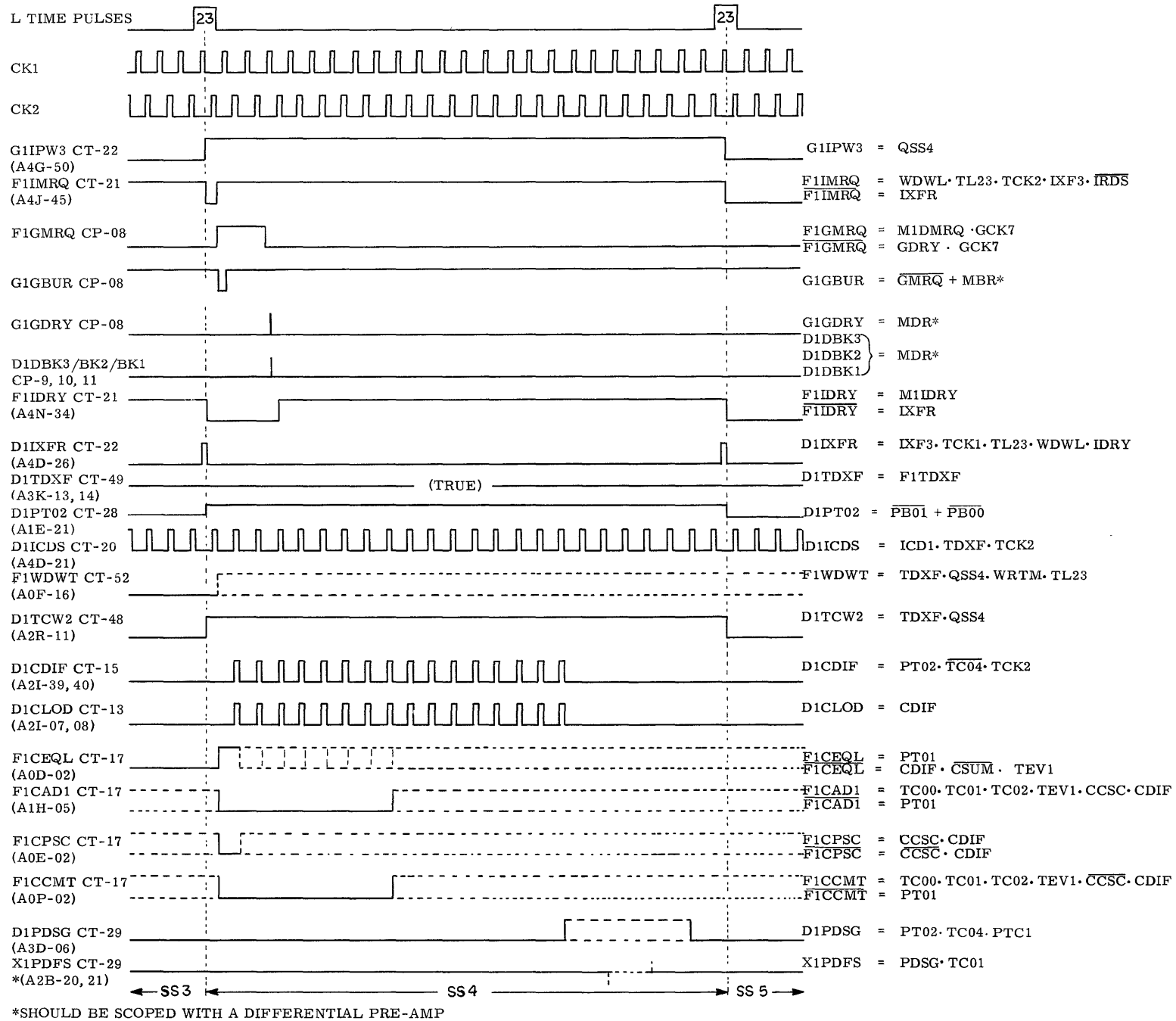


Fig. OUT 4.2. Block Diagram, OUT Command-SS4



\*SHOULD BE SCOPED WITH A DIFFERENTIAL PRE-AMP

Fig. OUT 4.3. Timing Diagram, OUT Command-SS4

## OUT COMMAND - SS5

During sequence state five, the count contained in the Block-Word Count Register is ring shifted and one subtracted from it. The head select and control select information is gated to the Disk. The cylinder select information is also gated to the Disk, if it was determined, in SS4 that the heads were not already positioned over the desired cylinder. If the operation is determined to be a write, i. e. the write flip-flop was set at the beginning of SS4, a memory request is made for the first data word at the end of SS5. If the results of the subtraction of one from the Block-Word Count Register reveal that it had contained all zeroes, the terminate flip-flop sets during this sequence state. The operation would then terminate at the beginning of SS6, after initiating a "seek operation" within the Disk. Since there are two possible sequences resulting from the subtraction of the Block-Word Count Register, there are two timing diagrams relating to this sequence state. Fig. OUT 5.2 relates to block-word count equal to zero, while Fig. OUT 5.3 refers to a block-word count not equal to zero. The basic block diagram, Fig. OUT 5.1, refers to both conditions. The Coupler, Controller and Disk logic should also be referenced as an aid to this description.

In the timing diagrams, it can be seen that the Block-Word Count Register is shifted during the time period of L00 - L13, by the word count shift signal, D1BWCS. As the bits are shifted from the register, they are sampled at the clear side of the carry flip-flop, F1BCRY and also at the AND inputs to the subtract gate, G1BSUB. The carry flip-flop was preset on the trailing edge of SS4. The one (1) side output of the Block-Word Count Register is gated through the gate, G1BSUB, as long as the carry flip-flop is set. This results in an inversion of the bit as it is shifted back into the register. If there is a "one" bit shifted from the register, the carry flip-flop is cleared and each remaining bit shifted from the register, after the first "one" detected, is gated back in its true form. This, effectively, decrements the register by one.

### BLOCK/WORD COUNT = 0

If, during the ring shifting of the register, no "ones" are detected, the carry flip-flop remains set at the end of the shift. This indicates to the Controller logic that the register contained all "zeroes". This identifies the Out command as a "seek only" and results in the terminate flip-flop, F1TTRM, being set at L16, as is shown in Fig. OUT 5.2. The carry flip-flop, F1BCRY, will be cleared on the trailing edge of L22 in this example. On the trailing edge of the CK1 of L23, the sequence state counter is incremented to SS6. At this time, the output of the terminate flip-flop is ANDed with SS6 to cause G0QRB1 to go true. G0QRB1 is inverted and ANDed with the following CK2 to enable the reset busy driver, D1QRBY, which clears the sequence state counter. This results in the Controller returning to SS0, and a "not busy" condition.

The enable time counter flip-flop, F1TETC, clears on the trailing edge of SS5, and remains clear during the brief period while the Controller is in SS6. Once the sequence state counter has returned to SS0, as mentioned

above, NOIBSY goes false and is ANDed with the next CK1 to again set F1TETC, which enables the time counter to begin counting again. During the period while TETC is cleared, the zero (0) output from the flip-flop is ANDed with CK2's to clear the timing counter during each CK2.

### BLOCK/WORD COUNT $\neq$ 0

A count of some increment of sixty-four in the Block/Word Count Register indicates that the Controller must execute some transfers, which will take place in SS7. Therefore, the operation will not terminate in SS6. As is shown in the timing diagram relating to a block/word count not equal to zero, Fig. OUT 5.2, the carry flip-flop, F1BCRY, clears (resets) upon detecting a "one" being shifted from the Block/Word Count Register and the terminate flip-flop, F1TTRM, does not set. The sequence state counter advances to SS6 at the CK1 of L23, and the enable time counter flip-flop, F1TETC, clears (resets), disabling the timing pulses.

The other portions of SS5 are identical to both conditions of the block count. Although the core data shift pulses, D1ICDS, are enabled throughout SS5, they are not effectively utilized. They do result, as in the two previous sequence states, in the shifting of the contents of the Coupler's DCR through the Controller's retime flip-flop. However, the output of the retime flip-flop is not gated to any of the registers, so the data contained in the DCR is lost. The data in this case was all "zeroes", since the DCR was filled with "zeroes" during the shift in SS4. During this shift in SS5, the DCR is again filled with "zeroes".

Information relating to the desired cylinder, the desired head, and other control information is sent to the Disk during this sequence state. This information transfer is under the control of the last count, D1PT03, of the position timing counter. The first information transmitted during this sequence state, the desired cylinder, will only be transmitted to the Disk if the current cylinder was not determined to be equal to the desired cylinder during SS4. If they were not equal, the cylinder select gate, G1PCSG, will be enabled during the period of L00 - L05. G1PCSG enables the output of the Desired Cylinder Register (DCR\*) to be gated onto the "Controller-to-Disk" bus lines. G1PCSG is also ANDed with D1TC01 to generate X1PDFS, during L02 and L03, which gates the bus lines into the Cylinder Address Register in the Disk.

The other two items of information are transmitted unconditionally to the Disk. This consists of the head select and control select information.

The head select gate, D1PHSG, is enabled during L08 - L13, and gates the output of the Desired Head Register onto the "Controller-to-Disk" bus lines. D1PHSG ANDs with D1TC01 to generate X1PHDS, which gates the bus lines to the Head Address Register in the Disk during L10 and L11.

The last group of information is the control select bits. Through these lines the operation (read or write), seek direction (forward or reverse) and the erase information (write only) are transmitted to the Disk. The return

to zero and head advance information is not normally transmitted at this time. The previously mentioned information is gated onto the bus lines from L16 - L21, by D1PCTG. D1PCTG also ANDs with D1TC01 to enable X1PCTS during L18 and L19, to gate the bus lines into their corresponding logic in the Disk. The read, write and erase information is not used by the Disk at this time, but the seek direction will be utilized in the following sequence state to control direction of seek and is, therefore, used in the Disk logic to clear the

stop flip-flop and determine if the reverse flip-flop should be set. The read, write and erase information will be enabled during SS7, as mentioned under the description for that sequence state.

If the write flip-flop, F1WDWT, was set at the beginning of SS4, a memory request will be made on the trailing edge of this sequence state. This will place the first word of information on the Coupler DCR during SS6, so that it will be available for transfer to the Disk in SS7.

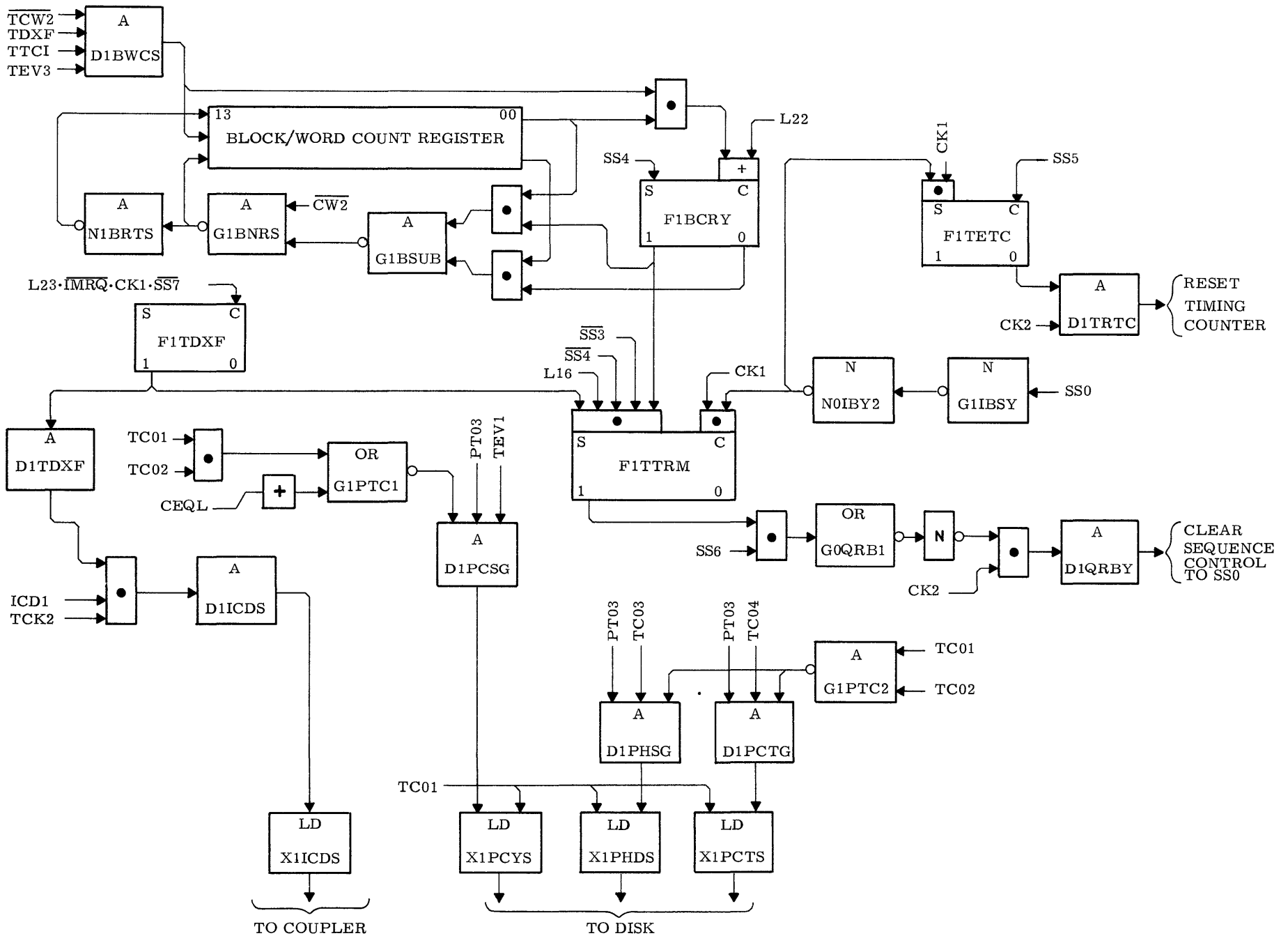
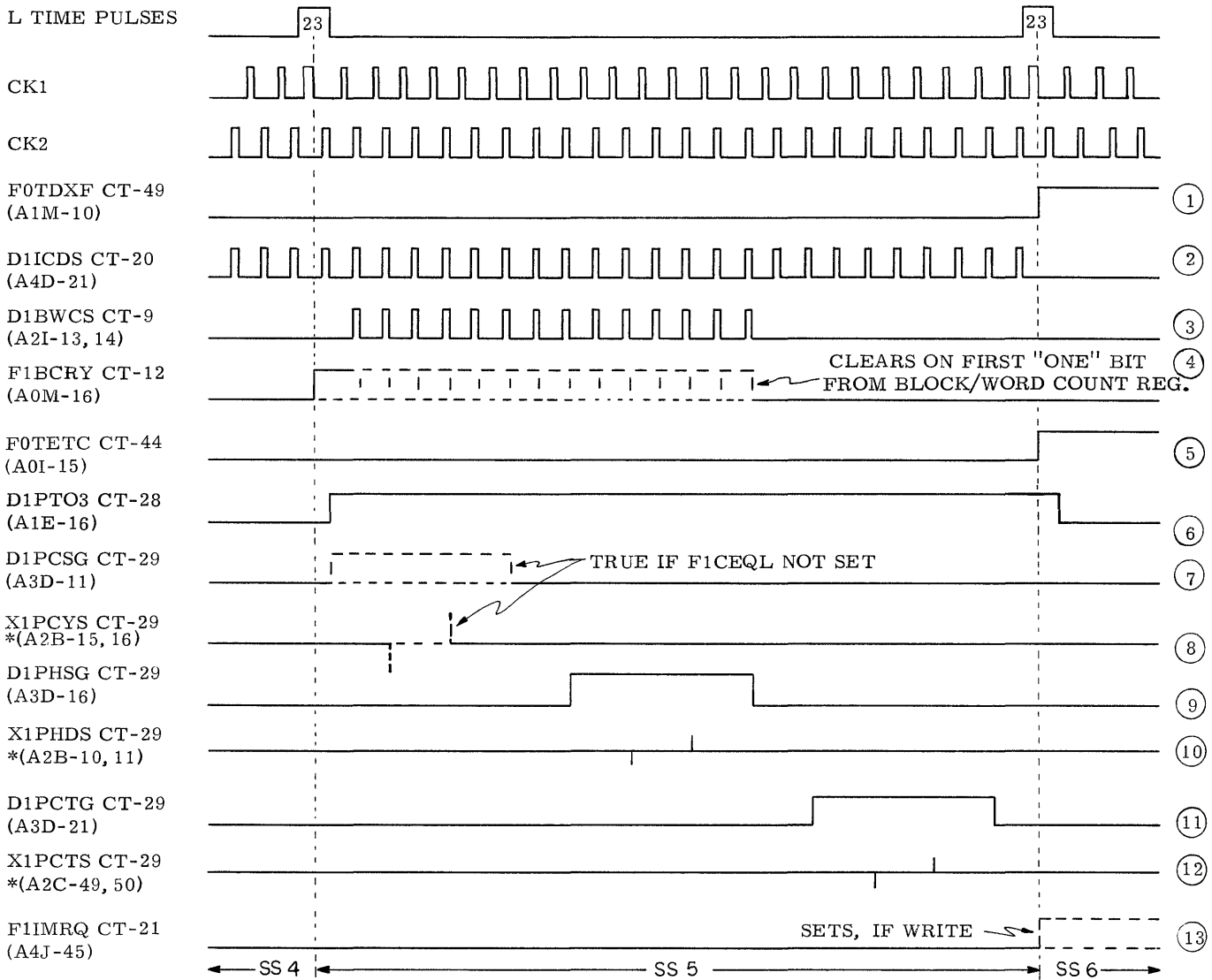


Fig. OUT 5. 1. Block Diagram, OUT Command-SS5



- ①  $F0TDXF = TL23 \cdot \overline{IMRQ} \cdot TCK1 \cdot \overline{QSS7}$  \* SHOULD BE SCOPED WITH DIFFERENTIAL PRE-AMP
- ②  $D1ICDS = IDC1 \cdot TDXF \cdot TCK2$
- ③  $D1BWCS = \overline{TCW2} \cdot TDXF \cdot TTCI \cdot TEV3$
- ④  $\overline{F1BCRY} = SS4$   
 $F1BCRY = BWCS \cdot BW00$
- ⑤  $F0TETC = SS5$
- ⑥  $D1PTO3 = PB00 \cdot \overline{PB01}$
- ⑦  $D1PCSG = PTC1 \cdot PT03 \cdot TEV1$
- ⑧  $X1PCYS = PCSG \cdot TC01$
- ⑨  $D1PHSG = PT03 \cdot TC03 \cdot PTC2$
- ⑩  $X1PHDS = PHSG \cdot TC01$
- ⑪  $D1PCTG = PT03 \cdot TC04 \cdot PTC2$
- ⑫  $X1PCTS = PCTG \cdot TC01$
- ⑬  $F1IMRQ = \overline{QSS5} \cdot WDWT$

Fig. OUT 5. 2. Timing Diagram, OUT Command - SS5 Block/Word Count  $\neq 0$

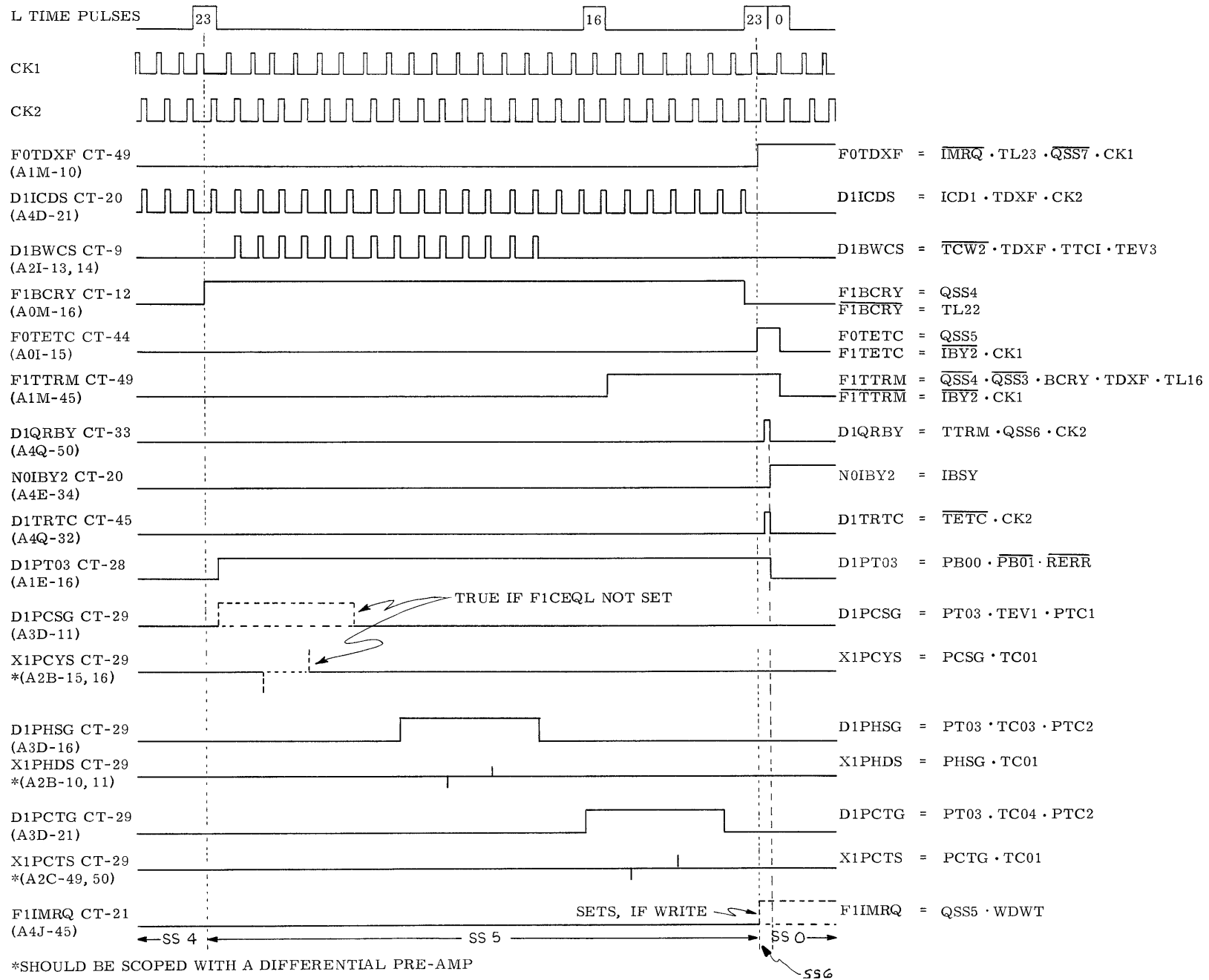


Fig. OUT 5.3. Timing Diagram, OUT Command-SS5 Block/Word Count = 0



## OUT COMMAND - SS6

Sequence State 6 is an asynchronous sequence state, whose time period is determined by the distance the heads must travel to reach the desired cylinder. This sequence state may be entered from SS5, during the initial sequencing of the command, or from SS7 when the end of a cylinder is reached before the block/word count has reached zero. These two sequences are described individually in the following text.

### INITIAL SEQUENCING (SS5-SS6)

During this period the Controller is awaiting a "seek complete" signal from the Disk. The actual signal which initiated this seek was generated during SS5, in the form of a "seek forward" or "seek backward" signal. When the Disk arrives at the desired cylinder, during SS6, it generates the "seek complete" signal, which advances the Controller to SS7.

### DESIRED CYLINDER INCREMENT (SS7-SS6)

If, during a transfer operation in SS7, it is determined that it is necessary to continue the operation past the end of the cylinder, a desired cylinder increment is needed. Since this involves a head movement operation, it is necessary for the Controller to return to SS6, during which address seek information is sent to the Disk and the "seek" to the next succeeding cylinder takes place. A timing diagram, Fig. OUT 6.1 is provided as an aid to this description. The Controller and Disk logic should also be referenced.

As may be seen in the timing diagram, the desired cylinder increment driver, D1CDCI, is enabled during L14 of the head update cycle at the end of the cylinder. The head address is incremented once each time the Desired Sector Register reaches a count of  $15_{10}$ . This is accomplished by setting the head update flip-flop, F1HHUD, which enables the head advance signal, N1LHAD, to be gated to the Disk during SS7. This signal increments the Head Register in the Disk to advance the data transfers to the next sequential track on the cylinder. When the Head Address Register in the Disk reaches a count of  $10_{10}$ , indicating that the last head of a cylinder is being utilized, an end of cylinder signal, (M1YEOC) is sent to the Controller. When the last sector of that cylinder has been addressed, the desired cylinder increment signal, D1CDCI, is generated to return the Controller to SS6 and move the heads to the next sequential cylinder.

The desired cylinder increment signal, D1CDCI, also starts the timing of the gray-code positioning counter which governs the transfer of control information to the Disk. The first count, initiated by CDCI, is

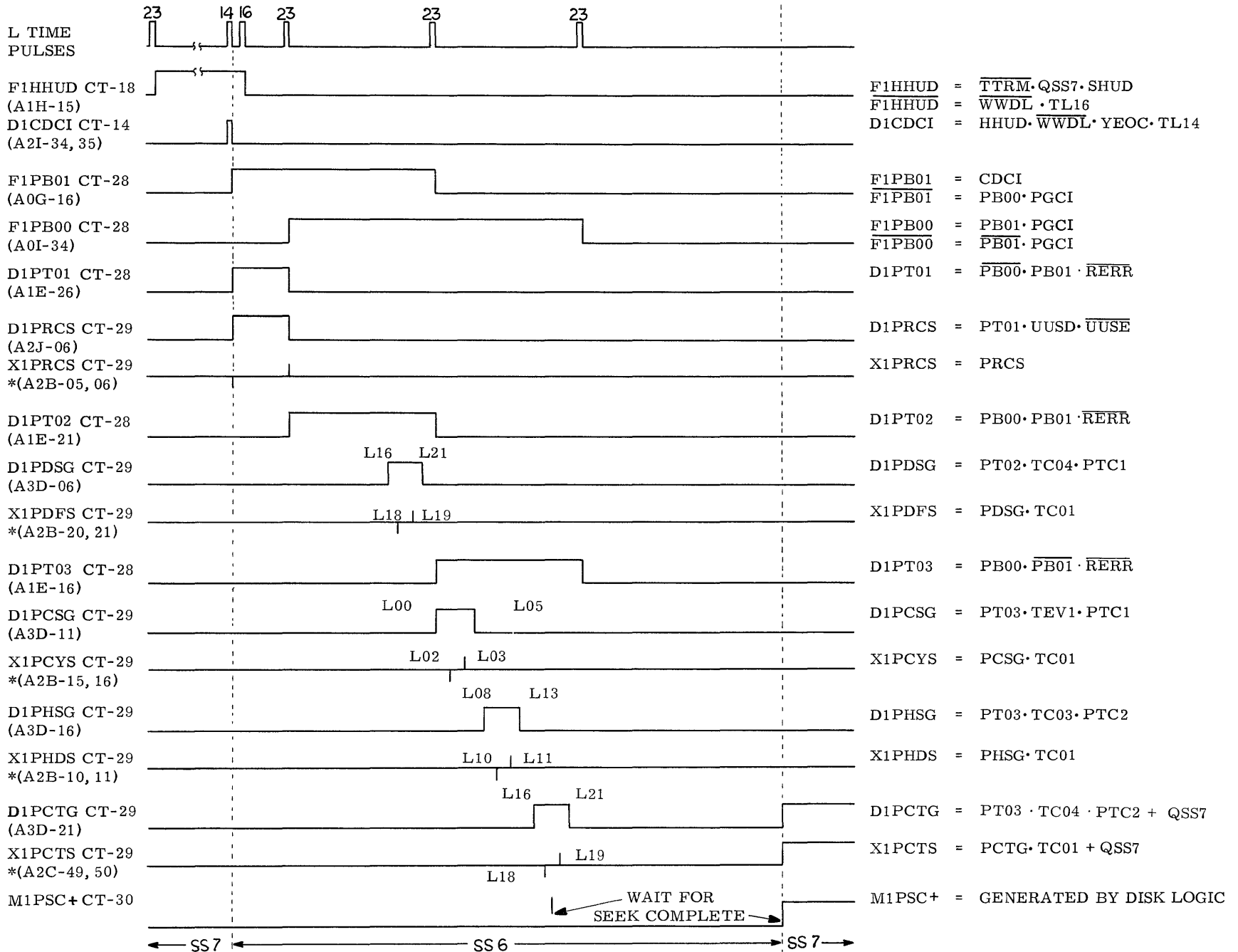
D1PT01. During this time, from L14 to L23, the read cylinder select signal, D1PRCS, enables the inputs from the Disk-to-Controller bus lines to the Current Cylinder Address Register (CCR). The signal, X1PRCS, is enabled at the same time to gate the contents of the Cylinder Address Register, in the Disk, to the bus lines. The current cylinder address of the Disk is, therefore, read into the Controller's CCR. During L16 of this period, the CCR is cleared, but the address from the Disk is enabled at the input of the CCR until L23.

Upon the incrementing of the gray-code counter at the end of L23, D1PT02 is decoded. This signal is enabled until the following L23. During L00 - L15, the CCR is ring-shifted twice, and the cylinder difference sum calculated as described under "Cylinder Difference", in SS4. From L16 until L21, the output of the CCR, which now contains the difference sum, is gated onto the Controller-to-Disk bus lines by D1PDSG. X1PDFS is sent to the Disk during L18 and L19 to enable the lines to be gated into the Decrement Counter in the Disk.

The gray-code counter is again incremented at L23, and decodes the count D1PT03. During L00 - L05, D1PCSG gates the output of the Desired Cylinder Register to the Controller-to-Disk bus lines. This address was incremented by one, by signal D1CDCI, just prior to leaving SS7. X1PCYS is enabled during L02 and L03 to gate these lines into the Disk Cylinder Address Register.

The desired cylinder increment pulse, D1CDCI, also clears the contents of the Desired Head and Sector Registers, so that the operation to take place during the next cylinder will begin at the first track and sector, of that cylinder, i. e., head 00, sector 00. The outputs of the Desired Head Register are gated onto the Controller-to-Disk bus lines from L08 to L13 by D1PHSG. X1PHDS then gates them into the Disk's Head Address Register during L10 and L11.

The seek direction is determined by the difference sum calculation, as in SS4. The seek forward driver, D1LSFW, is therefore enabled throughout D1PT03. Its condition is sensed at the Controller-to-Disk bus lines during L16 - L21, while D1PCTG is enabled. X1PCTS is enabled at L18 and L19 to gate the seek information to the Disk. This seek information initiates the head movement to the next cylinder. Upon reaching that cylinder, the Disk issues a seek complete signal, which is received by the Controller. The Controller then returns to SS7 to resume data transfers.



\*SHOULD BE SCOPED WITH DIFFERENTIAL PRE-AMP  
+ = 0 - 3

Fig. OUT 6.1. Timing Diagram, OUT Command-SS6

## OUT COMMAND SS7

During sequence state seven the actual read (Disk to Core), or write (Core to Disk), will take place. The sequence of events followed during this time, is governed by whether the operation to take place is a read or a write. The detailed description is, therefore, divided into two parts: Write operation and Read operation.

Upon entering sequence state seven, the Controller awaits the detection of a sector marker pulse, D1SSMP, before commencing operation.

On both the Read and Write operation, the first function performed at the beginning of each sector, after synchronizing the read circuitry from the tolerance gap and sync word, is the address read and comparison. This first three blocks of the sector format (see Fig. OUT 7.1) are recorded by a "Write Header" operation (Refer to "Write Header" description). The first of these blocks is the tolerance gap of 72 "one" bits. The next block, of sync bits (23 "zeroes" and a "one" bit), serves to synchronize the read circuitry in preparation of reading the address bits. The "one" bit, which is the actual sync bit, is recognized by the Controller logic (D1ZSRP). Following this bit, the read circuitry is enabled to gate the address and address check-sum bits to the Controller.

The address is read from the disk and compared with the desired address within the Controller. If there is not a direct comparison, the operation will wait for another sector pulse and repeat the above procedure. (If the address is not detected after a full revolution of the disk, an "address compare" error will be indicated). Once address comparison is achieved, the read or write operation will commence on that sector.

The first operation to be described is that of a Write.

### WRITE OPERATION

For the Write operation, a simplified block diagram of the data flow is given in Fig. OUT 7.2. Also included to aid the description of the Write operation are two timing diagrams: one illustrating the timing relationship of the major sequence of events, Fig. OUT 7.3; and the other, Fig. OUT 7.4, showing in somewhat more detail the actual transfer of data to the disk.

As may be seen on Fig. OUT 7.3, the non-data portion flip-flop, TNDP, is set on the trailing edge of the sector marker pulse received from the Disk. As a result of TNDP being set, the read gate, LRED, is enabled. This signal allows the data being read from the disk to enable the time control pulse driver, RTCP.

Until this point in the operation, the timing counter was exclusively under the control of the Controller's 1.25 MHz clock oscillator. The synchronization pulse, ZSRP, which is detected as a "one" bit following at least 16 "zeroes", sets the enable time counter flip-flop, TETC. The time counter increment flip-flop, TTIC, ANDs with TETC and RTCP, to initiate the incrementing of the time counter, under the control of the RTCP pulses being received from the disk. (This method of

clocking is described in more detail under the Read Operation portion of SS7). During the time that the read circuitry is enabled, the address and address check-sum bits are read from the disk.

As the address bits are read from the disk, they are compared with the desired starting address in the Controller. During timing envelope one, TEV1 (L00 - L07), the combined Desired Sector/Head Registers are shifted through the compare logic, D1WAEQ, as their corresponding bits, read from the disk, are shifted through. The desired cylinder address is shifted through during time count three, TC03 (L08 - L15), along with the cylinder address which is read from the disk during this time. The address equal flip-flop, WAEQ, which is set by the ZSRP pulse prior to the arrival of the address bits, is cleared if there is not a direct correspondence between the contents of the registers and the address data being received from the disk. (The check-sum portion of the address is used only during the Write Header operation).

If there is not address equality, the Write operation will not take place during that sector. The Controller will await the next sector and repeat the above procedure. Whenever the address read from the disk is the same as the desired starting address, the address equal flip-flop, WAEQ, remains set and the Write operation occurs during that sector.

Once address equality is ascertained, the writing of the sector commences. The timing increment control flip-flop, F1TTIC, clears (resets) at the end of the address portion of the sector, which is coincident with F1THDR. With TTIC cleared, the timing counter is again incremented under the control of the basic 1.25 MHz Controller oscillator. The time counter is then incremented on the trailing edge of each CK2.

G1LRED, the read signal for clocking the address from the disk, ends with F1THDR going false. At this time the write and erase signals, G1LWRT and D1LERA, are enabled to begin writing the tolerance gap. Since the erase head is mounted physically ahead of the write head, the information will pass underneath the erase head before reaching the write head. The write and erase currents are turned on at the same time and the area passing beneath the erase head is erased prior to its passing beneath the write head.

The first information written, once the write current is turned on, is the tolerance gap of 72 "one" bits. The control of the number of "one" bits to be written is governed by the gray-code counter, consisting of F1WDL1 and F1WDL0. If there is address equality, WDL1 sets on the trailing edge of the L23 of the address portion. On the trailing edge of the next L23, WDL0 sets. On the third L23, WDL1 clears (resets) and WDL0 remains set. During these three word times, when at least one of these flip-flops was in a set condition, the disk write line gate, G1WDWL, is enabled. G1WDWL ANDs with the output of the data-to-disk enable 1 gate, G1WDD1, which is enabled while THDR is false, to enable the "one" bits to be sent to the disk through D1W1BE. At the end of these three word times,

the gray-code counter ends up with both flip-flops cleared, which disables G1WWDL and terminates the writing of the tolerance gap.

The period following the tolerance gap is that of the sync bits. This consists of 23 "zeroes", followed by a "one" bit. The "zeroes" are sent to the Disk, from L00 through L22, since the retime flip-flop, F1WRTM, remains cleared. On the trailing edge of L22, the retime flip-flop is set and W1BE is enabled during L23, which generates the "one" bit for the sync word.

Following the sync bits are the actual data words, written in a continuous stream of 1536 bits. This constitutes 64 data words of 24 bits each. A timing diagram depicting the relevant sequence of the actual data word transfers is shown on Fig. OUT 7.4. The data flow during this time is also given in the basic block diagram, Fig. OUT 7.2.

The request for the first data word to be written on the sector, was actually made during sequence state five. The first word of data, therefore, is already contained in the Coupler's Data Buffer Register (DBR), when the write operation is begun. The data transfer flip-flop, F1TDXF, was set on the trailing edge of L22, during the sync word time. At CK1 of L23, the transfer pulse, D1XFR, is generated to cause the transfer of the contents of the DBR to the DCR. Beginning at the CK2 of L23, the Shift pulses (D1ICDS) are generated in the Controller to shift the contents of the DCR through the retime flip-flop, in the Controller. After shifting through the retime flip-flop, the bits are gated through the line driver, X1WDTD, to the write circuitry in the Disk Unit. This path is outlined on the simplified block diagram, Fig. OUT 7.2.

The data is received in the Disk Unit through a line receiver and fed to a gated toggle flip-flop. The toggle flip-flop changes state on the negative going edge of each pulse. The output of the flip-flop is then fed through an amplifier to a diode isolation card, and to the write head to be recorded on the disk. The erase driver is also enabled to provide current through the GAA diode board to the erase head. The center tap for these heads, which supplies the return line, is provided by the head select card. One of these head select cards (IOD's) is enabled as a result of a head address being placed into the Head Address Register during sequence state five.

As may be seen on Fig. OUT 7.4, the sixty-fourth request during the sector results in the first word of the next sector being loaded into the Coupler DBR. The Controller, however, continues to serially shift the last data word of the sector to the disk.

During the write operation, the Block/Word Count Register is ring shifted and decremented during each word time. The shifting takes place during L00 - L13. The carry flip-flop is pre-set at L23, prior to the shifting, and during the shift the output of the register is sampled. If a "one" bit from the register is detected during L00-L05 of the shift, the carry flip-flop, F1BCRY, is cleared, indicating the word count has not yet reached zero. The Controller continues receiving data from the Coupler and

transferring it to the disk until the word count (bits 00 - 05) are detected as being zero. During this shift, F1BCRY will remain set at the end of L05 and at L06 the header flip-flop, F1THDR, will set, indicating the next information to be written on the disk is the data check-sum bits.

The generation of the check-sum bits is accomplished by allowing the output of the retime flip-flop, F1WRTM, to be used as an incrementing signal whenever a "one" bit is received from the Coupler during the writing of data. Through the synchronization/check-sum count driver, ZSCC, the Check-Sum Register is incremented for each "one" bit that passes through F1WRTM during the time data is transferred.

Following the last data word of the sector, the computed check-sum is written on the disk. This is accomplished by serially shifting the Check-Sum Register, with shift signal D1ZSCS, and comparing its output at the one-bit enable gate, D1W1BE, during the period following the transfer of the last word of data. The time during which the check-sum is written on the disk covers 12 "L" times, L00 - L11. During this time, WDDL gates any "one" bit from the register through the one-bit enable driver, D1W1BE.

Upon completion of the writing of the data check-sum bits, D1W1BE is held true from L12 - L23 by TEV2 being false, which enables G1WDD1. During this time, 12 "one" bits are written on the disk. This period is followed by a block of 72 "one" bits, which cover three word times and is controlled by the gray-code counter and the sync flip-flop, F1WSYN.

The gray-code counter flip-flop, WDL1, was set on the trailing edge of the L23 that gated the last data bit through the Controller to the Disk. The counter enables WDDL for the subsequent three word times. This covers: the data check-sum (12 bits); 12 "one" bits; and two more word times of 48 "one" bits.

At L23 of the count where WDL1 is cleared and WDL0 is set, the sync flip-flop is set. Since the erase gate cleared on the L23, after the header flip-flop set for the second time in the sector, the sync flip-flop, F1WSYN, is permitted to enable D1W1BE, for the one word time that it is set. With the clearing of F1WSYN, after one word time, the address equality flip-flop, F1WAEQ, is cleared. The clearing of WAEQ causes the write gate to go false and terminate writing on the disk for the remainder of that sector. The remaining portion of the sector would already have been filled by "ones" during the Write Header operation.

When the required number of sectors has been recorded, the Block/Word Count Register will have decremented to zero. Therefore, after writing the last data word for the sector, the Block/Word Count Register will be ring shifted during the period of L00 - L13. Since the register contains all "zeroes", F1BCRY, which was pre-set at the L23, prior to the shift, will remain set throughout the shift. The condition of F1BCRY is sampled at L16 and since BCRY is set, the terminate flip-flop, F1TTRM, is set, indicating the required number of blocks has been written on the disk.

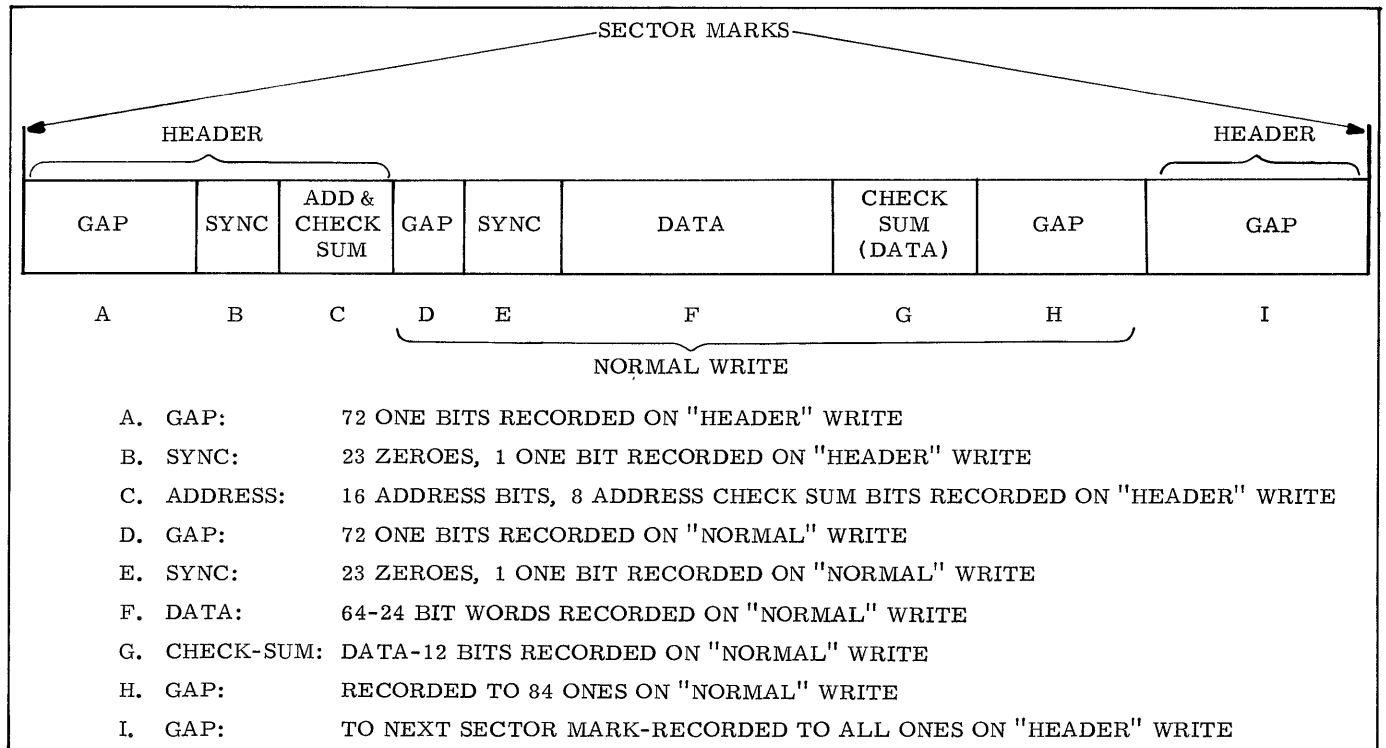


Fig. OUT 7.1. Sector Format

With F1TTRM set, the reset busy driver, D1QRBY, will be enabled and on the trailing edge of the sync signal, F1WSYN, the sequence counter will be cleared to zero, terminating the write operation.

## READ OPERATION

Visual references regarding the read operation include: Sector Format, Fig. OUT 7.1; Basic Block Diagram, Fig. OUT 7.5; and the timing diagrams, Figs. OUT 7.6 and OUT 7.7. Figure OUT 7.5 depicts the general flow of data as it is transferred from the read head of the Disk to Core Memory. The timing diagram, Fig. OUT 7.7 shows the main sequence of events which take place during a sector read. Fig. OUT 7.6 shows, in more detail, the data as it is transferred from the read head of the Disk to Core Memory. Reference should be made to the logic drawings involved: the Coupler, Controller, and Disk logic.

The read enable signal, G1LRED, is enabled throughout sequence state seven on a read operation. The read operation begins when address equality has been determined. Upon detection of the synchronization pulse, D1ZSRP, contained in the sync bits preceding the data area, the data transfer flip-flop, F1TDXF, is set. During the time TDXF is set, the information contained on the disk will generate strobes to shift the data to the Coupler as it is read from the disk. F1TDXF clears at the time of the last data bit being read from the disk. Although the data continues to enter the Controller after this time, it is not shifted to the Coupler and requests to transfer information to Memory are not generated. The data is read from the next sector in the same

manner. Each sector read results in 64 data words, containing 24 bits each, to be transferred to Memory. Sector reads will discontinue when the Controller logic detects that the requested number of blocks have been transferred (block/word count equal to zero).

The enable time counter flip-flop, F1TETC, which had been cleared at the end of sequence state five, sets on the trailing edge of sequence state six. Therefore, upon entering sequence state seven, the time counter increment driver, D1TTTCI, is enabled and the counter will begin counting again. At the arrival of the first sector marker, TETC is cleared and the non-data portion flip-flop, F1TNDP, is set. The read gate, G1LRED, which is enabled throughout sequence state seven, allows the data to be sensed on the head windings and coupled through the read circuitry to the Controller. G1LRED is sent to the Disk on the bus lines and is gated into the correct circuitry in the Disk by the control select signal, PCTS, which is enabled through-out sequence state seven. The read signal is received in the Disk, ANDed with the control select signal, inverted, and sent to the OVA zero cross detection circuit, where it is used to gate the data through the Disk read circuitry.

The first information to pass under the read head is the tolerance gap of 72 "one" bits. Following the tolerance gap are the sync bits. The tolerance and sync bits are read in the same manner as during the write operation. The address comparison is also identical with the write operation. Following the address bits is another group of tolerance gap and sync bits, which were written during the write operation. The read operation reads these bits in the same manner as those preceding the address

bits and, upon detecting the ZSRP pulse, the time counter is again enabled as the Controller prepares to receive the data portion of the sector.

This second ZSRP pulse ANDs with  $\overline{\text{TNDP}}$  and WAEQ to set the data transfer flip-flop, F1TDXF. As may be seen in Timing Diagram, Fig. OUT 7.6, the setting of TDXF permits the shift pulses, D1ICDS, to be generated by the D1RTCP pulses. Also shown in the timing diagram are some representative waveforms of the read signals and various control signals present during the data transfer from the Disk to the Coupler. The diagram is divided into three portions. The first portion is at the beginning of the first word to be transferred to Memory. It contains a combination of "ones" and "zeroes". The second portion of the diagram shows "one" bits being transferred at the end of the word time. The final portion of the diagram shows the end of the last word in the sector being transferred to Memory. This portion illustrates a string of "zero" bits being read.

The output of the selected read head is fed to the input of the ATB circuit in the Disk. The input is shifted ninety-degrees by the ATB circuit and amplified by the EUC circuit. The resulting waveform, from the EUC circuit, is fed to the EWA circuit where the zero crossings are detected and the output is a square wave whose polarity changes are at the point of the zero crossings of the input. From the EWA circuit, the pulses are sent to the OVA circuit, which generates negative pulses for each level change. The input to the OVA circuit is enabled by the combination of the control select signal and the read signal sent from the Controller. The output of the circuit is inverted and sent to the Controller through a line driver/line receiver combination. The output of the line receiver in the Controller is sent to the read recovery circuitry. It is fed to a single-shot which forms negative, 30 ns. pulses from the positive leading edges of the input. As may be seen in Fig. OUT 7.6, these negative pulses are at a 2.5 MHz rate while reading "one" bits and a 1.25 MHz rate while reading "zeroes". The output of the single-shot is inverted and ANDed with the output of another single-shot, S0RDST, at the input of G0RCLK. When the output of G0RCLK is enabled, it sends a 30 ns. pulse, which is inverted to S0RDST. The output of RDST then goes negative for a period of approximately 500 ns. This negative pulse disables G0RCLK for this period, thus allowing only the 1.25 MHz pulses to be gated to the single-shot, S0RCKP. S0RCKP forms 120 ns. pulses, which are inverted and clear the data flip-flop, F1RDTA, on their trailing edge. F1RDTA can only be set by its dc input, which occurs when G0RDTA is enabled, as a result of reading a "one" bit. G0RDTA is enabled by the inverted output of the single-shot, RDST, and is true during the 500 ns. period following the detection of a "zero" bit. The gate, therefore, detects any "one" bits being sensed in this interval. Therefore, the data flip-flop, F1RDTA, is set on the detection of a "one" bit and cleared by the normal clock pulses which were written between these bits.

