SDS TECHNICAL INFORMATION

Theory of Operation

COMPUTER
Model 910

SDS 900004D

SCIENTIFIC DATA SYSTEMS

$9.00

Theory of Operation

COMPUTER
Model 910

SDS 900004D

May 1966

This publication supersedes
SDS 900004C dated August 1964

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

LIST OF EFFECTIVE PAGES

Total number of pages is 288, as follows:

PREFACE

The SDS 900 Series Computers are solid state, general purpose, digital machines which use a parallel, random access, coincident current, magnetic core memory and serial arithmetic operation.

These computers are completely modular, contain all silicon components, and have positive true logic. Internal operations are binary and two's complement arithmetic is used.

This manual is presented in four sections. These cover Computer Logic, W Buffer and Input/Output Operations, Magnetic Core Memory, and BCD Typewriter and Tape Punch. Each section has its own lists of Contents and Illustrations which define the referenced pages.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont.)

LIST OF ILLUSTRATIONS (Cont.)

LIST OF ILLUSTRATIONS (Cont.)

SECTION I

COMPUTER LOGIC

INTRODUCTION

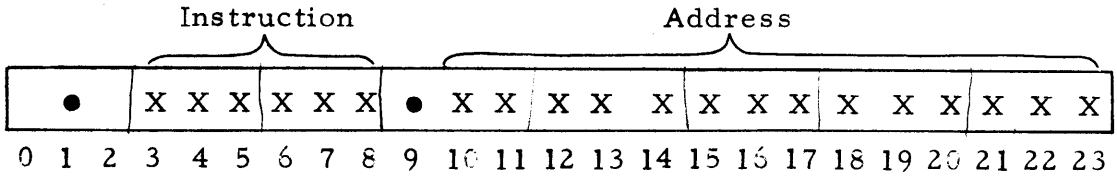
The Computer Logic Section describes the logical operation of the computer in detail. It is primarily intended for engineering, checkout, training and maintenance use. Descriptions are presented with the assumption that the reader has a working knowledge of Boolean algebra and basic digital operations.

For proper use of this publication, the reader should be familiar with the SDS Reference Manual for this computer.

BASIC COMPUTER STRUCTURE AND OPERATION

NOTE: The description below includes only the most common conditions. Further details and qualifications are given later in this section.

The computer uses a 24-bit, single-address instruction:



The instruction code occupies the six bit positions 3 through 8 and determines the operation which will be executed. For example, the binary configuration 101100 represents an octal "54", which is the designation for a subtraction instruction. The address field occupies the 14 bit positions, 10 through 23. In general, the address field represents the address of the operand. For example, it may represent the address of the number which is to be subtracted from the A register (accumulator) upon execution of a subtraction instruction.

The computer has nine registers, as represented in Figure 1. The P register (program counter) holds the 14-bit address of the next instruction to be executed. This 14-bit flip-flop register is incremented by one before reading a new instruction since the instructions are generally read in numerical sequence, i. e.,

10046
10047
10050
10051

The address held in the P register is transferred in parallel to the 14-bit S flip-flop register (memory address) where it sets up the address lines in the core memory for access to the word position addressed. The contents of the desired word are then transferred in parallel to the 24-bit M register from memory. During memory accessing or storing the S register will always hold the desired address and the M register will contain the operand or accept it from memory.

The instruction in the M register, after being received from memory, is transferred in parallel to the 24-bit C register where its fields are distributed as follows: The 6-bit instruction code field is transferred in parallel to the 6-bit flip-flop "O" (instruction code) register where it will be used to control execution of the instruction.