The shift pulse, to gate the information from the Coupler to the Controller, is derived from the data being received

from the disk. The clock pulse, RTCP, is generated by the "zeroes" which were written on the disk. As mentioned in the write operation, the basic clock frequency was recorded if there were no "ones" to be written. This serves as a clock when reading the data from the disk. The data, therefore, is self-strobing during the read operation. (If there were no pulses recorded on the disk, "ones" or "zeroes", there would be no strobe pulses to gate it to the Controller; and since there would be no F1RTCP pulses, no memory requests would be made).

The data is read serially from the disk and sent through the Controller to the Coupler to be assembled in the Coupler's Data Collection Register (DCR). As mentioned previously, the data is shifted into the DCR under the control of the D1ICDS pulses which originate in the Controller. After assembling a complete word in the DCR, a transfer pulse (D1IXFR) is generated in the Controller, which notifies the Coupler to shift the data parallel to the Coupler's Data Buffer Register (DBR). The Controller's memory request flip-flop, F1IMRQ, is set at this time, which informs the Coupler through a line driver/line receiver combination. With receipt of this information the Coupler's memory request flip-flop, F1GMRQ, is set, which notifies the memory that a cycle is needed to transfer data to Memory.

Since the data assembled in the DCR was transferred to the DBR, the DCR is free to continue receiving data, which is the next data word of the sector. This data word will be read in the same manner as the one just mentioned.

With acknowledgement of a request, the Memory will return a signal (MUGOMBR\*) to the Controller. The Memory will then write the information, contained in the DBR, into Memory. When the next data word is fully assembled in the DCR, it will also be transferred to the DBR, a memory request made, and upon acknowledgement, stored into Memory.

During the reading of each data word from Memory, the Block/Word Count Register is ring shifted (L00 - L13) and one subtracted from it. The carry flip-flop, F1BCRY, is pre-set at the L23 before this shift. If there is a "one" bit in bits 00 - 05, it is determined that all the words (64) of the sector have not yet been read. Therefore, F1BCRY will be cleared and the header flip-flop, F1THDR, will be inhibited from setting at L06 time. Once the word count (L00 - L05) reaches zero, F1BCRY will remain set through L05 and at L06 the header flip-flop will set. This informs the Controller logic that the next group of bits is the data check-sum.

As the data was being read from the disk, the "one" bits incremented the Data Check-Sum Register via D1ZSCC. During the reading of the data check-sum from the disk, this register is shifted and the count accumulated from the incrementing by each "one" bit is compared with the data check-sum being read from the disk. This data check-sum recorded on the disk, it should be remembered, was recorded as part of the write operation. The purpose of the comparison is to check the validity of the data read from the disk. The check-sum compare

flip-flop, F1ZCSC, is pre-set with the "one" bit of the sync word which precedes the data. If there was a non-comparison of the check-sum counter and the count read from the disk, ZCSC will be cleared. At the next L23, the data check-sum error flip-flop, F1RDCS, would be set in the Status Register, indicating an error.

Following the reading of the data check-sum, the read circuitry continues to detect information from the disk, but will not transfer it to the Coupler until the data area of the next sector.

The above operation is repeated through successive sectors until all of the bits (00 - 13) of the block/word count reach zero. With this condition, F1BCRY would still remain set at L16 of the word time and the terminate flip-flop, F1TTRM, would set. Then, with the clearing of the header flip-flop, F1THDR, which is true during the data check-sum read, the sequence counter

would be cleared to zero by D1QRBY. This would terminate the read operation.

If either a read or a write operation is carried to the end of a track before the block/word count reaches zero, the next successive head must be selected. This causes the Controller to go into a head update cycle. This condition is detected by the Controller when the Desired Sector Register reaches a count of 15<sub>10</sub>. The head update flip-flop, F1HHUD, is set and the Desired Head Register incremented by one. A head register increment pulse, N1LHAD, is also sent to the Disk to advance the Disk's Head Address Register by one. The Desired Sector Register, in the Controller, is merely incremented by one to return it from a count of 15<sub>10</sub> to 00<sub>10</sub>.

If the head update is on the last sector of the last track of a cylinder, a desired cylinder increment cycle will result. This will cause the heads to be moved to the next successive cylinder, as is discussed in SS6.

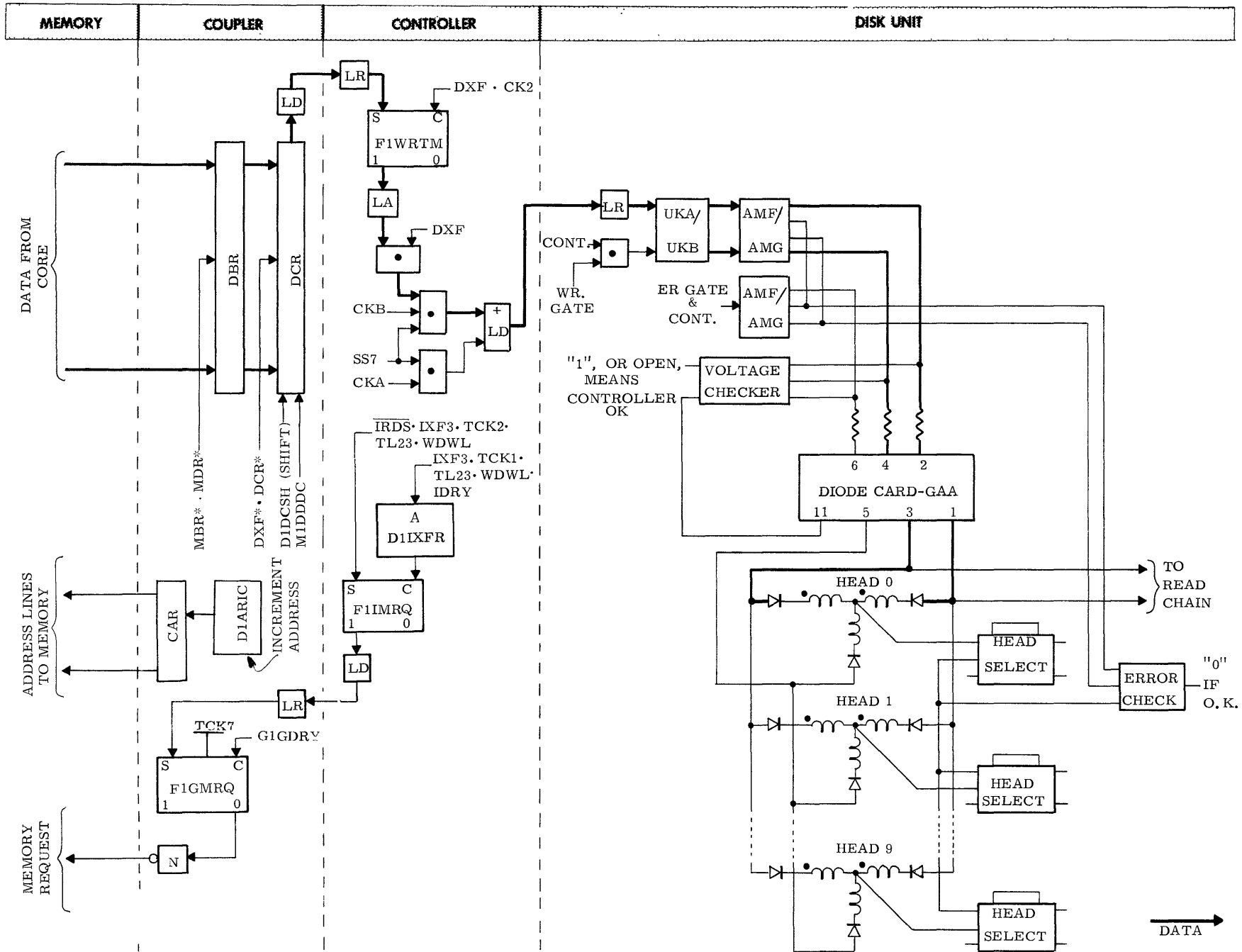
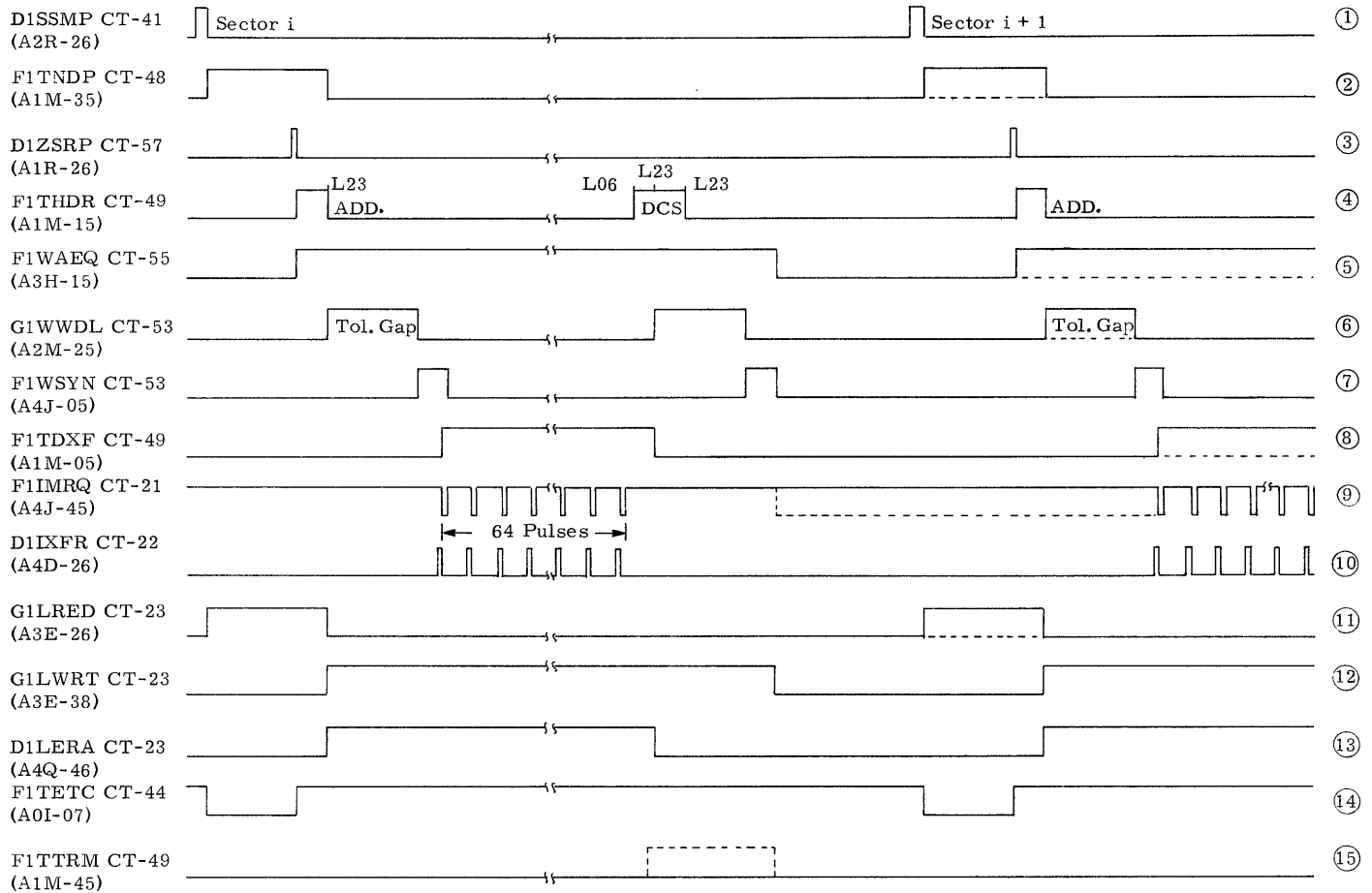


Fig. OUT 7.2. Data Transfer Path-WRITE



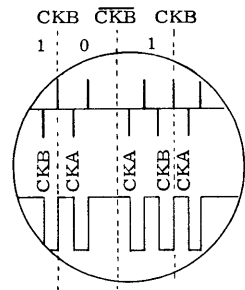
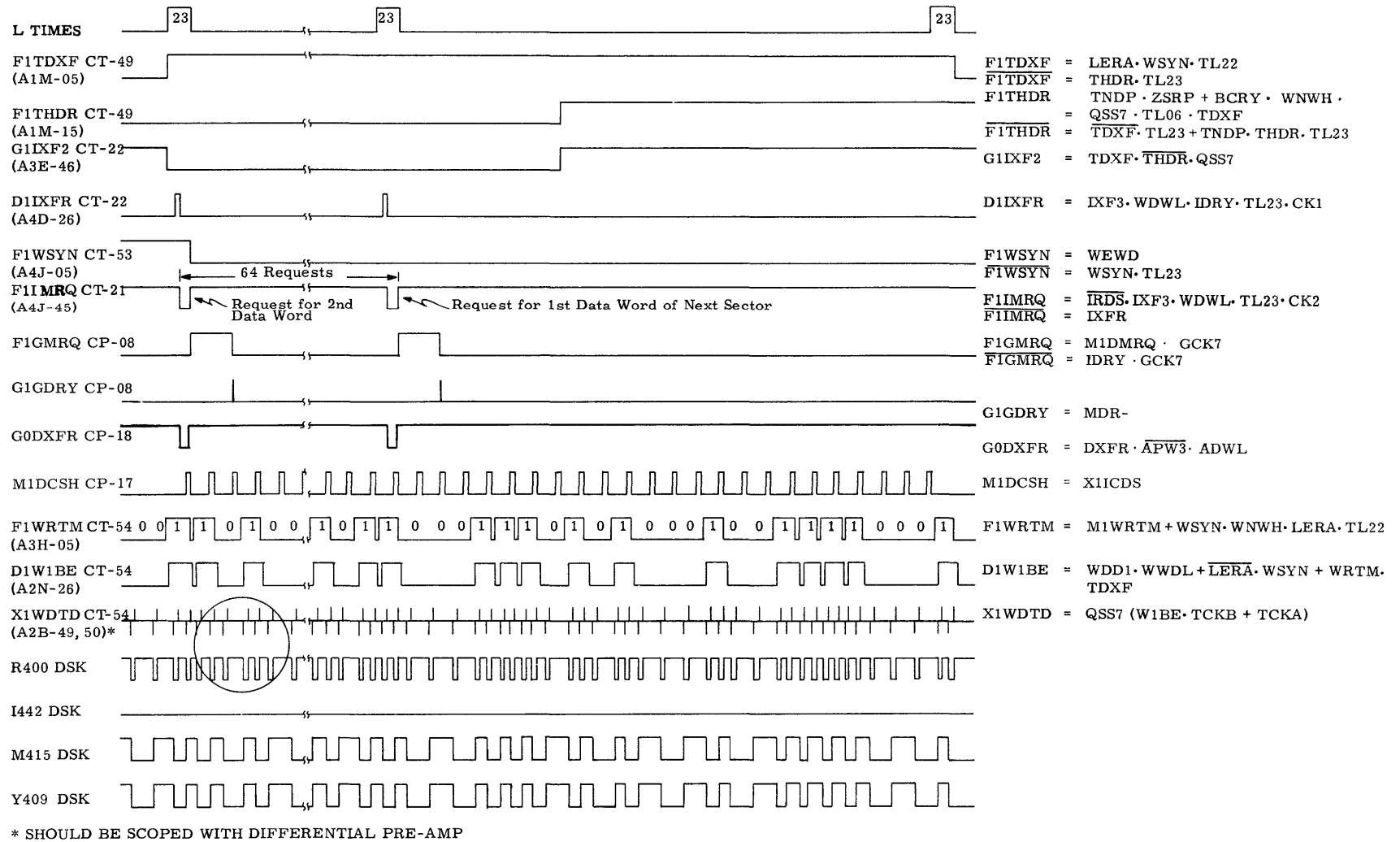


- ①  $D1SSMP = SSMP \cdot QSS7$
- ②  $F1TNDP = SSMP$   
 $\overline{F1TNDP} = \overline{THDR} \cdot \overline{IABP}$
- ③  $D1ZSRP = ZC04 \cdot \overline{TETC} \cdot RDTA$
- ④  $F1THDR = TNDP \cdot ZSRP + BCRY \cdot WNWH \cdot QSS7 \cdot TL06 \cdot TDXF$   
 $\overline{F1THDR} = \overline{TDXF} \cdot TL23 + \overline{TNDP} \cdot \overline{THDR} \cdot TL23$
- ⑤  $F1WAEQ = ZSRP \cdot TNDP$   
 $\overline{F1WAEQ} = \overline{THDR} \cdot \overline{TNDP} \cdot WAEG \cdot TTCI + WSYN \cdot \overline{LERA}$
- ⑥  $G1WWDL = WDL1 + WDL0$
- ⑦  $F1WSYN = WEWD$   
 $\overline{F1WSYN} = \overline{WSYN} \cdot TL23$
- ⑧  $F1TDXF = LERA \cdot WSYN \cdot TL22$   
 $\overline{F1TDXF} = \overline{THDR} \cdot TL23$
- ⑨  $F1IMRQ = \overline{IRDS} \cdot IXF3 \cdot WDWL \cdot TL23 \cdot CK2$   
 $\overline{F1IMRQ} = \overline{IXFR}$
- ⑩  $D1IXFR = IXF3 \cdot WDWL \cdot IDRY \cdot TL23 \cdot CK1$
- ⑪  $G1LRED = \overline{WWHD} \cdot QSS7 \cdot TNDP$
- ⑫  $G1LWRT = \overline{LRED} \cdot WDWT \cdot QSS7 \cdot WAEQ \cdot \overline{TNDP}$
- ⑬  $D1LERA = LERA \cdot LWRT$
- ⑭  $F1TETC = QSS6 + QSS7 \cdot ZSRP$   
 $\overline{F1TETC} = \overline{SSMP} \cdot WNWH$
- ⑮  $F1TTRM = TDXF \cdot BCRY \cdot \overline{QSS3} \cdot \overline{QSS4} \cdot TL16$   
 $\overline{F1TTRM} = \overline{IBY2} \cdot CK1$

NOTES:

- ① Pulse relationships not drawn to scale, intended for sequence reference.
- ② Dotted Lines Become true upon completion of disk write operation, i. e. Block/Word Count = 0
- ③  $D1SSMP \approx 800$  nsec.
- ④  $D1ZSRP$  is the one bit following the 23 zero bits.

Fig. OUT 7.3. Timing Diagram-WRITE



TRANSMISSION DELAY BETWEEN TOP AND BOTTOM SIGNALS NOT ILLUSTRATED. DELAY VARIES WITH DISTANCE THAT DISK IS LOCATED FROM CONTROLLER

Fig. OUT 7. 4. Timing Diagram-WRITE

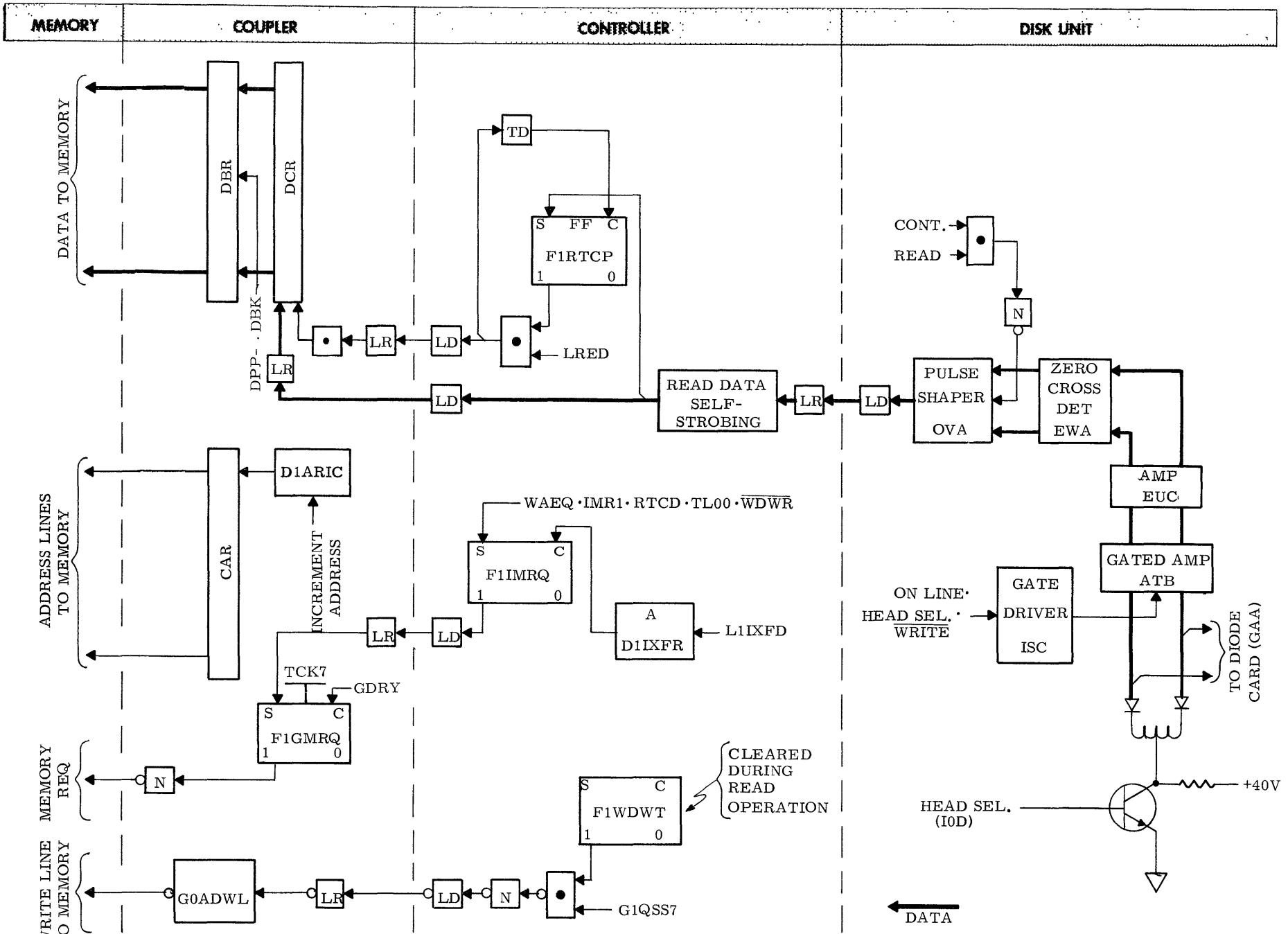
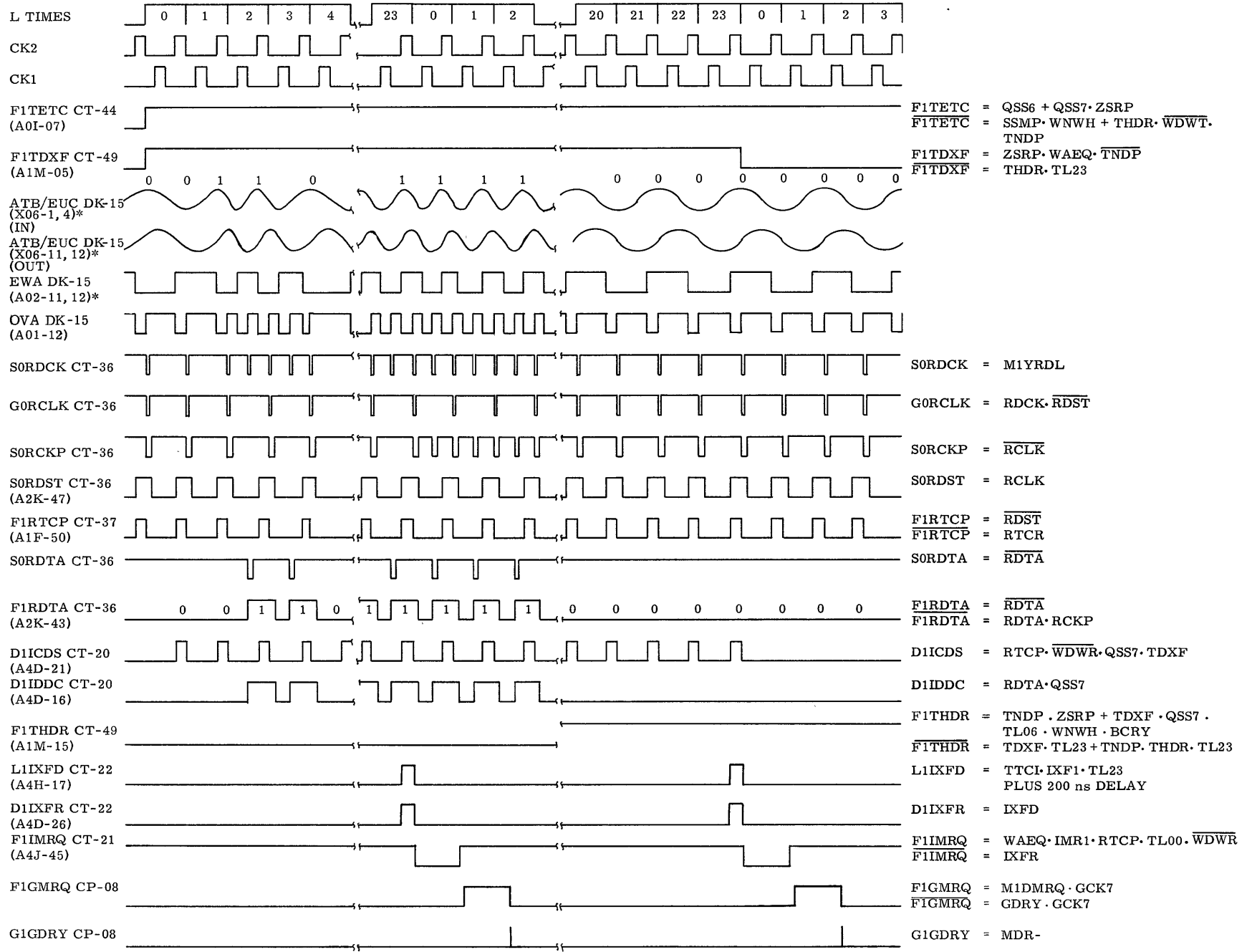
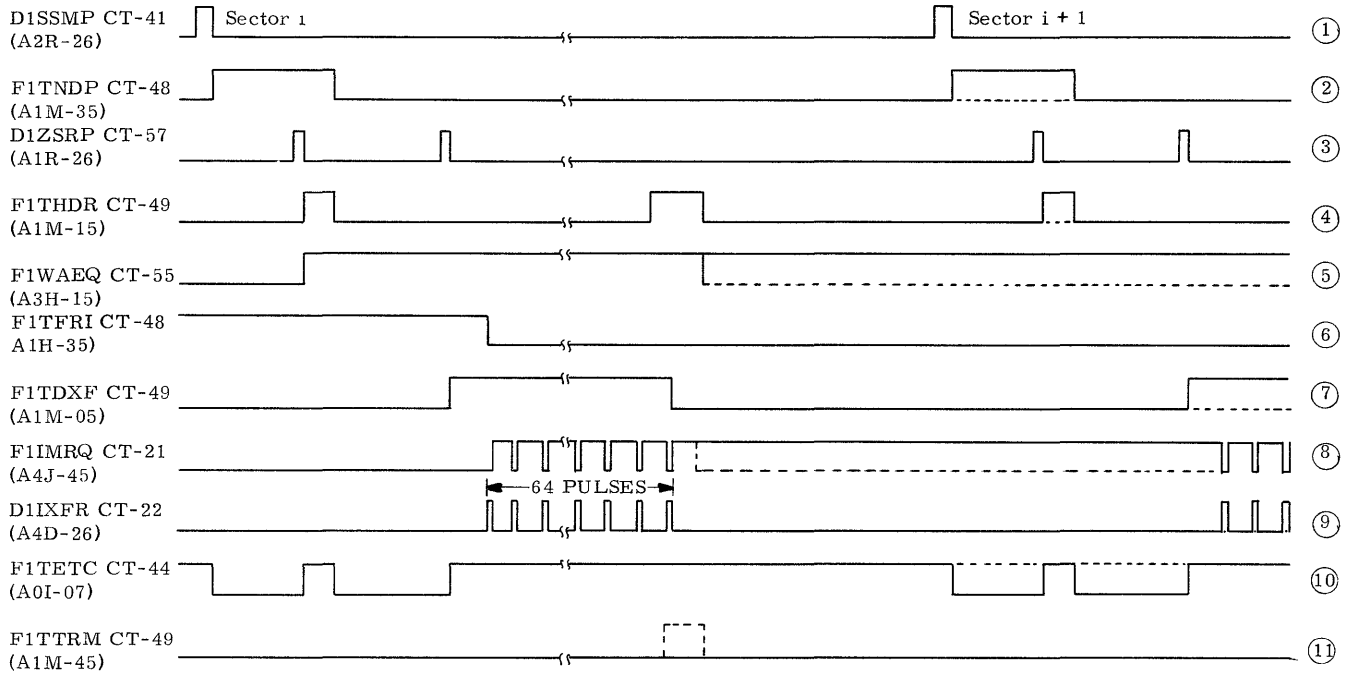


Fig. OUT 7.5. Data Transfer Path-READ



\*SHOULD BE SCOPED WITH A DIFFERENTIAL PRE-AMP  
- = 1, 2, or 3

Fig. OUT 7.6. Timing Diagram-READ



- ①  $D1SSMP = SSMP \cdot QSS7$
- ②  $F1TNDP = SSMP$   
 $\overline{F1TNDP} = THDR \cdot \overline{IABP} + QRBY$
- ③  $D1ZSRP = RDTA \cdot \overline{TETC} \cdot ZC04$
- ④  $F1THDR = QSS7 \cdot WNWH \cdot BCRY \cdot TDXF \cdot TL06$   
 $\overline{F1THDR} = TDXF \cdot TL23 + TNDP \cdot THDR \cdot TL23$
- ⑤  $F1WAEQ = ZSRP \cdot TNDP$   
 $\overline{F1WAEQ} = WSYN \cdot \overline{LERA}$
- ⑥  $F1TFRI = \overline{WDWT} \cdot QSS5$   
 $\overline{F1TFRI} = IXFR$
- ⑦  $F1TDXF = ZSRP \cdot WAEQ \cdot \overline{TNDP}$   
 $\overline{F1TDXF} = THDR \cdot TL23$
- ⑧  $F1IMRQ = WAEQ \cdot IMR1 \cdot RTCP \cdot \overline{WDWR} \cdot TL00$   
 $\overline{F1IMRQ} = IXFR$
- ⑨  $D1IXFR = IXFD$
- ⑩  $F1TETC = QSS6 + QSS7 \cdot ZSRP$   
 $\overline{F1TETC} = SSMP \cdot WNWH + \overline{WDWT} \cdot TNDP \cdot THDR$
- ⑪  $F1TTRM = QSS3 \cdot QSS4 \cdot BCRY \cdot TDXF \cdot TL16$   
 $\overline{F1TTRM} = \overline{IBY2} \cdot CK1$

Fig. OUT 7.7. Timing Diagram-Read

## WRITE HEADER OPERATION

The Write Header operation is a "special case" Out command, used to establish essential information, other than data, on a sector of a disk. This is accomplished by the Controller logic, using information provided by a "Write Header" program. The pre-address sync bits, the address bits, and the address check-sum bits are provided by the program. The remainder of the sector, including the data area, will then be filled with "ones" by the Controller during the Write Header operation.

While performing a Write Header operation, the "Normal/Header" switch, located on the maintenance console, should be in the "Header" position which differentiates the "header" write from a "normal" write. After the execution of the Write Header, a "normal" write should be performed to establish the pre-data sync bits for the recorded data. If this is not done, an address compare error will be detected the next time a Read operation is attempted on that sector.

The header information is written on a track-by-track basis, i.e. the "header" program will write all 16 sectors of a cylinder under a specific head, referred to as a track. By this method, a track containing bad header information may be restored.

The Write Header operation requires two data words from Memory during each sector. These words consist of: the pre-address sync bits, 23 "zeroes" and a "one"; the address bits, 16 bits; and the address check-sum bits, 8 bits. The order in which this information is recorded on the sector may be seen in Fig. WHD 7.1. Since there are 16 sectors to a track, this will require 32 data transfers during each track.

The Write Header operation is essentially the same as a "normal" Out command, specifying a write operation, until sequence state seven. The block/word count, however, should always be loaded with a decimal count of 32, in order to fill every sector of the track. Since a 32<sub>10</sub> word count is needed in the Block/Word Count Register, only the five least significant bits (04 - 00) are cleared following the loading of the register during sequence state four. This differs from a "normal" write in which the six least significant bits (05 - 00) are cleared, following the loading of the register. The starting cylinder address, contained in the control word one, will be that of the desired track to be recorded.

During sequence state seven the operation differs from the "normal" write and, therefore, will be described in the following text, while references may be made to sequence states prior to seven, as described under the heading "Out Command".

References should be made to Figs. WHD 7.1, 7.2, and 7.3, during the description. The write portion of the Out Command flow chart may also be followed.

During sequence states 1 through 6, the word count and address selection information would have been transferred to the Controller. The Controller, in turn, would have executed an address search (by the transmittal of seek forward or backward to the Disk, if the heads were not already positioned over the correct cylinder) and

upon reaching the desired track would have received a "seek complete" from the Disk. Upon detection of the index marker from the Disk, the Controller would then advance to sequence state seven.

With the setting of sequence state seven, the Controller prepares to begin writing on the disk. The enable time counter flip-flop, F1TETC, sets on the trailing edge of sequence state six. The control select signal, D1PCTG, is enabled throughout sequence state seven. D1PCTG enables the write and erase bus-lines to be gated to the Disk to enable the write and erase drivers. The write and erase bus-lines are enabled at the beginning of sequence state seven by G1LWRT and F1LERA, respectively. The non-data portion flip-flop, F1TNDP, sets on the trailing edge of sequence state six.

The first thing written on the disk is a gap of 72 "one" bits. The enabling period for these bits is governed by the word delay signal, G1WDDL, which initially goes true with the setting of the word delay line 1 flip-flop, F1WDL1. The combination of the two flip-flops, F1WDL1 and F1WDL0, form a gray-code counter which enables WDDL for three word times. The counter is incremented every L23 time during their count. The counter, therefore, enables the writing of "ones", through the one bit enable driver, D1W1BE, for a period of three word times, or 72 bits. This will be prior to the setting of the data transfer flip-flop, F1TDXF.

F1TDXF sets on the trailing edge of the L22, which occurs when the gray-code counter is in its third count (WDL0 • WDL1). At this time the Controller begins writing the information, it received from Memory, on the Disk. The first request for information from Memory, as during a "normal" write, would have been made in sequence state five. The first word (sync bits: 23 "zeroes" followed by a "one") would, therefore, be waiting in the Coupler's Data Collection Register (DCR).

The word is shifted serially from the DCR to the Controller, through the retime flip-flop, F1WRM, as during a "normal" write. The shift pulses from X1ICDS occur at each CK2, during the period that the data transfer flip-flop, F1TDXF, is set. The shift pulses cause the word to be shifted to the Controller, where it enables W1BE each time it receives a "one" bit from the Coupler. As described under the Write portion of the Out command, if "zeroes" are to be written on the disk, only CKA's will be gated through X1WDTD to enable the write drivers in the Disk. To write "ones", the CKB's are gated also, thus doubling the frequency of the write pulse. As mentioned previously, the pattern of data written during this word time is 23 "zeroes" and a "one" bit.

At the CK2, following the setting of the TDXF flip-flop, the memory request flip-flop, F1IMRQ, is set. The setting of IMRQ is detected in the Coupler and a memory request generated. This information received from Memory (assuming no Memory hold-up, which would result in an error) is gated into the DCR. Upon completing the writing of the pre-address sync bits, the Controller governs the shifting of the address bits and address check-sum bits from the DCR to the Controller.

During the time the address bits are gated into the Controller, they are also used to increment the check-sum counter. The counter is cleared on the trailing edge of the synchronization pulse, F1WSYN, which occurs just prior to the gating of the address check-sum bits into the Controller. As the address bits are gated through the Controller, to be written on the disk, they are also sampled on one of the OR inputs to the synchronization check-sum count pulse, D1ZSCC. If the output of the retime flip-flop is a "one" during this time, it increments the check-sum counter by one. Since each "one" bit received increments the counter by one, a count of the number of "one" bits received will be kept in the counter. The increment to the counter is only enabled during TC04 of this word time, which is from L00 through L15.

Following the address bits are the address check-sum bits, generated by the Write Header program, which are gated to the Controller from L16 through L23. During TC04 (L16 -L23) the check-sum count is shifted through an "exclusive OR" circuit, to be compared with the data arriving from the Coupler. The check-sum compare flip-flop, F1ZCSC, was set on the trailing edge of the synchronization pulse, WSYN, by D1ZRCS. During the shift, the bits from the Coupler are checked against those of the check-sum counter in the Controller and the output of the exclusive OR checking circuit is applied to the clear (reset) side of ZCSC. If the check-sum bits compare, the flip-flop remains set. If they do not compare, the output of the checking circuitry causes the flip-flop to clear, thereby indicating an address check-sum error. During the time the check-sum bits are used to verify the address bits, they are also gated to the write circuitry to be written on the disk, as were the address bits.

Upon completion of the writing of the address bits and the address check-sum bits, the header flip-flop, THDR, is cleared. This, in turn, causes the non-data portion flip-flop, TNDF, and the enable time counter flip-flop, TETC, to be cleared. The data transfer flip-flop, TDXF, will also be cleared at this time, inhibiting data from being requested from Memory to be written on the Disk, for the remaining portion of the sector.

The remaining portion of the sector is filled with "one" bits. The "one" bit enable driver, D1W1BE, is enabled during the period that the data transfer flip-flop, TDXF, is cleared.

On the trailing edge of the next sector mark, the non-data portion flip-flop, TNDF, is again set and the word delay gate, WWDL, enabled. This starts the writing of information on that sector in the same manner as the previous one. The Controller continues to write pre-address sync bits, address and address check-sum bits on the disk until the detection of the block/word count equal to zero. The sync bits are always the same pattern, but the address bits coming from Memory, to be written on the disk, will be incremented for each sector to correspond to the addresses of that sector. This is established by the program when it assembles its data to be transferred to the disk.

If the proper word count (3210) was used in control word two, the terminate flip-flop, F1TTRM, is set on the trailing edge of the first L16 of the data transfer period of sector 15. This allows one more sector to be written upon.

At this time, all 16 sectors of the track have been written on. Therefore, the next sector will be sector 00, which has already been recorded. This second pass will re-record the gap bits of that sector. The purpose for doing this relates back to the arrangement of the write and erase heads. Since, on each of the 10 head assemblies, the erase head precedes the write head physically, as well as in time, as the disk surface passes beneath, there will be a gap between the heads during recording. When the write was initiated at the beginning of sequence state seven, the write and erase currents were turned on simultaneously. Therefore, the portion of the disk surface that had already passed the erase head, but had not yet reached the write head would have been written on without having first been erased. The re-recording of this area corrects this situation.

The second recording of sector 00 is restricted to the gap of 72 "one" bits, only. This is accomplished by clearing the sequence counter, coincidentally with D1WEWD, by D1QRBY. This causes the TDXF and TNDF flip-flops to clear early. The disk write line goes false, with the disappearance of SS7 and the memory request flip-flop, IMRQ, clears with the D1IXFR pulse. F1IMRQ will not set again until the Controller receives another command. At the time of the IXFR pulse, the data ready flip-flop, IDRY, and the terminate flip-flop, TTRM, clear. The Controller, therefore, is returned to a ready condition.

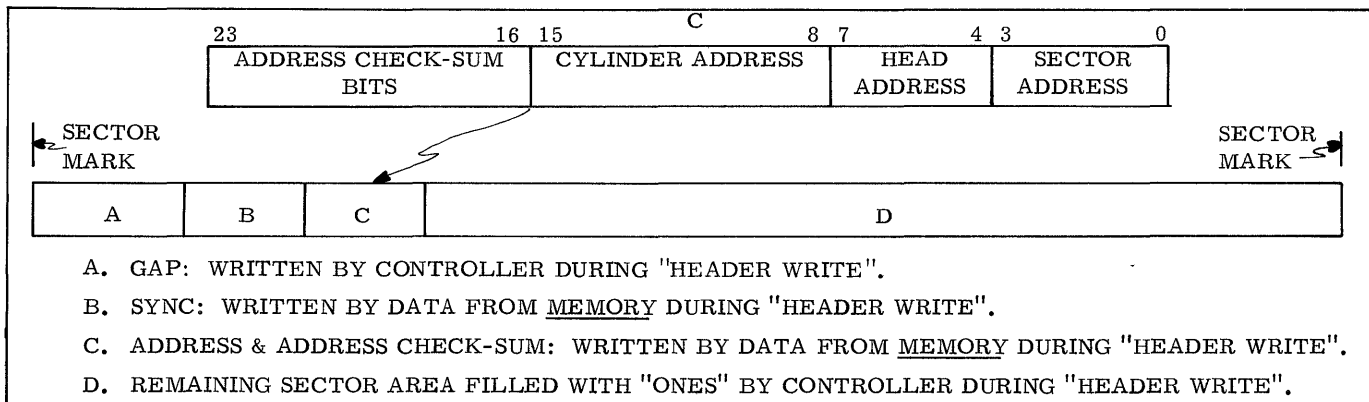


Fig. WHD. 7.1. Sector Format After Write Header





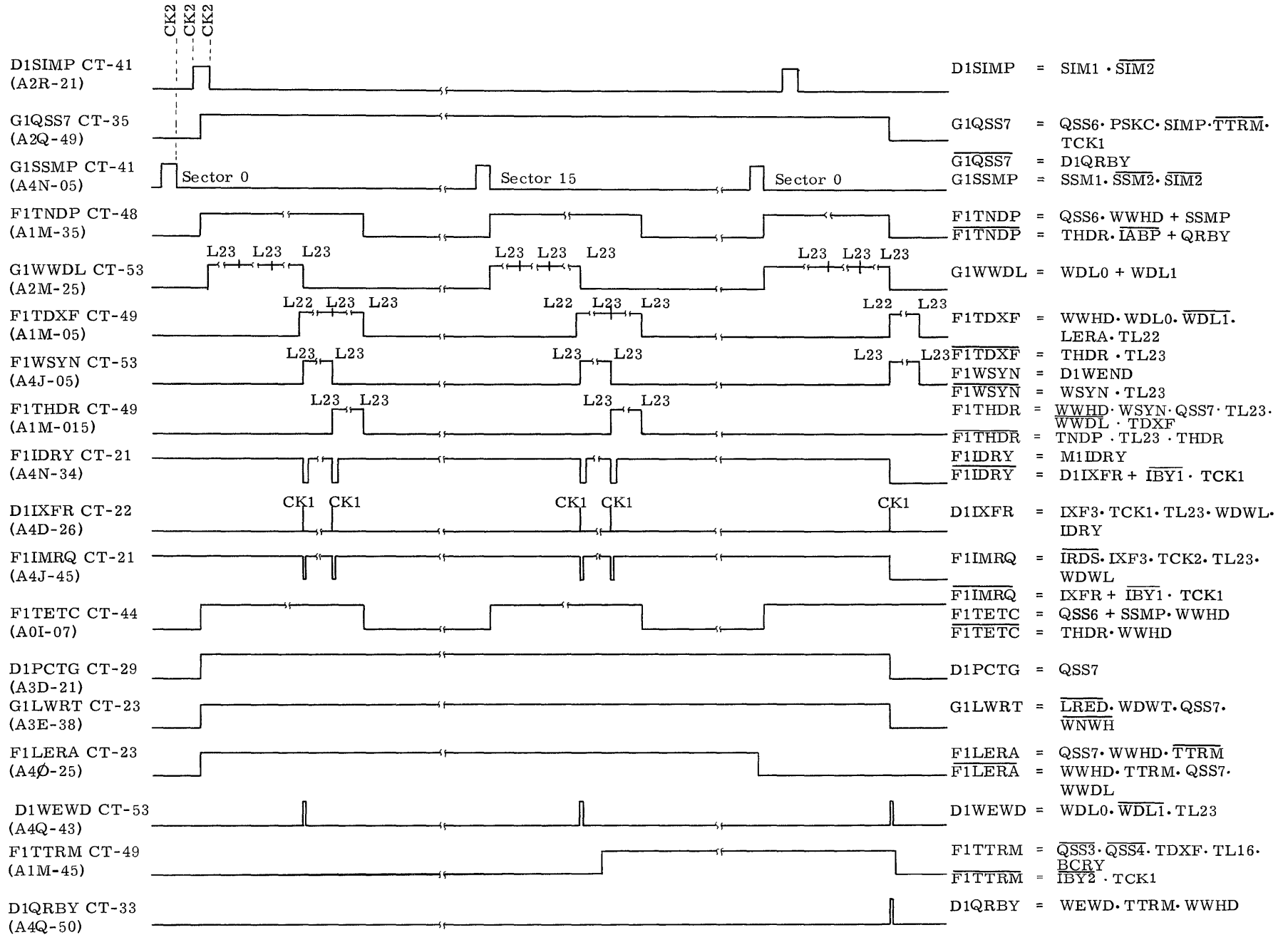
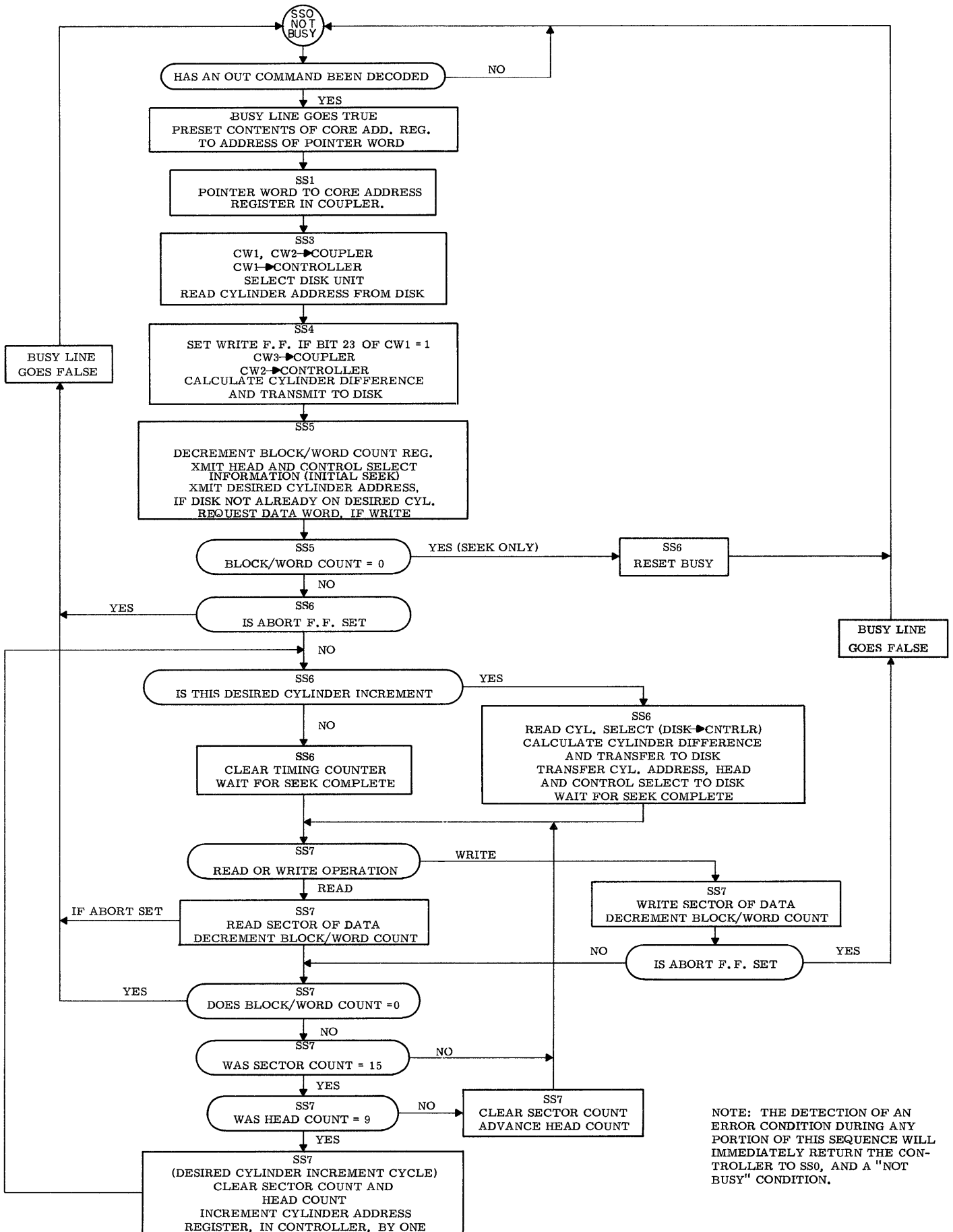


Fig. WHD 7.3. Timing Diagram, Write Header-SS7



NOTE: THE DETECTION OF AN ERROR CONDITION DURING ANY PORTION OF THIS SEQUENCE WILL IMMEDIATELY RETURN THE CONTROLLER TO SS0, AND A "NOT BUSY" CONDITION.

Fig. WHD 7.4 Flow Chart-OUT Command (Read/Write, Seek and Write Header Operation)

## READ STATUS SSO

During sequence state zero, upon receipt of a Read Status command, the Coupler presets the Core Address Register (CAR) to the address of the pointer word and clears the Data Collection Register (DCR). The read status flip-flop in the Controller is set to record the type of operation to be performed and the busy line goes "busy". The following text refers to Figs. RDS 0.2 and 0.3 as well as the Coupler and Controller logic. The flow chart, Fig. RDS 0.1 may be referenced throughout the description of the Read Status command.

As was mentioned in the description of the Out command, the Controller will be awaiting a command during sequence state zero. The decoding of a Read Status command, which is used to gate the status of the Disk Sub-System's error conditions to the pointer word in Core Memory, is detected in the Coupler by its "S" bits equal to an octal five.

At the  $\phi A$  pulse of the Read Status command, the command is recognized in the Coupler by the enabling of G0GRDS. This may only take place if the Controller is in a not busy (ready) condition. G0GRDS is inverted through D1APZX, which is used to enable clock driver, D0APZT, to preset the CAR to the address of the pointer word. D1APZX also enables clock drivers D0DPZX and D0DPZT to clear the DCR.

The output of G0GRDS is also inverted through G1GRDS and transmitted to the Controller to inform the Controller of a Read Status command by setting the read status flip-flop, F1IRDS.

With the setting of IRDS, the Controller's busy line goes true. This information is sent back to the Coupler through a line driver/line receiver combination and disables the read status gate, G0GRDS. The busy line is also sent to the AU for testing purposes and to the API to indicate the point at which the Controller becomes ready.

Within the Controller, an inverter on the zero (0) side of the read status flip-flop is used to disable the error test line during the execution period of the Read Status command. This inhibits an error from clearing the sequence counter to SS0 during the Read Status

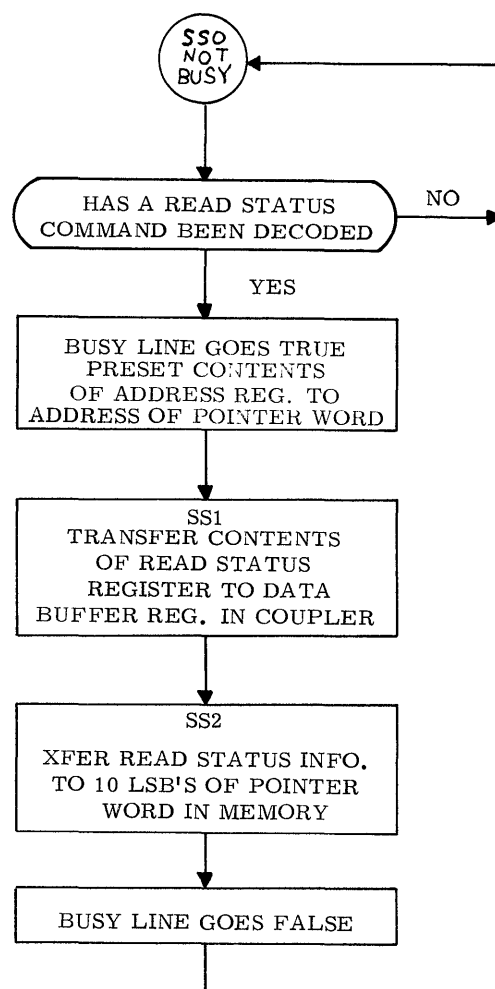


Fig. RDS 0.1. Flow Chart, Read Status Command

command. On the trailing edge of the first L23, after the recognition of the Read Status command, the sequence counter is incremented to SS1.

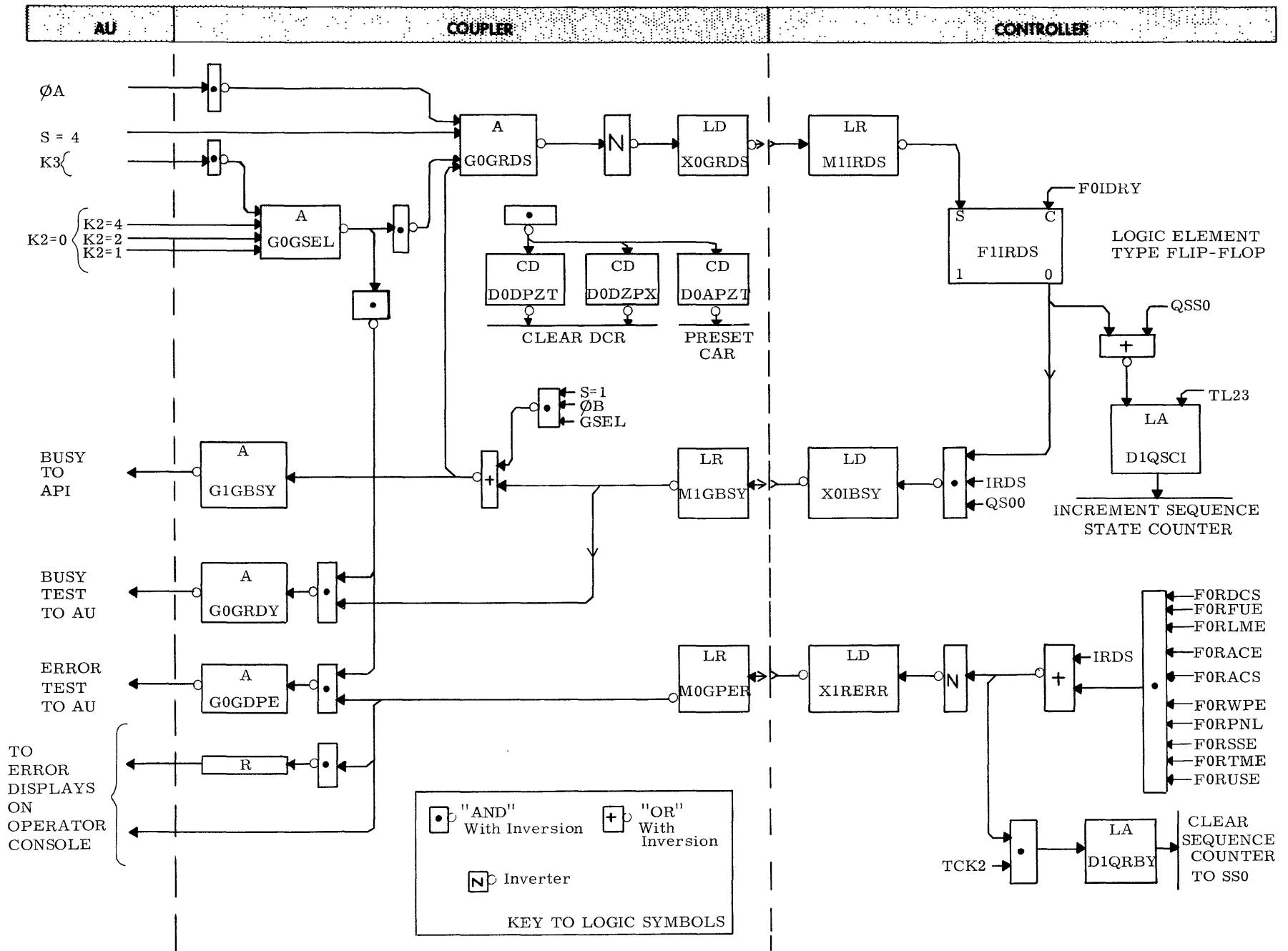


Fig. RDS 0.2. Block Diagram, Read Status-SS0

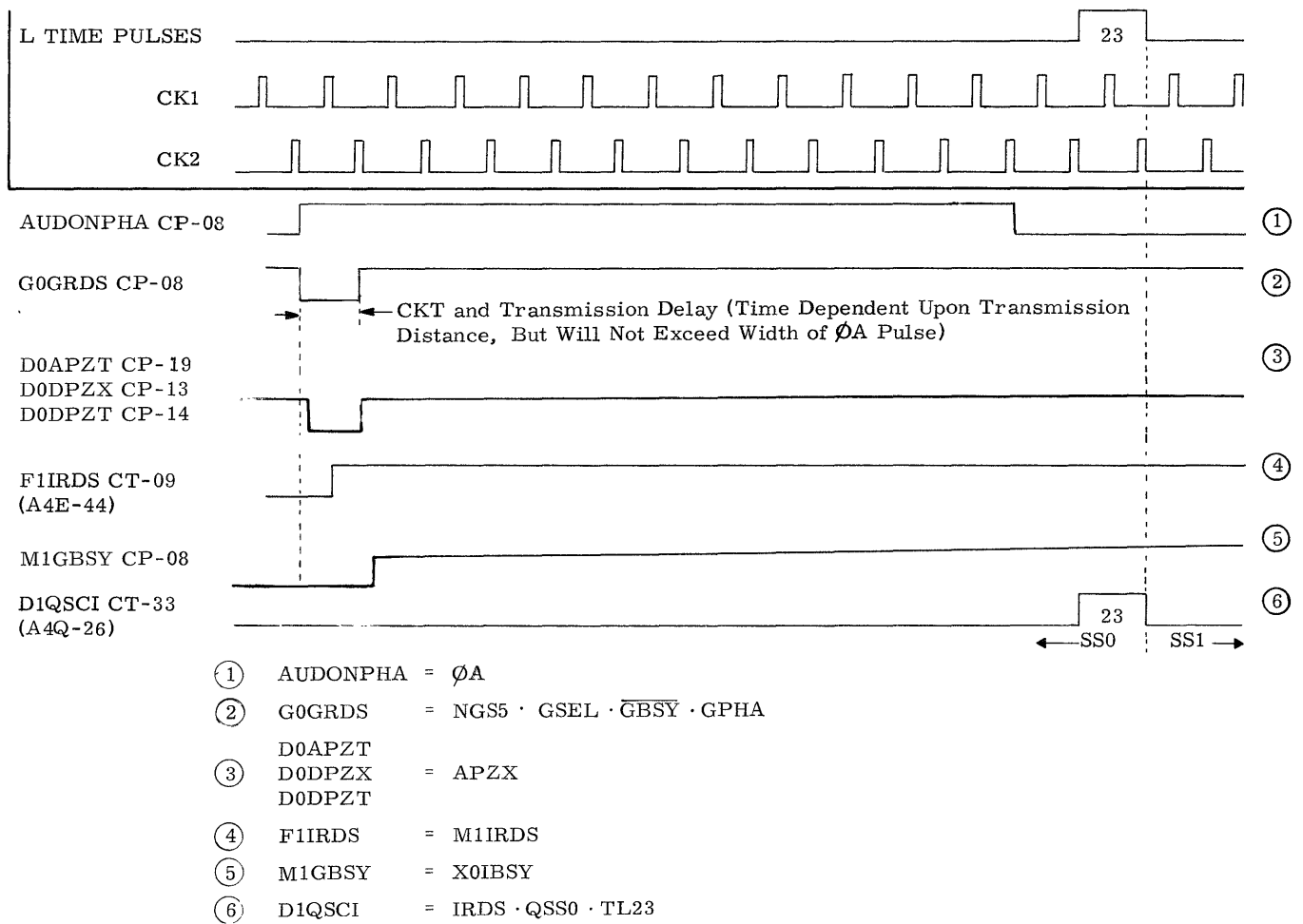


Fig. RDS 0.3. Timing Diagram, Read Status-SS0

# READ STATUS SS1

During sequence state one, the contents of the Controller's Status Register will be serially shifted to the Data Collection Register (DCR) in the Coupler. At the end of sequence state one, the transfer pulse, D1XFR, is generated and sent to the Coupler to clock the contents of the DCR into the Data Buffer Register (DBR). The sequence state counter is then incremented to sequence state two. References regarding this description should be made to the block diagram, Fig. RDS 1.2; the timing diagram, Fig. RDS 1.1; and the flow chart, Fig. RDS 0.1.

The shift signal, D1RSCC, is used to serially shift the contents from the Read Status Register. It is enabled by CK2's, during the period of L00 - L08 of this sequence state. As the register is shifted out its least significant end, it is presented to line driver, X1IDDC, and also at the inputs of the most significant flip-flop of the Read Status Register, F1RUSE. The register is, therefore, ring shifted at the same time as its contents are shifted to the Coupler. The information is received in the Coupler by line receiver, M1DDDC, which has a two-level output resembling that of a flip-flop. The

"one" and "zero" outputs from DDDC are ANDed with D1DCS1/D1DCS2/D1DCS3 at the set and clear inputs, respectively, of the most significant flip-flop (DC23) of the DCR. The information is clocked into the DCR by clock drivers, DCK4/DCK5/DCK6, which are enabled as a result of the shift pulses sent through the X1ICDS/M1ICDS combination to the Coupler. Twenty-four shift pulses will be generated to fully shift the contents of the Status Register into the DCR, with the 10 bits of the Status Register ending up in bits 00 - 09 of the DCR.

The outputs of the individual flip-flops of the DCR are ANDed with gating signals, DPPS/DPPX/DPPY, to arm the inputs of the respective bits of the DBR. The transfer pulse, D1XFR, which originates in the Controller at CK2 of L23, is sent to the Coupler to enable clock drivers, DCK4/DCK5/DCK6, which clock the bits from the DCR to the DBR.

Also at the end of L23, the sequence state counter increment signal, D1QSCI, is generated to step the sequence counter to SS2.

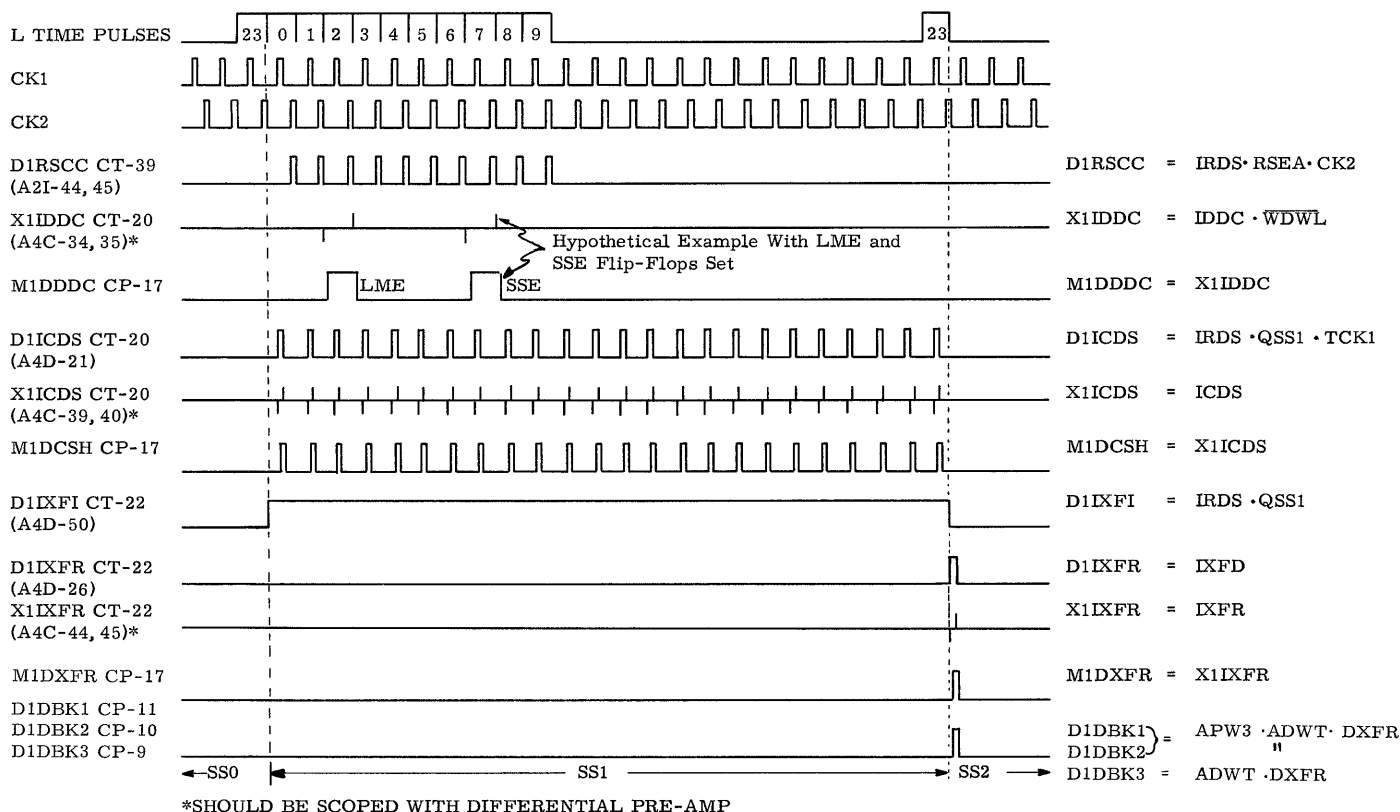


Fig. RDS 1.1. Timing Diagram, Read Status-SS1

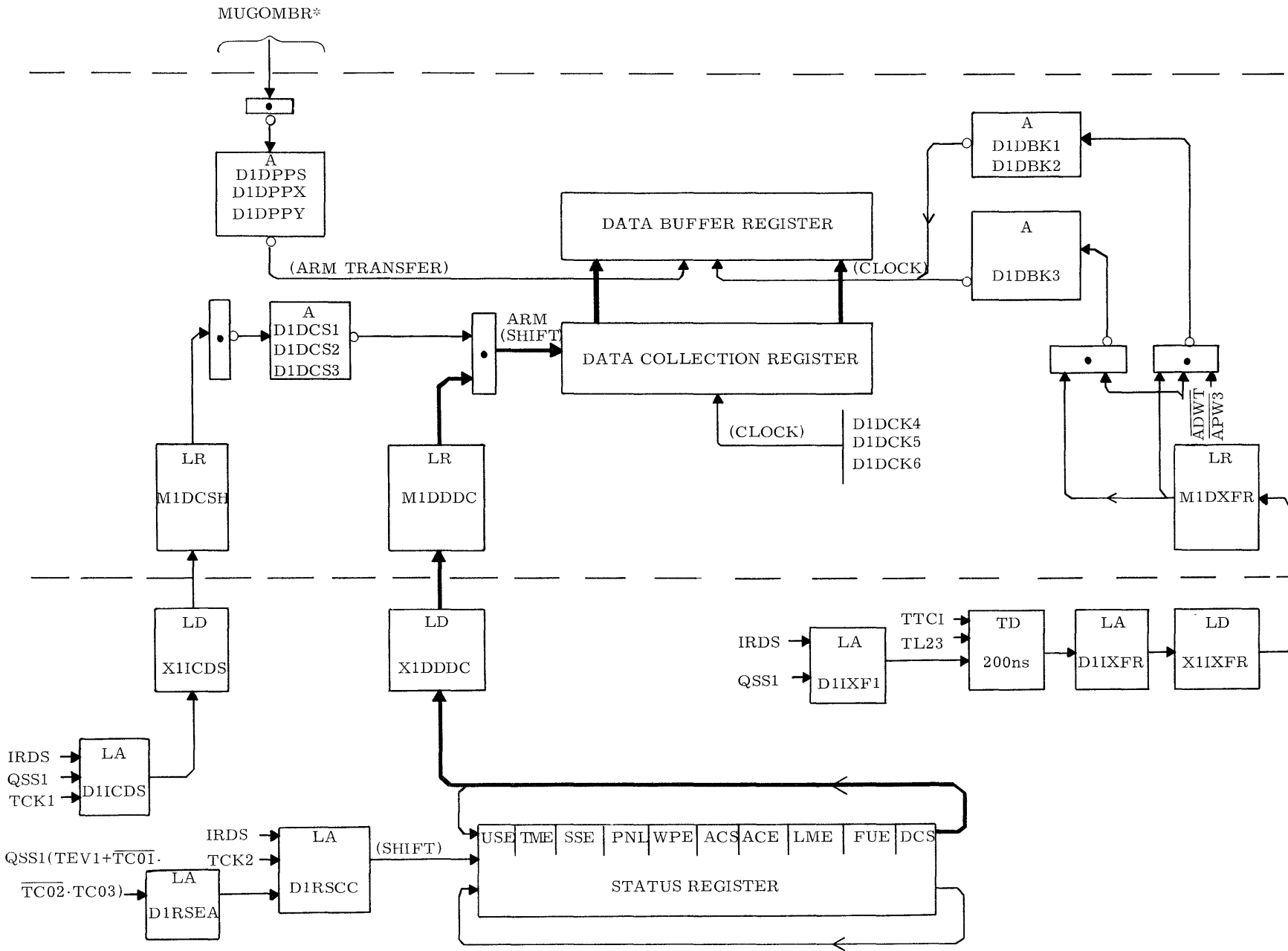


Fig. RDS 1.2. Block Diagram, Read Status-SS1

MEM.

COUPLER

CONTROLLER

# READ STATUS SS2

During sequence state two, the request to transfer the contents of the Data Buffer Register (DBR) to Memory is made. Upon the Memory becoming available, the information is transferred to the least significant ten bits of the pointer word in Memory. With the receipt of the "data ready" signal from Memory, the Controller terminates the Read Status command and the sequence counter returns to sequence state zero. References to the following description should be made to Fig. RDS 2.2 and 2.1 as well as the Coupler and Controller logics. Reference may also be made to the general flow chart of the Read Status command, Fig. RDS 0.1, included at the beginning of the Read Status description.

The request to transfer the contents of the DBR to Memory is originated in the Controller at L06 time with the setting of the memory request flip-flop, F1IMRQ. This request is recognized in the Coupler with the setting of F1GMRQ. The output of GMRQ is sent to Memory to request a Memory cycle. Memory is informed that the request is for a "write-to-Memory" cycle, by the fact that G0ADWL is disabled. The disk write line, G0ADWL, is disabled throughout the decoding of the Read Status command in the Controller.

The information to be stored in Memory had been transferred to the Coupler's DBR as a result of the IXFR

pulse, generated in the Controller at the end of sequence state one. The actual transfer of the information does not take place until the beginning of sequence state two, due to the delay time of the transfer pulse sent to the Coupler.

The Core Address in which this information is stored is specified by the contents of the Coupler's Core Address Register, which was preset to the address of the pointer word during sequence state zero.

Once Memory priority is established, the data and address information is accepted by the Memory and the "memory write" cycle is executed. At the point of the Memory cycle which the actual write begins, a "data ready" pulse is generated, which is sent to the Coupler. This pulse clears F1GMRQ in the Coupler and is also sent to the Controller to set the data ready flip-flop, F1IRDY. The "one" side of IDRY is ANDed with QSS2 to enable G0QRB1, which is inverted and ANDed with the succeeding CK2 to clear the sequence counter and return the Controller to a "not busy" condition. The "zero" side of IDRY is used to clear the read status flip-flop, F1IRDS, formed from logic elements. The QSS0, IRDS, and IOU signals, are ANDed and the output inverted to disable N0IBY1. N0IBY1 is ANDed with the next CK1 to clear the Controller's memory request and data ready flip-flops.

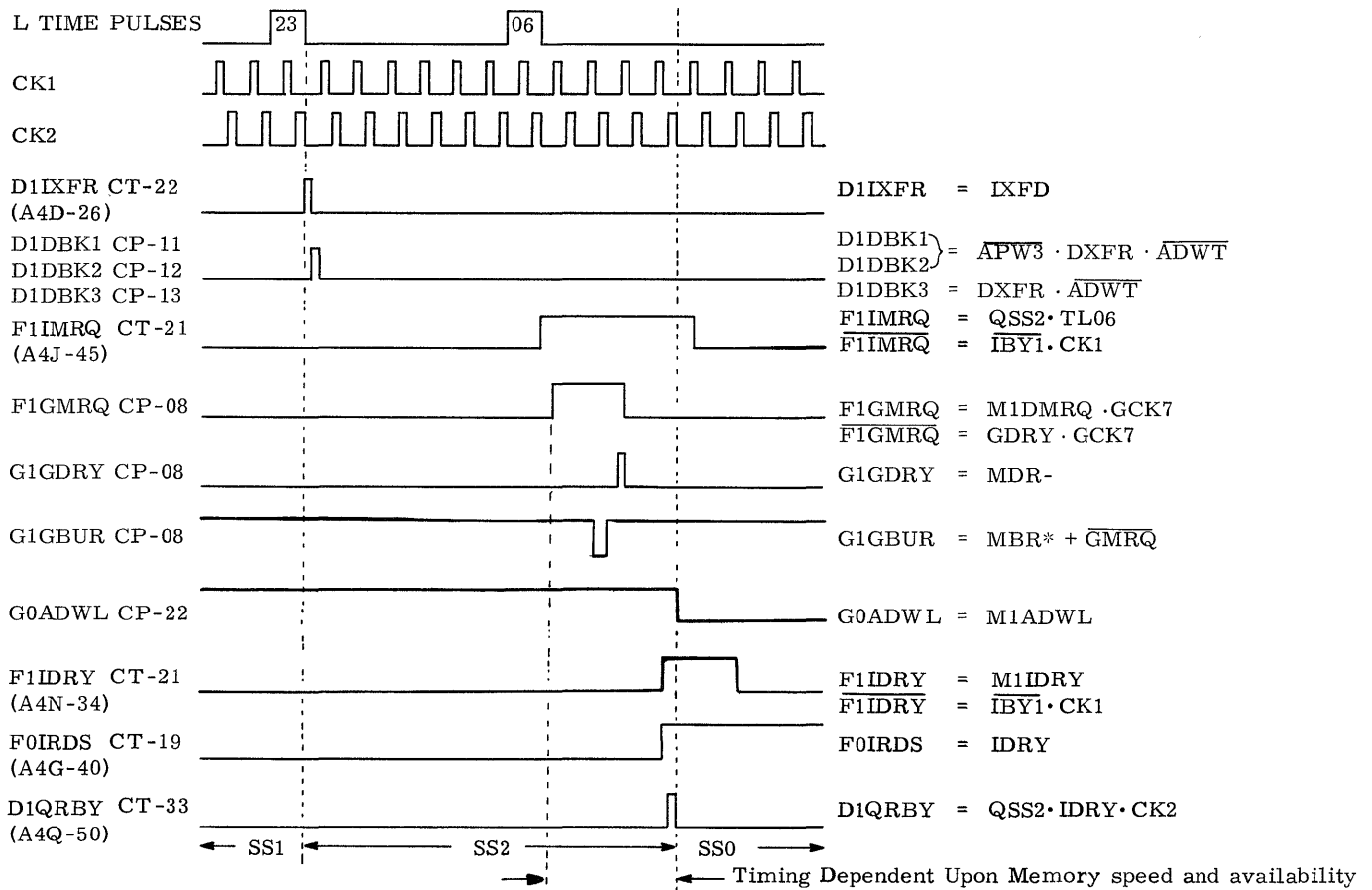


Fig. RDS 2.1. Timing Diagram, Read Status-SS2



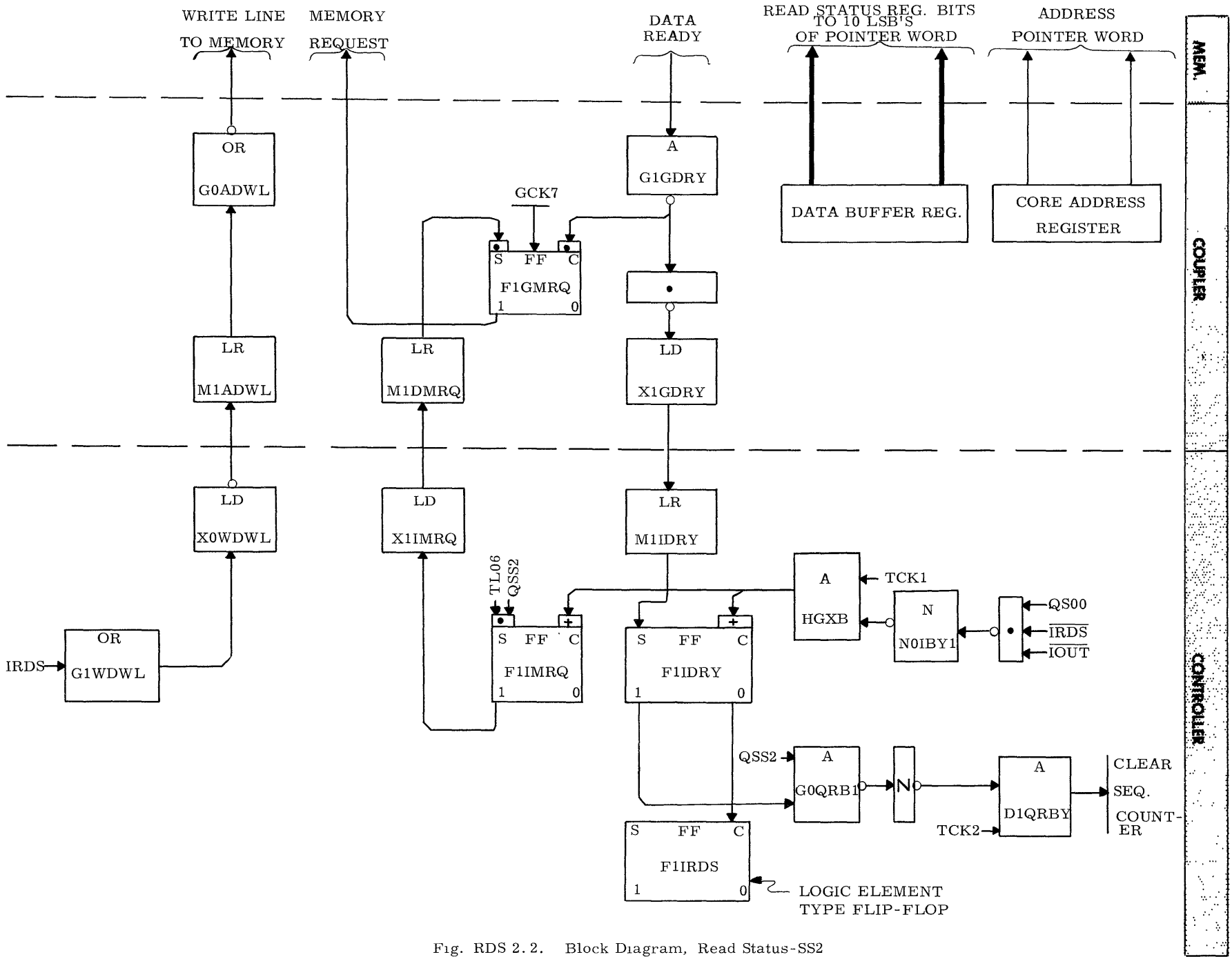


Fig. RDS 2.2. Block Diagram, Read Status-SS2

# NON-SEQUENCING COMMANDS

## RETURN TO ZERO SEEK

The return-to-zero seek command is normally used following a seek-initiating command which results in a seek error. The command serves two functions: to return the heads to a known cylinder, i. e. cylinder 00; and to clear the seek error flip-flop. The major gates involved during the presence of the RTZ command are shown in the block diagram, Fig. NSC.1. The timing of the generation of these signals during this time is shown in NSC.2. The actual seek operation, which will result from this command is shown by the flow chart and timing diagram for this command in the Disk Description Section.

The Return-To-Zero Seek command is decoded in the Coupler by line driver, X0GRTZ. Phase A ( $\phi A$ ) will determine the duration of this pulse, which is dependent upon the speed of the AU to which this device is connected. It can be seen in the block diagram, Fig. NSC.1, that the command is not decoded if the Controller is in a busy state.

With GRTZ decoded, the select unit (USLU), control select (PCTG), and bus line bit-six (LB06) gates are enabled in the Controller. The select unit driver, D1USLU, enables the unit select information, contained in the unit select flip-flops, to be decoded and sent to the Disk. The control select gate, D1PCTG, is

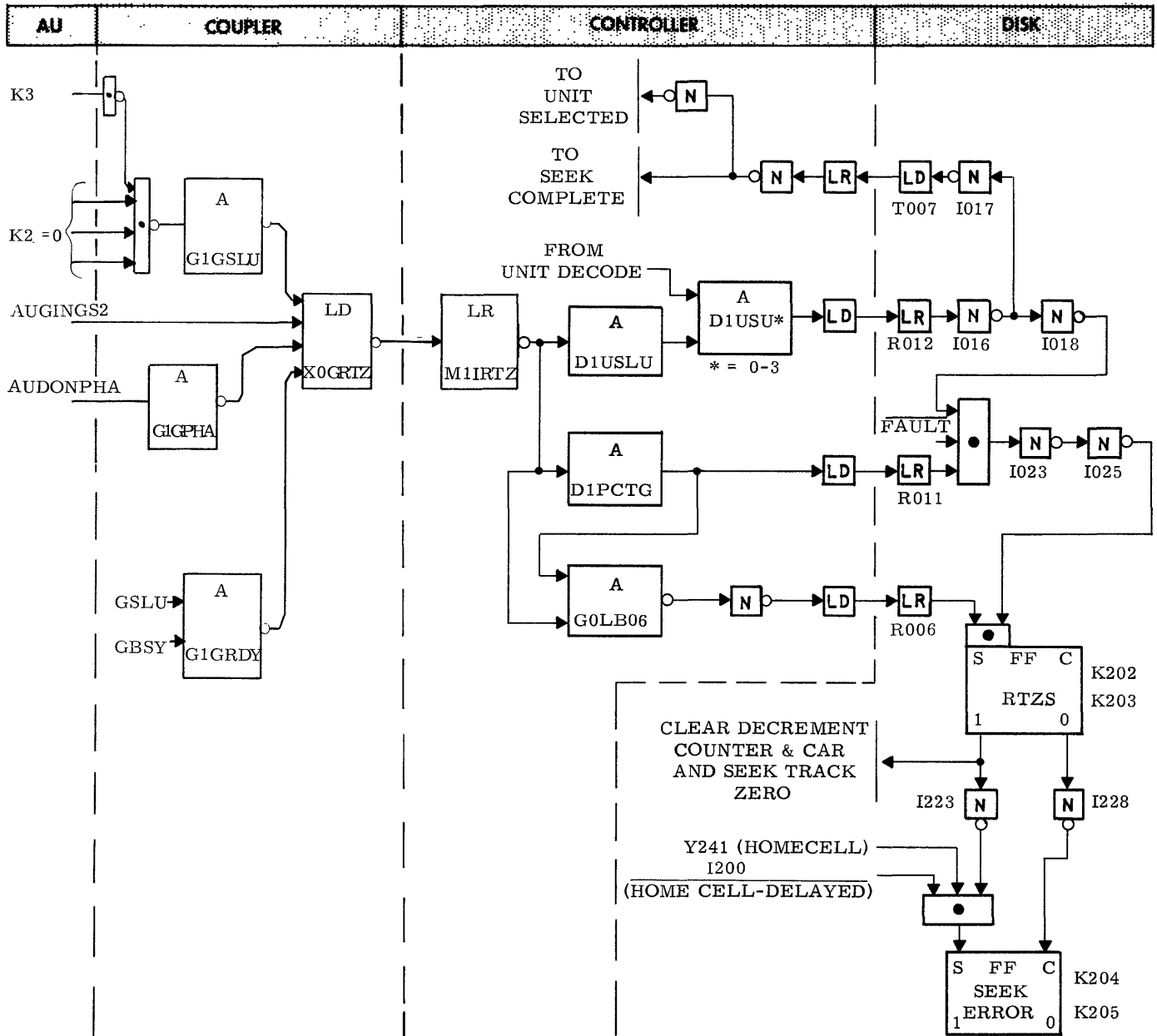


Fig. NSC.1. Block Diagram, Return-To-Zero Seek

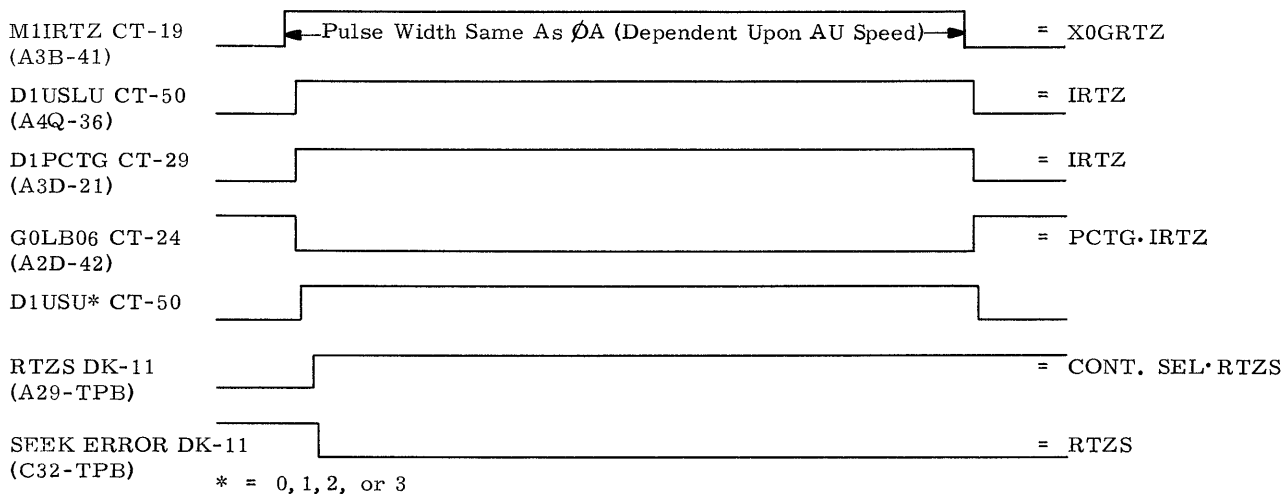


Fig. NSC. 2. Timing Diagram, Return-To-Zero Seek

also sent to the Disk to gate the data from the bus lines to the proper logic. Bus line bit-six is enabled to inform the Disk of a Return-To-Zero operation.

Within the Disk, the unit select signal from the Controller is ANDed with a FAULT condition and the control select signal. The control information is, therefore, gated into the desired Disk Unit. The ANDed signal is inverted twice and coupled with the received bus line bit six, to set the RTZ flip-flop. The setting of the RTZ flip-flop results in a track zero seek being initiated and the decrement counter, Cylinder Address Register and seek error flip-flop being cleared.

The unit select line, which was sent to the Disk, enables a return line which informs the Controller of receipt of the select. By the time the seek complete arrives at the Controller, however, the RTZ command, and consequently the unit select, is disabled. The seek complete is, therefore, ANDed with Unit Selected, to generate the seek complete. The seek complete interrupt is used to inform the program that the heads have settled on the requested cylinder. Since the RTZ command does not result in the Controller busy line going true, the RTZ should be followed by a programmed time delay of at least 155 ms, or should await the seek complete interrupt. This will ensure that a seek-initiating command does not begin while the Disk is still in a seek operation.

## ACTIVATE

The activate interrupt command will cause the busy line in the Coupler to go busy for the period of the phase A ( $\phi A$ ) pulse, providing the Controller is in a not busy condition. If the Controller were busy, the command would be ignored. The toggling of the busy line serves to cause an interrupt to take the program to the Disk operating routine. For a basic block diagram of the Activate command decoding within the Coupler, see Fig. OUT 0.2.

## ABORT COMMAND

The Abort command makes possible the interrupting of a Disk operation without endangering the data recorded on the Disk, since it may sometimes be necessary to interrupt a Disk operation in favor of a more important one.

In this case the program should determine the operation that was being performed, so that it may be repeated upon return to the Disk operation. The following text refers to the sequence of events resulting from the execution of an Abort command. References should be made to the block diagram, Fig. NSC. 3, and the timing diagram, Fig. NSC. 4, as well as the Coupler and Controller logic diagrams.

Upon receipt of the abort command, line receiver X0GABT, in the Coupler, is enabled for the duration of the Phase A ( $\phi A$ ) pulse. X0GABT is then sensed in the Controller by line receiver M11BSY. The Controller recognizes the command by setting the logic element flip-flop, F11ABT. F11ABT records the fact that an Abort command has been executed.

The setting of IABT enables the setting of F11APB in either sequence state six or sequence state seven. If the Controller is in sequence state six, (during which a seek takes place) the operation will terminate immediately, i. e. the Controller will go "not busy" (ready). However, the seek operation will continue in the Disk. For this reason, the program should ensure that a seek-initiating command is not executed within a 155 ms interval following an Abort command.

If the Controller is in sequence state seven at the time of the execution of the Abort command, acknowledgement will be dependent upon the operation being performed. If a read operation was in progress, it will terminate immediately. This is accomplished by the abort pulse flip-flop, F11APB, being set for one clock time. This pulse will enable the reset busy gate, D1QRBY to clear the sequence state counter. With the sequence counter returned to SS0, the busy line, G11BSY, will be disabled, informing the Coupler, and eventually the AU, that the Controller is "not busy" (ready). If a write operation is in progress, the operation can terminate any time from the end of one sector, WAEQ, until the beginning of the write portion of the next sector, TNDP.

If no operation was taking place at the time the Abort was executed, the Controller will not be affected.

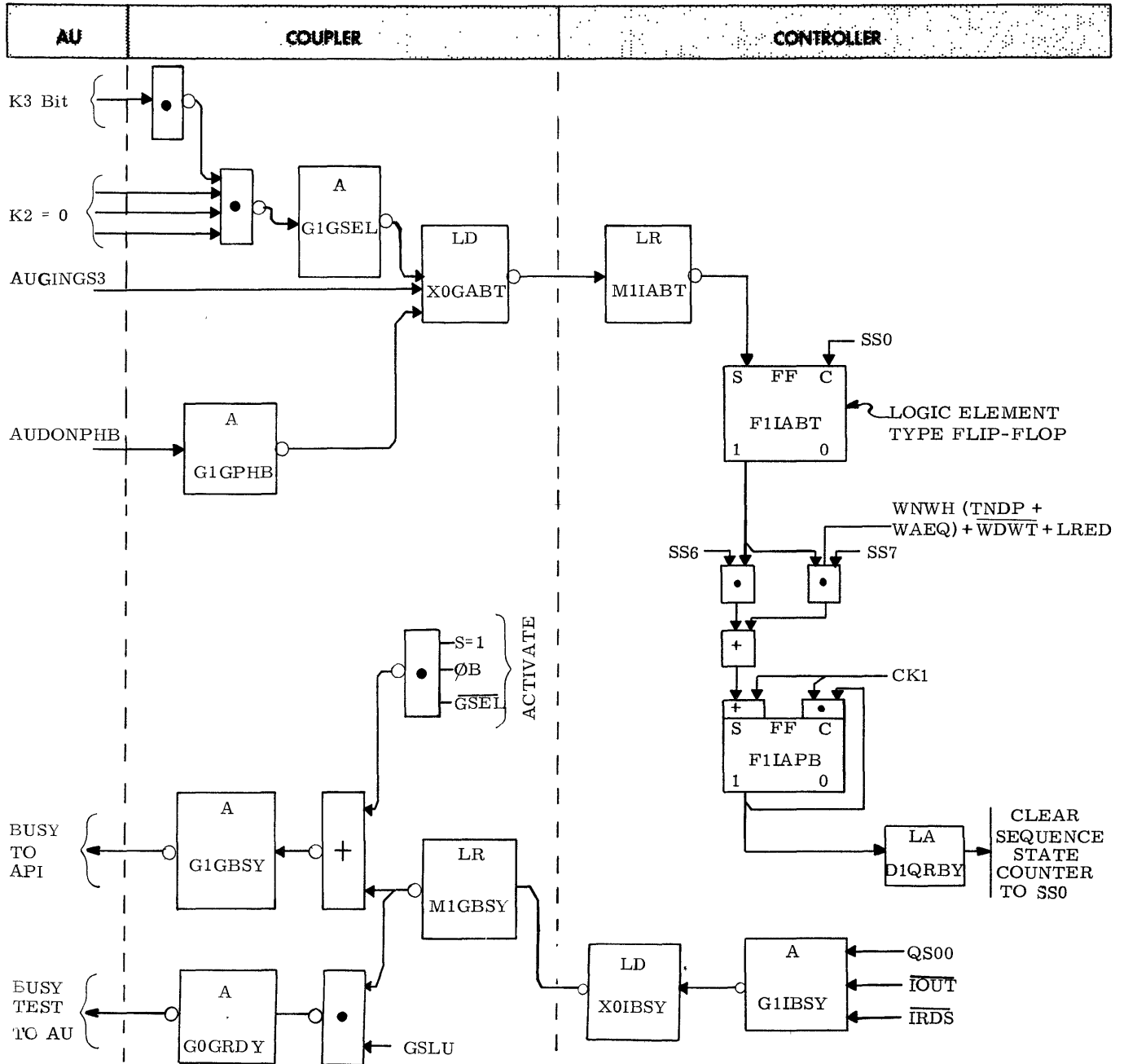


Fig. NSC. 3. Block Diagram, Abort

### JNE COMMAND

The JNE command is executed entirely within the AU. It samples the error line from the Coupler to determine if there is an error within the Disk sub-system. If there is an error, the JNE instruction will result in the contents of the P register, within the AU, being incremented by one. If there is no error, the P register will be incremented by two.

A general description of the conditions causing an error is listed below with the logic equations for the setting of

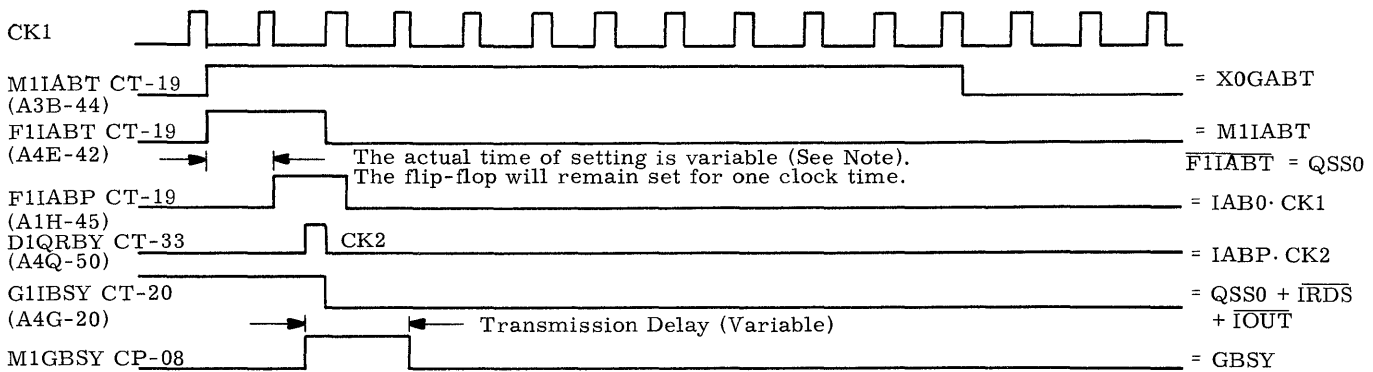
the error flip-flops. The errors are listed corresponding to the bit positions they would occupy in the pointer word after a Read Status instruction.

#### Bit 9 Unit Select Error

$$USE = PT01 \cdot TL23 \cdot (\overline{UUSD} + UUSE)$$

The USE may result from one or both of the following conditions:

- UUSE - More than one unit selected.
- $\overline{UUSD}$  - Unit requested was not selected.



NOTE: The time lapse before F1IABP sets is determined by the type of operation being performed at that time.

1. During a seek (SS6 or read (SS7)), it will become effective immediately. However, the seek operation in the disk will continue until completion.
2. During a write, the setting of F1IABP will await the completion of the current sector write.

The example above is typical of a Read or Seek. A write operation would delay the setting of F1IABP by a variable period of time.

Fig. NSC. 4. Timing Diagram, Abort

### Bit 8 Timing Error

$$TME = \overline{IMRQ} \cdot \overline{IDRY} \cdot QSS7 \cdot TL22$$

A timing error will result if any memory request during sequence state seven is not acknowledged by the end of the following L22.

### Bit 7 Selected Seek Error

$$SSE = YSSE$$

A selected seek error will result if the home cell is detected in the Disk unit (heads are at either extreme) during a direct seek. A direct seek is all seeks other than a return-to-zero, or initial (power up) seek.

The signal which is sent to inform the Controller of the error condition, YSSE, is ANDed with signal RSRE, which prevents the setting of the YSSE flip-flop during a Read Status operation. This is done on several other flip-flops in the register to prevent a flip-flop setting during the register shift, which could inject false error conditions.

### Bit 6 Pack Not On-Line

$$PNL = RPNL$$

This error indicates that the Disk unit is unavailable for one or more of the following reasons:

- The selected Disk's heads are not loaded and the motor is not up to speed.
- The Disk Pack is not on the unit.
- The Disk unit is in the off-line condition.

These errors are detected as a result of the conditions of two switches within the Disk unit. A simplified block diagram of the error detecting circuitry is shown in Fig. NSC. 5.

### Bit 5 Write Protect Error

$$WPE = WDWT \cdot RERE \cdot PT02 \cdot TL22$$

A write protect error may be detected in either sequence state four or sequence state six. It is generated as a result of a Disk write being requested for a Disk that is write protected by a switch on the maintenance console. It is also inhibited from being set during a Read Status instruction.

### Bit 4 Address Check-Sum Error

$$ACS = WWHD \cdot IHDR \cdot \overline{ZCSC} \cdot TL23$$

The address check-sum error can only be detected during a Write Header operation. The error detects a non-comparison of the count generated by the Controller Check-Sum register with that sent following the address bits. If the count does not compare with the count sent from Memory, the check-sum compare flip-flop, ZCSC, will be cleared. This will be detected by the logic and result in the setting of the ACS flip-flop.

### Bit 3 Address Compare Error

$$ACE = WACE$$

This error is generated within the Disk if the heads search an entire track twice without having a comparison of desired address with addresses read from the Disk.