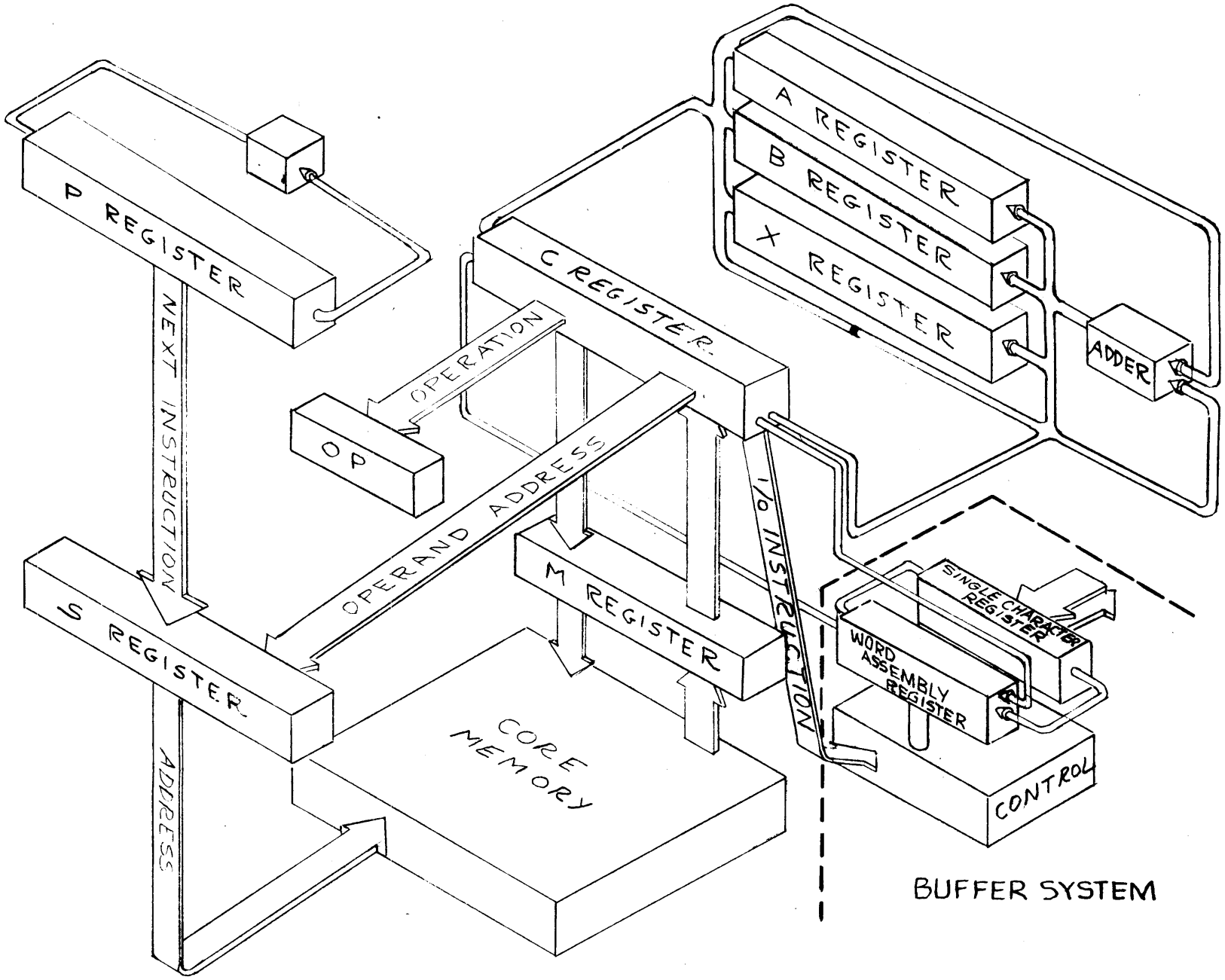


FIGURE 1

BASIC COMPUTER STRUCTURE

INFORMATION FLOW

Depending upon the instruction to be executed, the address field containing the memory address of the desired operand is transferred in parallel to the S register. For example, the M register accepts the operand from memory and transfers it in parallel to the C register. At this point the code of the operation to be performed is held in the O register and the operand, upon which the operation is to be performed, is held in the C register.

During the execution of the instruction the operand is serially shifted out of the C register and into the Word Assembly register (WAR), Adder, A register, B register, or X register.

During the execution of "Store" instructions the operand address of the instruction is transferred to the S register to set up the Store address. The data to be stored is then shifted into the C register from its present location, and the information is transferred in parallel to the M register which stores it in memory.

To communicate with external devices such as a photo-reader, magnetic tape unit, punch and typewriter, the computer uses the WAR with its associated single character register (SCR) as shown in Figure 1. During an input process 6-bit characters from the external device are transferred into the SCR in parallel. When the Buffer "control" detects that a character has been received, it shifts the character into the WAR, and when the WAR contains the desired number of characters, a signal is sent to the computer to store the information held in the WAR into memory.

On output, the inverse operation is used; the information is shifted from the C register to the WAR and the computer returns to its previous program. The WAR now shifts one character at a time into the SCR where it is sent out in parallel to the external device in operation. When the buffer control detects that the WAR is empty, it signals the computer to load the WAR with another word for output.

When studying the information which follows, frequently refer to the Reference Manual to properly integrate the programming nomenclature and discussion with that of the internal operation.

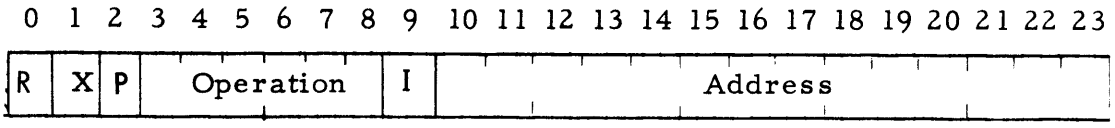
LOGICAL DESCRIPTION

INSTRUCTION AND WORD FORMAT

The instruction list for the computer is shown in Figure 2. The instructions are numbered from 00 through 77 octally and are listed with their respective mnemonic codes and execution times. The execution time for a specific instruction is the number of word times (machine cycles) which elapse during the accessing of the instruction and its complete execution, including the accessing or storing of any operands.

INSTRUCTION FORMAT

The instruction format is:



	0 :	Relative address bit (not used in logic)
	1 :	Index bit
	2 :	Program operator bit
3 ⟶	8 :	6-bit instruction code
	9 :	Indirect address bit
10 ⟶	23 :	14-bit address

The programming functions of the above fields are:

Index Register Bit:

A "1" in this position causes the contents of bits 10 through 23 of the index register (X register) to be added to the address portion of the instruction prior to execution.

Program Operator Bit:

A "1" in this position causes the program operator bit and the 6 bits of the instruction code to be interpreted as an entry address for a subroutine.

Instruction Code:

The 6 bits of the instruction code contain the 2-digit octal code for the instruction to be executed. This information is transferred to the "O" register flip-flops, 01, 02, 03, 04, 05, and 06.

Indirect Address Bit:

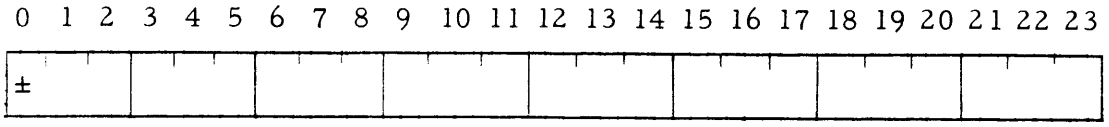
A "1" in this position causes the computer to interpret bits 10 through 23 of the instruction (possibly modified by indexing) as the memory location where the effective address of the instruction may be found. A "0" causes bits 10 through 23 of the instruction to be interpreted as the effective address.

Address:

The address is defined by bits 10 through 23 of the instruction. The address generally represents the memory location of the operand.

DATA WORD FORMAT

The data word format is:



Arithmetic is performed using the binary two's complement number system. The sign bit can then be considered to be integral with the data word. Data words consist of eight octal digits, each octal digit consisting of three binary digits (bits).

During all serial operations the 23rd bit position (least significant end) appears first in time.