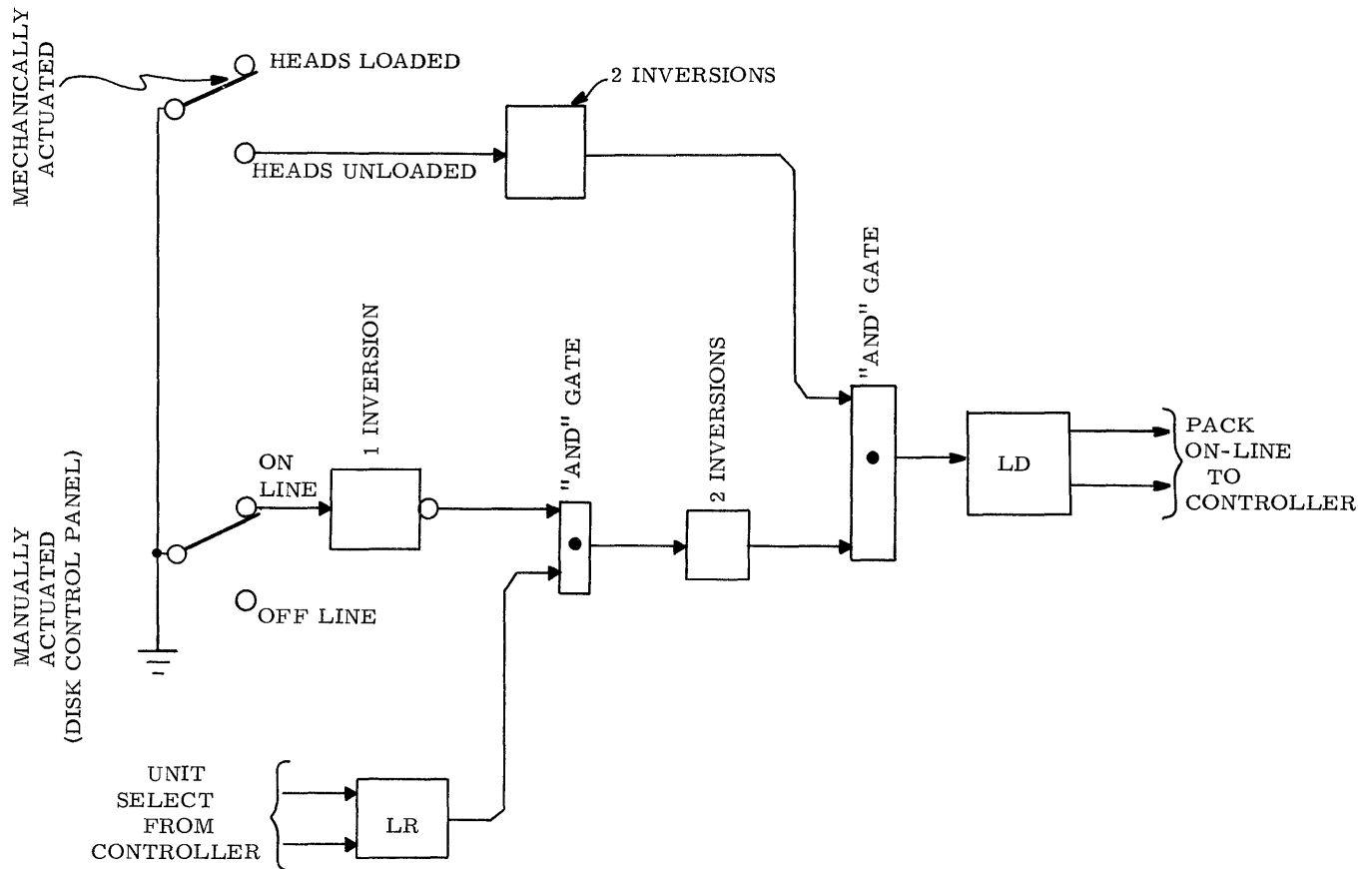


Fig. NSC.5. PNL Error Detection

### Bit 2 Limit Error

$$\text{LME} = \text{RLME}$$

The limit error is detected whenever the control word specifying the cylinder address contains a count of  $200_{10}$  or greater for a 200 track Disk Unit or  $100_{10}$  or greater for a 100 track Disk Unit.

### Bit 1 File Unsafe Error

$$\text{FUE} = \text{RSRE} \cdot \text{YFUS} + \text{RIHA}$$

The file unsafe error indicates either an illegal head address has been decoded by the Controller (RIHA) or a fault within the Disk unit itself (YFUS). A fault within the Disk unit may result from one or more of the following conditions:

- a. More than one head is selected.
- b. Select read and write exist at the same time.
- c. Select read and erase exist at the same time.
- d. Erase with no write driver selected.
- e. Erase with both drivers on.
- f. One or both write drivers on with no erase.
- g. Select read, write, or erase and not on cylinder.
- h. Bus level monitor signal from Controller.
- i. Voltage monitor circuit within Disk unit.

Since the detection of the first fault condition listed, item a, is relatively simple, only the general manner in which it is detected will be described. The detection is accomplished by monitoring the current flow through the head select transistors, whose collector circuits are common. If no more than one head is selected, the resultant current through the base circuit of Q02 (refer to the CDC circuit description of the AND type circuit, found in the circuit description manual) of the AND type circuit results in Q02 turning-on. As a result of Q02 turning on, a fault will be detected by the AND circuit feeding the input to the Fault flip-flop.

The detection of items b and c are accomplished by ANDing them, at the set input to the Fault flip-flop. The detection of items d through f are relatively more complex. A description of the manner of detection is, therefore, given in the following text. As an aid to the description, an equivalent circuit diagram, Fig. NSC.6, is given, with relative components shown. The first example to be discussed is item d; the detection of an erase driver turned-on without a write driver.

With both write drivers turned-off, point A of the diagram will be at a +40V potential. Prior to the turning-on of the erase driver, point B would also be at a +40V. The emitter circuit of Q05, at point D, will therefore be at a +40V. The base of Q05 will establish, approximately, a 39.03V potential, through the resistor and diode drop

in the base circuit. It is necessary for the base of Q05 to be approximately 1.77V negative, with respect to the emitter, in order to turn on Q05. (By turning on Q05, a fault condition would be indicated.) In this example, where the erase driver turns-on, point B will begin to drop in a negative direction, due to the current path through the erase head and head select transistor. (It is assumed that the head select is enabled; otherwise no fault condition would be detected.) As point B drops in a negative direction the diode between points B and F becomes forward biased.

As the collector potential of the erase driver continues to drop, the voltage at point F will follow. When the level drops sufficiently far, so that the level at the base of Q05 is approximately 38.23V, Q05 will turn-on, indicating a fault.

Once Q05 is turned-on, the collector voltage of the write drivers is provided with a current path through Q05. The emitter potential of Q05 will, therefore, try to follow that of the base but due to the base-emitter drop will remain sufficiently high to continue conduction.

The second example, item e, is that of the erase and both write drivers being turned-on. With both write drivers turned on, point A will drop at a greater rate than will point B, whose drop will be due to the erase

driver being turned-on. Due to the arrangement of the diodes, the higher potential, point B, will be seen at point D, while the lower potential, point A, will be seen at point F. This voltage differential will be sufficient to turn-on Q05. Once Q05 is turned-on, the fault condition will be detected by the setting of the Fault flip-flop.

The third example, item f, is that of one, or both, write drivers turned-on with no erase. The condition of one write driver being on with no erase is similar to that of item d, erase with no write driver selected. The circuit will react in the same manner, with different diodes establishing the bias references. In this example, the diode between points B and D will hold point D at a +40V until the level at point A, which forward-biases the diode between points A and F, is sufficiently low to turn-on Q05.

The condition of both write drivers being on with no erase is nearly the same, except for the fact that both write drivers being turned-on will establish a lower voltage level at point A than would the example with one write driver turned-on. The fact that the erase driver is not turned-on, will establish a +40V potential at point B and consequently at point D. As the level at point A drops, the level at point F and subsequently point E will follow.

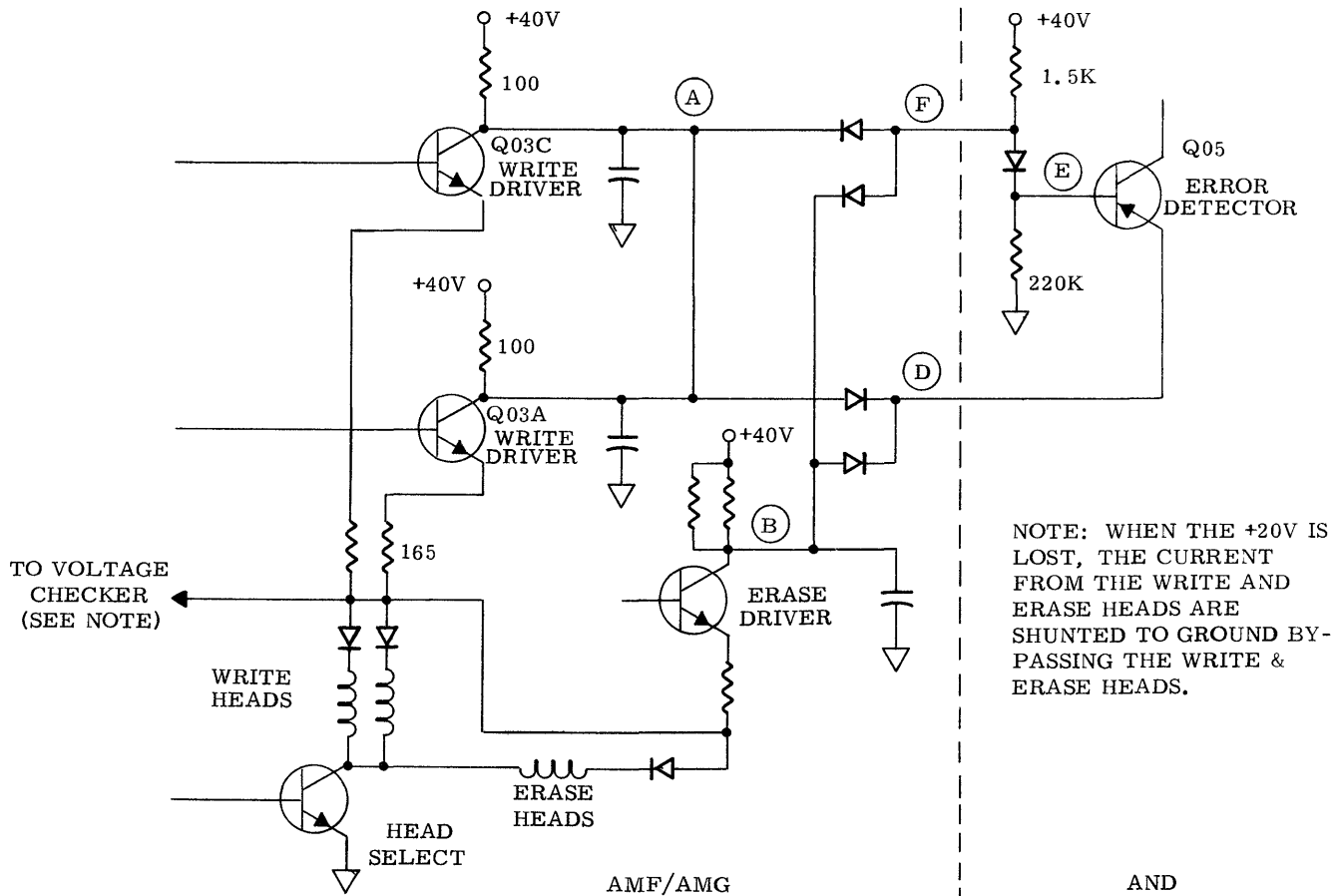


Fig. NSC.6. Write/Erase-Select Error Circuit

When the level at point E is approximately 38.23V, Q05 will turn-on, indicating a fault.

The desired condition of one write driver and one erase driver being turned-on, will not generate an error indication. The write drivers will switch alternately, in a push-pull arrangement. Since their collector circuits are common, the level at point A will remain close to that of a single write driver being turned-on. The switching time will have little effect, due to the relatively long time constant of the collector circuit. The erase driver, which turns-on simultaneously with one of the write drivers, will cause point B to drop at a rate approximately equal to that of point A, which goes negative as a result of a write driver turning-on. The levels of points A and B, and consequently points F and D, drop at approximately the same rate, which keeps Q05 turned-off.

The capacitors in the collector circuit of the write drivers will prevent the generating of a fault indication when the erase driver is turned-off prior to the write drivers, which occurs during a write operation, to prevent erasing of an area which will not be written upon. The relatively long time constant in the collector circuit will prevent the level at point A from rising too fast. If the write drivers are turned-off soon enough, as during a write operation, the level will not change sufficiently to indicate a fault.

The remaining fault conditions, g - i, are detected by logic. Item g is detected at the set input of the fault flip-flop by the ANDing of inverters I220 and I448.

The modification drawing, number 68A975864, should be referenced to show the detection of the remaining two fault conditions. The Bus Level Monitor (BLM) fault is generated if the voltage for the Controller logic should vary from prescribed limits. The BLM error detection is described in detail under the BUS LEVEL MONITOR, THEORY AND MAINTENANCE description, publication number 4791A-2. The output of the BLM is gated to the line driver, X0LPMP, in the Controller and the line driver indicates to the Disk unit the status of the BLM.

The remaining fault condition is detected if the power supply used for the transmission lines between the Controller and Disk should vary to the point where the +20V

drops below +17.6VDC or if the -20V gets closer to ground than -17.2VDC. The condition is detected by circuit Y408 located on page 15 in the Disk logic. The PWB is located at location X02 in the Disk unit.

Any of the error conditions mentioned in items a through g will cause the Fault light on the Disk unit to come on. Depressing the light/button would clear the fault and turn-off the light until a reoccurring fault condition.

### Bit 0 Data Check-Sum Error

$DCS = \overline{TDXF} \cdot \overline{TNDP} \cdot THDR \cdot WNW\overline{H} \cdot \overline{WDWT} \cdot TL23 \cdot \overline{ZCSC}$

During a write operation, the bits are sampled as they are written on the disk. For each "one" bit of data, the Check-Sum Generate Register is incremented once. After all of the data is recorded on the disk, the count contained in this register is written on the disk. Therefore, a verification of the data received during a read operation is made possible. If the data check-sum, generated by the "one" bits being read from the disk, does not compare with the data check-sum read from the disk following the data, a data check-sum error is detected.

This is accomplished by ring shifting the contents of the Check-Sum Generate Register, as the data check-sum is being read from the disk, and comparing the contents of the register with the check-sum read from the disk. The check-sum compare flip-flop is preset prior to the shift and will remain set as long as the two check-sums compare. If they do not compare, the flip-flop is cleared (reset) and upon completion of the shift the flip-flop is sampled at the input to the DCS error flip-flop in the Status Register. If the flip-flop is reset, an error will be indicated by the setting of the RD<sub>CS</sub> flip-flop.

### JNR COMMAND

The JNR command tests the condition of the busy line, from the Disk sub-system, within the AU. If the line indicates that the system is not busy (ready) the JNR command will cause the P counter, in the AU, to increment by one. If the line indicates that the sub-system is busy (not ready), the P counter will be incremented by two.

The busy line will be true any time the Controller has received an OUT, or RDS, command or is in another sequence state other than zero.



# MAINTENANCE CONSOLE DESCRIPTION

The Disk Sub-System maintenance console serves as an indicator for the status of various registers and flip-flops located in the Coupler and Controller. The console is mounted above row zero, on the page which contains the Controller logic. The signals are gated into the console through a diode matrix arrangement, which selects the inputs according to the enable signals provided by a five position rotary switch, located on the console. The selected information is displayed on the 24 indicators (23 - 0), located on the console. The indicators and switches provided by the console may be seen in Fig. MCS.1, and a brief description of these is included in the following text.

## SELECTOR SWITCH (CSW10)

### Position 1

- Bits 23 - 0

In this position the console indicators display the contents of the Data Buffer Register (DBR), which is located in the Disk Coupler, and through which control words and data pass enroute to/from Memory. Since, during an active Disk operation, the information is being transferred at a high rate of speed, the contents of the register can only be interpreted when the sub-system is in a passive state.

### Position 2

- Bits 23 - 16

These indicators do not have inputs connected to them and are therefore not illuminated in this position. The unused indicators in this position, as well as those in the remaining three switch positions may be utilized to display various outputs from the Coupler and Controller logic. This may be done using a temporary or permanent connection, depending upon requirements, between the logic output and the desired tie-point on the diode matrix board. The inputs should be restricted to the logic levels (+5V to light and 0V to turn off). The outputs from the logic should be within 100 ft. of the console and connected through a 20K ohm resistor to the tie-point of the diode matrix board. The tie-point may be determined by referring to the table provided on sheet 65 of the Controller logic, 68C972186. It should be remembered when selecting a point to display, that the positive logic level will light the indicator while the ground level will not.

- Bits 15 - 0

These bits display the contents of the Core Address Register, contained in the Disk Coupler. This information, as did the contents of the DBR, will vary during an active Disk operation (one which requires sequencing of the Controller) and can only be interpreted when the sub-system is in a passive state.

### Position 3

- Bit 23

Indicates the type of operation (read/write) being performed (while active) or was last performed (while passive).

- Bits 22 - 18

Not used. See Bits 23-16 of Position 2.

- Bits 17 - 16

Indicates the Disk Unit selected (while active) or the last Disk Unit accessed (while passive).

- Bits 15 - 8

Indicates the contents of the Desired Cylinder Register, located in the Controller. If an active Disk operation is being performed on more than one cylinder, the contents of this register will be varying and cannot be interpreted. If the operation remains on a specific cylinder or if there is no active operation, the contents of the register will indicate the present cylinder over which the heads are positioned.

- Bits 7 - 4

Indicates the contents of the Desired Head Register, located in the Controller. Unless an operation is repeated within a track of information, which retains the same head address, the contents of this register will vary during the operation and cannot be interpreted. If the operation is repeated on a track or if there is no active operation, the indication will be that of the head address being selected or the address of the last head used, respectively.

- Bits 3 - 0

Indicates the contents of the Desired Sector Register, located within the Controller. Unless a one-block (64 24 bit words, found on a sector) transfer is being repeated, the contents of this register will vary and cannot be interpreted. If the operation is being repeated on a sector or there is no active operation, the indication will be that of the sector involved in the transfer or the last sector address involved in a transfer, respectively.

### Position 4

- Bit 23

When lit, an indication is given that the Controller is presently being timed (L pulses) by the information being received from the disk. When not lit, the Controller is being timed by the 1.25 MHz<sub>Z</sub> oscillator, located in the Controller.

- Bits 22 - 21

Not used. See Bits 23-16 of Position 2.

- Bits 20 - 18

Indicates the binary representation of the sequence state counter. Since the sequence states which remain true the longest, during a data transfer command are SS6 and SS7, the predominate display during the operation will be an octal seven, with bits 20 and 19 being slightly brighter. Upon conclusion of the operation, the counter will return to SS0, during which time none of these indicators will be lit.

- Bits 17 - 10

Not used. See Bits 23-16 of Position 2.

- Bits 9 - 0

Indicates the contents of the Read Status Register, located within the Controller. The error to which the bit pertains is defined just below the indicator. Upon detection of an error, the respective flip-flop in the Read Status Register will set, lighting the corresponding indicator on the console (providing the rotary switch in in position 4). Errors may be cleared by execution of a Gen II Out command.

### Position 5

- Bits 23 - 14

Not used. See Bits 23-16 of Position 2.

- Bits 13 - 0

These bits indicate the contents of the Block/Word Count Register, located in the Controller. The contents cannot be interpreted during an active Disk operation, when data is to be transferred. Upon completion of the operation, the contents of this register should contain all ones and consequently all indicators in bits 13-0 of this position will be lit.

### NORMAL/HEADER SWITCH (CSW05)

This switch differentiates between the two types of Write operations which can be performed as a result of an Out command, with a bit 23 in control word two, being executed. The two types, as indicated by the switch, are: "Normal" and "Write Header". The switch should be left in the "Normal" position for all

operations other than a Write Header operation, used to restore address and synchronizing information on the disk. When the switch is placed in the "Header" position, the control logic will respond in a different manner than if the switch was in the "Normal" position. For this reason, care should be exercised, and the switch should be returned to the "Normal" position following a Write Header operation.

### MEMORY WRITE PROTECT SWITCHES

The following switches will protect the areas mentioned:

- CSW01 (Unit 0) - All addresses of Unit 0.
- CSW02 (Unit 1) - All addresses of Unit 1.
- CSW03 (Unit 2) - All addresses of Unit 2.
- CSW04 (Unit 3) - All addresses of Unit 3.
- CSW11 (Unit 0, 00 - 37<sub>8</sub>) - All cylinders from 00<sub>8</sub> through 37<sub>8</sub> of Unit 0.
- CSW15 (Unit 0, 00) - All addresses on cylinder 00 of Unit 0.
- CSW14 (Unit 0, 01) - All addresses on cylinder 01 of Unit 0.
- CSW13 (Unit 0, 02) - All addresses on cylinder 02 of Unit 0.
- CSW12 (Unit 0, 03) - All addresses on cylinder 03 of Unit 0.
- CSW19 (Unit 0, 04) - All addresses on cylinder 04 of Unit 0.
- CSW18 (Unit 0, 05) - All addresses on cylinder 05 of Unit 0.
- CSW17 (Unit 0, 06) - All addresses on cylinder 06 of Unit 0.
- CSW16 (Unit 0, 07) - All addresses on cylinder 07 of Unit 0.

Any of the above two-positioned switches placed in the "On" position will prevent that address, to which the switch pertains, from being written upon. Placing a switch in the "On" position will enable an AND gate at the time when that address or Unit to which that switch pertains is selected. This will enable driver, D1REPE, which will set the write protect error flip-flop F1RWPE, at the beginning of sequence state four, if the active command is determined to be a Write operation.

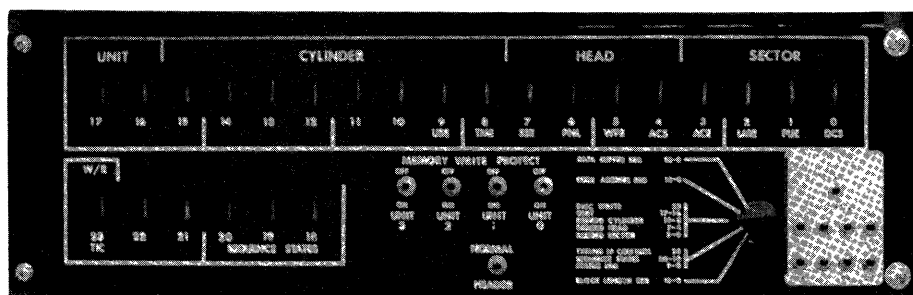


Fig. MCS.1. Maintenance Console

# TABLE OF CONTENTS

<b>INPUT/OUTPUT BUFFER</b>	4202A/B	PARITY CHECK AND GENERATE
<b>INPUT/OUTPUT BUFFER</b>	I/O	INPUT DEMAND
COMMAND DEFINITIONS		AUTOMATIC PROGRAM INTERRUPTS
IN - Input From Device		PACKAGING AND OPTIONS
OUT - Output To Device		SPECIAL CHARACTER FUNCTIONS
OPR - Operate Channel		Start of Text (STX) 002
JCB - Jump If Channel Busy (K1 = 01)		End of Text (ETX) 003
JDR - Jump If Data Ready (K1 = 10)		Delete Code (DEL) 117
JNE - Jump If No Error		Carriage Return (CR) 015
ABT - Abort Channel		Device Control - 1 (DC1) 021
INPUT FUNCTION		Device Control - 2 (DC2) 022
OUTPUT FUNCTION		Device Control - 3 (DC3) 023
		Device Control - 4 (DC4) 024

## LIST OF ILLUSTRATIONS AND TABLES

Fig. I/O. 1	Simplified Block Diagram	Fig. I/O. 7	Card Read Flow Chart
Fig. I/O. 2	ASCII Characters and Codes	Fig. I/O. 8	Paper Tape Read Flow Chart
Fig. I/O. 2a	ASCII Characters and Codes	Fig. I/O. 9	Simplified Diagram of the OUT Function
Fig. I/O. 3	Typer Common Peripheral	Fig. I/O. 10	Paper Tape Punch Flow Chart
Fig. I/O. 4	EIA RS-244 Characters and Codes	Fig. I/O. 11	Printers Flow Chart
Fig. I/O. 5	GEN II Command Format	Table I/O. 1	Peripheral Record Control Actions
Fig. I/O. 6	Simplified Diagram of the IN Function		

# INPUT/OUTPUT BUFFER

The 4202 I/O Buffer provides on-line control and synchronization of data transfers between the Central Processor and the system peripheral device channels. The basic buffer, as shown by the block diagram, Fig. I/O. 1, includes GEN II instruction decode logic, address drive and test line neckdown logic, parity check and generate logic, and 14 in/out data and parity transfer gates. The entire logic is implemented with integrated circuit elements which are mounted on two PBCA1 printed wire boards.

Each I/O Buffer is capable of controlling 16 channels, of which a maximum of 7 can be used for input. The channels are selectively coded from 00 through 17 octal. For the standard I/O Buffer, input channels will always start at 00<sub>8</sub> and output channels will always start at 07<sub>8</sub>. If a tenth output channel is required it will occupy the last input channel address (06<sub>8</sub>). Peripherals such as the ASR Teletypewriter Set and the IBM I/O Typer will require both an input channel and an output channel. For example, the first I/O device which will normally be the Typer will use input channel 00<sub>8</sub> and output channel 10<sub>8</sub>.

Since the I/O Buffer utilizes the ASCII code, all devices other than the Teletypewriter Sets will either operate in a free-running mode with blocked format or will be programmed in a record or blocked format. In addition, devices such as the IBM Selectric Typer which print a character, will require program translation prior to outputting to the device. Reading or writing a "record" at a time, makes more efficient use of the interrupt synchronized I/O instructions, TIM/TOM, available on the 4020, and IDL/ODL, available on the 4050/4060.

All peripherals operating with the 4202 I/O Buffer nominally use the ASCII/ISO character-code set summarized by Figs. I/O. 2 and I/O. 2a. Exceptions are:

- IBM Selectric typers which use the Common Peripheral code set summarized by Fig. I/O. 3.
- Card readers and punches which operate with the 12-bit column image code (Hollerith).
- Numerical Machine Tool control devices which operate with the EIA RS-244 code summarized by Fig. I/O. 4.

The basic I/O Buffer logic, referenced to GE Drawing No. 68C972190, and the GEN II command functions will be discussed in this section. The optional channel control and drive logic will be described separately at the end of this section. Mechanical and electrical theory, maintenance and parts information for the various optional peripheral devices will be discussed in subsequent sections of this manual or under separate cover.

## COMMAND DEFINITIONS

The various operations of the I/O Buffer are controlled by six basic commands: OPR, IN, OUT, ABT, and variations of JNR and JNE. These commands are all of the GEN II format, as shown by Fig. I/O. 5. Specific action is determined by the micro-coded "S" and "K" bits. The S bits are octally decoded into the various

command signals before entering the Buffer. One bit each from the K3 and K2 positions is connected to the Buffer Instruction Decode logic as a select code; the basic I/O Buffer of a typical system is assigned address 71<sub>8</sub>. For the 4040, 4050, 4060, the address is 11<sub>8</sub>. The K1 = 4<sub>8</sub> and K1 = 2<sub>8</sub> bits specify the test lines. The K0 bits plus the K1 = 1 bit are used to specify a particular channel or device address.

The basic function and decoding of each command is described in the following text. The particular application of the decoded commands are described by the separate Device Control write-ups at the end of this section. Refer also to the GEN II Command Description in the AU portion of Volume I.

### IN - Input From Device

The IN command transfers data from the addressed input channel to the A Register of the AU, in response to a "data-ready" interrupt. The data is right justified and unused bits of A are set to zero. In the I/O Buffer, IN disables the AU input gates (G1DOXX) which in turn enable the channel neckdown gates (G1DIXX) to drive the corresponding channel to AU clock drivers (D0DBXX), providing the K2, K3 bits are present. Also, D1GGIN is decoded to cycle the addressed device control logic.

### OUT - Output To Device

The OUT command transfers data from the A Register to the addressed channel in response to a "buffer-ready" interrupt. In the I/O Buffer,  $\overline{OUT}$  is or'ed with  $\overline{OPR}$  to disable the channel neckdown gates (G1DIXX) which in turn enable the AU input gates (G1DOXX) to drive the corresponding outputting clock drivers (D0DBXX), provided the K2, K3 bits are present. Also, D0GOUT is decoded to cycle the addressed device control logic.

### OPR - Operate Channel

The OPR command sets the addressed channel busy (not ready) and transfers the contents of the A Register to the channel. This information (least 7-bits) is decoded within the device control logic and used as control signals or in some devices it may be interpreted as data. The A Register is unchanged. The channel error flip-flop is cleared. Within the I/O Buffer,  $\overline{OPR}$  enables the outputting clock drivers (D0DBXX) in the same manner as  $\overline{OUT}$ . Also, D0GOPR is decoded and sent to the addressed device control logic.

### JCB - Jump If Channel Busy (K1 = 01)

If the addressed channel is busy, program control is transferred to the second sequential location. If the channel is not busy, control is transferred to the first sequential location. The command (JNR) is decoded within the AU which simply monitors the channel ready test point. Within the I/O Buffer,  $\overline{K1} = 2_8$  enables D1AK12 which enables the device control logic to output the channel-busy API status. The ready status of the addressed channel is presented to the AU, via the ready test neckdown logic. Refer to the device control logic for specific channel ready conditions. Normally, a channel is busy when one of the following conditions exists:

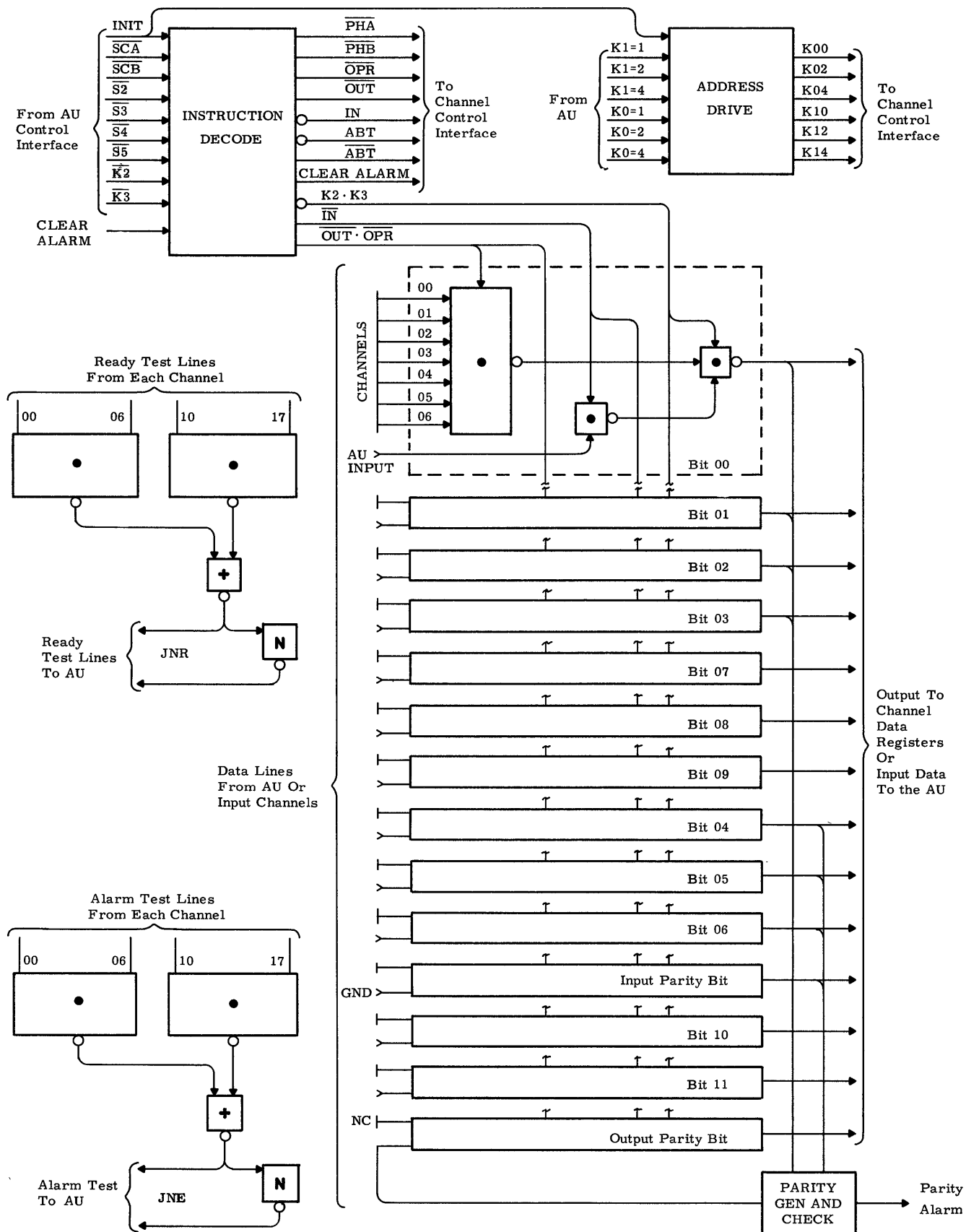


Fig. I/O. 1. Simplified Block Diagram

OCTAL CODE = C<sub>2</sub> C<sub>1</sub> C<sub>0</sub>

C <sub>2</sub> C <sub>1</sub> \ C <sub>0</sub>	0	1	2	3	4	5	6	7
00	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL
01	BS	HT	LF	VT	FF	CR	Ø	SI
02	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB
03	CAN	EM	SUB	ESC	FS	GS	RS	US
04	SP	!	"	#	\$	%	&	'
05	(	)	*	+	,	-	.	/
06	0	1	2	3	4	5	6	7
07	8	9	:	;	<	=	>	?
10	@	A	B	C	D	E	F	G
11	H	I	J	K	L	M	N	O
12	P	Q	R	S	T	U	V	W
13	X	Y	Z	[	\	]	^	_
14	`	a	b	c	d	e	f	g
15	h	i	j	k	l	m	n	o
16	p	q	r	s	t	u	v	w
17	x	y	z	{		}	~	DEL

LEGEND

Control Characters

NUL	Null	DC3	Device Control 3
SOH	Start of Heading (CC)	DC4	Device Control (stop)
STX	Start of Text (CC)	NAK	Negative Acknowledge (CC)
ETX	End of Text (CC)	SYN	Synchronous Idle (CC)
EOT	End of Transmission (CC)	ETB	End of Transmission Block (CC)
ENQ	Enquiry (CC)	CAN	Cancel
ACK	Acknowledge (CC)	EM	End of Medium
BEL	Bell (audible or attention signal)	SUB	Substitute Character
BS	Backspace (FE)	ESC	Escape
HT	Horizontal Tabulation (punched card skip) (FE)	FS	File Separator (IS)
LF	Line Feed (FE)	GS	Group Separator (IS)
VT	Vertical Tabulation (FE)	RS	Record Separator (IS)
FF	Form Feed (FE)	US	Unit Separator (IS)
CR	Carriage Return (FE)	DEL	Delete *
SO	Shift Out		
SI	Shift In		
DLE	Data Link Escape (CC)		
DC1	Device Control 1		
DC2	Device Control 2		

Graphic Characters

Symbol	Name	Symbol	Name
SP	Space (normally nonprinting)	/	Slant
!	Exclamation Point	:	Colon
"	Quotation Marks (diaeresis*)	;	Semicolon
#	Number Sign	<	Less Than
\$	Dollar Sign	=	Equals
%	Percent	>	Greater Than
&	Ampersand	?	Question Mark
'	Apostrophe (closing single quotation mark; acute accent*)	`	Grave Accent * (opening single quotation mark)
(	Opening Parenthesis	[	Opening Bracket
)	Closing Parenthesis	]	Closing Bracket
*	Asterisk	^	Circumflex *
+	Plus	_	Underline
,	Comma (cedilla*)	@	Commercial at
-	Hyphen (minus)	{	Opening Brace
.	Period (decimal point)	}	Closing Brace
			Vertical Line
		\	Reverse Slant

(CC) Communication Control. (FE) Format Effector.

(IS) Information Separator.

\* In the strict sense, DEL is not a control character.

Revised May 19, 1966.

Fig. I/O.2. ASCII Characters and Codes

TELETYPE 33 AND 35									ANELEX LINE PRINTER							
OCTAL CODE = C <sub>2</sub> C <sub>1</sub> C <sub>0</sub>									OCTAL CODE = C <sub>2</sub> C <sub>1</sub> C <sub>0</sub>							
C <sub>2</sub> C <sub>1</sub>	C <sub>0</sub>								C <sub>0</sub>							
	0	1	2	3	4	5	6	7	VFU <sub>0</sub>	VFU <sub>1</sub>	VFU <sub>2</sub>	VFU <sub>3</sub>	VFU <sub>4</sub>	VFU <sub>5</sub>	VFU <sub>6</sub>	VFU <sub>7</sub>
00		SOH	STX	ETX		ENQ										
01		HT	LF			CR	RED	BLK								
02		DC1	DC2	DC3	DC4								DC4			
03																
04	SP	!	"	#	\$	%	&	'	SP	!	"	#	\$	%	&	'
05	(	)	*	+	,	-	.	/	(	)	*	+	,	-	.	/
06	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
07	8	9	:	;	<	=	>	?	8	9	:	;	<	=	>	?
10	@	A	B	C	D	E	F	G	@	A	B	C	D	E	F	G
11	H	I	J	K	L	M	N	O	H	I	J	K	L	M	N	O
12	P	Q	R	S	T	U	V	W	P	Q	R	S	T	U	V	W
13	X	Y	Z	[	\	]	↑	←	X	Y	Z	[	\	]	↑	←
14																
15																
16																
17	DEL								DEL							

NOTES: VFU<sub>0</sub> Slew Paper To VFU Channel 0  
 ⋮  
 VFU<sub>7</sub> Slew Paper To VFU Channel 7.

Fig. I/O. 2a. ASCII Characters and Codes

LOGGING TYPEWRITER, MODEL B (PROPOSED CHARACTER SET)									SELECTRIC TYPWR							
OCTAL CODE = C <sub>2</sub> C <sub>1</sub> C <sub>0</sub>									OCTAL CODE = C <sub>2</sub> C <sub>1</sub> C <sub>0</sub>							
C <sub>2</sub> C <sub>1</sub>	C <sub>0</sub>								C <sub>0</sub>							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
00									0	1	2	3	4	5	6	7
01		HT				CR	RED	BLK	8	9	[	#	@	:	>	?
02					DC4				SP	A	B	C	D	E	F	G
03									H	I	&	.	]	(	<	\
04	SP				\$				↑	J	K	L	M	N	O	P
05			*		,	-	.		Q	R	-	\$	*	)	;	'
06	0	1	2	3	4	5	6	7	+	/	S	T	U	V	W	X
07	8	9							Y	Z	←	,	%	=	"	!
10		A	B	C	D	E	F	G	CR							
11	H	I	J	K	L	M	N	O								
12	P	Q	R	S	T	U	V	W	HT							
13	X	Y	Z						BLK	RED						
14																
15																
16																
17	DEL								DEL							

Fig. I/O. 3. Typewriter Common Peripheral Characters and Codes

Code For One Character Per Row  
Punched Perpendicular To The  
Longitudinal Axis Of The Tape

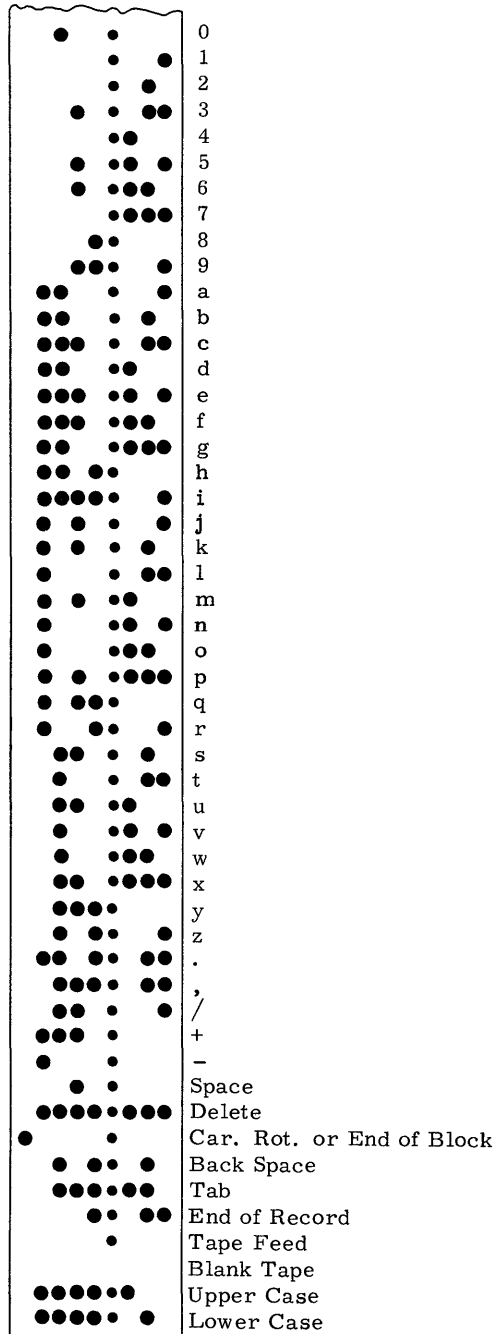


Fig. I/O. 4. EIA RS-244 Characters and Codes

- A data record is being transferred through the channel.
- An Input Demand exists for this channel.
- The companion channel of a pair of channels controlling an input/output peripheral is busy.

### JDR - Jump If Data Ready (K1 =10)

**Input Channel:** If the addressed input channel is busy and its data-ready test line is true (set), JDR transfers program control to the second sequential location. If the test line is false, control is transferred to the first sequential location. If the channel is not busy, control is always transferred to the first sequential location. The command (JNR) is decoded within the AU which monitors the data-ready test point and accordingly simulates a GEN II IN command, provided the channel-ready API is true. Within the I/O Buffer,  $\overline{K1} = 4_8$  enables D1AK14 which in turn enables the device control logic to output the "data-ready" status, via the ready test neckdown logic.

**Output Channel:** If the addressed output channel is busy and the "buffer-ready" line is true, JDR transfers program control to the second sequential location. If the test line is false, control is transferred to the first sequential location. If the channel is not busy, control is always transferred to the first sequential location. The  $K1=4_8$  bit will enable the device "buffer-ready" line in a manner similar to the input channel.

### JNE - Jump If No Error

The JNE command transfers program control to the second sequential location if the error API is false (no error). Control is transferred to the first sequential location if the error API line is true. The command is decoded within the AU which monitors the alarm test point, via the alarm test neckdown logic. Up to four error or alarm lines are available to determine the "status" of a specified on-line device:

- $K1=00X$  - Condition Alarm - Device is in halt, off-line, etc.
- $K1=01X$  - Device Alarm - Card jam and broken tape, etc.
- $K1=10X$  - Operator Alarm - Hopper empty, stacker full, etc.
- $K1=11X$  - Internal Alarms - Parity error, loss of synchronization (data loss, etc.).

JNE may also be used to sample the Demand flip-flop of a specified channel. If set, JNE clears the Demand.

### ABT - Abort Channel

The ABT command initializes the addressed channel. The ready and error lines are cleared within the device logic; therefore, the channel becomes ready. The I/O Buffer decodes the command during the phase A clock:

$$ABT = \overline{S3} \cdot \overline{\phi A} \cdot K2 \cdot K3.$$



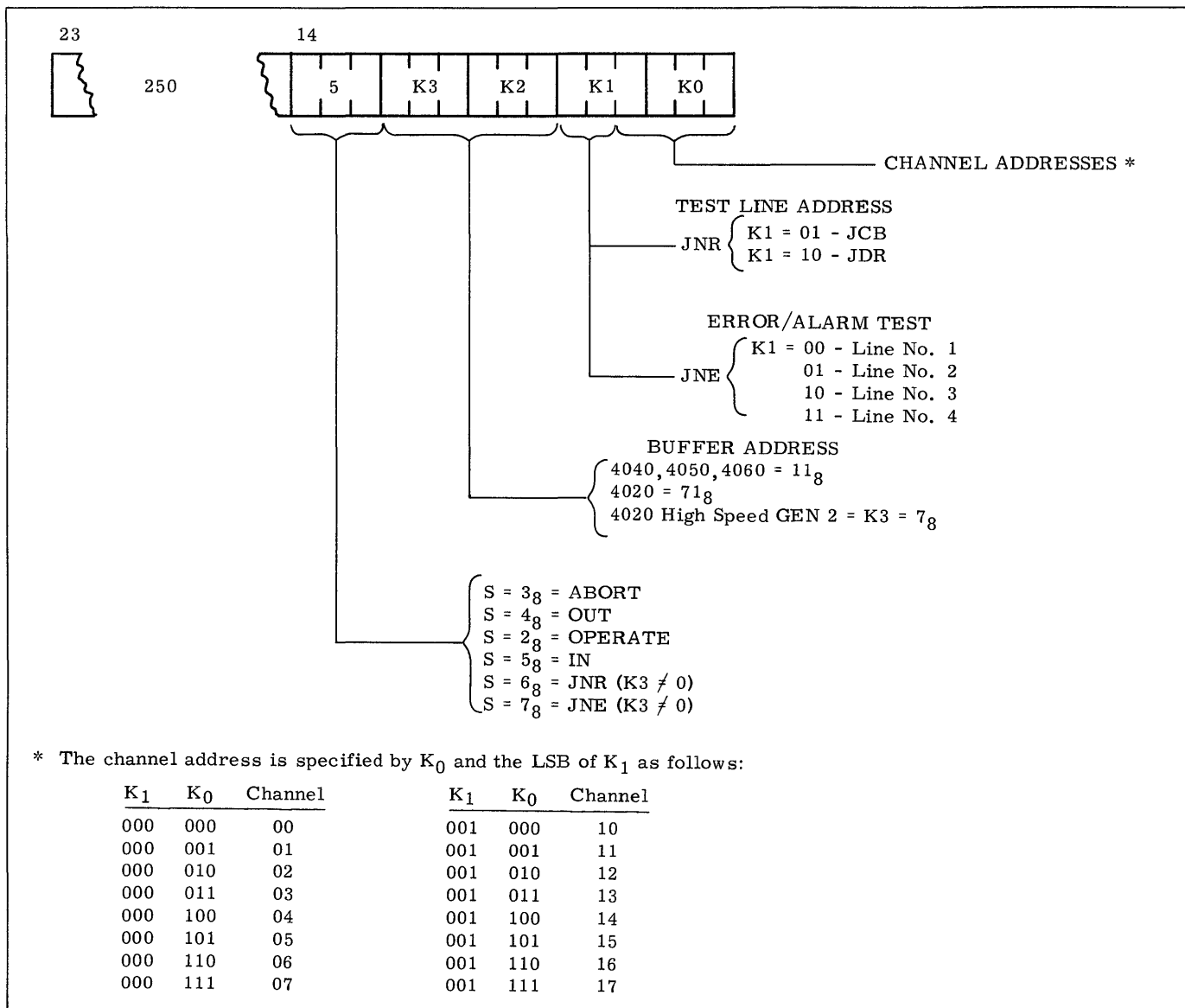


Fig. I/O. 5. GEN II Command Format

## INPUT FUNCTION

Input devices are characterized in that control of the physical movement of the data media (tape, cards, keyboard) and control of data transmission to the Central Processor (CP) are by the device rather than by data inputting instructions. Thus, the CP (program) must synchronize its data-input commands to the mechanical and/or electrical cycle of the peripheral.

The device's control logic initiates movement of the data medium upon receipt of a media enable code (see Table I/O. 1). This code is output from the CP via the OPR command and is decoded within the addressed device control and drive logic. The initiating action sets the reader's clutch flip-flop (or equivalent), provided

the device's ON-OFF switch (or equivalent) and its AC power is ON.

Tape movement on the Teletypewriter Set may be started manually by depressing the X-ON key if keyboard input is enabled; tape movement may always be started by moving the Start-Stop lever on the reader to the START position.

Media movement and inputting of data will continue until media-disable and/or data terminate action is initiated (see Table I/O. 1). For example, the Card Reader will always read at least one card before stopping. The Paper Tape Reader will input until the "end of text" character is detected; however, tape movement will continue until the media-disable code (23g) is detected.

Peripheral	Media-Enable Action	Data-Enable Action	Data-Terminate Action	Media-Disable Action	End of Record Indication	Inter-Record Code
Digitronics PTR	OPR of 021 <sub>8</sub> Code (DC1)	Advent of 002 <sub>8</sub> Code (STX)	INPUT of 003 <sub>8</sub> Code (ETX)	Advent of 023 <sub>8</sub> Code (DC3)	Advent of 023 <sub>8</sub> Code (DC3)	CR, LF
Teletype PTR	OPR of 021 <sub>8</sub> Code (DC1)	Advent of 002 <sub>8</sub> Code (STX)	INPUT of 003 <sub>8</sub> Code (ETX)	Advent of 023 <sub>8</sub> Code (DC3)	Advent of 023 <sub>8</sub> Code (DC3)	CR, LF
NCR PTP	OPR of 022 <sub>8</sub> Code (DC2)	Accomplished by Media Enable	OUTPUT of 024 <sub>8</sub> Code (DC4)	Accomplished by Data Terminate	024 <sub>8</sub> Code (DC4) Punched	CR, LF
Teletype PTP	OPR of 022 <sub>8</sub> Code (DC2)	Accomplished by Media Enable	OUTPUT of 024 <sub>8</sub> Code (DC4)	Accomplished by Data Terminate	024 <sub>8</sub> Code (DC4) Punched	CR, LF
Long Carriage Typewriter	OPR of 177 <sub>8</sub> Code (DEL)	Accomplished by Media Enable	OUTPUT of 024 <sub>8</sub> Code (DC4)	Accomplished by Data Terminate	024 <sub>8</sub> Code (DC4) Outputted	None
Selectric Typewriter	OPR of 177 <sub>8</sub> Code (DEL)	Accomplished by Media Enable	OUTPUT of 024 <sub>8</sub> Code (DC4)	Accomplished by Data Terminate	024 <sub>8</sub> Code (DC4) Outputted	None
Teletype Printer	OPR of 177 <sub>8</sub> Code (DEL)	Accomplished by Media Enable	OUTPUT of 024 <sub>8</sub> Code (DC4)	Accomplished by Data Terminate	024 <sub>8</sub> Code (DC4) Outputted	None
Line Printer	OPR of Line 177 <sub>8</sub> Code (DEL)	Accomplished by Media Enable	OUTPUT of Any Control Code	024 <sub>8</sub> Code (DC4) Outputted	024 <sub>8</sub> Code (DC4) Outputted	None
Selectric Keyboard	OPR of 177 <sub>8</sub> Code (STX)	Accomplished by Media Enable	INPUT of 100 <sub>8</sub> Code (CR)	Accomplished by Data Terminate	INPUT of 100 <sub>8</sub> Code (CR)	None
Teletype Keyboard	OPR of 002 <sub>8</sub> Code (STX)	Accomplished by Media Enable	INPUT of 015 <sub>8</sub> Code (CR)	Accomplished by Data Terminate	INPUT of 015 <sub>8</sub> Code (CR)	None
Card Reader	OPR of 0000 <sub>8</sub> Code	Advent of Column 1	End of Card Indication	End of Card Indication	End of Card Indication	None
Card Punch	OPR of 0000 <sub>8</sub> Code	Advent of Column 1	End of Card Indication	End of Card Indication	End of Card Indication	None

NOTE: Tape movement through the Digitronics reader ceases so that the frame following the DC3 code is positioned for reading. Tape movement through a Teletype reader ceases on the second or third frame following the DC3 code. In each case, the DC3 code and the codes in the following frames (except the code in the frame stopped on) are transmitted to the I/O Buffer. The stopped-on code would be the first code transmitted following another clutch-on action. An "inter-record" gap of at least two full frames is required following the DC3 code.

Table I/O. 1. Peripheral Record Control Actions

Fig. I/O. 6 shows the arrangement of the I/O Buffer during an input function. Assuming that the addressed channel is set busy and that media movement has been initialized, parallel transfer of data between the device control buffer register and the computer A Register is enabled each time the interrupt synchronized IN command is issued. The channel is set busy by the OPR command and remains busy until an "end of record" indication is detected. It then becomes ready to signal the end of record. The programmed operation of an input device, utilizing interrupts, will require the following steps:

- Test the channel-ready line via the JCB command to verify that the channel is not currently busy.
- Initialize the List (TIM/IDL) control word, specifying a core buffer area large enough to hold the entire record of input data.
- Initiate a programmed diagnostic count-down, considering the total elapsed time of the input record.
- Enable movement of the data media using the OPR instruction to output the device's media enable code.
- Data is transferred to the core input buffer via the TIM, or equivalent, (simulates the GEN II IN command) as synchronized by the Data-Ready interrupts.
- Detect the end of record via the channel-ready "echo" interrupt.
- Check for input transmission error using the JNE command.

If the diagnostic countdown underflowed prior to the end

I/O-8

I/O BUFFER

4202A/B-1

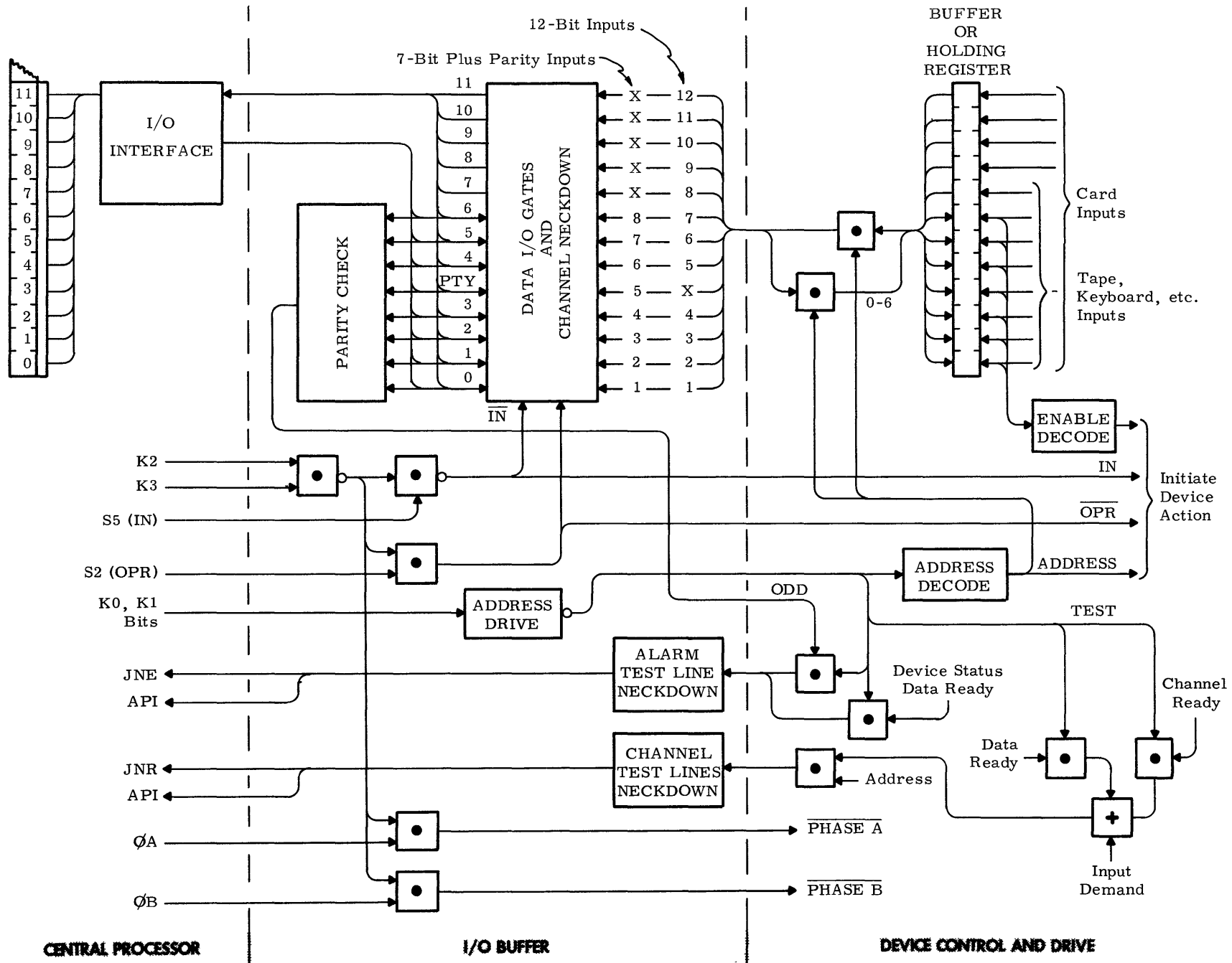


Fig. I/O. 6. Simplified Diagram of the IN Function

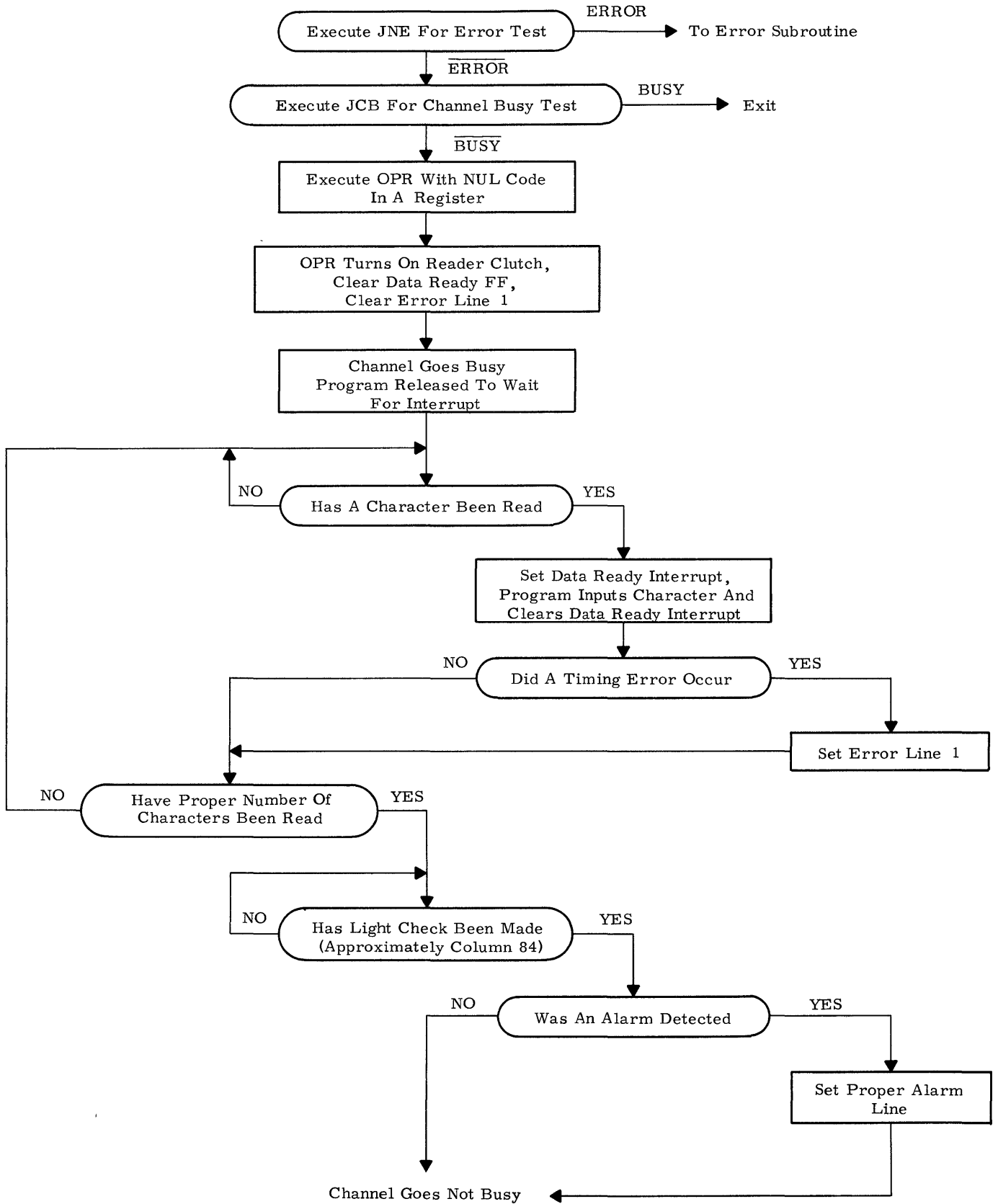


Fig. I/O. 7. Card Read Flow Chart

of record echo, the program will assume device failure and alarm this condition. If a transmission error occurred, the program should request manual operation so that the data media can be back spaced one record; the program should then re-initiate input of the record.

Programmed operation not utilizing interrupts is suitable for use only in an off-line non-process environment. The following program steps are necessary:

- Test for Channel Ready.
- Reserve a core input buffer.
- Enable data media movement.
- Use the JDR instruction to input data.
- Check for transmission error.
- Use the JCB instruction to detect the end of record. In addition, a character count should be maintained to insure that the input does not overflow the input buffer.

In summary, the Input Function enables the following action (see Figs. I/O. 8 and I/O. 9):

- Transfers data to the AU from the channel's buffer. The parity bit (if any) is separated from the data. The data is transmitted to the AU.
- Resets (clears) the channel's data-ready flip-flop.
- Sets the channel's data-error flip-flop if the data is found to have incorrect parity (character input devices only); otherwise, the flip-flop is left unaffected.
- Decodes the 015<sub>8</sub> or 100<sub>8</sub> code (CR) as the keyboard input media-disable code. The code is sent to the AU and the input channel-busy flip-flop is reset (cleared) to indicate "end of record".

## OUTPUT FUNCTION

The operation of output typers or other printing devices and paper tape punches is incremental, that is, data output and data media movement is controlled by the computer (program). Operation of the serial card punch is similar to the operation of input devices in that the computer must synchronize its output actions to the independently moving data media.

The output device's control logic will enable the recording of data on the medium upon receipt of a media-enable code (see Table I/O. 1). The code is not recorded. However, all subsequent codes, including additional media-enable codes, will be recorded until a media-disable code is received. The media-disable code is recorded, with the exception of card media. The 177<sub>8</sub> code (DEL) is a special "do nothing" code whose

only effect is to cycle the buffer-ready status of the addressed channel.

Fig. I/O. 9 shows the arrangement of the I/O Buffer during an output function. Assuming that the addressed channel is set busy and that media-enable has been initialized, parallel transfer of data between the computer A Register and the addressed device control buffer register is enabled each time the interrupt synchronized OUT command is issued. The channel is set busy by the OPR command and remains busy until the "end of record" is detected via the channel-ready "echo" interrupt. Note, that the channel-ready interrupt provides (with the exception of paper tape punches), the "end of record" echo. TOM/ODL echo generators are provided only for the paper tape punches; this echo is the OR of the channel-ready test signal and the TOM table full echo. The programmed operation of an output device, utilizing interrupts, will require the following steps:

- Test the channel-ready line via the JCB command to verify that the channel is not currently busy.
- Initialize the List (TOM/ODL) control word.
- Initiate a programmed diagnostic count-down, considering the total elapsed time of the output record.
- Enable the data media using the OPR instruction to output the device's media-enable code.
- Data is transferred from the core output buffer via the TOM, or equivalent, as synchronized by the buffer-ready interrupts.
- Detect end of record via the channel-ready "echo" interrupt.
- Check for output transmission error using the JNE command (not meaningful to all devices).

If the diagnostic count-down underflowed prior to the end of record echo, the program will assume device failure and alarm this condition. If a transmission error occurred, the program should re-output the last record.

Programmed operation not utilizing the interrupts is suitable only in an off-line, non-process or test environment. The following program steps are necessary:

- Test for Channel Ready.
- Enable data media.
- Use the JDR instruction to output data.
- Use the JCB instruction to detect the end of record.
- Check for output transmission error, if applicable.

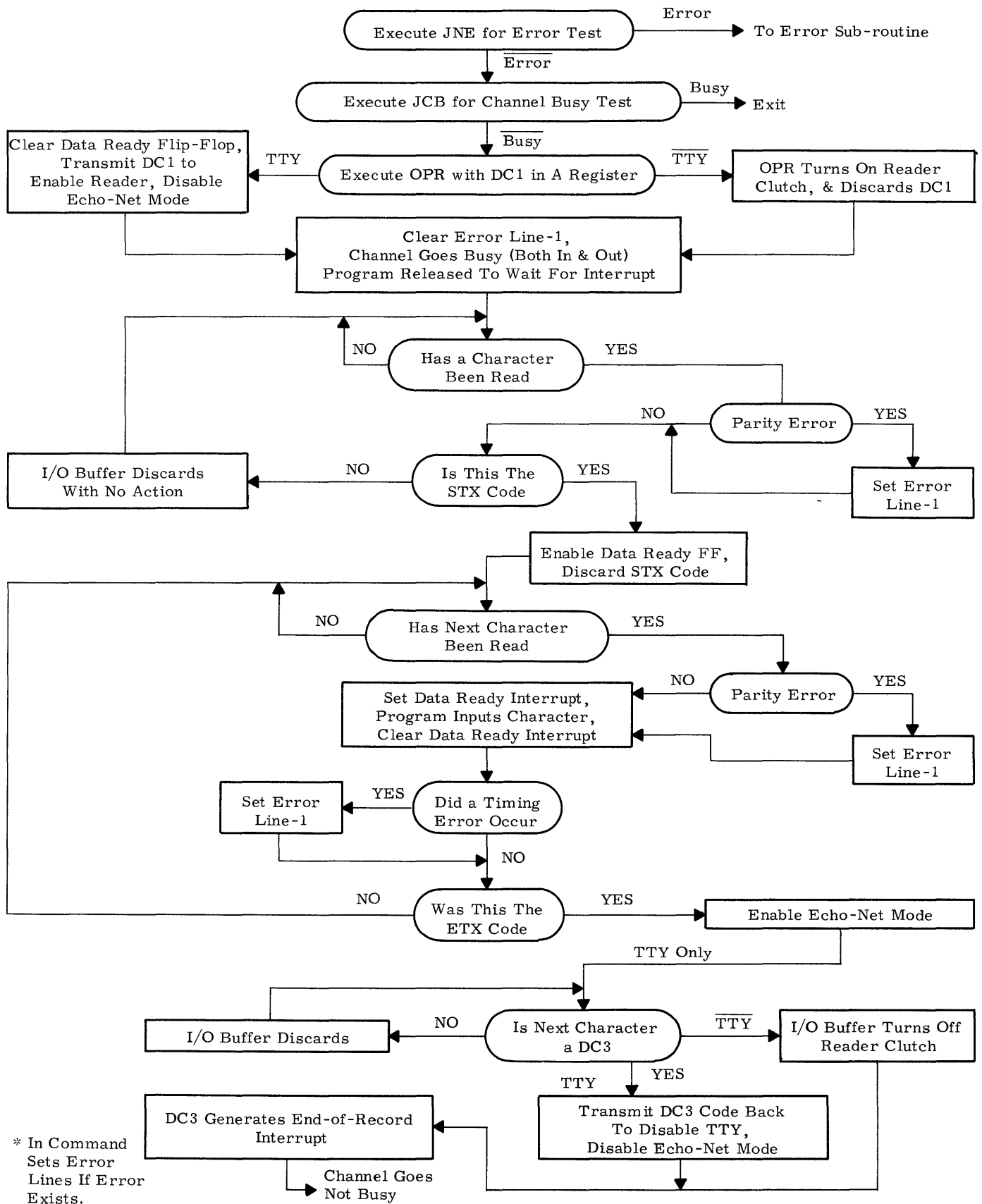


Fig. I/O. 8. Paper Tape Read Flow Chart

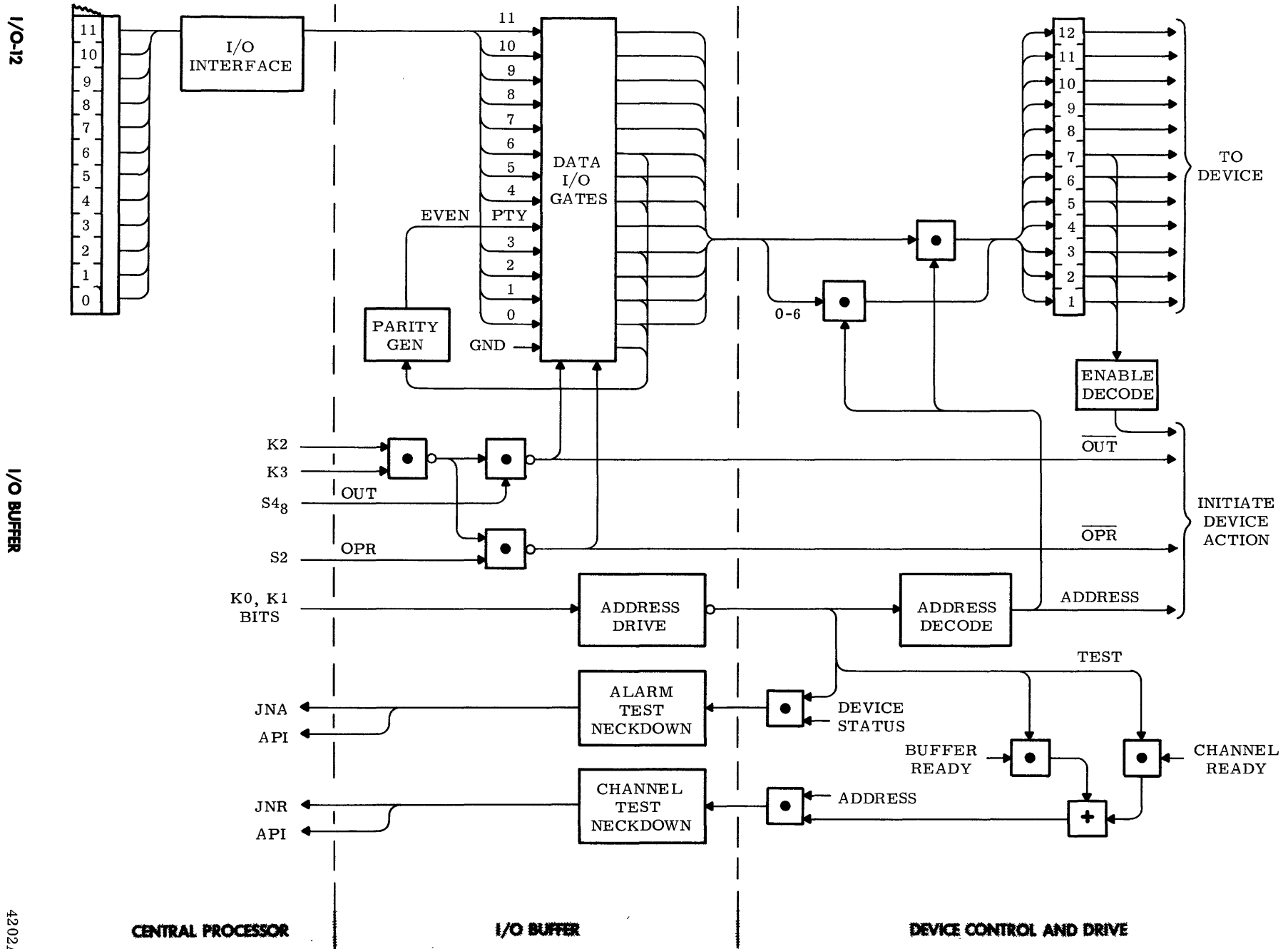


Fig. I/O. 9. Simplified Diagram of the OUT Function

In summary, the Output Function enables the following action (see Figs. I/O. 10 and I/O. 11):

- Data is transferred to the channel's buffer from the AU.
- The parity bit (if required) is generated and appended to the data.
- The channel's buffer-ready flip-flop is reset.
- Decodes the 177g code (DEL) as a legal "do nothing" output code. It does not affect the device or data media. However, the buffer-ready signal is cycled and becomes set at the end of the output action.
- Initiates (incremental devices only) one data-write operation followed by a data media movement operation. The free running devices utilize data in the buffer independently of the output action.
- Decodes media-disable codes which cause the output channel-ready flip-flop to reset (clear).

## PARITY CHECK AND GENERATE

The I/O Buffer parity logic is capable of checking parity on input data and generating parity on output data. The input parity test is only meaningful if parity is part of the data input (e. g. , an even parity bit is included with a paper tape character but not with a character from a card reader). To limit the error test to devices which provide a parity bit, the error signal, D0PPAB (area "P"), is routed to the device control logic where it is ANDed with the channel address to set the alarm flip-flop.

When parity checking, the eight least significant bits, including the channel parity bit of the input data, are sampled by the quarter-adder summation circuit. If an even number of 1-bits are present, the output will be 0V (D0PPAB). If an odd number of 1-bits are present, the output will be +3.5V (D0PPAB), indicating parity error.

When outputting, the same logic becomes a parity generator simply by forcing the input parity channel level to 0V and driving the output parity channel with D0PPAB. Note, the input to G0DOPT (area "D") is always grounded. The output data is sampled in the same manner as for parity checking; however, if the output word (bits 00-06) contains an even number of 1-bits, D0PPAB is enabled and a 0-bit is placed in the output parity channel. Should the output data word contain an odd number of 1-bits, D0PPAB will place a 1-bit in the output parity channel.

## INPUT DEMAND

The input demand function allows an operator to manually initiate a data transfer sequence from any input device. An input demand is initiated by:

- Teletypewriter - Depress the BREAK key on the keyboard (ASR, KSR). This action will open the transmission line to the device's control logic and lock all keyboards on that line. If the channel is not busy, the control logic will cycle the channel-ready API.
- Other Input Devices - Depress the Input Demand pushbutton. If the channel is not busy, the device's control logic will cycle the channel-ready API.

The computer program will recognize the interrupt as an input request and enable the input device or print a message indicating that keyboard input will be accepted.

## AUTOMATIC PROGRAM INTERRUPTS

The following interrupt signals provide standard API communications between the program and the I/O Buffer channels:

- Input Channels
  - Individual data-ready API's assigned as non-inhibitible interrupts.
  - Individual channel-ready API's assigned as inhibitible interrupts.
- Output Channels
  - Individual buffer-ready API's assigned as non-inhibitible interrupts.
  - Individual channel-ready API's assigned as inhibitible interrupts.

The interrupt signal level is such that the interrupt flip-flop is set when the signal is true. The channel-ready API functions as the "end of record" echo. The data-ready and buffer-ready API's are used to synchronize data transfer between the AU and the addressed device.

An API input signal from an output-only or input-only device is the same as the corresponding JCB test signal. The API input signals from an I/O device differ from the JCB test signals in that both test signals will indicate busy whether either channel is in use; only the API for the channel actually in use will indicate busy.

## PACKAGING AND OPTIONS

There are two basic models of the I/O Buffer which are electrically and functionally identical, but mechanically different:

- Model 4202A - This model is used with the 4020 computer system. It is packaged in a GEAPS (General Electric Advanced Packaging System) page-mounting module. The basic buffer consists of three 10-inch printed wire boards (PWB) plus one or two boards for each attached device.
- Model 4202B - This model is specially packaged as a GEAPS module for use with the 4040, 4050,



and 4060 computers; also with a 4020 computer system that requires the I/O Buffer module be mounted in a GE/PAC page. The 4202B has a total capacity of twelve channels with a maximum of seven inputs or twelve outputs. The module is implemented for five spare PWB slots.

The only I/O Buffer options is the number and kinds of peripheral channels. Each channel has its own buffer and drive hardware. In some cases multi-purpose boards are used. For example, the drive hardware used for the Paper Tape Reader may also be used for the Card Reader; however, the corresponding channel connectors must be interchanged.

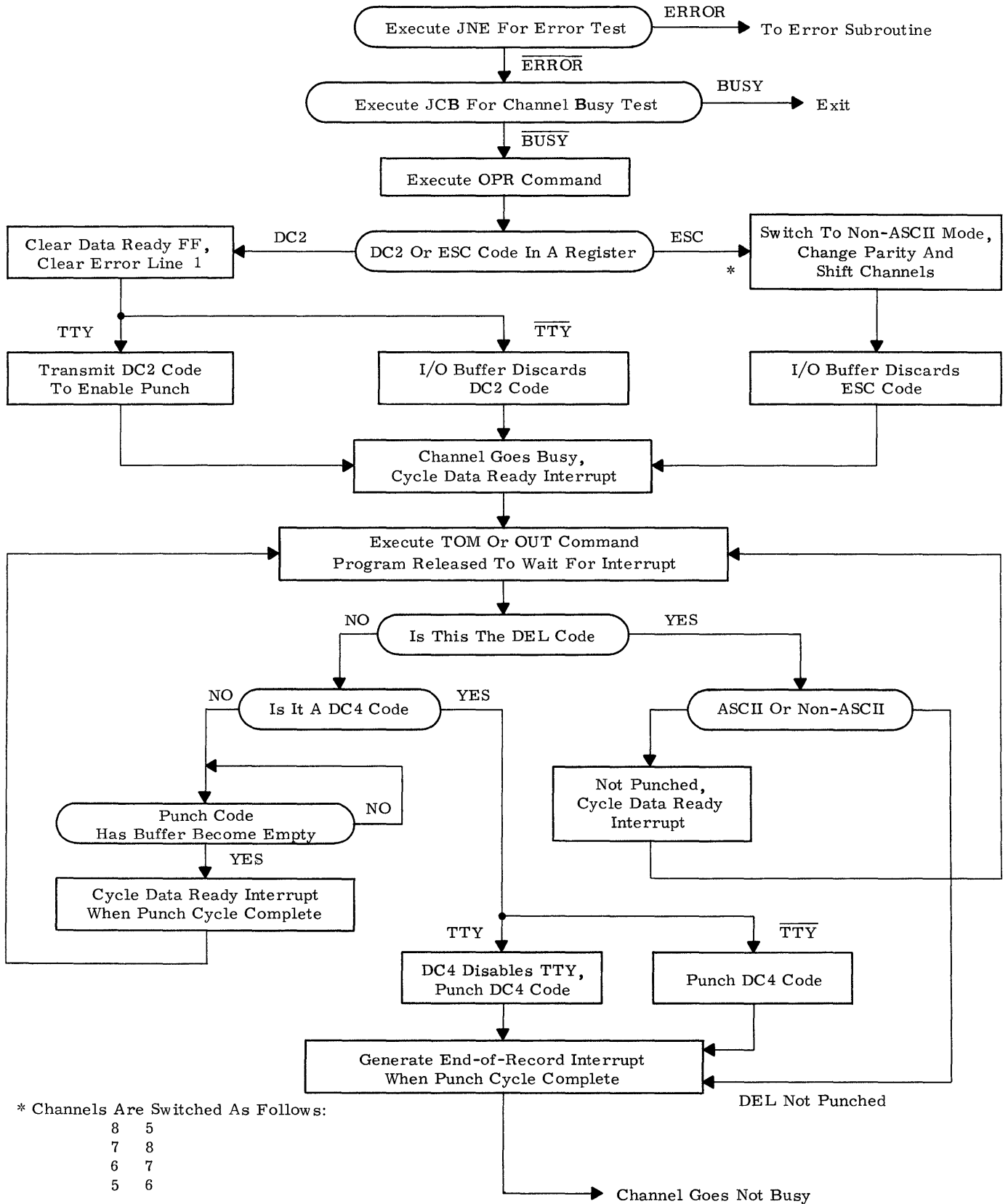


Fig. I/O. 10. Paper Tape Punch Flow Chart

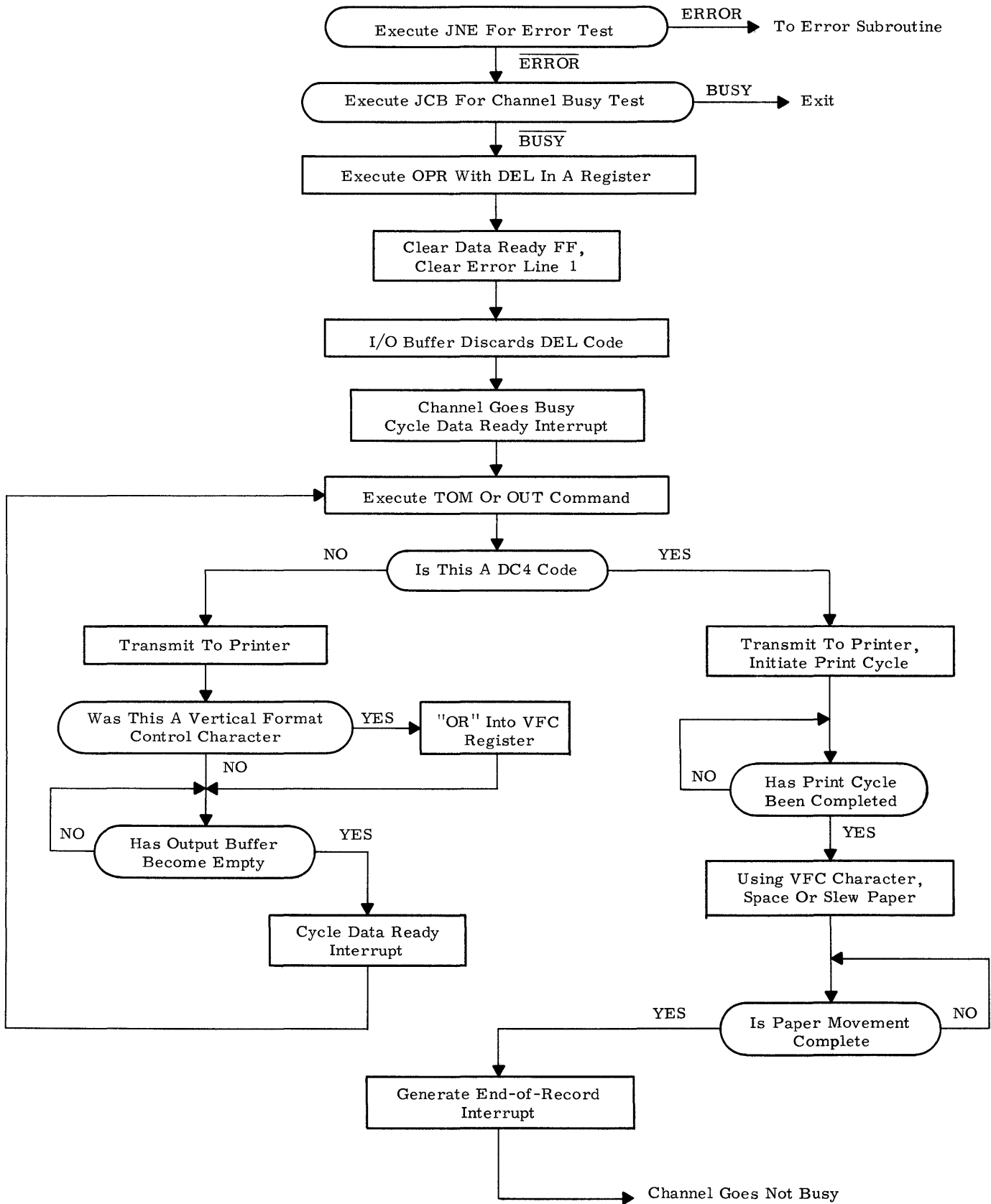


Fig. I/O. 11. Printers Flow Chart

## SPECIAL CHARACTER FUNCTIONS

In order to control the action (media enable, etc.) of the attached peripheral devices, certain special characters are recognized by the I/O Buffer drive logic or by the device itself.

### Start of Text (STX) 002

The STX code is used when operating Paper Tape Readers and Keyboards. It becomes the data enable code when reading paper tape, i. e. , it informs the associated drive that the next incoming character is the first data character. In this case STX is discarded by the drive logic.

When keyboard input is requested, the program will normally execute an OPR instruction with STX in the A Register. The STX character in this case will enable the Echo-net mode so that all characters transmitted from the keyboard to the I/O Buffer are also sent back to the typer to provide the operator with hard copy of his data. STX also unlocks the keyboard on the Selectric I/O Typer.

### End of Text (ETX) 003

The ETX code is used when operating Paper Tape Readers. When decoded, the drive logic will switch to Echo-net mode so that the next character received (normally a DC3) will be transmitted back to the paper tape reader to disable it and prevent further data from being read. The ETX character is stored in core (list) as the last character of the message.

### Delete Code (DEL) 117

When OPR or OUT is executed with DEL in the A Register, it is decoded within the device drive as the channel buffer ready signal for Typers and Line Printers. This signal will enable the Data Exchange Ready API

interrupt. DEL is not transmitted.

### Carriage Return (CR) 015

The CR code is used with Keyboard input. It is decoded by the device drive to enable the End of Record API and the non-echo-net mode (after echoing). CR (100g) locks the keyboard on the Selectric I/O Typer. The ASR Teletype receives and decodes CR as the RETURN function.

### Device Control - 1 (DC1) 021

The DC1 code is used as the media-enable code for Paper Tape Readers. When the OPR instruction is executed with DC1 in the A Register, paper tape movement is enabled. The ASR Teletypewriter receives and decodes DC1 as the X-ON function.

### Device Control - 2 (DC2) 022

The DC2 code is used as the media-enable code for Paper Tape Punches. When the OPR instruction is executed with DC2 in the A Register, paper tape feed is enabled. The ASR Teletypewriter receives and decodes DC2 as the TAPE ON function.

### Device Control - 3 (DC3) 023

The DC3 code is used as the media-disable code for Paper Tape Readers. DC3 normally follows ETX and serves as the end of Record API. The drive is switched to the non-echo mode after echoing the code back to the reader to turn it off. The Teletypewriter decodes DC3 as the X-OFF function.

### Device Control - 4 (DC4) 024

The DC4 code is used with Typers, Paper Tape Punches, and Line Printers to generate the End of Record API and to turn off the devices. The Teletypewriter decodes DC4 as the OFF function.

# ASR AND KSR TELETYPEWRITER DRIVE AND CONTROL

The 4202A/B 061 and 062 Serial Transmitter and Receiver Drives are capable of exchanging recorded communications (full or half duplex), with or without Echo-Net, between the computer and ASR/KSR Teletype equipment, via the I/O Buffer at a rate of 10 characters per second (110 bits per second). The optional drive functions include:

- 061 - 110 BPS, direct drive (no modem) with provisions for Echo-Net. Requires one input and one output channel.
- 062 - Same as 061, except no Echo-Net provisions.

Note, on some GE/PAC systems, the 061 drive was also used in place of the 060 direct drive for RO Teletype equipment including Answer Back - only the output channel hardware is used.

For this application and discussion, when the computer (CPU) is outputting (OUT), the I/O Buffer drive is transmitting and the Teletype Set is receiving. Conversely, when the CPU is inputting (IN), the drive is receiving and the Teletype is sending.

Both Teletype Models 33 and 35 (GE/PAC Model 4233C) respond to American Standard Code for Information Interchange (ASCII). This is an 8-level code with an 11-bit transmission pattern, i. e. , one start bit, seven data bits, one parity bit, and two stop bits. The start and stop bits ensure synchronization between the sending and receiving equipment.

Since all 4233C Teletype equipment will terminate into duplex (two pair) signal lines, the full or half duplex option is determined primarily by the software or program requirements. Full duplex operation permits receiving messages and transmitting them at the same time without interference between the two signals. This is accomplished by electrically separating the sending and receiving loops of the set by changes in the wiring on the basic facilities assembly in the electrical service unit and connecting the loops to the appropriate duplex signal lines.

The Echo-Net option (061 Drive) enables the I/O Buffer to output to the same inputting device, the data which it is sending to the computer. For example, when inputting from a keyboard, the input data which is serially shifted into the message receive register can be immediately

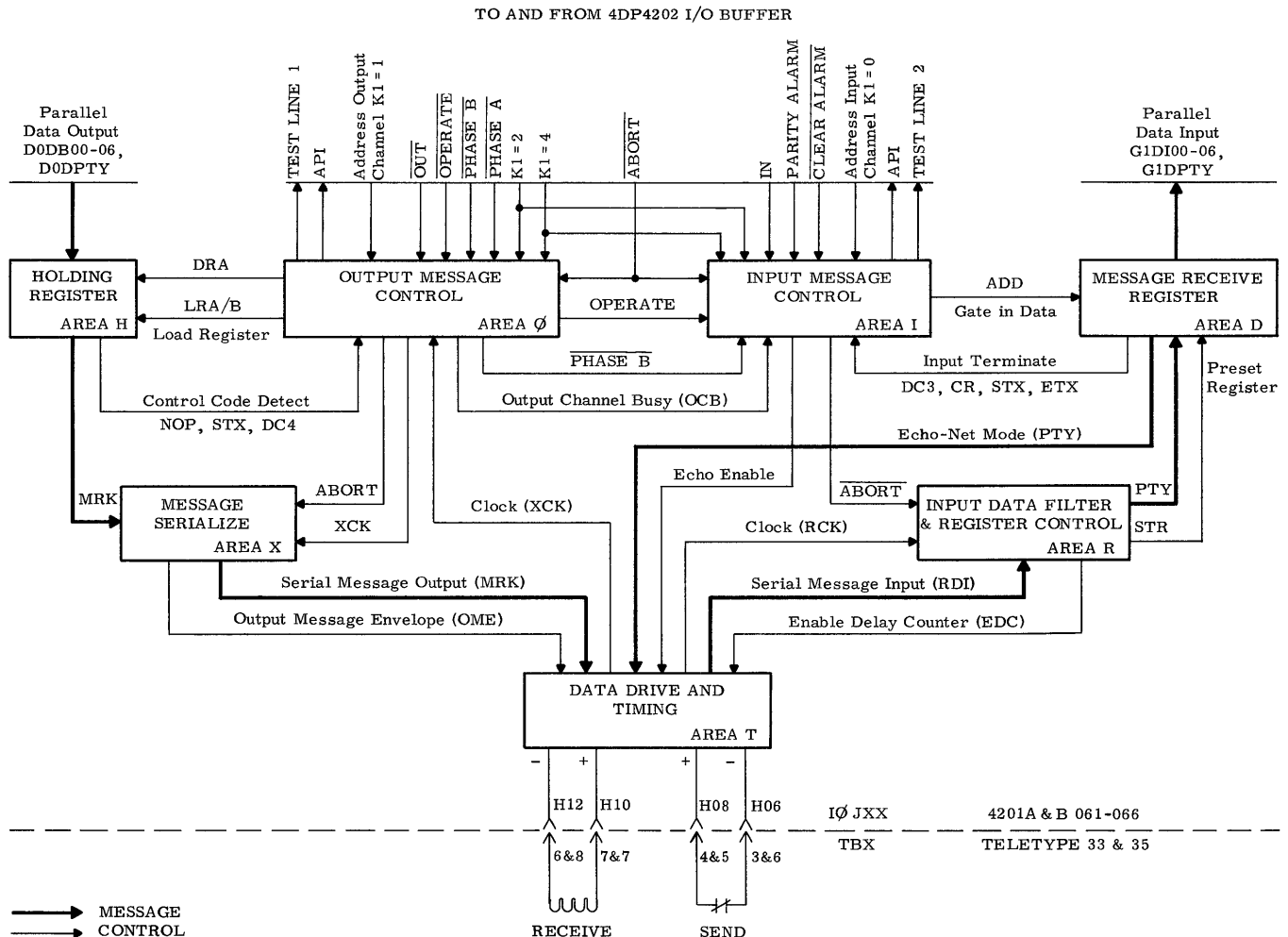


Fig. TTY. 1. Simplified Block Diagram - Teletype Drive ASR/KSR

(within 9 milliseconds) sent back to the typing unit for hard copy. The switching from Echo-Net to Non-Echo-Net mode or vice-versa is via the decoding of special characters, inserted by the programmer.

The ASR/KSR Teletype drive logic, as shown by Fig. TTY.1, is implemented entirely with integrated circuit (IC) elements which are mounted on two printed wire boards (PWB), PTR A1 and PCMA5. The following text describes this logic and timing, referenced to GE Drawing 68C972157. The GEN 2 instructions, special character codes, and general input/output functions are described in the preceding basic I/O Buffer section of this manual.

### CLOCK TIMING

The free-running unijunction oscillator, Q1TCLK (Area T), which is implemented for an output of 880 cps, provides basic timing signals for both transmit and receive functions, as shown by Fig. TTY.2. Timing stability and frequency division is achieved by driving a 3-stage time counter with CLK and then ANDing the output of the counter with CLK. Since CS0 will toggle at 440 cps, CS1 at 220 cps, and CS2 at 110 cps, the output of transmit clock gate, G0TXCK, is a dc pulse (4V peak) with a repetition rate of 110 cps. R1 is adjusted so that the distance between XCK pulses is 9.09 milliseconds, one bit time. The receive strobe clock, G0RSC3 (Area R), is generated in much the same manner as XCK. However, the strobe clock is not free running, but is enabled only during enable strobe clock (G1RESC) time.

### OUTPUT (CPU TO TELETYPE)

Initially, for any output operation (punch or print), the assigned output channel must be ready (not busy). The output channel busy (OCB) flip-flop (Area O) is normally cleared (reset) by the media-disable code (DC4) from the previous operation, or by initializing the computer or by the Abort device instruction:

$$F1OOCB = DC4 + ABT.$$

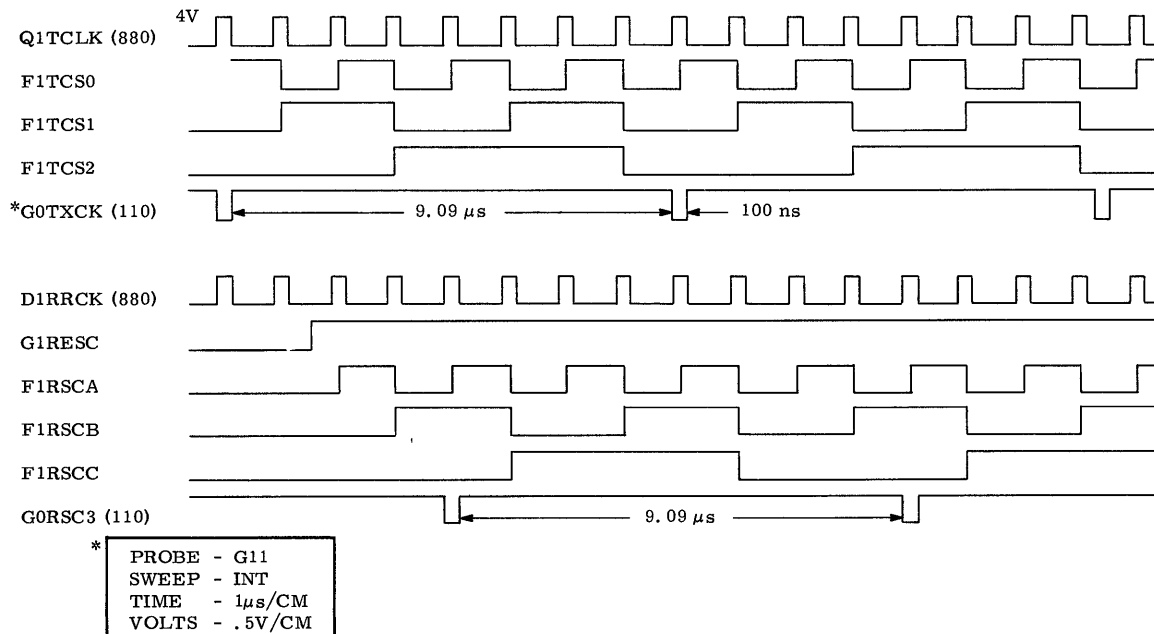


Fig. TTY.2. Transmit/Receive Clock Timing

A channel ready test is made by the program via the JCB (K1=2g) instruction. If the channel is not busy (ready), the test line 1 gate, G0OTL1, is enabled (0V), provided the associated input channel is ready and there is no input demand. A demand or Break will make the input channel appear busy. With TL1 enabled, the program will step to the next sequential instruction:

$$G0OTL1 = F1OOCB \cdot ADDRESS \cdot K12 \cdot G0ICRI.$$

### Media-Enable

Assuming that a punch paper tape operation is desired, the media-enable, special character code, DC2 (022g), is first placed in the A Register, then the OPR instruction is executed. At ØB time, G0OOPR will set the OCB flip-flop and clear the channel buffer ready (CBR) flip-flop. At ØA time, the DC2 code is gated into the output holding register, F1HD00 - F1HD06 (Area H). Note, the polarity of the data bits which are gated into this register is such that a 1-bit is 0V and the 0-bit is +3.6V.

CBR will enable the send request line driver, D1TSRQ (Area T), and arm the transmit time line receiver, D1TXMT. During ØA, the load register gate, G0OLRA, besides enabling the clock driver, D1HGDI (Area H), to gate the DC2 code into the holding register, also clears the delay counter, DC0 and DC1 (Area T) via the G0TEDC gate. As shown by the Output Timing Diagram, Fig. TTY.3, the third XCK pulse following ØA will load the delay counter, enable XMT, and set the output message envelope flip-flop, OME (Area X).

At the start of OME, the 4-bit, base-11 counter, F1XSS0-SS3, which is implemented so that it is self-clearing after the count of 10, will always hold a count of zero. Since there is no gating element associated with the count of zero, the marking gate, G0XMRK, will always be disabled at this time. As a result of MRK and OME both being positive, the send data gate, G0TSDO, is enabled; thus sending the "start" no current (spacing) pulse.