TWO'S COMPLEMENT ARITHMETIC

Considering the machine to be a fixed point, fractional computer, using two's complement arithmetic, any number stored in memory can be expressed by the following general formula:

$$N = A_0 (-2^0) + A_1 (+2^{-1}) + A_2 (+2^{-2}) + - - - + A_{23} (+2^{-23}),$$

or it can be expressed as:

$$N = \sum_{i=1}^{23} A_i (2^{-i}) - A_0$$

From this it can be seen that the largest allowable positive number is:

0. 11111111111111111111111

The minimum positive number is:

0. 00000000000000000000000

The maximum negative number is:

1. 00000000000000000000000

The minimum negative number is:

1. 11111111111111111111111

Numbers of negative magnitude are expressed in two's complement form and arithmetic operations, where the operands and answers are within the allowable range, maintain the negative numbers in two's complement form.

Time	Code	Mne-monic	Function	Time	Code	Mne-monic	Function
1	00	HLT	Halt	1, 2	40	SKS	Skip if M not set
1	01	BRU	Branch to M	1, 2	41	BRX	(X) + 1 → X, BR if X9 = 1
1	02	EOM	Energize M		42		
	03			2	43	BRM	P → M, BRU → M + 1
	04				44		
	05				45		
	06			1	46	*	
	07				47		
2+	10	MIY	(M) → Y when ready		50		
	11			2	51	BRR	(M) + 1 → P (BRU)
2+	12	MIW	(M) → W when ready		52		
3+	13	POT	Parallel out, when ready	2, 3	53	SKN	Skip if (M) negative
2	14	ETR	Extract	2	54	SUB	(A) - (M) → A
	15			2	55	ADD	(A) + (M) → A
2	16	MRG	Merge: (A) or (M) → A		56		
2	17	EOR	Exclusive OR		57		

* Instructions where the address is used in addition to the instruction code to define the specific operation. (see page 1. 12)

FIGURE 2

INSTRUCTION LIST

Time	Code	Mnemonic	Function	Time	Code	Mnemonic	Function
1	20	NOP	No operation				
	21			3	61	MIN	(M) + 1 → M
	22				62		
1	23	EXU	Execute		63		
	24			2	64	MUS	Multiply Step
	25			2	65	DIS	Divide Step
	26			N+2	66	*	
	27			N+2	67	*	
3+	30	YIM	(Y) → M when ready	2	70	SKM	Skip if (A) = (M), on B mask
	31			2	71	LDX	(M) → X
3+	32	WIM	(W) → M when ready	2, 3	72	SKA	No skip if (A) (M) = 1 anywhere
4+	33	PIN	Parallel in, when ready	2, 3	73	SKG	Skip if (A) > (M)
	34				74		
3	35	STA	(A) → M	2	75	LDB	(M) → B
3	36	STB	(B) → M	2	76	LDA	(M) → A
3	37	STX	(X) → M	2	77	EAX	Eff. address → X

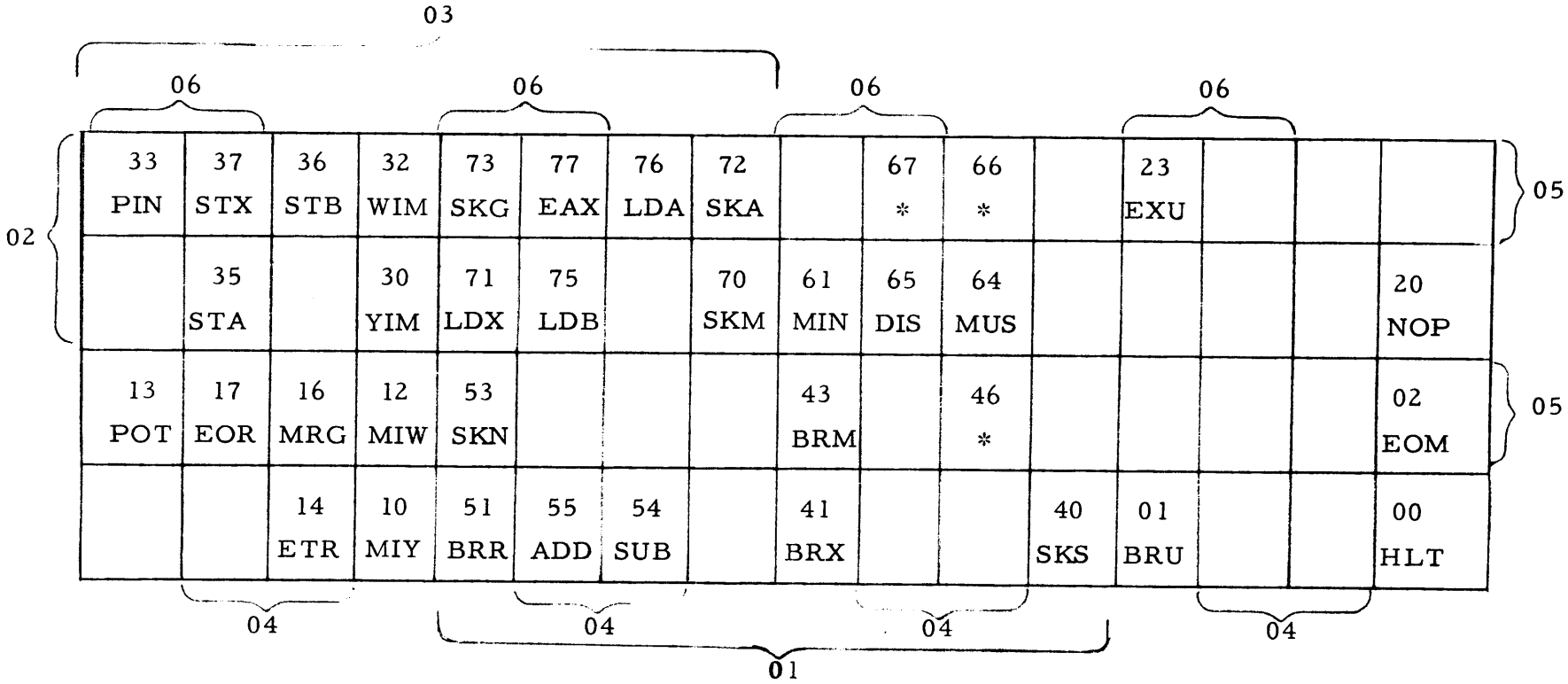
* Instructions where the address is used in addition to the instruction code to define the specific operation. (see page 1.12)