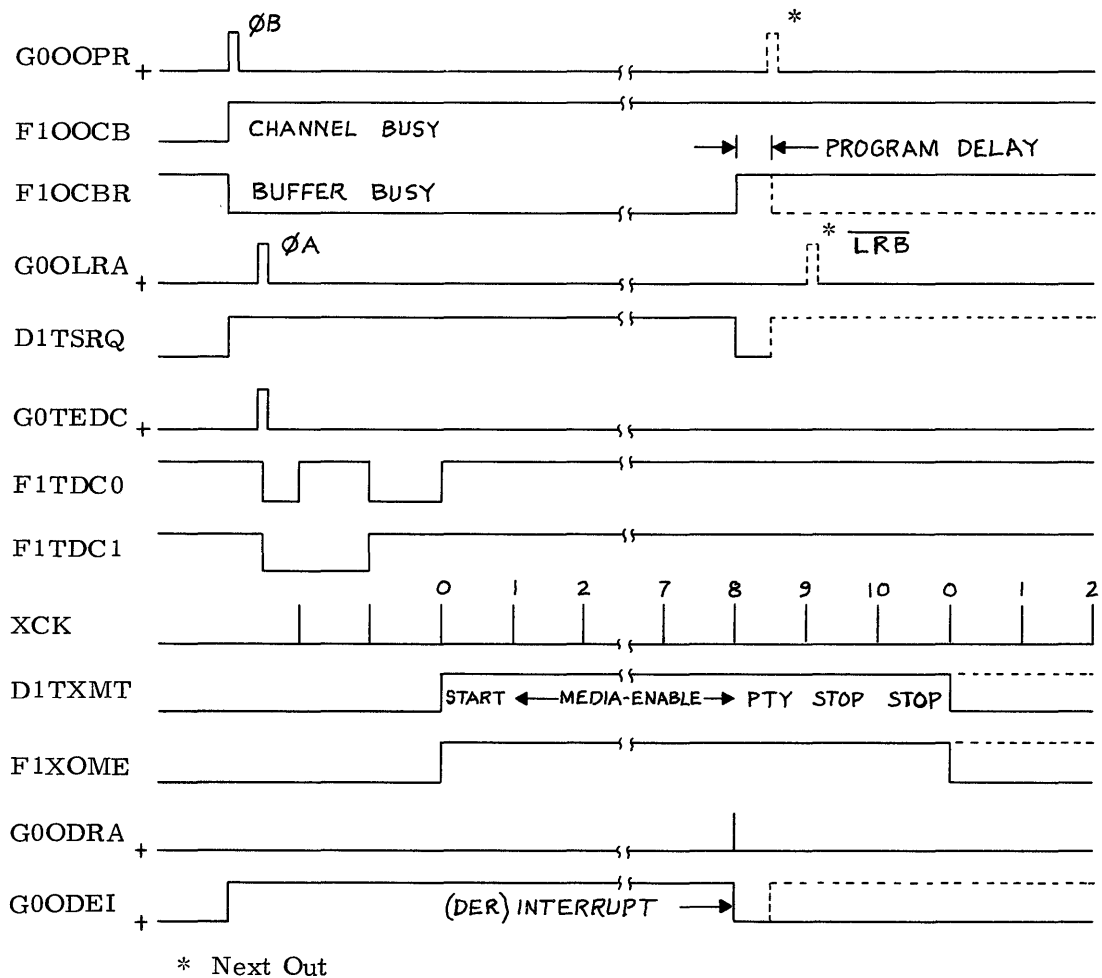


Fig. TTY. 3. Output Timing Diagram

As shown by the Transmit Control Timing Diagram, Fig. TTY. 4, the clock pulses following OME will toggle the counter which progressively samples the data, beginning with the least significant bit, by enabling the associated gate when the data is true. In effect, the data bits are shifted serially through the G0XB03 gate. At the count of 8, the parity bit (if any) is sampled by the G1XPTY gate. At the same time, the data ready available gate, GOODRA, is enabled:

$$G1XSTP = F1XSS0 \cdot F1XSS1$$

$$G1XPTY = G1XSTP$$

$$GOODRA = F1XSS3 \cdot G1XPTY \cdot XCK.$$

At the count of 9, G1XSTP is ANDed with F1XSS3 to hold the output of G0XMRK at 0V, generating the first "stop" bit. This condition is also true for the second "stop" bit, at the count of 10:

$$G1XSTP = F1XSS0 + F1XSS1$$

$$G1XMRK = G1XSTP \cdot F1XSS3.$$

After transmission of the parity bit, the CBR flip-flop is set as a result of DRA which was enabled (0V) with the count-of-8 clock. DRA goes positive with the fall of the clock and generates the data exchange ready interrupt by enabling G0ODEI. As a result of the API and/or the test line 1 level change (JDR, K1=4g), the program, at this time, will execute the next output instruction.

In summary, during XMT, the "start" bit is generated and transmitted, followed with each succeeding XCK pulse by bits of the DC2 code (least significant bit first) and two "stop" bits. This 11-bit transmission pattern is received by the Teletype equipment as a series of "spacing" (no current flow) and "marking" (current flow) pulses, and interpreted as the TAPE function which turns on the paper tape punch.

For a typer operation (print), the logic events are similar to those described for the punch. However, the special character code, DEL (177g), is not transmitted to the Teletype, but is decoded by G0HNOP (Area H) as soon as it is gated into the holding register. The output of NOP (0V) will simply set the CBR flip-flop which in turn enables the data exchange interrupt. Note, as soon as NOP loses its positive drive, CBR sets as a result of G0IDEC. DEC is always positive except when an input OPR with STX (Echo-Net) is in progress.

### Data-Enable

The OUT instruction which is executed as a result of the data exchange ready API, clears the CBR flip-flop at ØB. At ØA, OUT transfers data plus parity from the A Register, via the I/O Buffer channel, to the holding register. The OUT which follows punch turn on must occur prior to the count-of-9 XCK or within 8 milliseconds of the interrupt. Since XMT and OME can only be

cleared coincident with CBR and XCK, they will remain positive. As a result, the start bit for the output data will be generated with the count-of-zero, or immediately following the last stop bit.

### Data-Terminate and Media-Disable

For both the punch and typer, output of the DC4 code will clear the OCB flip-flop after the code is transmitted:

$$\begin{aligned} \overline{OCB} &= DC4 \\ DC4 &= 024_8 \cdot DRA \\ DRA &= \text{Count-of-eight} \cdot PTY \cdot XCK. \end{aligned}$$

The Teletype equipment will interpret the DC4 code as the ~~TAPE~~ function which turns off the punch. With DRA and OCB, the CBR flip-flop will set disabling SRQ and XMT. The DEI interrupt will indicate end-of-record.

A channel test is made by the program via the JCB (K1 = 2g) instruction. If the channel is not busy (ready), the test line 2 gate, G0ITL2, is enabled (0V), provided the associated output channel is ready and there is no input demand. With TL2 enabled, the program will step to the next sequential instruction:

$$\begin{aligned} G0ITL2 &= \text{ADDRESS} \cdot \overline{OCB} \cdot G0ICRI \cdot K12 \\ G0ICRI &= \overline{ICB} \cdot \overline{BREAK}. \end{aligned}$$

### Media-Enable

The next sequential instruction would be OPERATE, addressed to the input channel, with the appropriate media-enable code. OPR sets the input channel busy, clears the data ready enable, DRE flip-flop (Area I), and disables G0IDEC allowing CBR to clear. For the paper tape reader, the DC1 (021g) code is transmitted to the Teletype in the same manner as described under "Output Media-Enable". The Teletype decodes DC1

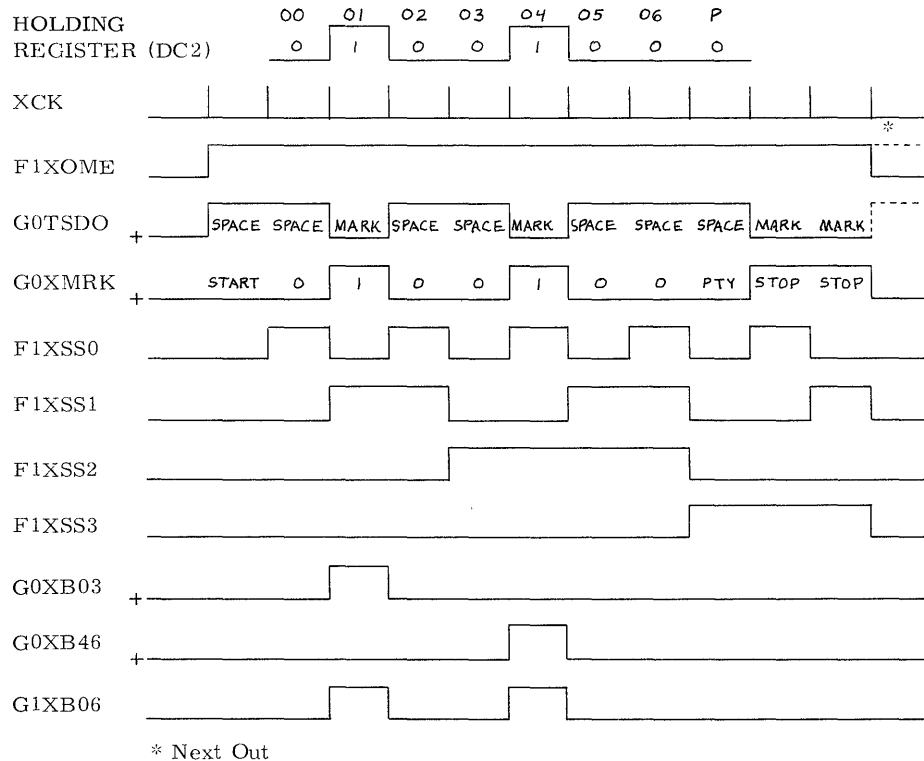


Fig. TTY. 4. Transmit Control Timing Diagram

### INPUT (TELETYPE TO CPU)

Initially, for any input operation (tape or keyboard), both the assigned input channel and output channel must be ready (not busy). The input channel busy (ICB) flip-flop (Area I) is normally cleared (reset) by the media-disable code, DC3, for the tape reader, and CR for the keyboard, from the previous input operation. The channel is also cleared by initializing the computer or by the Abort device instruction:

$$F1ICB = DC3 + CBC + ABT.$$

as the X-ON function which turns on the reader, i. e. , starts tape feed.

The reader sends the first character code which will always be STX (002g). As shown by the Input Timing Diagram, Fig. TTY. 5, STX is registered with the seventh shift pulse decoded (G0DSTX), and used to set DRE with the last shift pulse:

$$F1IDRE = STX \cdot D1DSHB.$$

DRE, in effect indicates that the next character re-



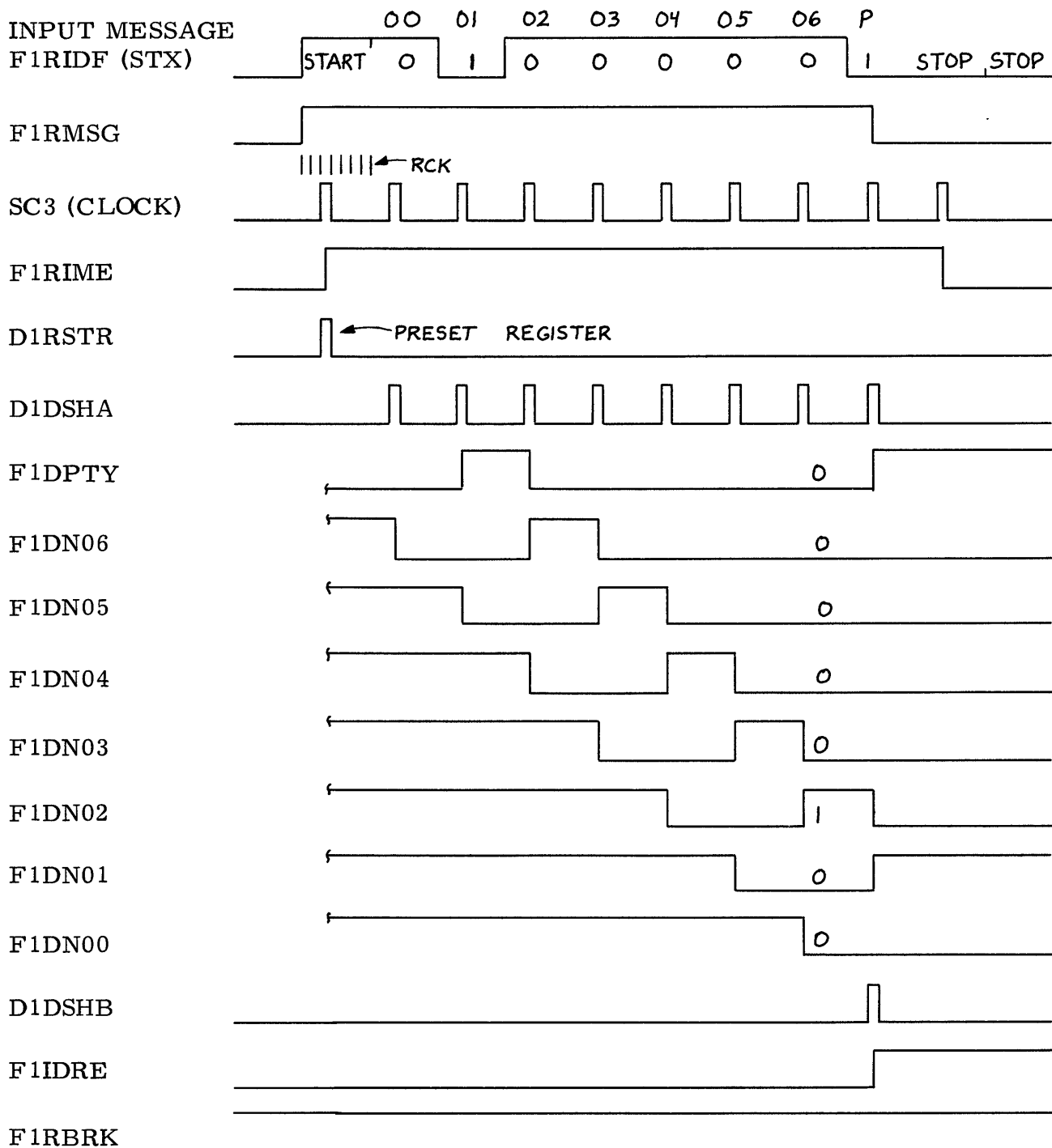


Fig. TTY. 5. Input Timing Diagram

ceived from the tape reader will be meaningful data, and that an interrupt will be enabled when the character is registered. No further use of STX is made.

For keyboard input, STX is gated into the holding register, decoded (G0HSTX), and used to unconditionally set DRE:

$$\begin{aligned} \text{GOIDES} &= \text{OPR} \cdot \text{STX} \\ \text{F1IDRE} &= \text{GOIDES}. \end{aligned}$$

Here again, DRE sets up the input control logic so that the next input from the keyboard will generate an interrupt.

#### Data-Enable

The next data character or input message following media-enable is shifted into the serial receive register, F1DN00-06 and F1DPTY, as shown for STX (Fig. TTY. 5). However, with DRE set, the last shift pulse (SHB) will set the data ready, DRY, flip-flop when the

last input bit is registered. DRY will enable the API drive which informs the computer to execute the IN instruction.

Upon receipt of the start (spacing) pulse which precedes the next data character, the send data enable line receiver, G1TSDE (Area T) is enabled (+5V). As a result, the input data filter, IDF, flip-flop is set with the free running RCK clock. Next, IDF is ANDed with BRK (set during IDF) to enable G1RESC which turns on the strobe counter, F1RSCA-CC. The next RCK clock will set the message, MSG, flip-flop and toggle the strobe counter.

The third RCK clock following IDF (see Fig. TTY. 1), will enable SC3. This first strobe clock will set the input message envelope, IME, flip-flop and disable the start clock driver, D1RSTR, which presets the receive register to all ones except for PTY which is cleared.

With IME and MSG, the next SC3 (9.09 milliseconds later) will generate the first shift pulse, G0RSHF, which shifts the register to the right one place, PTY now holds the least significant bit of the input message. The succeeding shift pulses continue to shift in the data bits until the "0" bit which was preset in the PTY flip-flop clears N00. This will always occur with the seventh shift pulse.

As a result of N00, the ninth SC3 clock clears MSG and enables the eighth shift pulse. This shifts the register one more time, so that it now contains all of the data bits plus parity. The eighth shift pulse also enables D1DSHB, which sets DRY.

$$F1IDRY = \overline{N00} \cdot DRE \cdot ICB \cdot SHB.$$

MSG arms the clear side of the IME flip-flop and disables G0RSHF so that the tenth SC3 will not generate a shift pulse, but will clear IME which then disables G1RESC. ESC (0V) unconditionally clears the strobe counter and holds it in the clear state, terminating the SC3 clock.

Data ready is ANDed with MSG to enable the data exchange interrupt, DEL, which then enables the computer API drive or JDR test point. The computer program responds by executing the IN instruction.

If for any reason, the MSG flip-flop failed to clear, the interrupt is aborted and the alarm gate, G0IALM, is enabled. This sets the alarm flip-flop which arms the any alarm/error gate, G1LJNA, and provides an alarm display output.

The input channel address bits which accompany the IN instruction are decoded to enable D1IADD (Area I) which then gates the data out of the receive register, in parallel, to the computer via the I/O Buffer channel data lines. Note that the data off the tape is inverted as it is shifted into the receive register and inverted again as it is gated out; thus, a hole or "0" bit is seen at the I/O Buffer as a 0V level or true signal.

At 0B, IN is decoded to unconditionally clear DRY.

Also, if an odd number of "1" bits were present during parity check, D0PPAB will be disabled, allowing IN to set the alarm flip-flop. PAB (3.6V) indicates parity error. The alarm flip-flop, F1IALM, is cleared by depressing the console clear switch.

## Data-Terminate and Media-Disable

Data input will continue until data-terminate action is initiated. For the paper tape reader, the ETX (003g) code is read off the tape, decoded (G0DETXX), and sent to the computer as the last character of the message. ETX clears DRE with the last shift pulse and unconditionally sets the input echo enable, F1IEE, flip-flop.

The next and last character received is the DC3 (023g). This is decoded as G0DDC3, and used to clear the ICB flip-flop which enables the end-of-record interrupt. It is not sent to the computer, since DRE is cleared, but is echoed back to the Teletype where it is decoded as the X-OFF function to turn off the reader. DC3 also clears the IEE flip-flop, after echoing.

For the keyboard, the operator simply strikes the carriage return key which inputs the CR (015g) code and generates G0DICR. ICR clears both DRE and IEE with the last shift pulse. DRE is ANDed with IEE to enable G0ICBC which clears ICB. ICB then enables the end-of-record interrupt.

## Echo-Net

The input logic is implemented so that the input message is sampled by SDO, via the N0DPTY gate, as it is shifted into the first stage of the receive register. SDO transmits this message back to the Teletype, provided the enable echo envelope flip-flop, F1REOE, is set.

The echo-net option requires that the clear side of the EOE flip-flop be connected to +3.6V. This enables EOE to set during MSG or IME time (G0RTEE disabled), provided the start echo envelope gate, G0RSEE, is enabled.

To enable SEE, the following conditions must be met simultaneously:

1. The OCB flip-flop must be cleared and the CBR flip-flop must be set. This will always be true during an input function.
2. The IEE flip-flop must be set. This will only occur as a result of ETX, or STX ANDed with OPR.
3. D1RSTR must be disabled (0V). This will occur at the same time that the receive register is preset.

When the preceding conditions are true, EOE will set, clearing the delay counter and enabling SDO. Since F1DPTY is always preset to clear, the first echo bit will always be a start bit (0V out of SDO). Echo transmission will continue until G0RTEE is enabled as a result of MSG and IME. At this time, EOE will clear, disabling SDO which then echoes the stop bits.

# PAPER TAPE READER CONTROL MODULE

The 4202A010 control module is used to control, monitor, and operate Model 4212C or 4213C Paper Tape Readers. Paper tape input operations are performed in a record format. The paper tape readers are free-running devices that are started by the program and stopped when the End of Record code is detected by the buffer.

## INTERRUPTS

Paper tape readers require two interrupts. A Data Exchange interrupt is used to signal the processor each time a character is available for input. An End of Record interrupt is required to signal the processor that the record has been completed. The End of Record interrupt is caused by the DC3 code on the tape. The Data Exchange interrupt is normally non-inhibitible and the End of Record is normally an inhibitible interrupt.

## RECORD FORMAT

The incoming record will contain the following data in the format and sequence shown below.

<u>Name</u>	<u>Octal Code</u>	<u>Function</u>
NUL		Tape Leader
OPR-DC1*	021	Device Enable Code
STX	002	Start of Text Code
TEXT		
↓		
TEXT		
ETX	003	End of Text Code
DC3	023	Device Disable Code
CR	015	Inter-Record Code
LF	012	Inter-Record Code
DC4	024	Inter-Record Code

\* The DC1 code is not punched in the tape. It is sent from the A Register by the OPR instruction.

## SPECIAL CHARACTER DEFINITIONS

As shown in the record format, several special characters are recorded in the tape record. These special characters are defined below.

- NUL - The NUL code is used as tape leader and does not get read into memory.
- DC1 - The Device Control 1 code serves no functional purpose on paper tape input.
- STX - The Start of Text code enables the Data Exchange Ready line which means data is to follow.
- ETX - The End of Text code serves no functional purpose on paper tape input. The ETX code is read into memory.

- DC3 - The Device Control 3 code disables or turns off the reader and generates the End of Record signal. The DC3 code must follow the ETX code.

After the DC3 code is detected, the reader will stop, ready to read the next character which is the CR code. When the next OPR instruction is executed, all characters preceding the next STX are ignored by the I/O Buffer and, therefore, serve only as inter-record codes on paper tape input.

- CR-LF-DC4 - Serve as inter-record codes.

## ALARMS

The following alarm indications can be detected by using properly encoded JNE instructions.

- Alarm 1 - Device off-line.
- Alarm 2 - Power off, out of tape, or broken tape.
- Alarm 3 - Not used.
- Alarm 4 - Parity or timing error.

Parity is checked as each character is received.

## CONTROL LOGIC DESCRIPTION

The 4212C and 4213C paper tape readers are capable of reading 8-channel (maximum) tape at speeds of either 100 or 200 characters per second, depending on the model. Characters are read as a result of the IN command. One character at a time is gated from the tape to the A Register of the AU each time a character passes over the photo-diodes of the reader device. This action continues until the media-disable code (DC3) is detected. The ETX code that precedes the DC3 code is read into memory as data. Tape movement ceases so that the frame following the DC3 code is positioned for reading.

The basic IN command is discussed earlier in this section. The 4212C and 4213C readers are fully described elsewhere in this manual. The following text is devoted to the logic and timing of the reader input cycle, referenced to the simplified block diagram of Fig. PTR. 1 and to the logic schematics of G-E Drawing 68C972544.

Initially, the channel must be ready (not busy). The channel busy flip-flop, F1CCBY, is normally cleared (reset) by the media-disable code (DC3) from the previous input operation. The Abort instruction will also clear the channel busy condition.

A channel test is made by the program via the JNR instruction. If the channel is not busy, the test line 1 gate, G0ATL1, is enabled and the program will step to the next sequential instruction:

$$G0ATL1 = \overline{CBY} \cdot \overline{DEN} \cdot K12 \cdot ADD$$
$$ADD = K0$$

$\overline{\text{DEN}}$  = No Manual Demand.

A demand for an interrupt may also be manually initiated by momentarily depressing the READ pushbutton on the paper tape reader device. Flip-flop F1DDEN sets, the channel busy flip-flop resets, and a channel ready interrupt occurs.

$$\text{F1DENV} = \text{B1DENV} \cdot \text{CLK}$$

$$\text{F1DDEN} = \text{F1DENV} \cdot \text{CLK}$$

$$\text{G0DRCB} = \text{F1DDEN} \cdot \overline{\text{F1DENV}} \cdot \text{CLK}$$

$$\text{G0ACRI} = \overline{\text{F1CCBY}} \cdot \text{F1DDEN}$$

### Media-Enable

The next sequential instruction is OPERATE, addressed to the input channel. The DC1 code (021g), if present in the A Register, is disregarded by the reader control logic. OPR sets the input channel busy flip-flop,

F1CCBY; clears the data ready enable flip-flop, F1CDRE; sets the clutch flip-flop, F1CCLH, if the device is on-line; and clears test line 1. The program is released to await an interrupt.

$$\text{F1CCBY} = \text{OPR} \cdot \text{PHB} \cdot \text{ADD}$$

$$\text{F1CCLH} = \text{CBY} \cdot \text{ONL}$$

### Data-Enable

Relay driver D0CCLH activates the tape feed mechanism. The sprocket hole associated with each frame of information is used as one of the timing signals. All inter-record codes prior to the STX code are gated to the data register but are discarded because only the STX code can set the data ready enable flip-flop which is necessary to generate a data exchange interrupt.

When the STX code passes over the photo diodes of the reader, its sprocket signal, RT1 at zero volts, sets

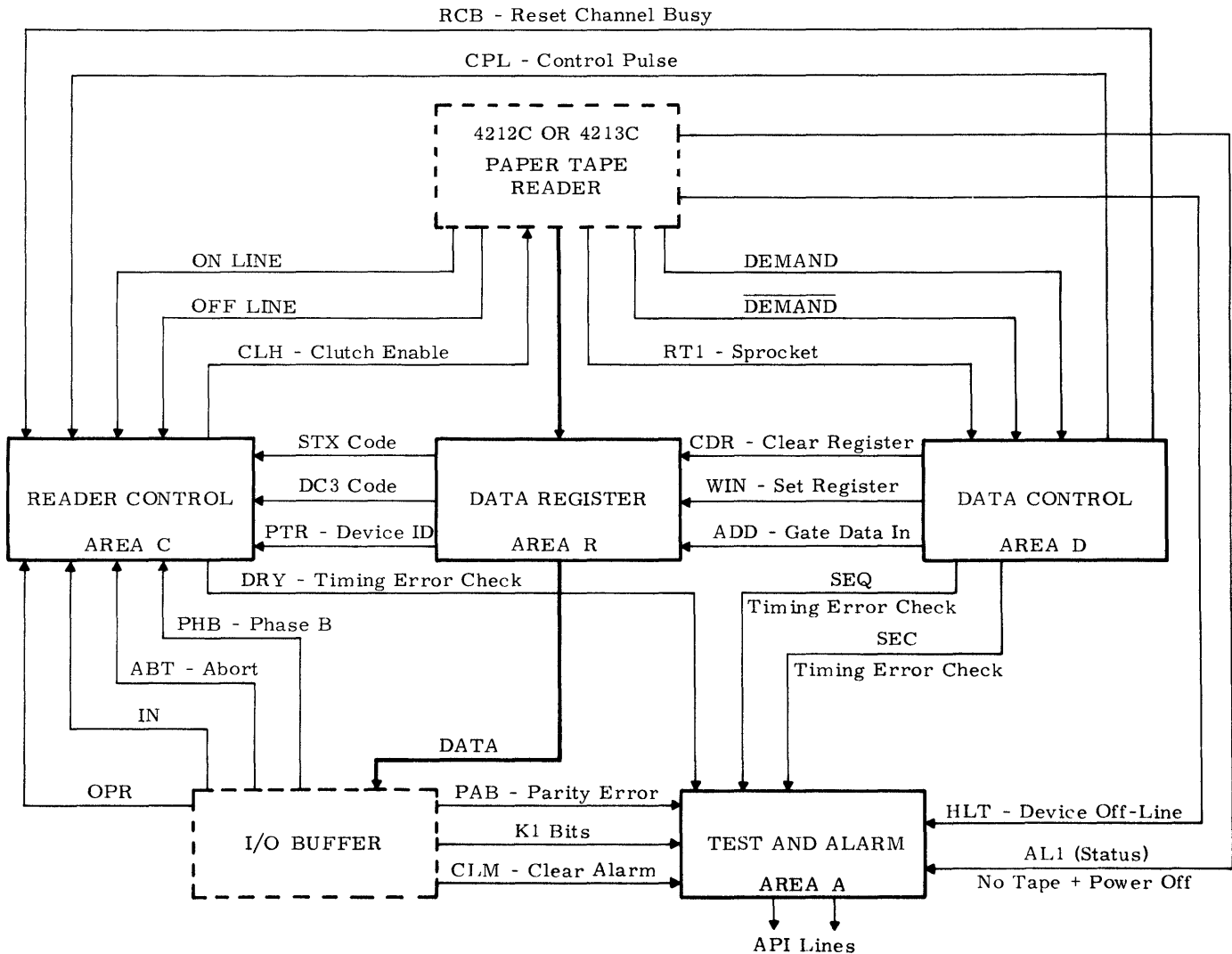


Fig. PTR. 1. Simplified Block Diagram - Paper Tape Reader Control

sequence flip-flop F1DSEQ. Sequence control flip-flop F1DSEC sets at the following clock pulse and the data register is cleared. Window driver D1DWIN is enabled with the setting of F1DSEQ.

$$\begin{aligned} F1DSEQ &= RT1 \cdot F1CCLH \cdot CLK \\ F1DSEC &= F1DSEQ \cdot CLK \\ D1DWIN &= F1DSEQ \\ D0DCDR &= CLK \cdot F1DSEQ \cdot \overline{F1DSEC} \end{aligned}$$

The STX code (002<sub>g</sub>) is gated to the data register.

$$DR \text{ SET} = CODE \cdot D1DWIN.$$

The STX code is decoded by gate G0CSTX and the signal is inverted by N1CSTX. The data ready enable flip-flop F1CDRE sets, indicating that the next data received from the reader will be meaningful.

$$\begin{aligned} F1CDRE &= N1CSTX \cdot F1CCBY \cdot N1DCPL \\ N1DCPL &= F1DSEC \cdot \overline{F1DSEQ} \cdot CLK \end{aligned}$$

Data ready flip-flop F1CDRY sets at the next control pulse, enabling data exchange interrupt gate G0ADEI. By this time the first data character has set the data register.

$$G0ADEI = F1CDRY \cdot \overline{F1DSEC}.$$

The program is interrupted and either an IN command is executed or a TIM operation is started. The data is transferred from the data register, through the I/O Buffer, to the AU. Flip-flop F1CDRY is cleared to await the next character.

Each data character, including the ETX code, is transferred to memory in a like manner, i. e., an interrupt and an IN command are required for each character.

### Media-Disable

When the DC3 code (023<sub>g</sub>) has been detected by the reader and is in the data register, it is decoded by gate G0CDC3. Channel busy flip-flop, F1CCBY, and clutch flip-flop, F1CCLH, are cleared. With F1CCBY cleared, gate G0ACRI is enabled and its output goes from 3.6V to 0V. This signal is recognized by the AU as the End of Record interrupt.

### PARITY ERROR CHECK

In addition to the seven data characters on the paper tape, there is an even parity bit. This bit, like data, is gated to the data register and subsequently to the I/O Buffer. If the parity checking circuitry of the I/O Buffer detects a parity error, signal D0PPAB will be 3.6V. Alarm gate G0AALM is enabled and the alarm flip-flop, F1AALM, is set.

$$G0AALM = \overline{D0PPAB} \cdot N1RPTR \cdot N1COIN.$$

### TIMING ERROR CHECK

Alarm gate G0AALM is also enabled if a timing error has occurred.

$$G0AALM = \overline{SEC} \cdot SEQ \cdot DRY.$$

If, for any reason, the AU has not serviced the Data Ready interrupt, the alarm condition can indicate that a data character has been lost.

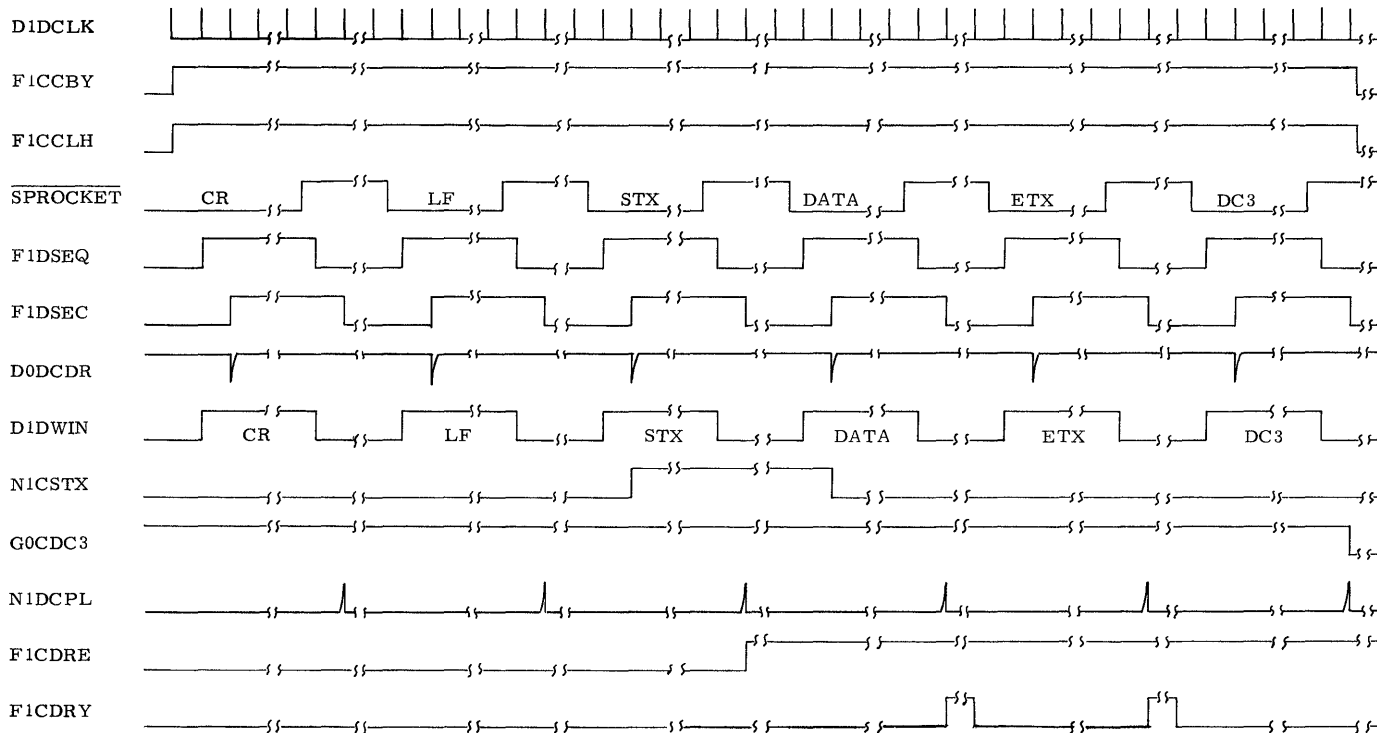


Fig. PTR. 2. Timing Control Diagram

# CARD READER CONTROL MODULE

The 4202A010 Control Module is used to control, monitor, and operate Model 4244C Card Readers and is used in conjunction with the Model 4202A/B Input/Output Buffer. The card reader is a unit record device. Therefore, each 80 column card represents one record of information. The card reader is operated on a demand basis, i. e., each OPR instruction causes only one card feed cycle.

## INTERRUPTS

Card readers require two interrupts. A data exchange interrupt signals the computer each time a character is available for input. An end-of-record interrupt is required to inform the computer when the record has been completed. The end-of-record interrupt is generated when the card reader makes a light check at about column 84. The data exchange interrupt is normally non-inhibitible while the end-of-record is usually an inhibitible interrupt.

## TRANSFER FORMAT

Data from the card reader enters the central processor as illustrated in Fig. CR.1. Row 9 is transferred to the least significant bit of the A-register and row 12 goes to the A<sub>11</sub> position. When the TIM function is used, row 9 goes to either bit position 0 or 12, depending on the character packing mode.

## PUNCHED CARD CHARACTER BIT CODES

A definition of card codes for ASCII characters has not as yet been determined by the American Standards Association. On an interim basis, the General Electric Company recognizes the following minimal subset. Although irrelevant in a discussion of the control module, the information is given for convenience.

Character	Punches	Character	Punches
A	12-1	X	0-7
B	12-2	Y	0-8
C	12-3	Z	0-9
D	12-4	\$	11-8-3
E	12-5	(	0-8-4
F	12-6	)	12-8-4
G	12-7	*	11-8-4
H	12-8	+	12
I	12-9	,	0-8-3
J	11-1	-	11
K	11-2	.	12-8-3
L	11-3	/	0-1
M	11-4	0	0
N	11-5	1	1
O	11-6	2	2
P	11-7	3	3
Q	11-8	4	4
R	11-9	5	5
S	0-2	6	6
T	0-3	7	7
U	0-4	8	8
V	0-5	9	9
W	0-6	=	8-3

## SPECIAL CHARACTERS

The card reader and control module do not recognize any special character codes. The OPR instruction initiates the operation that causes one card feed cycle. When the OPR instruction is executed, the contents of the A-register can be any code.

## ERROR ALARMS

Each of four OR inputs of gate G0ATL2 is used to sense a particular alarm condition. The following conditions can be detected by using a properly encoded JNE instruction.

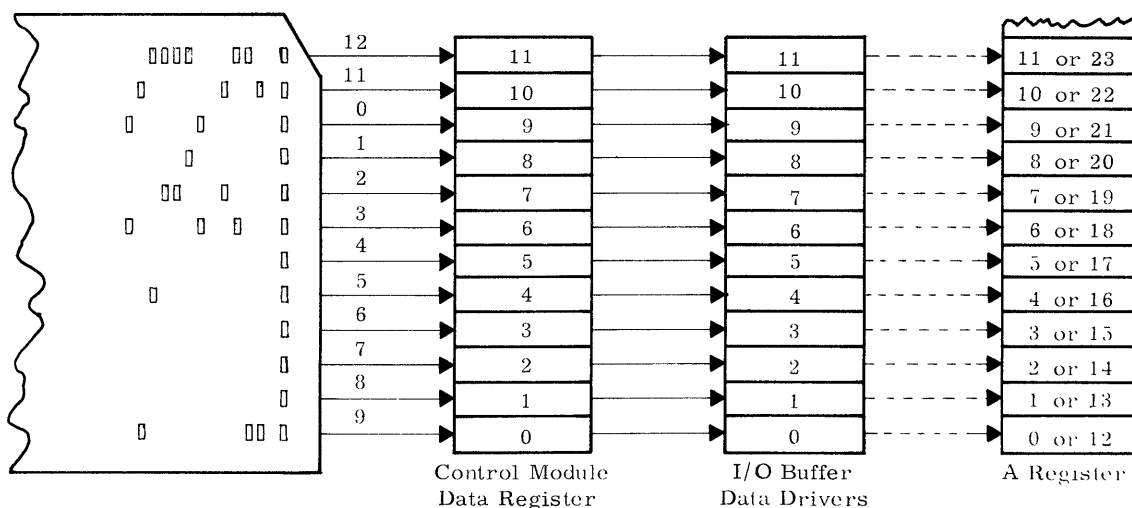


Fig. CR.1. Data Transfer Format

- Alarm 1 HALT key activated or device off-line
- Alarm 2 Card jam, feed failure, power off, or light check failure (bad photocell or lamp out).
- Alarm 3 Hopper empty or stacker full
- Alarm 4 Timing error

### CONTROL LOGIC DESCRIPTION

The 4244C card readers are capable of reading 80 column, 12 row cards at speeds of 100, 200, or 300 cards per minute, depending on the particular model. Characters are read as a result of IN commands; one is required for each of the 80 characters. Each time a character passes over the photocells of the reader device, it is gated from the punched card to the A-

register of AU. This action continues until the entire card has passed over the photocells and the end of the card is detected. Continuous card reading is made possible by properly spacing the OPR instructions in the program.

Model 4244C card readers are fully described elsewhere in this manual. The following text is devoted to the logic and timing of the reader input cycle, referenced to the simplified block diagram of Fig. CR.2, the timing diagram of Fig. CR.3, and to the logic schematics of GE drawing 68C972544.

Initially, the channel must be ready (not busy). The channel busy flip-flop, F1CCBY, normally had been cleared at the end of the previous record. The ABT instruction will also clear the channel busy condition.

A channel test is made by the program via the JNR instruction. If the channel is not busy, the test line 1

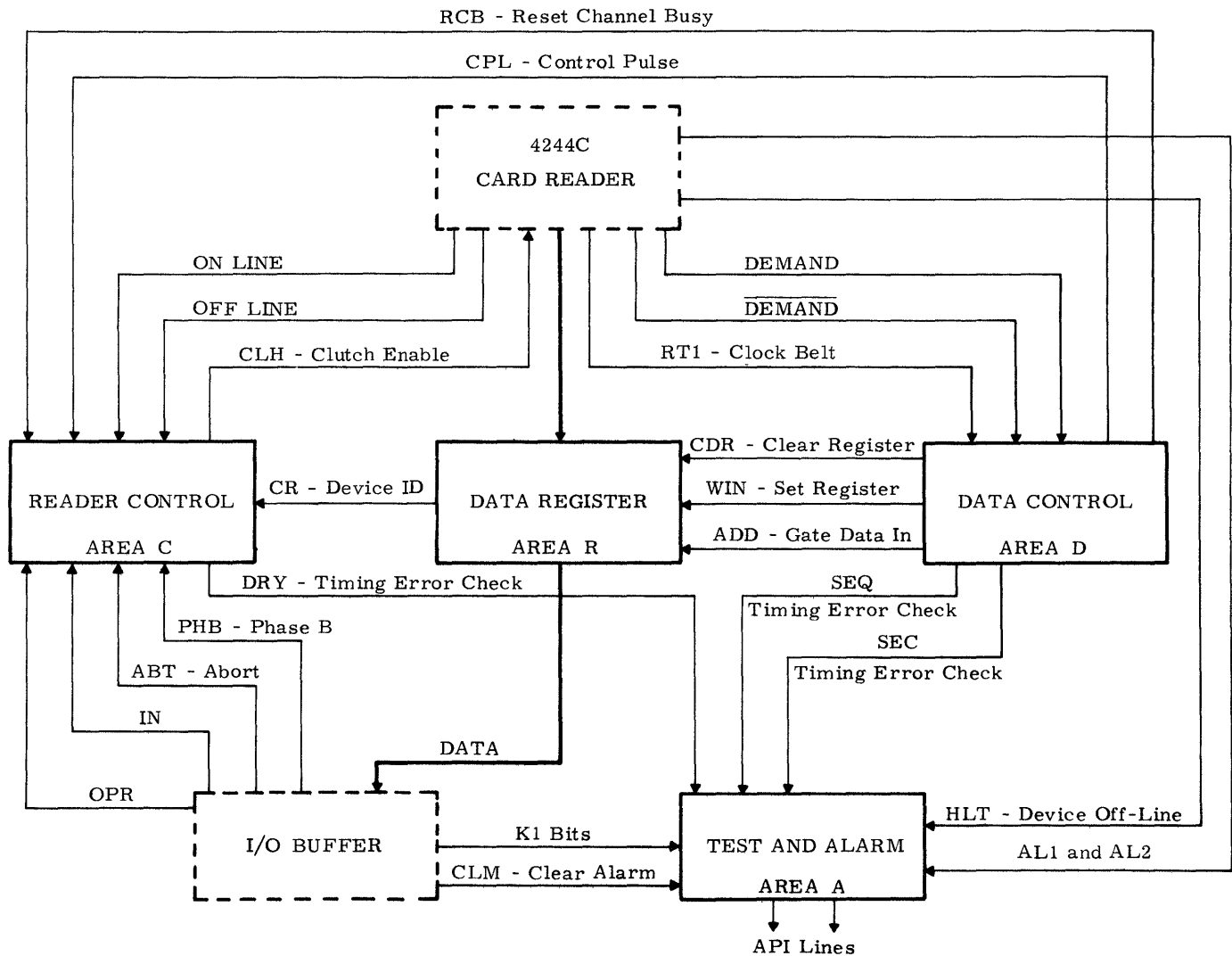


Fig. CR.2. Simplified Block Diagram - Card Reader Control

gate, G0ATL1, is enabled and the program will step to the next sequential instruction.

$$G0ATL1 = \overline{CBY} \cdot \overline{DEN} \cdot K12 \cdot ADD$$

$\overline{F1DDEN}$  = No manual or prior demand

A demand for an interrupt can be manually initiated by momentarily depressing the DEMAND pushbutton on the card reader device. Flip - flop F1DDEN sets, the channel busy flip - flop resets, and a channel ready interrupt occurs.

$$F1DENV = B1DENV \cdot CLK$$

$$F1DDEN = F1DENV \cdot CLK$$

$$G0DRCB = F1DDEN \cdot \overline{F1DENV} \cdot CLK$$

$$G0ACRI = \overline{F1CCBY} \cdot F1DDEN$$

### Media Enable

The next sequential instruction is OPR, addressed to the input channel. Any code that may exist in the A-register is disregarded by the reader control logic. OPR sets the input channel busy flip - flop, F1CCBY, which sets clutch flip - flop F1CCLH, if the device is on-line, and clears test line 1. The program is released to await an interrupt.

$$F1CCBY = OPR \cdot PHB \cdot ADD$$

$$F1CCLH = CBY \cdot ONL$$

### Data Enable

Relay driver D0CCLH activates the reader's card feed mechanism. The timing hole on the reader's clock belt that is associated with each data column generates a signal (D0TCLK) that is used as one of the timing signals (RT1) in the reader control logic.

When the first column passes over the photocells of the reader, its clock signal, at zero volts, sets sequence flip - flop F1DSEC sets at the following control circuit clock pulse and the data register is cleared. Window driver D1DWIN is enabled with the setting of F1DSEQ.

$$F1DSEQ = D0TCLK \cdot F1CCLH \cdot CLK$$

$$F1DSEC = F1DSEQ \cdot CLK$$

$$D1DWIN = F1DSEQ$$

$$D0DCDR = CLK \cdot F1DSEQ \cdot \overline{F1DSEC}$$

The first data character is gated to and sets the data register.

$$DR SET = DATA \cdot D1DWIN$$

Data ready enable flip - flop F1CDRE is set continuously, the only condition being that the card reader device is connected to the control module. Data ready flip - flop F1CDRY sets with a control pulse that is generated as a result of the space between clock holes on the reader clock belt being sensed. Data exchange

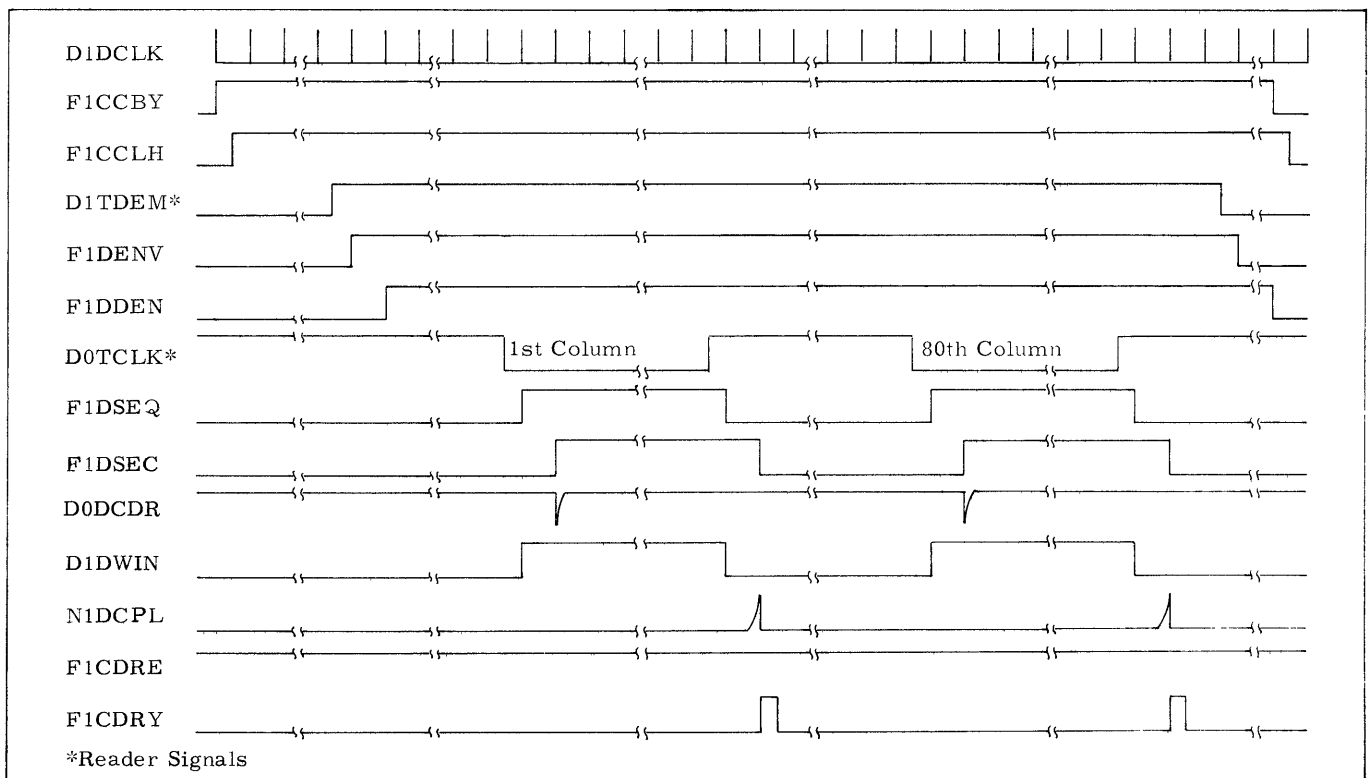


Fig. CR.3. Timing Control Diagram - From Advent of OPR



interrupt gate G0ADEI is enabled.

$$F1CDRY = F1CDRE \cdot N1DCPL$$

$$N1DCPL = F1DSEQ \cdot \overline{F1DSEC} \cdot CLK$$

$$G0ADEI = F1CDRY \cdot \overline{F1DSEC}$$

The computer responds to the interrupt with an IN command. The data is transferred from the data register, through the I/O Buffer, to the AU. Flip-flop F1CDRY is cleared to await the next character. All 80 characters are transferred to memory in a like manner, i.e., an IN command is required for each character.