FIGURE 2
INSTRUCTION LIST (CONTINUED)

The specific operation to be executed for instruction codes 46, 66, and 67 is specified by the address configuration as shown below:

Timing	Code	Mnemonic	Function
1	4600000	XAB	A $\leftrightarrow$ B
1	4610000	BAC	B $\rightarrow$ A, 0 $\rightarrow$ B
1	4620000	ABC	A $\rightarrow$ B, 0 $\rightarrow$ A
1	4630000	CLR	0 $\rightarrow$ A, 0 $\rightarrow$ B
N+2	66000XX	RSH	Right Shift AB
N+2	66200XX	RCY	Right Cycle AB
N+2	67000XX	LSH	Left Shift AB
N+2	67200XX	LCY	Left Cycle AB
N+2	67100XX	NOD	Normalize and decrement X

The last two digits of the address, for instructions 66 and 67, specify the number of shifts or cycles to take place.



* Instructions where the address is used in addition to the instruction code to define the specific operation (see page 1.12).

FIGURE 3

INSTRUCTION DECODING

PULSE COUNTER							
Gray Count	Pulse Time	Q1	Q2	Q3	Q4	Q5	Q6
0	T1	0	0	0	0	0	1
1	T0	0	0	0	0	1	0
6	END Tp	0	0	1	0	1	1
7	START T24	0	0	1	0	0	0
8	T23	0	1	1	0	0	1
9	T22	0	1	1	0	1	0
10	T21	0	1	1	1	1	1
11	T20	0	1	1	1	0	0
12	T19	0	1	0	1	0	1
13	T18	0	1	0	1	1	0
14	T17	0	1	0	0	1	1
17	T16	1	1	0	0	1	0
18	T15	1	1	0	1	1	1
19	T14	1	1	0	1	0	0
20	T13	1	1	1	1	0	1
21	T12	1	1	1	1	1	0
22	T11	1	1	1	0	1	1
23	T10	1	1	1	0	0	0
24	T9	1	0	1	0	0	1
25	T8	1	0	1	0	1	0
26	T7	1	0	1	1	1	1
27	T6	1	0	1	1	0	0
28	T5	1	0	0	1	0	1
29	T4	1	0	0	1	1	0
30	T3	1	0	0	0	1	1
31	T2	1	0	0	0	0	0

FIGURE 4

PULSE SEQUENCE

PULSE COUNTER

A pulse counter consisting of flip-flops Q1, Q2, Q3, Q4, Q5 and Q6 that counts the pulses that make up the twenty-six pulse times of each computer word.

The pulse counter is a modified gray code counter with Q1 as the most significant bit.

The gray code is off-set from the 26-bit pulse count and certain counts are eliminated to facilitate decoding and counter design.

Note that computer pulse times are designated in reverse order, with respect to time.

Q6 acts as a straight binary counter generating the output which is counted by the gray code counter Q1 through Q5.

$$sQ6 = \overline{Q6}$$
$$rQ6 = Q6$$

The least significant gray code stage Q5 counts in gray sequence with the addition of (Q1 + Q3) on the reset side to force a skip of gray counts 15 and 16.

$$sQ5 = \overline{Q5}\ Q6$$
$$rQ5 = Q5\ Q6\ (Q1 + Q3)$$

Q4 counts in gray sequence with the addition of (Q1 + Q3) on the set side to force the skip of gray counts 2, 3, 4 and 5.

$$sQ4 = \overline{Q4}\ Q5\ \overline{Q6}\ (Q1 + Q3)$$
$$rQ4 = Q4\ Q5\ \overline{Q6}$$

Q3 counts in gray sequence with the addition of T0 to force the skip of gray counts 2, 3, 4 and 5.

$$sQ3 = \overline{Q3}\ Q4\ \overline{Q5}\ \overline{Q6} + T0$$
$$rQ3 = Q3\ Q4\ \overline{Q5}\ \overline{Q6}$$

Q2 and Q1 count in a gray sequence but the forced skip of gray counts 2, 3, 4, 5, 15 and 16 simplifies their set and reset expressions.

$$sQ2 = \overline{Q1}\ \overline{Q5}\ \overline{Q6}$$
$$rQ2 = Q1\ \overline{Q4}\ \overline{Q5}$$

$$sQ1 = \overline{Q3}\ Q5\ Q6$$
$$rQ1 = \overline{Q3}\ \overline{Q4}\ \overline{Q5}$$

PULSE DECODING

Several multiple and single pulse times are decoded from the pulse counter for use in the computer logic:

Data word, $\overline{Tp}\ \overline{T24} = Q1 + Q2 + \overline{Q3}$

Most significant or Sign bit, $T0 = \overline{Q1}\ \overline{Q2}\ \overline{Q3}\ Q5$

Least significant bit, $T23 = \overline{Q1}\ Q2\ \overline{Q4}\ \overline{Q5}$

Guard and first bit of word, $T24 = \overline{Q1}\ \overline{Q2}\ Q3\ \overline{Q5}$

Guard and last bit of word, $Tp = \overline{Q1}\ \overline{Q2}\ Q3\ Q5$

Address field, $(T23 - T10) = Q2$

Indirect address bit, $T9 = Q1\ \overline{Q2}\ Q3\ \overline{Q4}\ \overline{Q5}$

For additional requirements, the following pulse times are also decoded:

$(T22-T17) = \overline{Q1}\ Q2\ \overline{T23}$

$(T21+T22) = \overline{Q1}\ Q2\ Q3\ Q5$

$T14 = Q1\ Q2\ \overline{Q3}\ \overline{Q5}$

$T10 = Q1\ Q2\ \overline{Q4}\ \overline{Q5}$

$(T5-T0) = \overline{Q2}\ \overline{Q3}$

$T1 = \overline{Q1}\ \overline{Q3}\ \overline{Q4}\ \overline{Q5}$

PHASE CONTROL

The various internal operations, such as transfers of information, clearing registers, read from memory, etc., which must take place during the reading and execution of instructions, are controlled by eight phases, designated phase Ø0 through phase Ø7 . A general description of these eight phases is given below, followed by more detailed discussions on control of the phase sequence by the instruction code.