### Media Disable

At the end of a card (approximately column 84), the on-line demand signal D1TDEM goes to zero volts. This condition clears the channel busy flip-flop, clears the clutch flip-flop, and generates an end-of-record interrupt. The computer must respond with another OPR before the next card will feed.

### TIMING ERROR CHECK

If, for any reason, the AU fails to service the data ready interrupt, alarm gate G0AALM is enabled, indicating that a timing error has occurred and that a data character has been lost.

$$G0AALM = \overline{SEC} \cdot SEQ \cdot DRY$$

# TABLE OF CONTENTS

## COMMUNICATIONS COUPLER 4306A

### INTRODUCTION

INT

#### GENERAL DESCRIPTION

Function

Transmission Sequence

Reception Sequence

Errors

Options

### OPERATION

OP

#### LOGIC ORGANIZATION

AU Interface

A Register Data Lines

Outputs to A Register

Timing Sources

Baud-Rate Counter

Receive/Transmit Control

Shift Register

Data To and From External Devices

#### TRANSMISSION OPERATION

Baud-Rate Counter

OUT Instruction

#### RECEPTION OPERATION

#### ERROR DETECTION

Parity Error

Carrier Disappears

New Word Interrupts Reception

DSR False While Receiving

DSR False While Transmitting

OUT While Receiving

OUT While Serializing

IN While Receiving

IN While Serializing

## LIST OF ILLUSTRATIONS

Fig. INT.1 Coupler Block Diagram

INT

Fig. INT.2 A Register and S Register Formats

Fig. OP.1 Baud-Rate Counter Timing

OP

Fig. OP.2 Generation of Start, Parity & Stop Bits

Fig. OP.3 Complete Reception Sequence

# INTRODUCTION

## GENERAL DESCRIPTION

The Model 4306A Communications Coupler, operating in a simplex or half-duplex mode, serves the purpose of adapting the standard GE/PAC\* input/output control lines (GEN II Commands) to the needs of Bell data sets (Models 103F, 201A, 201B, 202C and 202D) or other equivalent communications modems. It enables the GE/PAC central processor, with either 4DP4020A, or 4DP4022A or 4DP4025A Arithmetic Units, to effect one-way or two-way communication with a remotely situated modem. Full duplex operation is possible when two simplex Model 4306A modules are employed.

There are nine separate, but inter-related, functions performed by the coupler in the course of one complete, two-way communication with a modem, as listed below:

- On command (OUT, ODL or TOM) from the central processor, receives parallel data in its Shift (S) Register. (For the remainder of this discussion, OUT command will imply that an ODL or a TOM may be used.)
- Serializes the data, adding start, parity and stop bits.
- Transfers data, least significant bit first, to a communications modem.
- Receives serial data from the modem, least significant bit first.
- Parallelizes the data in the S register.
- On command (IN, IDL or TIM) transfers data to the central processor. (Henceforth IN also implies IDL or TIM.)
- On command (JNR) from the central processor, indicates its Ready status.
- Provides signals for the Automatic Program Interrupt when the coupler transits between alternate Ready/Busy states, one each for transmit and receive.
- On command (JNE) from the central processor, indicates the presence of an error in the incoming data.

Reference to Fig. INT. 1, Coupler Block Diagram, will amplify the above information by providing data and control paths in simplified form.

### Function

The Model 4306A functions as an interface between a GE/PAC central processor and a telephone communications Modem (modulator/demodulator). Optionally, one-mile line drivers and line receivers may be used instead of modems. An OUT command gates the contents of the A Register or the adder to the S Register

of the coupler, where it is received in parallel and retained until a ready signal is received from the modem at the near end of the transmission line. The data is then serialized and fed to the modem with start, parity and stop bits added. After conditioning and modulating the string of data and control bits, the near-end modem transmits it to the modem at the far end of the communications line.

Conversely, when a signal travels in the opposite direction, it is demodulated by the near-end modem into digital information and fed in a serial string to the S Register of the coupler. Here it is parallelized, parity is checked, and generation of an automatic program interrupt causes the execution of an IN command which reads the word in the S Register of the coupler into the A Register or the adder, of the Arithmetic Unit. It is apparent, at this point, that the "half-duplex" mode of operation is necessitated by the fact that both serializing and parallelizing are accomplished by the shift register of the communications coupler, which can perform only one of these functions at a time. Only the use of two simplex 4306A modules can provide full duplex operation.

The communications coupler is addressed by two signals, one derived from the three K3 lines and the other from the three K2 lines. Data lines, clock lines, initialize and clear alarms, and six control lines comprise the remainder of the interface between the AU and the coupler. The control lines are made up of the various values of the "S" bits which determine the nature of the GEN II commands that control the operation of the coupler.

- ACT (S = 1) - When the coupler's GOCODL (Sheet 13) is true, ACT forces it false for 5 to 8 microseconds, and then true again, causing an interrupt which generates an OUT in instruction. When GOCIDL (Sh. 12) is true, ACT forces it false for 5 to 8 microseconds, and then true again, causing an interrupt which permits generation of an IN instruction.
- ABORT (S = 3) - This command terminates the coupler's "Request-to-Send" signal to the modem.
- OUT (S = 4) - This instruction transfers data from the A Register to the S Register, initiates serialization and transmission to the modem and forces the Coupler-Busy signal true.
- IN (S = 5) - This instruction transfers parallel data from the S Register to the A Register and forces the "Ready-to-Read" signal false.
- JNR (S = 6) - This instruction is not used directly in the coupler, but a test line is made available to the AU to sample the ready status of the coupler (0V = Coupler-Ready).

\* Registered Trademark of General Electric Company

- JNE (S = 7) - An error line is made available to the AU and the API modules. A zero volt signal indicates the presence of one or more of several possible types of error. The error flip-flop is reset by the JNE command (also by initialize or clear alarm).

### Transmission Sequence

As a condition for transmitting a word to the coupler, the program must first ascertain if the coupler is ready to accept it. Either a JNR instruction or an ACT (in conjunction with "Ready-to-Output" interrupt) will accomplish this purpose. An OUT command selects the coupler (K3K2 bits), causes the Coupler-Busy signals to go true, and then initiates the transmission of a word of data (previously placed in the A Register) to the communications coupler.

The shift register of the Model 4306A can accept as few as five bits or as many as twenty-four, depending on option. As indicated by Fig. INT. 2, at the moment of transmission of the data to the near-end modem, the appending of start, parity and stop bits is a function of the 27-flip-flop S Register. The coupler then transfers one bit at a time to the modem for modulation and transmission, beginning with the start bit and continuing with the data bits (least significant first), the parity bit and finally the stop bits (one to three, according to option). Transmission is in synchronism with the transmit/receive clock (either coupler-local or modem supplied, according to option).

Following transmission of the last stop bit to the modem, the Coupler-Busy signals go false and the data output to the modem is held in a "mark" condition (true) until the next OUT initiates a new transmission sequence.

#### NOTE

The Modem-Clear-To-Send signal must be true for the above transmission sequence to take place. This condition is initially brought about by the first OUT instruction, which activates a Request-To-Send line to the modem. The modem's response to this signal is a Clear-To-Send signal which prevails so long as Request is held true. Termination of Request-To-Send is accomplished by means of a ABORT command (S = 3), which is activated no less than 1 ms after transfer of the last stop bit of the last word to the modem.

### Reception Sequence

As a condition for reception of a word from a modem, the Coupler-Busy signal must be false. Incoming data is not permitted to interrupt a current serializing sequence or to destroy unread data. If this happened, the new data would be lost and the error flip-flop would be set. Reception begins when the data line from the modem transits from true to false (from Mark to Space). This is occasioned by reception of the Start bit, signifying that data is to follow. The data line must remain

false for at least half a bit time for the Start bit to be shifted into the parallelizer by the transmit/receive clock, thus ensuring that a valid start bit and not noise has been received.

Data bits are clocked serially into the coupler until the start bit has reached the last position in the shift register, at which time the "Ready-to-Read" signal goes true. This serves to inform the central processor (by way of an interrupt or an ACT) that a word has been assembled in the coupler. Coincidentally, parity is checked on the assembled data bits, and the error flip-flop is set if a discrepancy exists. In the absence of a parity error, the execution of an IN instruction (S = 5) causes the parallel transfer of the data in the S Register to the A Register, and causes the "Ready-to-Read" signal to go false. The format in the S Register, immediately prior to read-in, is identical to that shown in Fig. INT. 2.

### Errors

An error test line is made available to the processor for sampling by the JNE instruction. The coupler may be jumpered to check or generate either odd or even parity. Among the list of error conditions tabulated below, those preceded by a single asterisk should be followed by a request to re-transmit. For those with a double asterisk, instructions are blocked and the conditions described are not permitted to occur.

- \* Word/character parity error, either odd or even.
- \* Carrier signal disappears before a complete word/character is received and assembled.
- \* A new word/character begins to arrive before the word/character previously received is read into the AU (the new word is blocked).
- \* Incoming data is not at "mark" when a complete word/character has been received. (Phone line opened.)
- \* Data-Set-Ready signal goes false while receiving.
- Data-Set-Ready signal goes false while transmitting.
- An OUT command is executed while the Data-Set-Ready signal is false.
- \*\* An OUT instruction is executed while receiving.
- \*\* An OUT instruction is executed while serializing.
- \*\* An IN instruction is executed while receiving.
- \*\* An IN instruction is executed while serializing.

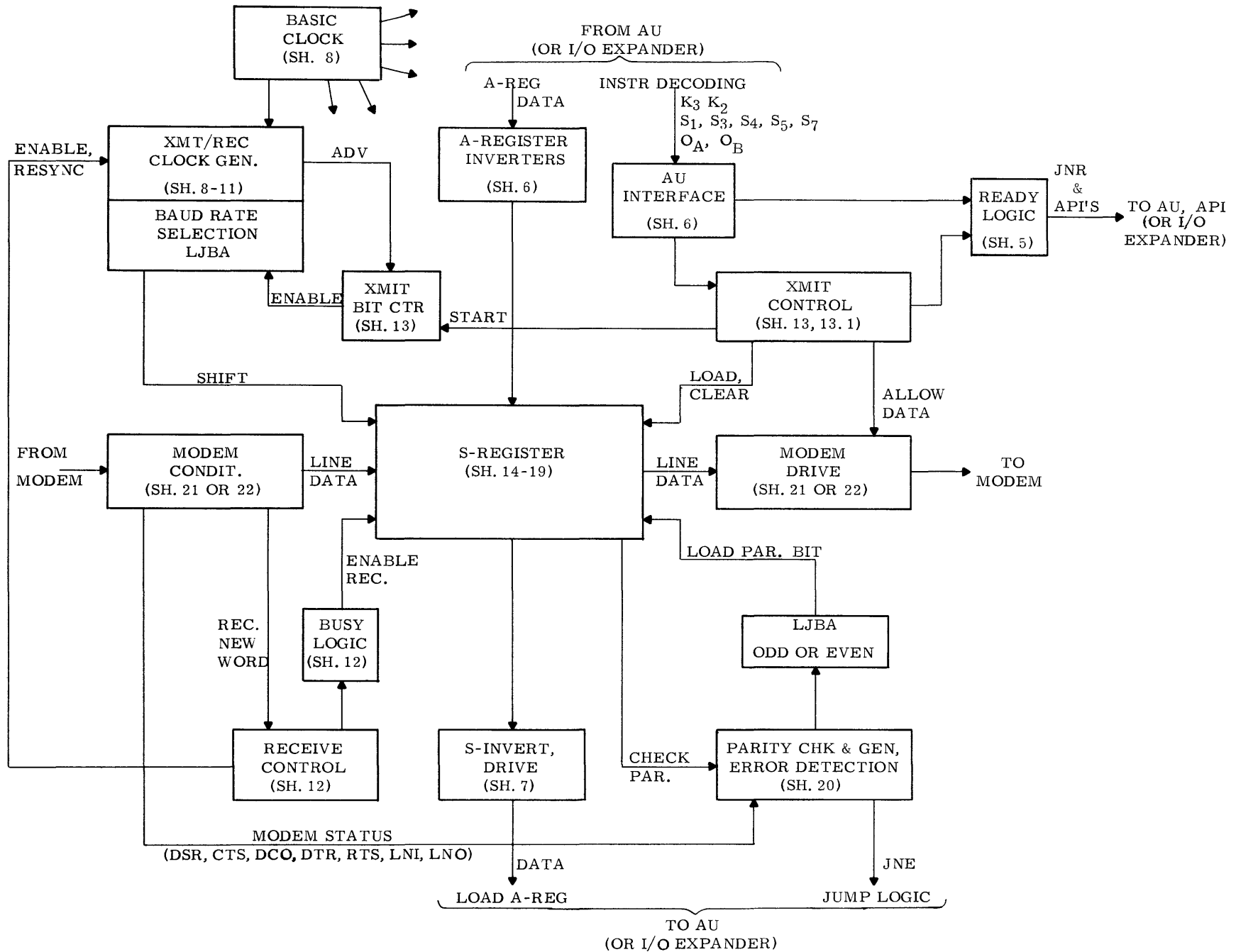


Fig. INT.1. Coupler Block Diagram

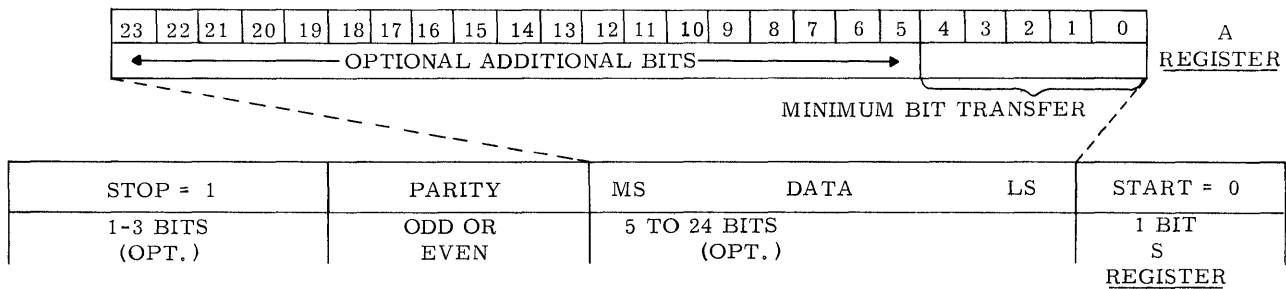


Fig. INT. 2. A Register and S Register Formats.

### Options

The Model 4306A Communications Coupler is available in a number of optional configurations, the different options being designated by the values given to the last four characters in the model number (4DP4306AWXYZ). These represent the number of transferrable data bits per word (W and X), three possible transmit/receive clock options (Y), and the mode of operation (Z). In addition, certain other operational characteristics may be selected and implemented by the use of a plug-in, jumper printed wire board. Among these are odd or even parity, a choice of eight baud rates and one to three stop bits.

- The number of data bits, designated by W and X, are limited by the W values of 0, 1, and 2, plus the fact that no less than five nor more than 24 bits may be used. Thus the W and X values in the model number vary from 4306A05YZ to 4306A24YZ.
- The Y values, which refer to Transmit/Receive clock, are available as shown below:

- Y = 0 Asynchronous operation, local clock (103, 202)
- Y = 1 Asynchronous operation with synchronous modem, local clock (201)

Y = 2 Asynchronous operation with synchronous modem, modem clock (201)

Y = 3 One-mile line drivers/receivers (no modems) local clock

- The Z values, referring to the mode of operation, appear in four configurations:

Z = 0 2-wire, half-duplex operation

Z = 1 4-wire, half-duplex operation

Z = 2 2-wire, simplex transmission (or transmission half of 4-wire, full-operation)

Z = 3 2-wire, simplex reception (or reception half of 4-wire, full-duplex operation)

As an example, a communications coupler bearing the model number 4306A2120, would have the following characteristics:

W = 2 2X data bits

X = 1 W1 data bits, or 21 data bits

Y = 2 Asynchronous operation with 201-type modem, modem clock

Z = 0 2-wire, half-duplex operation

# OPERATION

## LOGIC ORGANIZATION

During the ensuing discussion of the operation of the Model 4306A Communications Coupler the logic contained in G-E Drawing No. 68C995981 will be used extensively. It is understood that possible future alterations may change the sheet numbers quoted, but that the circuit paths can easily be traced from the logic mnemonics.

### AU Interface

Clock lines, address lines, control lines and Initialize/Clear Alarm, arriving from the Arithmetic Unit (AU) or the I/O Coupler (at ground level in the true condition) are developed on Sheet 5. With the exception of certain Initialize loads which remain at zero volts, these lines are received by MLEA inverters. A second inversion, using similar elements, combines the desired clock lines ( $\phi A$  and  $\phi B$ ) and the address lines (K3 and K2) with each of the instruction inputs. In the case of the ABORT (S = 3) command, this is accomplished by the delay timer T1A1MS.

### A Register Data Lines

Similarly, data bits A00 thru A23 are brought from the A Register and inverted by MLEA elements. Sheet 6 shows that these bits are received at ground level in the true state, groups them according to option and labels the loads as F1SS00 thru F1SS23, the Shift (S) Register. As noted, this sheet depicts a circuit flow used during a transmission sequence (OUT to a modem) and is not employed by models in which Z = 3 (receive-only models).

### Outputs to A Register

Data moving in the opposite direction during a reception sequence passes through the inverters shown on Sheet 7. Here the signals are received at +5V from the S Register, after parity has been checked, and are inverted to 0V for transfer to the A Register. As with the previous sheet, the elements are grouped according to option and it is noted that models with Z = 2 (transmit only) do not use this logic.

### Timing Sources

It will be seen in following this logic that many elements have alternate inputs and/or outputs, depending on the optional model under consideration. A case in point is clock-enable flip-flop, F1TCLK, shown on Sheet 8. One possible set input is F0MCLK, but is only used in options having a Y value of 2 (synchronous operation with a modem clock). The alternate set input is the baud counter element F1T04K and is used in options employing the local (internal) clock. Each sheet of logic, therefore, incorporates the necessary footnotes to enable the reader to apply those areas of logic which refer to the option under study, and to reject those which do not apply.

Also on Sheet 8 are pictured the Basic Clock, consisting of a temperature-regulated, crystal-controlled clock generator (720KC) and two logic-amplifier elements (D1TADV and D1TOSC), and the Transmit/Receive

Clock, which is another pair of HLAC elements, D1TCLK1 and D1TCLK2, plus expander elements. The basic clock provides the advance pulses for the Baud-Rate Counter, shown on Sheets 9 thru 11, thru its D1TADV output, and, in conjunction with  $\phi A$  and  $\phi B$  provides clear and set signals to the S Register, the transmit control logic and the Preset, Parity and Over-flow flip-flops, thru its D1TOSC output. This output also pulses the modem clock, the First Bit, Error and others.

### Baud-Rate Counter

Jumper board points are provided for selection of the baud (bits per second) rate desired. These points are shown at the top of sheets 9, 10 and 11 with pin pairs lettered between B and L. A table at the bottom of sheet 10 contains combinations of pin pairs which can be jumpered to obtain eight baud rates ranging from 150 bits per second to 5000 bps. (Any baud rate that can be divided an even number of times into the number 360,000 can be obtained by using jumpers.) Note that these three sheets of logic do not apply to any option with a model number in which Y = 2. These options employ a synchronized modem clock, instead of the baud-rate counter which functions as a local, or internal, clock.

### Receive/Transmit Control

Sheet 12 portrays the Busy-Receiving Flip-flop, the First-Bit flip-flop and the Enable-Receive flip-flop, the major control elements involved in a reception sequence. Transmit control elements are pictured on sheets 13 and 13.1. Aside from the jumper board LJBA which determines the number of stop bits employed, sheet 13 pictures a bit-counter composed of five flip-flops which is preset to count data bits in accordance with the WX option selected. This is accomplished by grounding the set AND gates as shown in a table at the bottom of the sheet (a wired option).

The set signal D1ASET (sheet 5) is derived from the OUT instruction and the K3K2 lines which select the coupler. It is pulsed by D1TOSC at Phase B Time. Each element of the bit counter which is not grounded causes the counter to be preset to a degree dictated by its position in the counter. For instance, if the set AND gates were grounded on F1C016 and F1C004, then F1C008, 02 and 01 would be preset. The bit counter would thus start its count at 12 and continue thru 31, the maximum for a 5-element counter, and then drop to zero. Since this adds up to 20 counts, this configuration fills the requirements for Model No. 4306A19YZ, being wired to transmit 19 data bits and a parity bit.

### Shift Register

Sheets 14 thru 19 illustrate the Shift Register's twenty-seven flip-flops plus associated logic and footnotes. These represent the stop bit, F1SSBI, the parity bit, F1SPAR, 24 data bits, F1SS23-00 and the start bit, F1SSTB. A table at the bottom of Sheet 14 illustrates the relative bit positions occupied by the parity bit in each of the allowable options. Associated logic such as the Parity Check/Generate flip-flop, the Error flip-flop

and the Parity Select jumper board are pictured on Sheet 20.

### Data To and From External Devices

While the Communications Coupler deals mainly with modems, it has been mentioned earlier that an option is available in which one-mile line drivers and receivers may constitute the external devices with which the coupler communicates. Sheet 21 depicts the ready signals, data and clock elements which relate to the use of modems. Their purposes are clearly identified in the drawing. Sheet 22, Data In/Out, refers to the use of other than modems as driving and receiving elements. Line connection details are furnished in drawings specifically generated for systems in which such devices are employed.

### TRANSMISSION OPERATION

In the ensuing discussion of a complete transmit sequence, and in the subsequent study of a receive operation, no consideration will be given to the size of the word transferred. This is entirely a function of the bit counter (sheet 13) which has been preset in accordance with the option selected, and its size in no way affects the method by which transfer of data is accomplished.

While it is also true that the selection of one or another bit-per-second option does not change the basic method of transmission, the operation of the baud-rate counter involves so much of the timing and control logic that it warrants special discussion. To this end Fig. OP. 1 presents the timing relationships of the essential elements associated with the baud-rate counter.

#### Baud-Rate Counter

The basic timing is derived from D1TOSC and D1TADV (sheet 8), elements of the 720 KC clock. D1TADV pulses both set and clear sides of the 13 flip-flops which

comprise the counter. D1TOSC arrives at the counter by way of D1TZRO and D1TL0D. D1TZRO pulses the clear sides of the flip-flops, while D1TL0D feeds the set sides. It will be noted by the timing diagram that D1TL0D is of greater duration than D1TZRO, permitting time for loading (pre-setting) the baud-rate counter after D1TZRO (which has cleared it) falls.

Presetting the baud-rate counter for the desired option is accomplished by the same means that are used in pre-setting the bit counter, except that instead of wiring certain AND gates to ground, jumper boards are used. Thus the baud-rate may be readily changed by arranging the jumpers in a new configuration in accordance with the table at the bottom of Sheet 10. Presetting the counter to a low starting figure, requiring a large count to proceed before F1T04K sets, results in a low baud-rate. The longer it takes to run through the count, the fewer bits per second are transmitted. Conversely, starting the count at a high figure requires less time to set 04K, thus giving a high baud rate.

Fig. OP. 1 indicates that the next D1TADV pulse following the 4096-count is missing. This occurs because ADV (sheet 8) requires the term NOTL0D (sheet 9), which is not up during the interval between the fall of D1TZRO and the fall of D1TL0D, thus inhibiting D1TADV. At the next D1TOSC, synchronized by F1TPRE, D1TADV resumes counting since NOTL0D is again present and will remain up until inhibited at the end of the count.

F1TCLK, also shown in Fig. OP. 1, which times the setting of F1CXMT, the Transmit flip-flop (sheet 13.1), is set every other time the count reaches 4096. Since F1TCLK requires F1T04K, and this flip-flop remains clear throughout a full count, F1TCLK must wait for the next count of 4096. F1T04K, on the other hand, is set during every other entire count but can not set

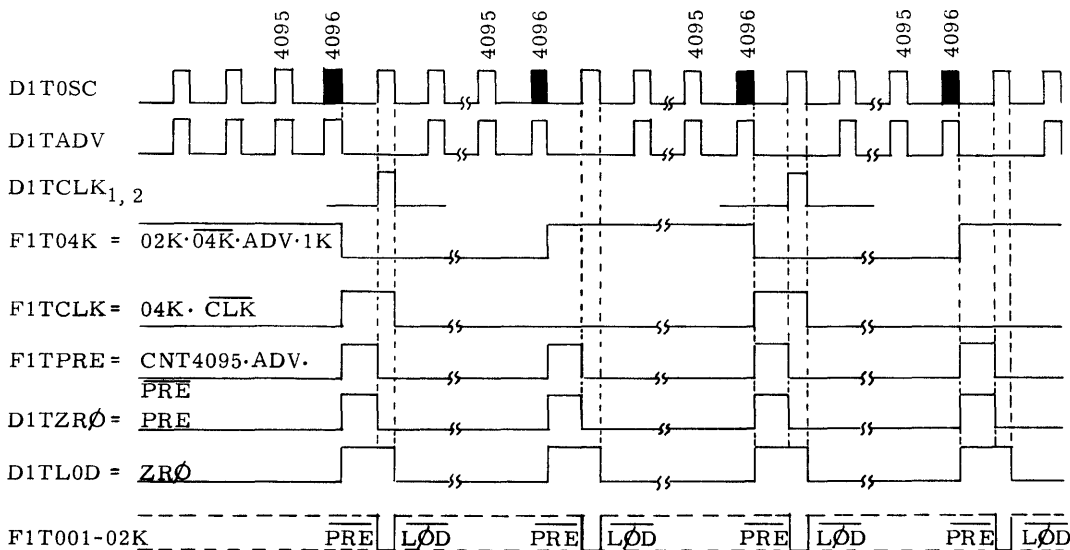


Fig. OP. 1. Baud-Rate Counter Timing



immediately after the baud-rate counter has been cleared (including 04K) because of the absence of D1TADV and F0TCLK at that time.

## OUT Instruction

The OUT instruction may be initiated only after it has been determined that the coupler is ready for a transmission sequence. This is done either by the use of an ACT command (S = 1) which activates an interrupt for an OUT instruction, or by sampling the JNR test line to check the status of G0ARTO (sheet 5). The ACT instruction, arriving in the coupler at 0V = true also selects the coupler with true K3K2 lines and at Phase A time causes G0AACT to emit a true signal to G0CODL (sheet 13). If the other input to this inverter is at +5V, meaning that there is an overflow in the bit counter and N1CHLT is inhibiting transmission, an interrupt is activated. With N1CHLT outputting 0V (F1COFL clear because a transmission is in progress) no interrupt is forced until the end of transmission. The JNR instruction (S = 6) is not used in the coupler logic, but serves to test the status of G0ARTO (Sheet 5, Coupler Ready).

Examination of the inputs to G0ARTO reveals that this ready signal involves not only the status of the coupler, but of the modem as well. Aside from F0CDEL, the inhibiting one-bit-time delay, plus N1CHLT, F0CBRC and F0CFIB, it is necessary that the data set be ready (M1MDSR) before a transmission can take place.

A zero volt signal indicating that the coupler is ready, an interrupt is generated for an OUT instruction. There was, prior to the JNR, a word of data for transmission already in the A Register. The OUT command performs the prime function of transferring this data to the coupler, though it also initiates serialization in the S Register and transmission of data to the modem, and forces the coupler's busy signals true.

Received by N1AS04 at zero volts (sh. 5), the signal is inverted and enters G0AOUT coincidentally with the inverted K3K2 coupler-select signals. After two more inversions D1ACLR is enabled at Phase A by the OUT signal and D1TOSC, an element of the basic 720KC clock shown on sheet 8. D1ACLR is a logic amplifier used to clear the bit counter, the S Register and the Parity Check/Generate flip-flop. Reference to Fig. OP. 2 will clarify the relationship between Phase A-B, the D1TOSC pulses and several other timing elements involved in the transmit sequence.

During Phase B, D1ASET is enabled by the OUT instruction, when a receiving or transmitting condition does not impose the inhibiting +5V signal, G1TROX. It will be seen (sh. 8) that the latter element is derived from the Receiving and Transmitting flip-flops, a further insurance against transferring data while the coupler is busy. D1ASET, now that the S Register has been cleared, loads it in accordance with the data transferred from the A Register. D1ASET merely provides the set pulses for the S Register flip-flops during Phase B time. Whether any particular FF is set or remains clear depends on the value of the corresponding bit in the A Register. The data bits arrive by way of 24

inverters (N1AA23-00) shown on sheet 6, and a logic one in any of them will cause the S Register flip-flop with the corresponding number to be set by a +5V signal.

D1ASET also contributes to loading the bit counter by providing the set pulses for those gates which are not grounded (according to option). In addition, it furnishes +5V signals which may set F1CRTS (sh. 13.1) or F1PERR (sh. 20) and will clear the overflow flip-flop F1COFL (sh. 13). Clearing F1COFL will set the Transmit flip-flop (F1CXMT) (sh. 13.1) when N1CCTS and Clock Enable are present.

The term N1CCTS (Coupler Clear To Send, sh. 13.1) is not to be confused with the Modem Clear To Send (M1MCTS), a related function. N1CCTS can not be enabled without M1MCTS (sh. 21) and F1CRTS (Coupler Request To Send), which, when set, also enables X1MRTS (Sh. 21).

Fig. INT. 2 illustrates the composition of the A Register and the position occupied by the data bits in the S Register, after their parallel transfer. Appending the Start Bit (F1SSTB, sh. 19), at the least significant end of the register, the Parity Bit (F1SPAR, sh. 14), immediately following the most significant bit called for in the option, and from one to three Stop Bits (F1SSBI, sh. 14), is accomplished as the serial transfer from the coupler to the modem progresses. The timing chart in Fig. OP. 2 illustrates the generation of these bits and includes the numbers of the logic sheets containing the terms involved. This chart depicts a minimum option of five data bits, F1SS00-04, and one stop bit, but may be expanded to include the maximums of 24 data bits and three stop bits.

Transmission of the S Register to the modem is synchronized with the Transmit/Receive Clock, D1TCLK<sub>1</sub> (or the modem clock in the Y = 2 option), and when completed F1COFL is set, causing the coupler busy signals to go false and halting transmission. The Clear To Send signal, however, still prevails and will do so until Request To Send, which initiated it, is terminated. This is accomplished by a Special Out command (S = 3, Abort), shown on Sheet 5. The single-shot T1A1MS provides a 1 ms delay before clearing F1CRTS, and disabling N1CCTS.

## RECEPTION OPERATION

As with the transmission sequence, the coupler must have completed its previous cycle in order to successfully receive a word from the modem. Incoming data must not be permitted to interrupt a current serializing sequence or to destroy a previously received word. Instead, if a receive sequence be erroneously initiated, the new word will be lost and the error flip-flop will be set. This will be discussed later under Error Detection.

Reception begins when the data line (M1MLNI, sh. 21) transits from mark to space (true to false). This occurs with reception of the start bit and signifies that data follows. Because of the possibility of noise erroneously triggering a reception sequence the coupler

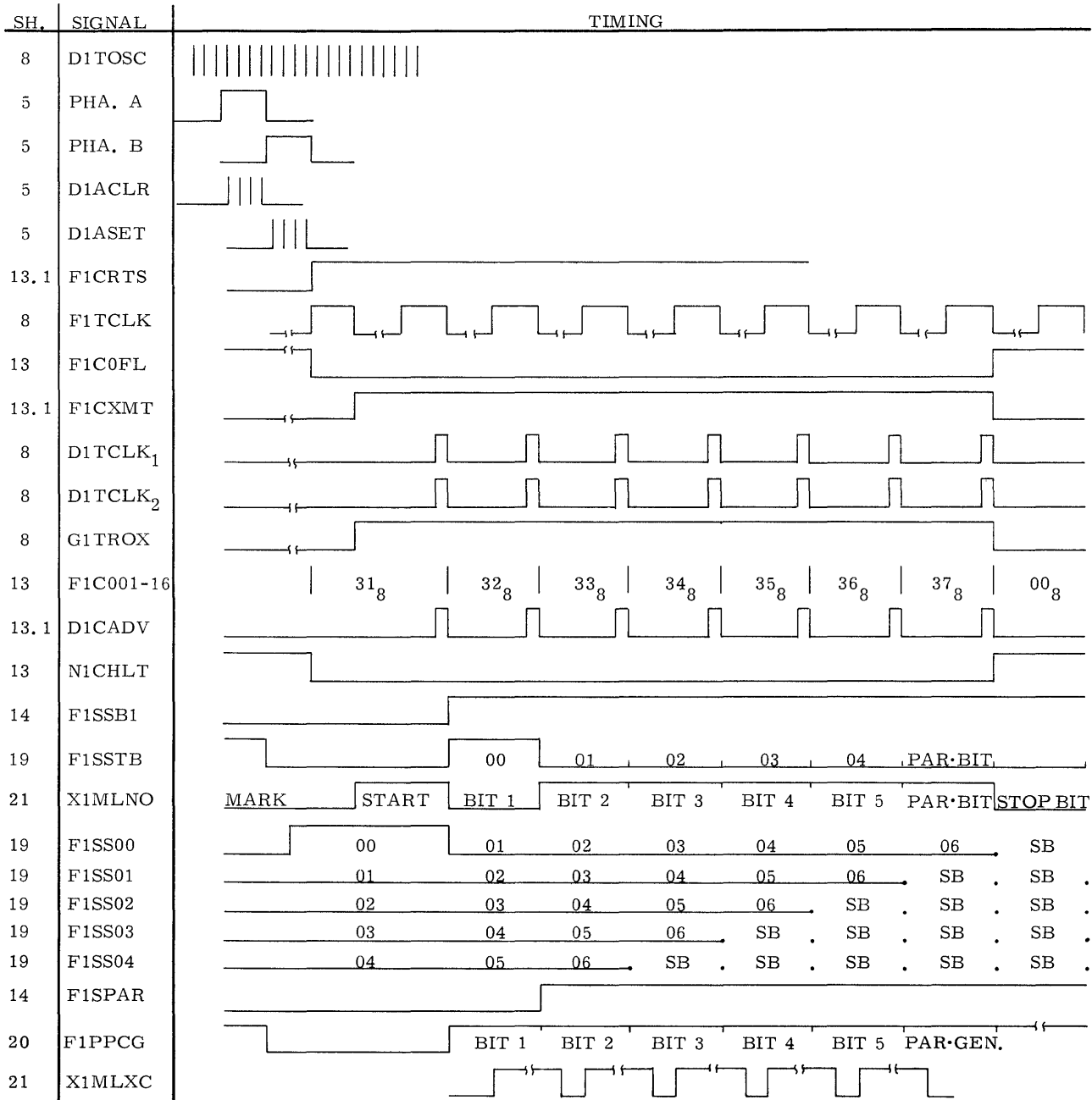


Fig. OP.2. Generation of Start, Parity & Stop Bits

only recognizes a signal which remains at "space" for at least half a bit time, at which time the start bit is shifted into the S Register by the transmit/receive clock (D1TCLK).

As shown on sheet 14 of the logic, the input from M1MLNI enters the S Register through the Stop Bit flip-flop, F1SSBI. This serialized string of bits continues to shift through the S Register from this point (F1SSBI, F1SPAR, and those data positions called out in the

option, from F1SS23 to F1SS00) until the start bit arrives at F1SSTB. At this point, each affected flip-flop (determined by option) will have assumed the value of the corresponding bit from the modem, as transferred through M1MLNI. This is insured by the fact that the transfer has been synchronized with the transmit/receive clock (D1TCLK) either through the modem clock (F1MCLK, sh. 21) or through the baud-rate counter synchronize-flip-flop (F1TSYN, sh. 11) which controls the internal clock through the preset-enable

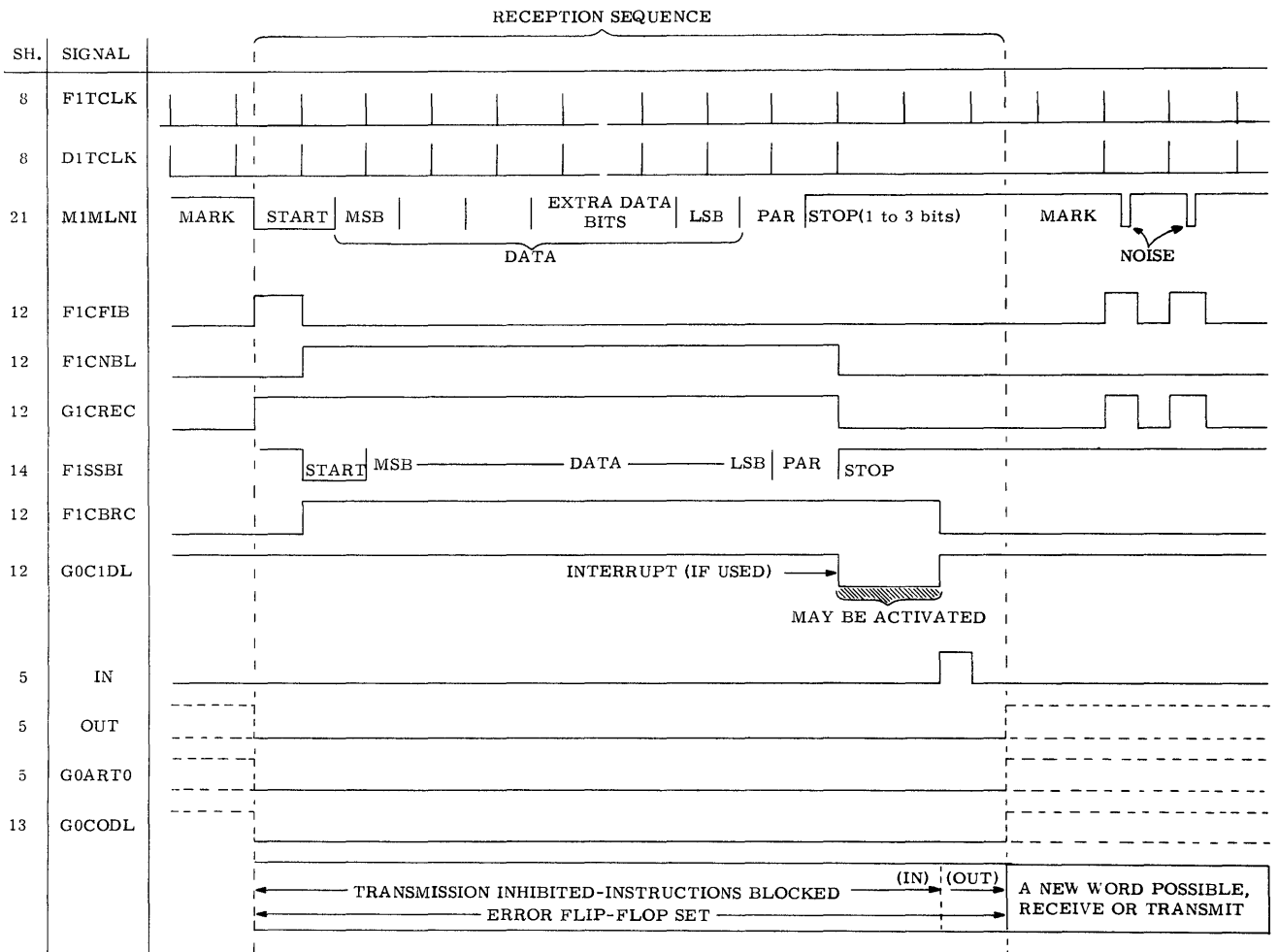


Fig. OP. 3. Complete Reception Sequence

flip-flop (F1TPRE, Sh. 9).

At the time of the reception of the start bit, M1MLNI also set F1CFIB, the first-bit flip-flop shown on Sheet 12, which remains set for about half a bit time, enabling G0PCPY (sh. 20) which serves to clear the parity-check/generate FF and also D1TSSR, (Sh. 8) which sets the S register to all ones. At the fall of F1CFIB, the busy-receiving flip-flop, F1CBRC, is set and remains up during the entire receive sequence. Coincidentally, the enable-receive flip-flop (F1CNBL) is set and will clear at the time that parallelizing is complete. It is at this point that parity is checked (F1PPCG, sh. 20) and an API (if used) is generated, or an ACT instruction may activate an interrupt through G0CIDL (Sh. 12).

During the transfer from the modem to the S register, transmission of a new word is inhibited, GODIKL (sh. 12) can not be activated so long as F1CNBL is set; G0CODL (sh. 13) is inhibited by N1CHLT, and G0ART0 (sh. 5) will not give a ready signal to a JNR until one bit time after reception is complete (F0CDEL, sh. 12).

Additional insurance that an OUT will not interrupt the reception sequence lies in the provision that G0ART0 also requires a clear Busy-Receiving flip-flop (F0CBRC, sh. 12). Upon the completion of the serial transfer from the modem (the start bit is shifter into F1SSTB and clears it), F1CNBL is cleared. G0CIDL may now be activated by an ACT command (S = 1) and the data in the S register (F1SS23-00) may make a parallel transfer to the A register, by way of the inverters N0AS23-00 shown on sheet 7, when an IN or IDL is initiated. Fig. OP. 3, Complete Reception Sequence, illustrates the timing relations of the more significant logic elements involved in a transfer of data from a modem, through the coupler, to the A Register.

The reception completed, the coupler now reverts to a "mark" condition and idles, ready for another reception sequence in which a new start bit may arrive from the modem, or an OUT may originate from the central processor. Because of the asynchronous nature of the remote scanning operation (in which the coupler plays a part) the next word may arrive immediately from either

direction (either Transmit or Receive), or a considerable lapse of time may occur, during which the coupler remains idle and ready, in the mark condition. "Space" will not occur until a new word is ready for transfer, either from the A Register or the modem.

## ERROR DETECTION

The JNE command (S = 7) enters the coupler (0V = true) through the inverter N1AS07 and is inverted back to ground potential by G0ACER coincident with Phase B and the coupler-selecting signals K3 and K2, as shown on sheet 5. The OR gate G1ACER receives this 0V signal, inverts it to +5V and clears the error flip-flop (sh. 20). Clear Alarm will also clear F1PERR in the same fashion.

N0AERR indicates a line error when it receives a binary one signal from the set side of F1PERR. Optionally an error display may be activated through I1AERR at the same time. N0AERR inverts the +5V set output of F1PERR and presents a 0V = Error signal for sampling by the JNE command.

### Parity Error

Odd or Even parity is optional, determined by the jumper board shown on sheet 20, and checked/generated by F1PPCG. Parity is generated during serialization in a transmit sequence, and checked at the completion of the parallellizing process in a reception sequence. A parity error then, is made apparent when all data bits have been shifted to their parallellized positions in the S Register. The clear side of the least significant bit (F1SSOO, sh. 19) determines the presence or absence of a parity error.

Since the choice of even or odd parity rests with whether the set (even) or clear (odd) side of F1PPCG is jumpered to feed the set side of F1PERR through its OR expander, the value of F0SSOO (1 or 0) will determine if a parity error is present, and whether F1PERR will be set or remain cleared.

### Carrier Disappears

The modem carrier signal (M1MDCO, sh. 21) must be present continuously during the reception and assembly of a word. If the carrier should fail, the output of N0PDCO (sh. 20) will rise to +5V, and since N1CNBL also outputs +5V during a receive sequence, G0PSEER will be enabled through its OR expander and F1PERR will be set through its clear output.

### New Word Interrupts Reception

When a new word starts arriving from the modem before the previously received word is read into the AU, the Error Flip-Flop is set, and reception inhibited. The start bit F0SSTB (cleared) is emitting +5V when parallellizing is complete and F1CBRC (Busy Receiving) is at the same level. These signals are present at the input of an OR expander of G0PSEER. So long as M0MLNI remains at 0V there is no effect at F1PERR, but when a new word arrives (sh. 12) G0PSEER is enabled and sets F1PERR through its clear output.

### DSR False While Receiving

M1MDSR (Data Set Ready, sh. 21) must be enabled (+5V) during a reception sequence. If it goes false, the signal N0PDSR (sh. 20) is no longer inhibited and begins to emit +5V to an OR expander of G0PSEER during a period when G1TROX (Receiving or Transmitting, sh. 8) is also delivering +5V. Once again, F1PERR is set by way of its clear output.

#### NOTE

The foregoing types of errors should be followed, when detected, by a request to re-transmit.

### DSR False While Transmitting

Setting the Error Flip-Flop when the Data Set Ready signal goes false during a transmission is accomplished exactly as described in the preceding paragraph.

### OUT While Receiving

D1ASET (sh. 5), a function of the OUT command, appears as an input to the set AND gate of the Error Flip-Flop (sh. 20). A positive signal here, together with a positive signal at the other input (F1CBRC, sh. 12, which is set during a reception sequence) will set F1PERR. Thus, if an OUT command should be executed while the coupler is receiving, a line error will be indicated to the JNE instruction and the optional display will light. Insurance against the possibility of this condition arising lies in the fact that G1TROX outputs +5V when the coupler is either receiving or transmitting, and as a result G1AOUT is inhibited, which disables both D1ACLR and D1ASET and prevents clearing and loading the S register with the new word.

### OUT While Serializing

The OR gate G1PIOO, shown on sheet 20, has a zero volt input from G0AOUT (sh. 5), a function of the OUT command. The OR gate, G1TROX (sh. 8), receives a zero-volt signal from F0CXMT when F1CXMT is set, a condition which prevails when the coupler is serializing. These two signals meet, at a +5V level, at the AND gate of G0PSEER, and if an OUT command should be executed during serializing, F1PERR would be set at Phase A time. Insurance against this possibility exists in the fact that both G0ARTO (Ready to OUT, sh. 5) and G0CODL (ODL or OUT, sh. 13) require that N1CHLT output +5V in order that a JNR or an ACT may initiate either an OUT or an ODL. Since F1COFL is cleared during a transmission, N1CHLT is disabled, outputting zero volts.

### IN While Receiving

The Error Flip-Flop will be set if an IN command should be executed while the coupler is engaged in a reception sequence. As in the preceding paragraph, this is accomplished by the coincidence of G1PIOO and G1TROX. In common with the previous two error situations, this condition should not be permitted by the program to happen. Since F1CNBL is set during reception, and N0CNBL is outputting 0V, G0CIDL is presenting

a busy signal for the ACT instruction.

### **IN While Serializing**

The coincidence of G1PIOO and G1TROX will also set F1PERR should an IN command be executed while the

coupler is serializing. As insurance against such an occurrence, G0CIDL is held in a busy state by the fact that F1CBRC is clear during a transmit sequence and therefore provides a 0V signal from its set side, rather than the +5V required to activate an interrupt or initiate an IN or IDL.

# TABLE OF CONTENTS

## REMOTE SCANNER 4307A

### INTRODUCTION

INT

#### GENERAL DESCRIPTION

Analog Inputs  
 Digital Inputs  
 Digital and Analog Outputs  
 Peripheral Buffer Inputs and Outputs  
 Timed Contact Outputs

#### FUNCTION

Reception Sequence  
 Transmission Sequence  
 Errors  
 Options

### OPERATION

OP

#### LOGIC ORGANIZATION

Timing

Clock  
 Control  
 R Register  
 Test Panel  
 Module Interface  
 Stall Alarm  
**RECEPTION OPERATION**  
**TRANSMISSION OPERATION**  
**ERROR DETECTION**  
 Unsuccessful Compare  
 Data Carrier Disappears  
 DSR False While Receiving  
 Word Parity Error  
 Erroneous Reception  
 Phone Line Open  
 DSR False While Transmitting  
 Module Errors

# LIST OF ILLUSTRATIONS

Fig. INT.1 Data Transfer, Remote Scanner

INT

Fig. INT.2 Reception Format-Module  
 Addresses

Fig. INT.3 Transmission Format

Fig. OP.1 Test Panel

OP

Fig. OP.2 Module Interface Plug Board

Fig. OP.3 Timing Chart, Reception/  
 Transmission

Fig. OP.4 Transmission Sequence -  
 Analog Scan.

# INTRODUCTION

## GENERAL DESCRIPTION

The Model 4307A Remote Scanner (Remote Scanner Controller) serves its purpose in the GE/PAC\* Computer family when communication between a centrally-located computer and remotely-located input/output modules is required. It operates in a half-duplex mode, using a four-wire system.

The scanner utilizes a pair of modems (devices which modulate an outgoing signal and demodulate an incoming signal). One of these is situated adjacent to the central processor and the other is adjacent to the remotely-situated modules. There is literally no limit to the distance between them, aside from the requirement that telephone lines must connect them. However, the distance usually regulates the maximum baud-rate permissible. The scanner is capable of controlling the operation of one, all, or any combination of the below-listed controllers. The only limitation is that only one of each type may be operated by one remote scanner controller.

- Model 4100 Analog Input Controller, together with a Model 4130 Analog to Digital Converter.
- Model 4400 Digital Input Controller.
- Model 4300 Multiple Output Controller.
- Model 4302 Timed Output Controller.
- Model 4201B Peripheral Buffer.

The scanner interface is comprised of a Model 4306A Communications Coupler, a pair of modems (one at each end of the communications line) and a Model 4307A Remote Scanner Controller. The coupler is described under Publications No. 4306A and GE Drawing 68C99581. The remote scanner is of modular design and will accommodate any or all of the listed controllers. The printed circuit boards required are shown on sheet 2 of GE Drawing 68C972150, Card Arrangement. As a field retrofit, it is thus possible to add any one or all of the drive and control hardware necessary to operate each additional controller.

Incorporating a Communications Coupler and a Remote Scanner Controller into a GE/PAC system permits the system to perform as described below and illustrated in Figure INT. 1, Data Transfer, Remote Scanner.