PHASES

A phase chart is shown on page 1.19 which indicates the phase sequencing for the various operation codes. Usually, each dot on the diagram represents one cycle time from Tp until the next Tp.

Phase 0 (Ø0)

This phase is the beginning of all instructions. Instructions start at Ø0 T24, with the new instruction in the C register and the instruction register cleared to "NOP" (20).

This phase steps directly to phase Ø5 (at Ø0T24) for some instructions requiring only one cycle time for execution. For others, indexing and indirect addressing are processed in phase Ø0. Also, the memory regenerates the instruction (writes it back in memory because of destructive read-out) and fetches the operand. This process may require more than one cycle time, as in the case of indirect addressing, executing and transferring.

At Ø0 T24 the instruction code is transferred to the OP code register (01, 02, 03, 04, 05, 06).

Phase 1 (Ø1)

This phase is the set-up or preparation phase for the shift instructions. This phase is not entered until the end of the first T9 in phase Ø0 and it lasts through Tp. Phase 1 inhibits pulsing the memory for an operand.

Phase 2 (Ø2)

This phase is the wait phase for the input/output instructions, (10), (12), (13); (30), (32), (33). The computer idles in this phase until the data is ready. If the data is ready before the instruction is given, then phase Ø2 is by-passed, except for the parallel input/output instructions, (13), (33), where phase Ø2 will last only one cycle time for data to be transferred.

Phase 3 (Ø3)

This phase is the execution phase for shifting and usually requires several word times. The duration of this phase is dependent upon the number of shifts required.

Phase 4 (Ø4)

This phase is the second cycle time of three-cycle instructions which write data into the memory. These three cycles are Ø0 → Ø4 → Ø7. During phase Ø4 the P register is incremented for accessing the next instruction, and the word to be stored is shifted serially into the C register.

Phase 5 (Ø5)

This phase is the execution phase for some single-cycle instructions. The C register does not shift and parity is not checked.

Phase 6 (Ø6)

This phase is the main execution phase of instructions requiring an operand, but no memory modification. These instructions are the two-cycle instructions, and include conditional skips which may require three cycles.

Phase 7 (Ø7)

During this phase memory regeneration (or new storage) and next instruction access are performed. The P register is not incremented unless the Skip flip-flop is high.

End Cycles

An End is defined as the last cycle of execution for the particular instruction being performed. The End term designates that the next phase is to be phase Ø0 unless the instruction causes a skip.

$$\text{End} = Ø5 + Ø6 + Ø7 + Ø0\ (- - - -)$$

The last term is for unconditional branches.

PHASE COUNTER

A three-flip-flop counter, F1, F2 and F3, controls the sequence and advancement of phases:

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

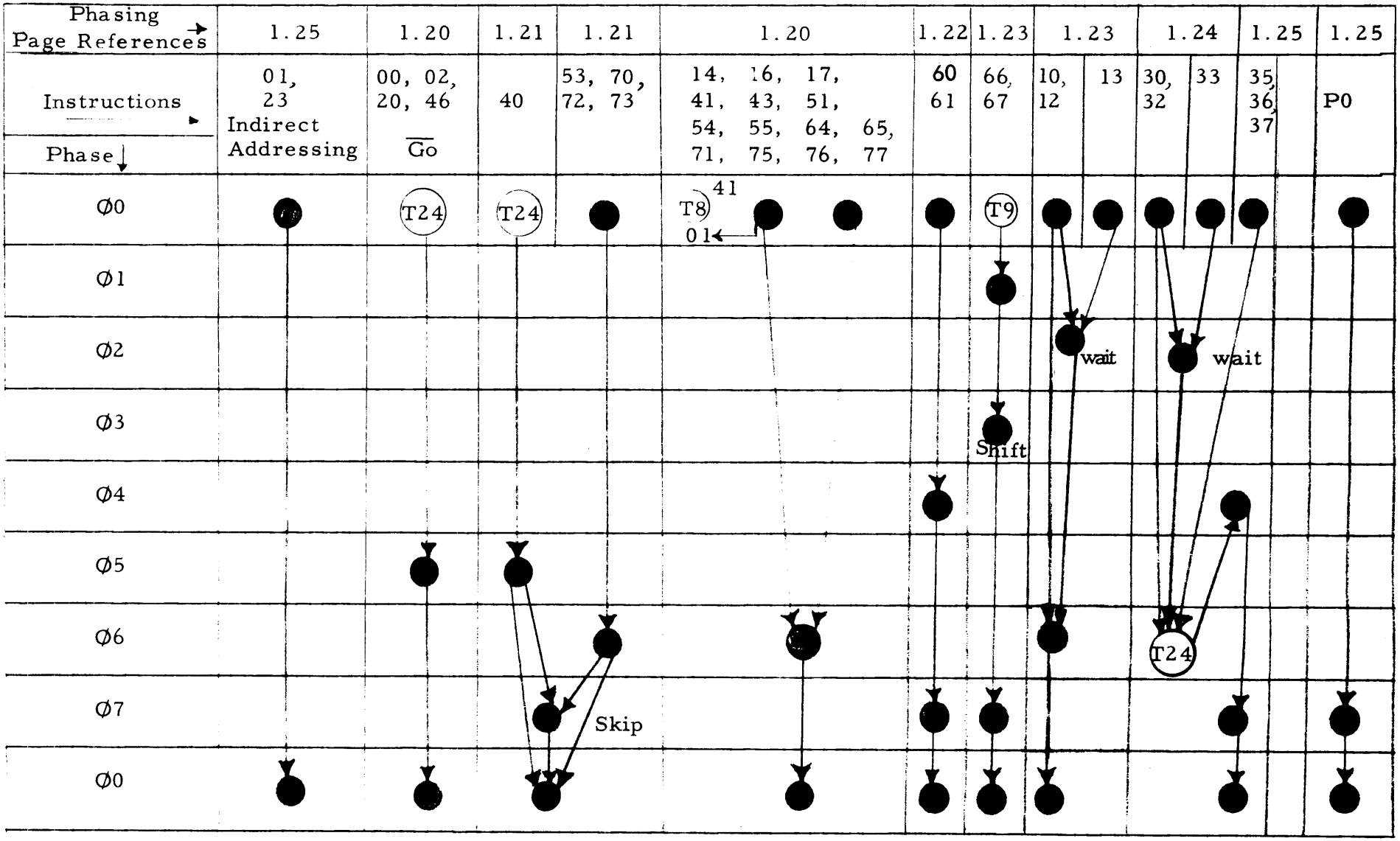


FIGURE 5

PHASE SEQUENCE CHART

The following **discussion** indicates the actions of the phase counter during the execution of the various instruction groups shown in the phase diagram.

SINGLE-CYCLE INSTRUCTIONS

Phasing: $\phi 0 \xrightarrow{T24} \phi 5 \longrightarrow \phi 0$

Instructions 00, 02, 20 and 46, and the idle state, $\overline{Go}$, advance directly to phase $\phi 5$ from phase $\phi 0$,

$$sF1 = T24\ \overline{Ia}\ \phi 0\ Go\left[\overline{C5}\ \overline{C8}\ \overline{C2}\ (\overline{C3} + \overline{C4})\right] + T24\ \overline{Go} + - - -$$

$$sF3 = T24\ \overline{Ia}\ \phi 0\ Go\left[\overline{C5}\ \overline{C8}\ \overline{C2}\ (\overline{C3} + \overline{C4})\right] + T24\ \overline{Go} + - - -$$

The output of the C register flip-flops must be used for the instruction code, as the transfer to the O register is not made until T24.

The $\overline{Ia}$ term represents the indirect address flip-flop and need not be considered for this instruction group. Go is the control term which allows computation.

To reset to phase $\phi 0$, F1 and F3 are reset at the first Tp,

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

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

where:

$$End = F1\ F3\ \overline{Ts} + F1\ F2\ \overline{Ts} + - - -$$

Ts is a time-share signal and freezes the internal operation of the computer for two machine cycles while external devices control access to memory. Sk is the flip-flop which signals for a skip condition and need not be considered in this instruction group.

TWO-CYCLE INSTRUCTIONS

Phasing: $\phi 0 \longrightarrow \phi 6 \longrightarrow \phi 0$

Instructions 14, 16, 17, 41, 43, 51, 54, 55, 64, 65, 71, 75, 76, and 77 advance to phase $\phi 6$ from $\phi 0$ at the first Tp, if there was no indirect addressing.

$$sF1 = Tp\ \overline{Ia}\ \phi 0\ 03\ 04 + Tp\ \overline{Ia}\ \phi 0\ 01\ \overline{04} + Tp\ \overline{Ia}\ \phi 0\ 01\ \overline{05} + - - -$$

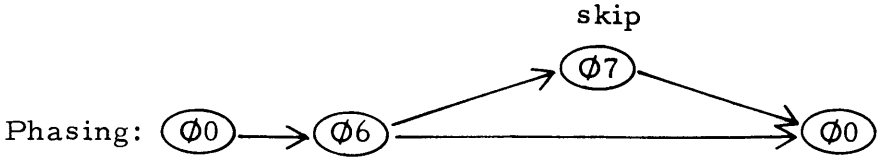
$$sF2 = Tp\ \overline{Ia}\ \phi 0\ (\ 03 + \overline{03}\ 04\ \overline{05}) + Tp\ \overline{Ia}\ \phi 0\ 01\ \overline{02} + - - -$$

If there had been an indirect addressing bit, Ia would be true and phase $\phi 0$ would have been extended one or more additional word times until Ia was reset by the effective address. The computer resets to phase $\phi 0$ at the first Tp of phase $\phi 6$.

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

$$rF2 = Tp\ End\ \overline{Sk} + - - -$$

THREE-CYCLE CONDITIONAL SKIP INSTRUCTIONS



Instructions 53, 70, 72 and 73 advance to phase Ø6 at the end of phase Ø0.

$$sF1 = Tp\ \overline{Ia}\ Ø0\ 01\ (\overline{04} + \overline{05}) + - - -$$

$$sF2 = Tp\ \overline{Ia}\ Ø0\ 03 + - - -$$

$$rSk = Ø0\ T0 + - - -$$

These instructions are conditional skip instructions where the Sk (skip) flip-flop is left in a set condition at the end of phase Ø6, if a skip is to occur. If Sk is true, the phase counter advances to phase Ø7.

$$sF3 = Tp\ Sk\ \overline{Ts} + - - -$$

Sk will be reset at the start of phase Ø7.

$$rSk = Ø7 + - - -$$

If Sk was left in a reset condition at the end of phase Ø6, the phase counter will reset directly to phase Ø0.

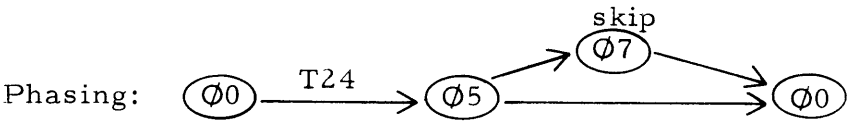
$$rF1 = Tp\ End\ \overline{Sk}$$

$$rF2 = Tp\ End\ \overline{Sk} + - - -$$

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

Note that phase Ø6 and Ø7 are both end cycle phases, and if a skip occurs, End will be true for two cycles. The conditions under which Sk is set during phase Ø6 are discussed in the material on "Skip and Branch Instructions".

TWO-CYCLE CONDITIONAL SKIP INSTRUCTIONS



Instruction 40 advances directly to phase Ø5 at Ø0 T24.

$$sF1 = T24\ \overline{Ia}\ Ø0\ Go\left[\overline{C5}\ \overline{C8}\ \overline{C2}\ (\overline{C3} + \overline{C4})\right] + - - -$$

$$sF3 = T24\ \overline{Ia}\ Ø0\ Go\left[\overline{C5}\ \overline{C8}\ \overline{C2}\ (\overline{C3} + \overline{C4})\right] + - - -$$

This phase advancement is identical to the single-cycle operation discussed under Phase Control.

During pnase Ø5, instruction 40 tests to see if M (external signal) is in a true or false condition. If the signal is false, then the Sk (SKIP) flip-flop will be set and a skip will occur.

$$sSk = T0\ \emptyset5\ 01\ \overline{04}\ \text{(external signal)} + - - -$$

If Sk is set at T0, then the phase counter advances to phase Ø7 at Tp to perform the skip.

$$sF2 = Tp\ \overline{Ts}\ Sk - - - + - - -$$

At the completion of phase Ø7, or at the completion of phase Ø5, (if the skip condition were false and Sk was not set) the phase counter is reset to phase Ø0.

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

$$rF2 = Tp\ End\ \overline{Sk} + - - -$$

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

Sk is reset at the start of phase Ø7.

$$rSk = \emptyset7 + - - -$$

Note that phases Ø5 and Ø7 are both end-cycle phases, and if a skip occurs, End will be true for two cycles.

THREE-CYCLE MEMORY INCREMENT AND DECREMENT INSTRUCTIONS

Phasing: Ø0 ⟶ Ø4 ⟶ Ø7 ⟶ Ø0

Instructions 60 and 61 advance to phase Ø4 at the completion of phase Ø0. During Ø0 indexing or indirect addressing may have been performed.

$$sF1 = Tp\ \overline{Ia}\ \emptyset0\ 01\ (\overline{05} + \overline{04}) + - - -$$

At the completion of phase Ø4 (one cycle), during which the contents of memory are incremented, the phase counter is advanced to phase Ø7.

$$sF2 = Tp\ \overline{Ts}\ \emptyset4 + - - -$$

$$sF3 = Tp\ \overline{Ts}\ \emptyset4 + - - -$$

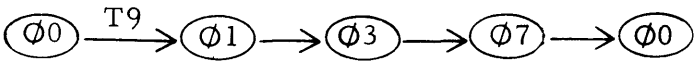
During phase Ø7 the modified number is stored in memory and the next instruction is accessed. At the completion of phase Ø7 the counter is reset to phase Ø0.

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

$$rF2 = Tp\ End\ \overline{Sk} + - - -$$

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

N-CYCLE SHIFT INSTRUCTIONS

Phasing: 

Instructions 66 and 67 terminate phase Ø0 at pulse time T9 to advance to phase Ø1 for pulse times T8 through Tp.

$$sF3 = T9\ \overline{Ia}\ Ø0\ 02\ \overline{03}\ 04\ 05 + - - -$$

During phase Ø1 (10 pulse times) the registers and shift counter are prepared for shifting. At Tp the counter advances to phase Ø3.

$$sF2 = Tp\ Ø1 + - - -$$

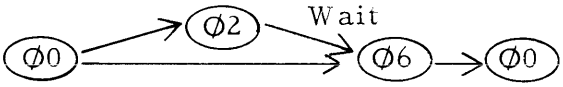
The shift operation occurs during phase Ø3 and is terminated when the Sk flip-flop is set, indicating that the shift counter has reached zero or that a normalized condition exists. Upon termination, the phase counter advances to phase Ø7 from phase Ø3 (or phase Ø1 if the shift count was originally zero).

$$sF1 = Tp\ Sk\ \overline{Ts} + - - -$$
$$sF2 = Tp\ Sk\ \overline{Ts} + - - -$$

During phase Ø7 the next instruction is accessed and at Tp the counter is reset to phase Ø0.

$$rF1 = Tp\ End\ \overline{Sk}$$
$$rF2 = Tp\ End\ \overline{Sk} + - - -$$
$$rF3 = Tp\ End\ \overline{Sk}$$

THREE-CYCLE PLUS N OUTPUT INSTRUCTIONS

Phasing: 

Instructions 10, 12 and 13 are output instructions which are interlocked with a flip-flop designated Rf, which indicates when the external device is "ready" for output. Instructions 10 and 12 advance the counter to phase Ø6, if Rf is set during or before phase Ø0.

$$sF1 = Tp\ \overline{Ts}\ \overline{01}\ 03\ \overline{04}\ \overline{Ia}\ \overline{F1}\ \overline{F3}\ Rf + - - -$$
$$sF2 = Tp\ \overline{Ia}\ Ø0\ 03 + - - -$$

If Rf was not set, the counter is advanced to phase Ø2 (a wait phase).

$$sF2 = Tp\ \overline{Ia}\ Ø0\ 03 + - - -$$

Instruction 13 always advances the counter to phase Ø2 for at least one cycle time.