### Analog Inputs

The analog input function operates in a manner essentially similar to that which prevails when the Model 4100 Analog Controller is attached directly to the central processor. It is, however, slower in operation and the scanner command format is compressed and merged with the module address. Also at variance with the standard method of operation is the treatment of matrix overload and converter overflow. Instead of combining these signals in bit 00 of the converter's C register, they are handled separately and their indication is returned instead of data if they should occur. A maximum of 512 analog input points may be scanned.

\* Registered Trademark of General Electric Company

The Model 4101 Analog Input Controller with automatic group advance is not used with the remote scanner.

### Digital Inputs

Because of the necessity for conserving line time, the digital input function operates differently in a remote scanning configuration than when attached directly to the central processor. The input function operates in conjunction with the analog scanner and also separately, and in either mode will read a maximum of eleven contacts (plus validity) per group. In those options which include the Digital Input Controller, the remote scanner provides the line receivers for digital inputs. A maximum of 64 groups of eleven contacts each (plus validity) may be scanned.

### Digital and Analog Inputs

With the exception that the command word is compressed and merged with the module address, digital and analog output functions operate essentially the same as they do when the Multiple Output Controller is connected directly to the central processor. Digital outputs are restricted to a maximum of 32 groups, 8 bits per group. Analog generators are restricted to eight (10-bit) for each remote scanner. These generators are assigned the first eight addresses. If only (8-bit) analog generators are required, they are treated as normal 4ms digital outputs.

### Peripheral Buffer Inputs and Outputs

The Peripheral Buffer (Model 4201B) only operates in a manner resembling its operation when connected directly to the central processor, but is subject to several restrictions. Speeds are slower; device types are limited to those which are incremental in nature (TTY ASR, KSR and the 4242 card reader excluded) and separate module addresses are required for input and output. Further, outputs are limited by line speed and program. Slower devices, therefore, should be employed in a remote scanner operation.

### Timed Contact Outputs

The timed contact output function is reduced in speed and employs a compressed command word format, merged with the module address. Normally, the Model 4302 Timed Output Controller employs eight bits to hold the count received from bits 16 through 23 of the A register (eight bits). In a remote scanner operation the count is restricted to the six least significant of these bits. The contact address, normally held in the six bits A0 through A05 (64 maximum address) is reduced to 32, or five bits. Aside from the foregoing minor differences, the Model 4302 operates as if it were attached to the central processor.

## FUNCTION

The Remote Scanner Controller was designed primarily to adapt the standard GE/PAC input/output controllers so that they could operate I/O devices in a remote location, employing communications modems to act as interconnectors between the locally-situated central processor and the remotely-situated devices.

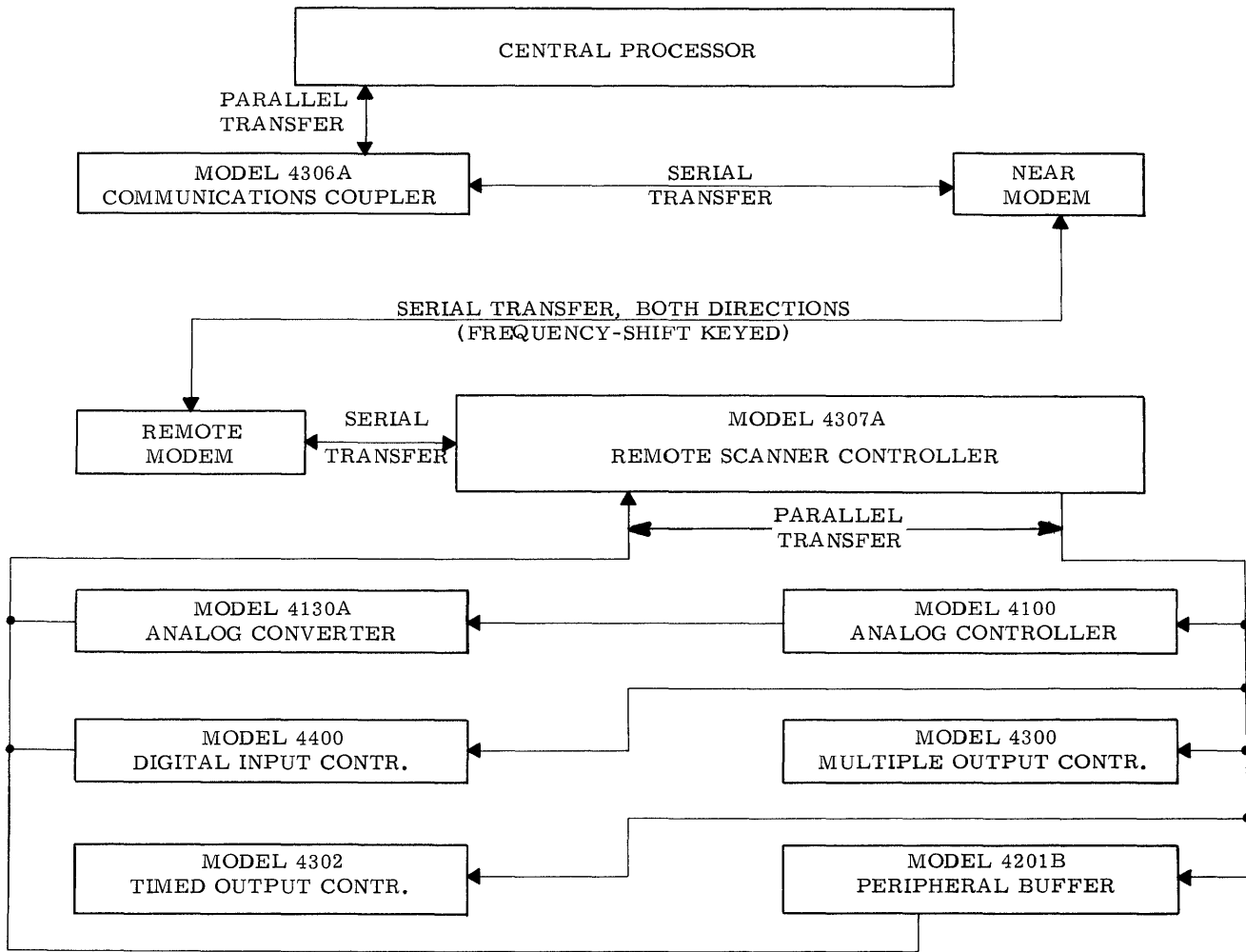


FIG. INT. 1. Data Transfer, Remote Scanner.

The items listed below constitute a summary of the functions performed by the remote scanner controller:

- Receive data in serial form from a communications modem and transfer it in parallel form to an input/output controller, generating its own simulated GEN II instruction.
- Receive data from a controller in parallel and transfer it in serial form to a data communications modem.
- Indicate in the return message the source of the data and the status of the controllers.
- Decode the module address and provide the necessary timing and sequencing to operate any and all of the attached input/output controllers.
- Send an error message whenever line data is received containing a parity error or other line errors.

- Send an error message if there is a check-failure on a repeated command word.
- Generate and transmit parity with each word transferred to the modem.
- Initiate action of any I/O Controller manually, by means of an optional test panel which permits visual observation of the data generated.

#### Reception Sequence

The successful reception of a word from the modem requires that the scanner have completed the receive-reply (replies) sequence. It is not permissible for incoming data to interrupt the current serializing sequence, nor may a previously received word be destroyed. Should a new word be processed prematurely (before the reply has been transmitted) it would be lost and the error flip-flop set.

As with the Communications Coupler, Model 4306A, reception from the modem begins when the data line



transits from true to false ("mark" to "space"), an event which accompanies the reception of the "start" bit, indicating that data is to follow. Because of the possible presence of noise-initiated pulses of short duration, the scanner is capable of distinguishing a start bit by refusing to recognize a change from mark to space which endures for less than one-half bit time. The valid start bit is shifted into the parallelizer (R Register) by the transmit/receive clock. It is important to note here that the same register performs the function of parallelizer during the reception, and that of serializer during transmission. For this reason it is an unacceptable condition for transmission and reception to occur simultaneously in one scanner.

The start bit, entering the R Register by way of the most significant bit position, continues to shift right with each clock pulse, followed by the rest of the new word, until it reaches the least significant bit position. At this time a signal is generated to inform the control logic that a word has been assembled in the scanner. Parity is now checked on the data bits and the error flip-flop is set if a discrepancy exists.

The assembled word is also checked to confirm that the scanner is properly addressed, after which (if the scanner is in an ON-LINE mode) it decodes, addresses an

I/O module, and initiates a simulated GEN II command. (The interface between the scanner and the controllers is equal to the GEN II outputs from the AU of the central processor.) The situation at this point in the reception sequence is analogous to that of the Communications Coupler at the parallelization-complete stage of its reception sequence, with certain format difference illustrated in Fig. INT. 2.

Though there is little correspondence in the positions they occupy in the various formats, the "L" areas serve the same functional purposes as the similarly numbered K bits of the GEN II command. Depending upon the module addressed, the control will assume one of three modes during the GEN II instruction.

- Outputs (non-data response required)
- Digital or Peripheral Buffer Input
- Analog Input (initiates a digital input first)

Assuming that the control is in the ON-LINE condition, the address and status bits are set and a transmission is initiated following the completion of the GEN II command.

### Transmission Sequence

A transmission (as oriented in this discussion) is defined as the transfer of the contents of the R register of the scanner controller to the remote modem, which transmits to the near modem, through the communications coupler, to the central processor. From the standpoint of the coupler, this would involve the Reception Sequence described in the Model 4306A discussion. Fig. INT. 1 clarifies the two points of view and the interchangeable terminology, whereby a coupler transmission is a signal going out from the central processor, while a scanner-controller transmission is a signal travelling toward the central processor. The words "near" and "remote" as applied to the two modems always mean "near to" and "remote from" the central processor.

Prior to transmission, data has been loaded into the serializer as illustrated in Fig. INT. 3, except in the instance when a repeat is being returned, in which case it is an exact duplicate of the command word. The R register is now loaded and one start bit, odd parity and three stop bits are appended to the data, address and status information, and the entire register is transferred, one bit at a time, to the modem. This serial transfer begins with the start bit and progresses in synchronism with the transmit/receive clock.

At this point, there are two possible conditions which may exist. Either the scanner-controller is finished with the reception/reply sequence, or it waits for completion of an analog conversion to initiate an additional reply. In either case, the data output to the modem will be held in a mark condition until the next transmission sequence is initiated or a new reception sequence begins.

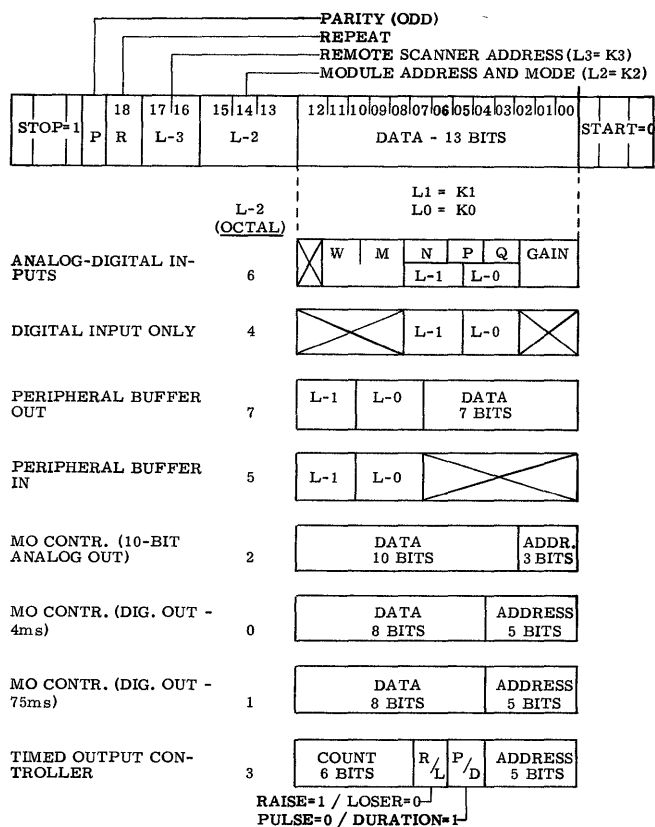


FIG. INT. 2. Reception Format - Module Addresses

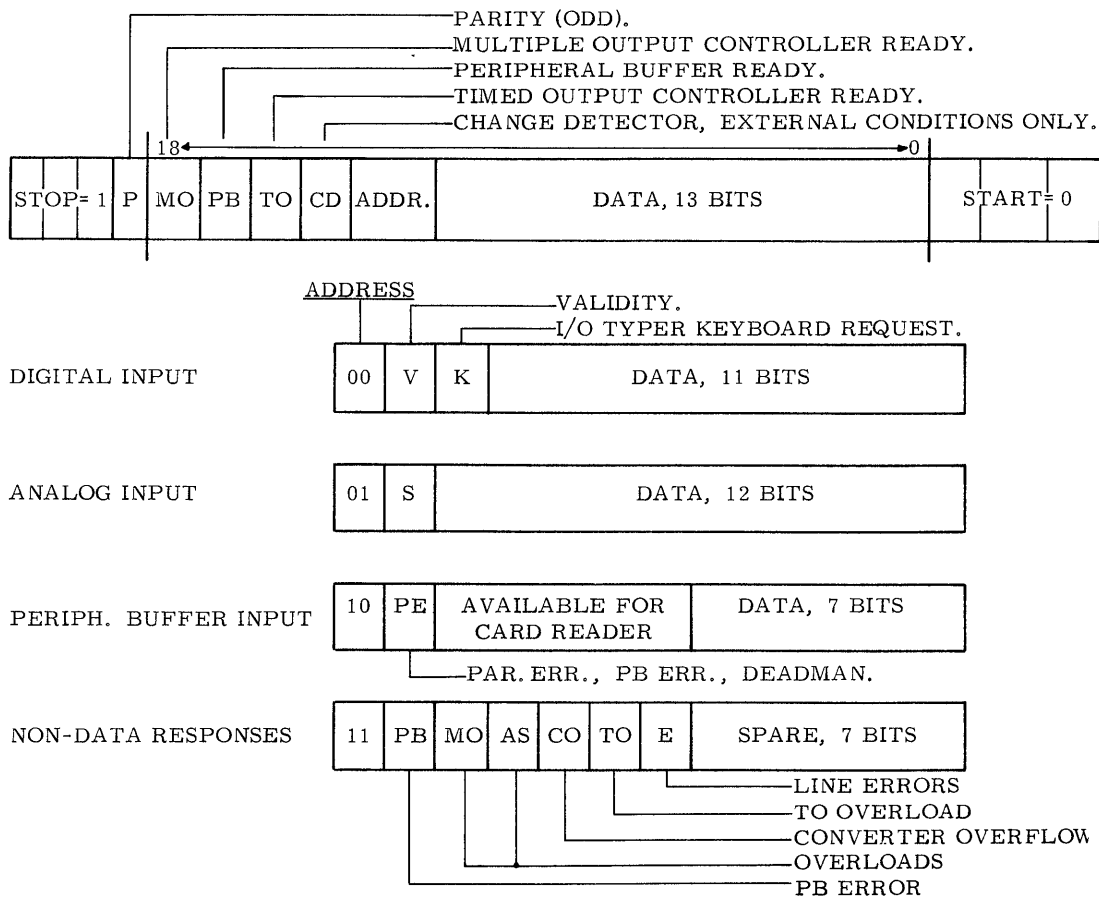


FIG. INT.3. Transmission Format.

## Errors

Errors which are detected in association with the operation of the remote scanner controller are listed below.

- Unsuccessful "compare" between two receptions of the same command word.
- Word parity error.
- Data-carrier signal disappears before a complete word has been received and assembled.
- Incoming data is not at "mark" when a complete word has been received. (Phone line opened.)
- Data-Set-Ready signal goes false while receiving.
- A new word starts arriving in the parallelizer before the previously received word is acted upon and its transmission response completed. The new word will be lost.
- Data-Set Ready signal goes false while transmitting. This should cause an error on the received data at the computer.

The errors shown below are detected in the I/O Controllers and are indicated as separate bits in the Non-data response format:

- Timed Output overload.
- Multiple Output overload.
- Analog Scanner overload (error and no overflow)
- Peripheral Buffer error.
- Converter overflow (error and overflow)

## Options

The Model 4307A Remote Scanner is available in a number of optional configurations, the different options being designated by the values given to the last three characters in the model number (4DP4307AXYZ). These represent the method of communications between the Communications Coupler and the Remote Scanner (X), the digital input accommodations (Y) and the presence or absence of the Stall Alarm and Test Panel (Z). In addition, certain other operational characteristics may be selected by the use of a plug-in jumper printed-wire

board. Among these is a choice of eight baud rates and four terminal addresses.

- Method of coupler-scanner communications:

X = 1 Standard, 19-bit controller for use with the successive approximation analog to digital converter (Model 4130A) and a pair of data sets (modems capable of more than one-mile transmission).

X = 2 Same as X = 1 except that this option uses line drivers and receivers capable of transmission up to one mile, instead of data sets (special private cabling necessary).

- Digital Input Accomodations:

Y = 0 No Digital Input Controller or line receivers at the remote end.

Y = 1 A 16-group Digital Input Controller (4400B) and line receivers.

Y = 2 A 64-group Digital Input Controller (4400B) and line receivers.

Y = 3 Line Receivers only (for use with the 4400A, not included).

- Test Panel, Stall Alarm (Deadman):

Z = 0 No test panel, no stall alarm.

Z = 1 No test panel, but stall alarm built into controller to supply drive to relay whenever valid data is not received for 0-5 seconds.

Z = 2 Test panel, no stall alarm.

Z = 3 Test panel, stall alarm.

# OPERATION

## LOGIC ORGANIZATION

During the ensuing discussion of the operation of the Model 4307A Remote Scanner Controller, the logic contained in GE Drawing No. 68C972150 will be used extensively. It is understood that possible future alterations or minor adaptations applicable to specific systems may change the sheet numbers quoted, but in either case the circuit paths can be easily traced from the logic mnemonics.

### Timing

The operation of the scanner is timed from two main sources, the basic clock illustrated on sheet 4 and the baud-rate (bits per second) counter shown on sheet 5-7. The basic clock consists of a temperature-regulated, crystal controlled, 720 KC clock generator and two logic-amplifier elements, D1TADV and D1TOSC. Directly or indirectly this clock provides the timing pulses for every function performed by the scanner, including that of the baud-rate counter.

The baud-rate counter, though dependent on the basic clock, performs a fundamental function in its control of the speed of the serial transfers which comprise a large part of the scanner's operation.

### Clock

Also shown on sheets 5-7 are various control elements associated with the baud-rate counter. F1TPRE, the Preset-Enable flip-flop, controls D1TZRO which clears the counter, and D1TLOD, which loads the counter. The choice of available baud rates is shown at the bottom of sheet 6. F1TSYN, the Synchronize flip-flop, ensures that the incoming data is synchronized with the internal clock (baud-rate counter).

### Control

As with the timing area, the control area of the logic deals with practically every phase of scanner operation. Most of the control elements, however, are grouped on sheets 8 through 12. F1CFIB (First Bit flip-flop, sh. 8) is set at the same time that the start bit of the new word enters the R Register. Falling, F1CFIB sets F1CNBL, which, at the end of the word shift, affects F1CCWH (Control Word Here), F1CSEL (This Terminal Selected), G0CCL2 (Clear L2 Buffer) and N1CLL2 (Loan L2 Buffer). The precise interaction of these control elements will be discussed later under Reception Sequence Operation.

Control elements involved in a transmission sequence are shown on sheets 11 and 12. The Transmit Flip-flop, F1CXMT, is always set during a transmission sequence. F1CLDR, D1CCLR and D1CSET are also involved in the parallel loading of the R Register, preparatory to the serial transmission to the modem. D1CNDR, D1CPBR, D1CAIR, D1CDIR and F1CCNV, however, only feature in those transmissions in which they have been specifically called for by the command word from the central processor. D1CNDR (Non-data response) provides such information as multiple output, timed output and analog scanner overloads, peripheral buffer and line transmission errors, and converter overflow. D1CPBR (Peripheral Buffer Response), D1CAIR (Analog Input Response)

and D1CDIR (Digital Input Response) are utilized in data transmissions by way of the indicated controllers.

The Transmit/Receive Bit Counter, pictured on sheet 12 is preset by G1CPBC in accordance with the signals received by its OR gate inputs. During a transmission sequence, the bit counter is governed by the clear side of the Load R Register flip-flop, while during a receive operation F0CFIB provides the presetting control. One set AND gate on both F1C004 and F1C016 are grounded, providing the 19-bit capability of both the X = 1 and the X = 2 controller options.

### R Register

The next area of logic consists of the 23 high-speed flip-flops and associated expanders which comprise the R Register. It is equipped to receive serial data from a modem, check parity, decode, address, generate a simulated GEN II command and transfer data in parallel to any one of five remote modules. It can also accept incoming data from one of these modules in parallel, generate start, stop and parity bits and transmit the entire contents of the register serially to the modem. Stop and parity bits are pictured on sheet 13, and sheets 14 through 17 contain the nineteen status and data bits, ending with the least significant bit of the register, F1RSTB, the start bit. The inputs to this register are many and varied, though they fall into a few specific groupings. In the area of data bits (F1RR00-12), the inputs N1MC06-17 (A/D Converter Register), N1MD00-10 (Digital Input data bits) and N1MN00-11 (Peripheral Buffer data bits) are present. Whether or not any certain group is being transferred at any particular moment is determined by the nature of the command being processed.

D1TCLK<sub>1</sub> supplies the clock pulses; D1CCLR, the clear inputs; D1CSET loads the register, while N1MMOR, N1MTOR, and N1MDVR provide ready signals. There are also inputs indicating error conditions, as shown under Non-data Responses and Peripheral Buffer Input in Fig. INT. 3, Transmission Format. This illustration, as well as Fig. INT. 2, Reception Format, are invaluable as references in following the logic of the Remote Scanner Controller. The entire operation of the scanner is inseparably involved, both in its capability and its limitations, with the content of these formats. They represent the content of the R Register in all of its functions as serializer, parallelizer, transmitter, receiver, generator or checker of parity bits, controller of remote modules and devices, et cetera.

### Test Panel

The Parity Check/Generate flip-flop, F1PPCG, the Error flip-flop and other error detection elements shown on sheet 18 will be discussed later in this publication. The test panel, furnished in options in which Z = 2 and Z = 3, provides the means for manually operating a remote scanner controller and observing the data generated. This option first appears in the 68C972150 logic on sheet 8 where the panel's on-line switch may either permit or inhibit the setting of F1CCWH by F1CNBL. The status of this switch also controls the loading of the L2 Buffer by G0CCL2.

Modules may be addressed manually by setting the Function Selector Switch on the test panel to the desired address and loading the data by way of the 13 data switches on the panel. The addresses selected correspond to the L2 (K2) values shown in Fig. INT. 2, and are to be found in the logic as inputs to the L2 Buffer on sheet 10. The Function Selector, shown near the lower left corner of Fig. OP. 1, can be positioned for any one of eight addresses, listed octally below:

- 0 MO Short (Multiple Output, 4ms delay)
- 1 MO Long (Multiple Output, 75ms delay)
- 2 AO (Analog Output)
- 3 TO (Timed Output)
- 4 DI (Digital Input Only)
- 5 PBI (Peripheral Buffer Input)
- 6 A & DI (Analog and Digital Input)
- 7 PBO (Peripheral Buffer Output)

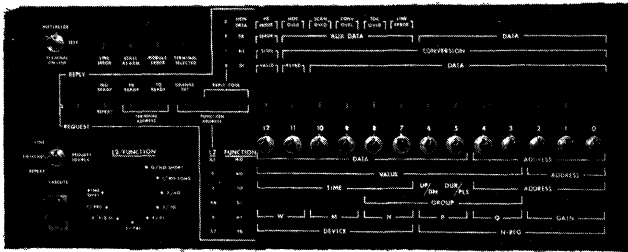


FIG. OP. 1. Test Panel

Repeat and Execute Peripheral Buffer signals from the panel are shown on sheet 10, while sheet 13 displays the indicator gates which enable the display of the R Register data and several non-data indications on the panel.

### Module Interface

Because the remote scanner may utilize only one input/

output module, or as many as five, a plug board is used as a flexible means of interconnection. This board shown in Fig. OP. 2, also simplifies the retro-fit of additional modules in the field. Sheet 20 lists the R Register outputs to the plug board. It is noted in this area that when the remote scanner is situated on the same page as the output modules, the plug board is not used. In this case, the outputs shown are wired directly to the indicated module terminals.

The SD-2 plug board appears throughout Area M of the logic, wherever MO, TO and SC outputs are shown. Sheet 21 features the K3K2, Phase A-B and GEN II drives to these modules, as well as Peripheral Buffer and Digital Input Controller drives. Status signals involving all modules are pictured on sheet 22. The outputs of these elements confirm the previous observation that Figures INT. 2-3 are the keys to understanding the operation of the remote scanner. As examples, the inverters N1MSCE and N1MCOF handle the "scanner error" and "converter overflow" signals respectively. Their outputs are bits 10 and 9 of the R register. Reference to Fig. INT. 3 reveals that these bits in the non-data response format hold the scanner error and converter overflow signals. Scanner overload is indicated by scanner error with no converter overflow. Similarly the format at the top of the illustration places the ready signals in bits 18-16, which correspond to the outputs of N1MMOR, N1MDVR and N1MTOR.

The module interface area continues through sheets 23-25, featuring the digital input data bits on sheet 23 and the inputs from the peripheral buffer and A/D converter on sheet 24, I/O typer and change detection buffers on sheet 25. Note that while there are 12 data bits from each of these modules, those from the analog converter are C Register bits 17-06, but the buffer transfers N Register bits 11-00 (the 5 most significant all zeros, except when inputting from the card reader). These are shown in pairs on sheet 24, each pair outputting to the R Register bit which corresponds to the similarly numbered buffer bit.

### Stall Alarm

This option, sometimes called "Deadman", is the standard Model 4091 and features an indicator lamp on the test panel in scanner options in which Z = 3. The alarm

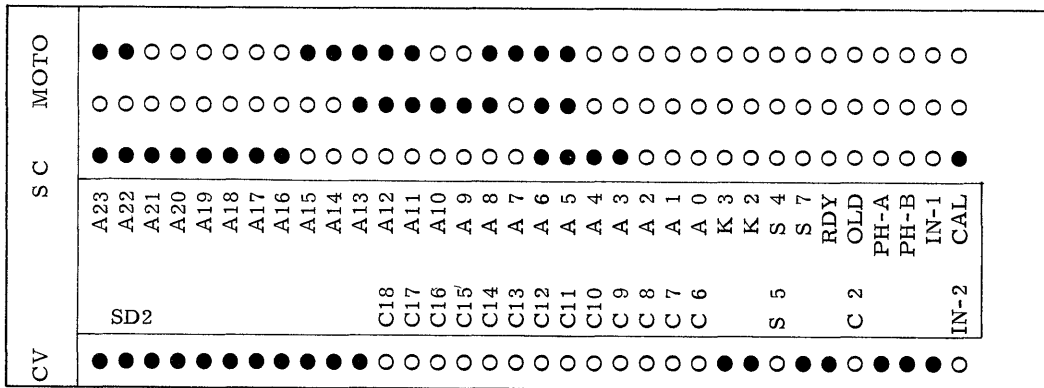


FIG. OP. 2. Module Interface Plug Board

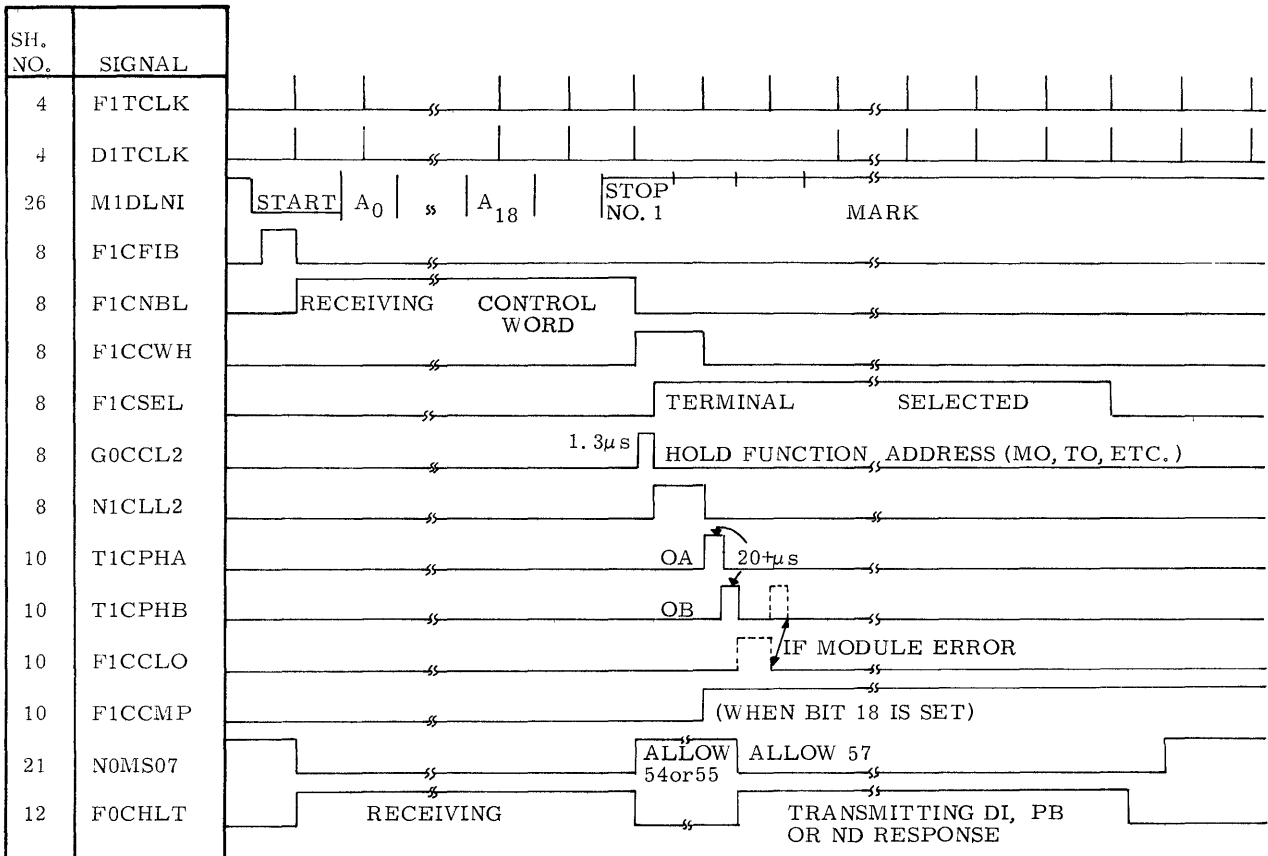


FIG. OP. 3. Timing Chart, Reception/Transmission.

is a timing device utilizing a single-shot MSSA element. It has the property of being set under a no-alarm condition and only resets (alarms) when permitted to complete a timing cycle. Normal programming procedures provide the issuance of the Set Stall Alarm command (OUT or IN) at short intervals, shorter than the one-to-four-second delay for which the stall alarm is adjusted.

### RECEPTION OPERATION

Discussion of the operation of the reception sequence will require frequent reference to the logic contained in GE Drawing No. 68C972150 and also to the timing chart, Fig. OP. 3. The timing relationships between the various terms shown are accurate with regard to sequence, though space limitations make a true scale drawing impractical. In both this discussion and that of the transmission sequence to follow, consideration will be confined to those areas peculiar to the operation of the remote scanner controller. The functions of the basic clock and the local clock (baud-rate counter) are described in the Model 4306A Communications Coupler publication, and since they operate in essentially the same manner in the scanner, these descriptions will not be repeated.

The term M1DLNI, the first element in the scanner to receive the incoming start bit from either the modem or 1-mile line receiver, appears on sheets 26 and 27. The loads on these elements, however, are identical. Since

the uses of the line-driver/receiver option are limited, the modem option will be explored in this discussion.

As the start bit arrives, M1DLNI goes down, since the start bit is always a zero. The data line has changed from a true (mark) condition, to false (space). F1CFIB (sh. 8) sets at the next D1TOSC (no more than 1.38µs later), provided M1DDSR (data set ready, sh. 26) and M1DDCO (carrier on, sh. 26) are true. This sets F1CNBL, which delivers 0V from its clear output, enabling G1CRCV and N0CRCV (receiving a word), plus G1TROX and N0TROX (receiving or transmitting, sh. 4). The prime purpose of the last two elements is to enable the transmit/receive clock (D1TCLK1, 2).

F1CNBL cannot be cleared until the bit counter (sh. 12) has completed its count at the end of the serial transfer of data into the R Register, and is cleared, causing N0CHLT to put out +5V. At this time F1CNBL goes down and sets F1CCWH (control word here, sh. 8), if not inhibited by the test panel's "On-Line" switch. Setting F1CCWH enables G0CCL2 which clears the L2 Buffer (sh. 9), and approximately 1.3µs later F1CSEL (terminal selected) is set, disabling G0CCL2. This enables G0CLL2 and N1CLL2, loading the L2 Buffer with the contents of bits 15-13 of the R Register (the module address).

Two other critical functions occur at this time, either of

which may stop the reception sequence. The first began when F1CFIB set and caused F1PPCG (Parity-Check/Generate, sh. 18) to clear. As the line is shifted into the R Register, every binary one shifted from the modem (M0DLNI to N1DLNI to F1PPCG/F0PPCG) toggles this flip-flop. This includes the binary one which represents the first stop bit. Starting cleared, an odd number of toggles (correct parity) leaves the flip-flop cleared. But if there should be an even number of ones, F1PPCG is set at the end of reception and the error flip-flop is also set (through its set input). This occurs when N1CLL2 goes true.

The second critical function is the scanner address check. Bits R17 and 16 (sh. 14), jumpered through the LJBA board shown on sheet 8 (according to the code at the bottom of the sheet), are brought into an OR expander of F1CSEL (This-Terminal-Selected). Should the wrong scanner address exist in R17-16 at the time parallelization is complete, F1CSEL would remain clear, T1CPHA would not be enabled, and there would be no transfer to a module. In effect, the reception sequence would be terminated.

Assuming that the selected module is ready (G1CRDY, sh. 10), F1CCWH is cleared. Also assuming that F1CCNV (Analog-Scanner-Converting, sh. 11) is not set, N0CCNV is delivering +5V to the OR expander of T1CPHA (Phase A, sh. 10). If F1CCNV were set, N1MSCC would be true (F1CSEL is true in both cases) and T1CPHA would be enabled through its AND gate. At the end of 20 $\mu$ s T1CPHB is triggered by T1CPHA and it also produces a 20 $\mu$ s signal. These signals are generated immediately following completion of the "load L2" function, which places them within that period of time after the end of the serial reception of the control word during which a simulated OUT command may initiate a parallel transfer to the MO, TO or analog scanner. To initiate a transmission sequence, an IN command may also be generated during this period. The contents of the L2 Buffer, plus the status of the converter, will determine the choice (G0MS04, G0MS05, sh. 21).

Also of interest on sheet 21 are the gates which furnish the K3K2 drives for the various controllers. Each has a separate gate directed to G0CRDY, the Selected-Module-Ready signal. Note that these are ORed so that G1CRDY provides a +5V signal to the clear side of F1CCWH only if the selected controller is in a ready state. This insures that the GEN II instruction is not executed until the addressed module is ready. Note that N0MS07 is not enabled at this time.

Since a zero volt = true signal is required for the inputs of the controllers to which the R Register's data output is directed, bits R12-00 are fed through the inverters shown on sheet 20. It will be noted that when in a manual mode, the signals arriving from the optional test panel are already at zero volts and are not inverted. The interface for these signals may be found on the test panel drawing, 68C972191, sheet 3.

## TRANSMISSION OPERATION

A transmission sequence always follows a reception

sequence, even though it may be confined to a response involving only status information. In one instance, when L2 = 6<sub>8</sub>, a second transmission will take place before a new word arrives from the modem. The first transmission is a digital input and the second is an analog conversion.

Non-data responses are required when the scanner is in an output mode. This occurs when the module address equals octal 0, 1, 2, 3 or 7 (Fig. INT. 2). As the format (Fig. INT. 3) indicates, overload, overflow and error information is transferred in parallel to the R Register, from the module addressed, via the medium speed inverters shown on sheet 22. These signals are fed directly and are not part of the scanner's error detection logic.

The transmit cycle is initiated when F1CLDR (sh. 11) sets. This occurs in the interval between the end of a reception sequence and the beginning of a transmission sequence, since it is required that the scanner be doing neither (N0TROX, sh. 4, outputting +5V). Armed by N0TROX, F1CLDR is set during Phase B, by D1TOSC, enabling N1CLDR by way of the zero volt output of its clear side. This results in the setting of F1RSB2-1 (stop bits, sh. 13), at the same time clearing F1RPAR (parity bit) and the Parity Check/Generate flip-flop, F1PPCG (sh. 18).

Since the reception sequence is finished, no repeat operation can be in progress, and therefore F1CCMP has been cleared. F0CCMP and F1CLDR enable D1CCLR, clearing the R Register. D1CSET, having been enabled by D1CCLR, remains up for about .4 $\mu$ s after D1CCLR falls, being held by its own +5V output, ANDed with D1TOSC. It is during this brief period that the parallel loading of the R Register is accomplished. F1RR18-15 are set directly from D1CSET, while F1RR14-00 are loaded via D1CDIR (digital input response), D1CAIR (analog input response), D1CPBR (peripheral buffer response) or D1CNDR (non-data response), whichever has been selected.

Reference to Fig. OP. 4, Transmission Sequence, reveals that when F1CLDR falls at the end of Phase B, F0CHLT rises. This occurs via G1CPBC (preset bit counter, sh. 12), which puts out +5V to clear F1CHLT when it receives a zero volt signal from F0CLDR. F1CXMT (transmit flip-flop, sh. 11), now armed by F0CHLT, needs only a clock pulse to set, in the case of the 1-mile line driver/receiver option. The modem option, however, also requires a Clear to Send signal (M1DCTS, sh. 26) from the modem.

Setting F1CXMT begins the serial transfer to the modem. The start bit (F1RSTB, sh. 17) is now added to the least significant end of the data field and is the first bit transferred, followed by the data bits (F1RR00-12), the address bits (F1RR13-14), the status bits (F1RR15-18) and finally the three most significant bits (which were generated during N1CLDR), F1RPAR (parity bit, 0 = odd) and the stop bits, F1RSB1-2. These signals, one by one, pass through the Data Phone Interface element, X1DLNO (line out, sh. 26), and then to the modem

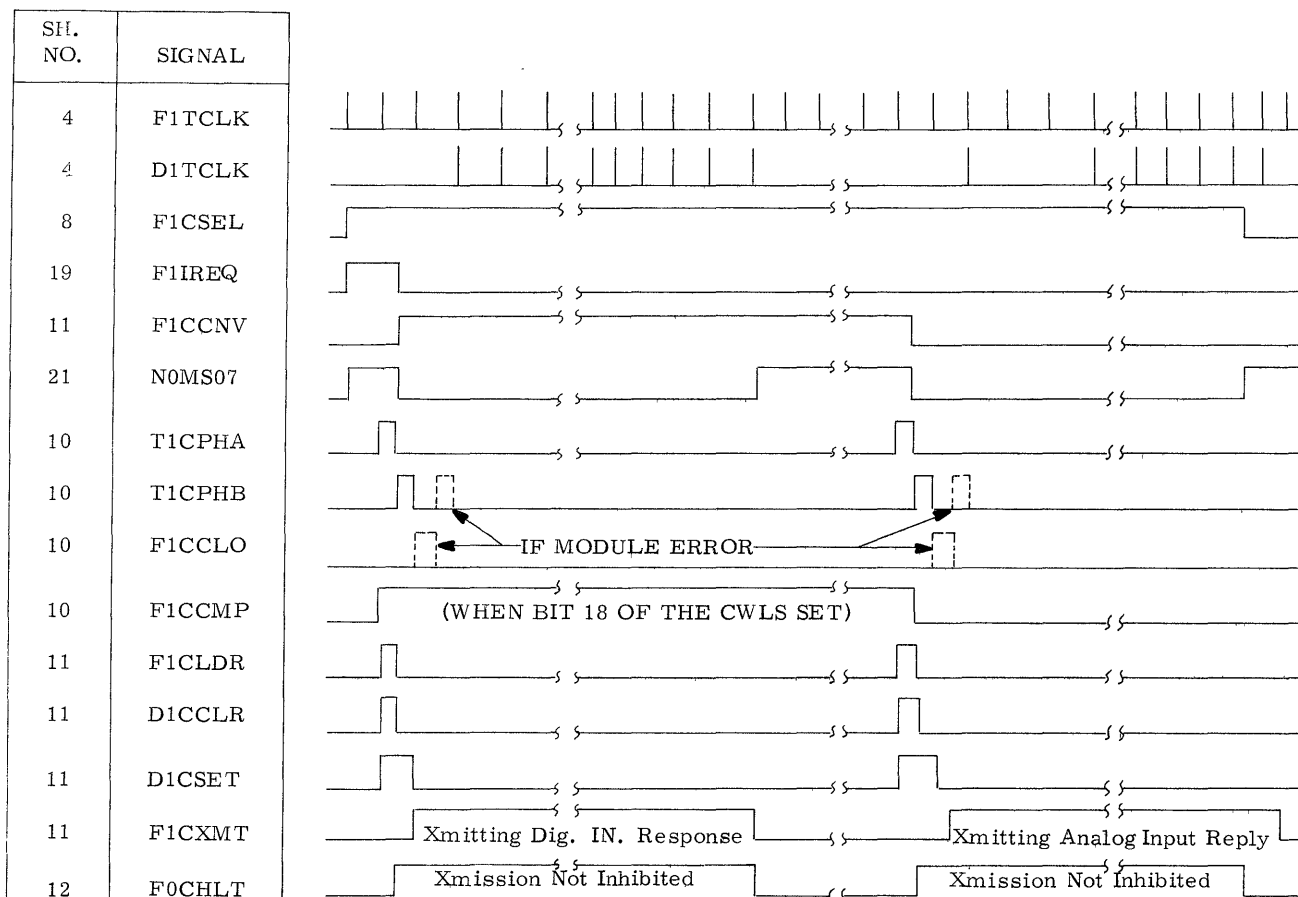


FIG. OP. 4. Transmission Sequence - Analog Scan.

For transmission. After the last bit has left X1DLNO, the remote scanner is, with one exception, finished with the receive-respond cycle, and is ready for the next reception sequence. The exception exists when the previous reception sequence carried an address of octal six (analog and digital input). In this case the scanner has only completed one-half of its transmission sequence (the digital input half) and must wait for a conversion-complete signal (N1MSCC, sh. 22) to initiate a second transmit sequence.

For an analog input to be made, it is necessary that F1CCNV (Scanner-converting, sh. 11) be set. This occurs during the coincidence of F1CL22 and D1CDIR, which is enabled by N1MDIK (DIC K2 drive, sh. 21), an inverter that needs F1CL24 set and F1CL21 cleared. This condition only exists when the L2 buffer is in an octal six configuration. In the same figure F1CSEL is shown to remain set, after an analog scan request, through the entire first reply and doesn't clear until the analog conversion has been completed and the digital result transferred to the modem.

### ERROR DETECTION

There are two main areas of error detection to be considered. The first, comprising those errors directly associated with the operation of the scanner, sets the

error flip-flop and returns a non-data response with bit 07 carrying a binary one. (See Fig. INT.3). These are line transmission errors and will be handled as indicated below. The second error area is composed of module errors, which are detected in the I/O controllers. These also return a non-data response.

### Unsuccessful Compare

An unsuccessful comparison between two receptions of the same command word causes the Line-Error flip-flop (F1PERR, sh. 18) to set. Detection is enabled during the normal ring-shifting of the R Register when the Compare-Control-Word flip-flop (F1CCMP, sh. 10) is set by the presence of a binary "one" in bit R18. A one in R18 (Repeat, Fig. INT.2) is placed by the program and causes comparison of bits R00 through R17 of the control word just received (containing R18 = 1) and a second reception of the same word.

During comparison, the clear side of F1CCMP, putting out zero volts, inhibits G0MPHA and G0MPHB (GEN 2 Phase A-B drive), preventing transfer to the module. At the end of the successful comparison F1PERR remains clear, which permits N0PERR to put out +5V and G0MPHA-B are enabled. If unsuccessful, F1CCMP is set and at the next clock pulse G0PSER (sh. 18) is enabled through its OR gate and sets F1PERR. This has the



effect of enabling a non-data response, but inhibiting digital input, analog input and peripheral buffer responses. (D1CNDR, D1CDIR, D1CAIR and D1CPBR, sh. 11).

The mechanics of the comparison involve two groups of expanders feeding the OR input of GOPSER (sh. 18). As the second reception of the control word is routed through N0DLNI or N1DLNI (dependent on the value of each bit), the first reception (which had been ring-shifted back into the R Register at the same time that it had been returned to the computer) is entering the two circuits from F1RSTB/F0RSTB. F1CCMP, F1CNBL and F1TCLK are shared by both circuits. If simultaneous "ones" are arriving, F1RSTB is at +5V, and N0DLNI is true (zero volts). At the same time N1DLNI(+5V) is true while F0RSTB supplied zero volts. GOPSER is inhibited. The same occurs when both bits are zeros. If dissimilar binary values are present, however, GOPSER is enabled and sets F1PERR through its clear output.

### Data Carrier Disappears

Should the Carrier-On signal (M1DDCO, sh. 26) go false before a complete word is received and assembled, the Error flip-flop will be set. Although F1CNBL delivers a +5V signal to the set input of F0PERR (via its expander) during the entire reception sequence, N0PDCO delivers zero volts so long as the carrier remains on. If the carrier fails, the expander sets F1PERR. As in all cases when F1PERR is set, a non-data response is enabled, but digital input, analog input and peripheral buffer responses are inhibited.

### DSR False While Receiving

When the Data-Set-Ready signal (M1DDSR, sh. 26) goes false while receiving it delivers zero volts to the input of the inverter N0PDSR (sh. 18). This signal, coincidental with G1CLNE (sh. 8) and G1TROX (sh. 4), both of which are true when on-line and receiving, will set the error flip-flop, via GOPSER. A non-data response is initiated to indicate a line error.

### Word Parity Error

Parity is checked during a reception sequence by the Parity-Check/Generate flip-flop (F1PPCG, sh. 18) at the completion of parallelization. All bits in the R Register have been shifted to their parallelized positions before a parity error becomes apparent. F1PPCG is either set (parity error) or cleared (odd parity) at this point, depending on the number of bits that have passed through (and toggled) the flip-flop. If a parity error exists it will be set and together with N1CLL2 (true at this time) will set the error flip-flop through its set input. A non-data response will be returned indicating a line error.

### Erroneous Reception

Should a new word start to arrive before the previous word is acted upon and its response completed, the

error flip-flop will be set and the new word lost. The modem line, in a mark condition since reception of the previous word, drops to "space" with the arrival of the zero-volts start bit. N0DLNI (sh. 26) now puts out +5V, and together with F1CSEL (still set) and the next N1TCLK, sets F1PERR through its set input. It will be noted that the resultant non-data response cannot occur until after the response to the previous word has been transmitted back to the central processor, because F1CLDR (sh. 11) can not set until after reception/transmission is complete (N0TROX).

### Phone Line Open

When the incoming data is not at "mark" (true) when a complete word has been received, it is usually an indication that an open has occurred in the telephone line during reception. N0DLNE, now at +5V ("space" condition) is ANDed with T1CPHA and N0CCNV, also at +5V because conversion can not begin until after a digital response, setting the error flip-flop. A non-data response is returned, after the phone line has been re-stored. This is insured by the Clear-To-Send (M1DCTS) input to F1CXMT (sh. 11). This signal, originated in the remote modem as a result of Request-To-Send from the near modem, will be present after the line is re-stored.

### DSR False While Transmitting

When the Data-Set-Ready signal (M1DDSR, sh. 26) goes false while transmitting, it delivers zero volts to the input of the inverter N0PDSR (sh. 18). This signal, ANDed with G1CLNE (sh. 8) and G1TROX (sh. 4), both of which are true when the scanner is on-line and transmitting, sets the error flip-flop. This error, occurring during transmission to the central processor, should (but not necessarily) cause a parity or other error to show in the received data.

### Module Errors

Overload, overflow and peripheral buffer errors are detected in the I/O modules themselves and are returned in the non-data response format, as shown in Fig. INT. 3. Received from the modules by the inverters N1MMOO, N1MTOO, N1MSCE and N1MPBE shown on sh. 22, these signals deliver +5V to the individual R Register flip-flops indicated in the format. The ground level signal as received from the controllers, however, is also Ored in G1IMER (Module Error, sh. 19) which provides an indication to the optional test panel and also sets F1CCLO (Clear-Module-Error, sh. 10).

Fed by its own set output and the term F0CXMT (this occurs immediately before the transmit-sequence) F1CCLO clears and triggers T1CPHB, which generates an extra Phase B and routes it to the driver G0MPHB (phase B drive, sh. 21). This signal, at zero volts, is ANDed within the addressed module with the zero-volt output of N0MS07 (GEN II S7 Drive, sh. 21) which serves to clear the module error in the addressed module.