When Rf is set during phase Ø2 the counter advances to phase Ø6.

$$sF1 = Tp\ \overline{Ts}\ \overline{01}\ 03\ \overline{04}\ \overline{Ia}\ \overline{F1}\ \overline{F3}\ Rf + - - -$$

At the completion of phase Ø6, during which output was executed, the counter is reset to phase Ø0.

$$rF1 = Tp\ End\ \overline{Sk}$$
$$rF2 = Tp\ End\ \overline{Sk} + - - -$$
$$rF3 = Tp\ End\ \overline{Sk}$$
$$rRf = Tp\ End\ \overline{Sk}\ \overline{Ts} + - - -$$

FOUR-CYCLE PLUS N INPUT INSTRUCTIONS

Instructions 30, 32 and 33 are input instructions which are interlocked with the Rf (Ready) flip-flop.

If Rf is not set during or before phase Ø0, the counter advances to phase Ø2 (automatically for instruction 33).

$$sF2 = Tp\ \overline{Ia}\ Ø0\ 03\ + - - -$$

When Rf is set during phase Ø2, or if Rf was set during or before phase Ø0, the counter advances to phase Ø6.

$$sF1 = Tp\ \overline{Ts}\ \overline{01}\ 03\ \overline{04}\ \overline{Ia}\ \overline{F1}\ \overline{F3}\ Rf + - - -$$
$$sF2 = Tp\ \overline{Ia}\ Ø0\ 03 + - - -$$

The counter remains in phase Ø6 for one pulse time (T24) and is reset to phase Ø4 where the input is executed.

$$rF2 = Ø6\ \overline{01}\ 02\ 03 + - - -$$

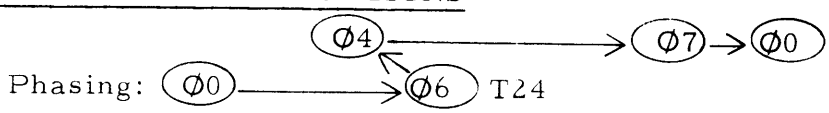
At the completion of phase Ø4 the counter is set to phase Ø7.

$$sF2 = Tp\ \overline{Ts}\ Ø4 + - - -$$
$$sF3 = Tp\ \overline{Ts}\ Ø4 + - - -$$

At the completion of phase Ø7 the counter is reset to phase Ø0 and Rf is reset.

$$rRf = Tp\ End\ \overline{Sk}\ \overline{Ts} + - - -$$

THREE-CYCLE STORE INSTRUCTIONS



Instructions 35, 36 and 37 advance to phase Ø6 at the completion of phase Ø0.

$$sF1 = Tp\ \overline{Ia}\ Ø0\ 03\ 04 + - - -$$

$$sF2 = Tp\ \overline{Ia}\ Ø0\ 03 + - - -$$

$$rF2 = T24\ Ø6\ \overline{01}\ 02\ 03 + - - -$$

After one pulse time in phase Ø6 the counter is reset to phase Ø4 where the information is shifted to the C register for storage. At the completion of phase Ø4 the counter advances to phase Ø7.

$$sF2 = Tp\ \overline{Ts}\ Ø4 + - - -$$

$$sF3 = Tp\ \overline{Ts}\ Ø4 + - - -$$

During phase Ø7 the data is stored in memory and the next command is accessed. At the completion of phase Ø7 the counter is reset to phase Ø0.

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

$$rF2 = Tp\ End\ \overline{Sk} + - - -$$

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

ONE-CYCLE UNCONDITIONAL BRANCH INSTRUCTIONS

Phasing: (Ø0) ⟶ (Ø0)

The 01 (Unconditional Branch) and 23 (Execute) instructions occur within phase Ø0, and there is no change in the phase counter. A similar operation occurs when there is an indirect address bit in the instruction and phase Ø0 is extended as a new address is read from memory.

PROGRAM OPERATOR INSTRUCTIONS

Phasing: (Ø0) ⟶ (Ø7) ⟶ (Ø0)

All instructions affected by the program operator bit act as follows:

The program operator acts as the instruction code and the counter is advanced to phase Ø7.

$$sF1 = Tp\ \overline{Ts}\ P0 - - - + - - -$$

$$sF2 = Tp\ \overline{Ts}\ P0 - - - + - - -$$

$$sF3 = Tp\ \overline{Ts}\ P0 - - - + - - -$$

During phase Ø7 the program operator bit and the instruction code are combined as an entry address to a subroutine and are transferred to the P register, which contains the address of the next instruction to be accessed. At the completion of phase Ø7 the phase counter is reset to phase Ø0.

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

$$rF2 = Tp\ End\ \overline{Sk} + - - -$$

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

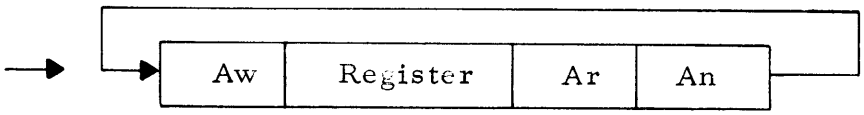
REGISTERS

There are nine registers in the SDS 910 Computer. These registers are implemented in three different methods. The S, M and O registers use the usual RS-type flip-flops. The P and C registers use flip-flops designated as "repeaters". These flip-flops will automatically reset, if there is no set input. However, these flip-flops will not set or reset unless they receive an "enable" signal.

The A, B, X and Word Assembly registers are one-word recirculating registers using dynamic, serial, shift circuits with repeater flip-flops at the "read and write" ends. To hold information, these registers must constantly shift and usually recirculate in a fashion similar to delay lines.

The first or "write" flip-flop (repeater) of the register is designated "w", i.e., Aw, and the last flip-flop, which may or may not be driven by an intermediate "read" flip-flop, is designated by an "n", i.e., An, which means "now". The "read" flip-flop is designated by "r", i.e., Ar.

During normal recirculation the output of the "n" (now) flip-flop is fed to the "w" (write) flip-flop which, in turn, feeds the dynamic register stages. The output of the last register stage either feeds the "n" (now) flip-flop directly or feeds an intermediate "r" (read) flip-flop which, in turn, feeds the "n" (now) flip-flop. In this case, the number of dynamic stages would be reduced by one to allow for the added delay of the "read" flip-flop.

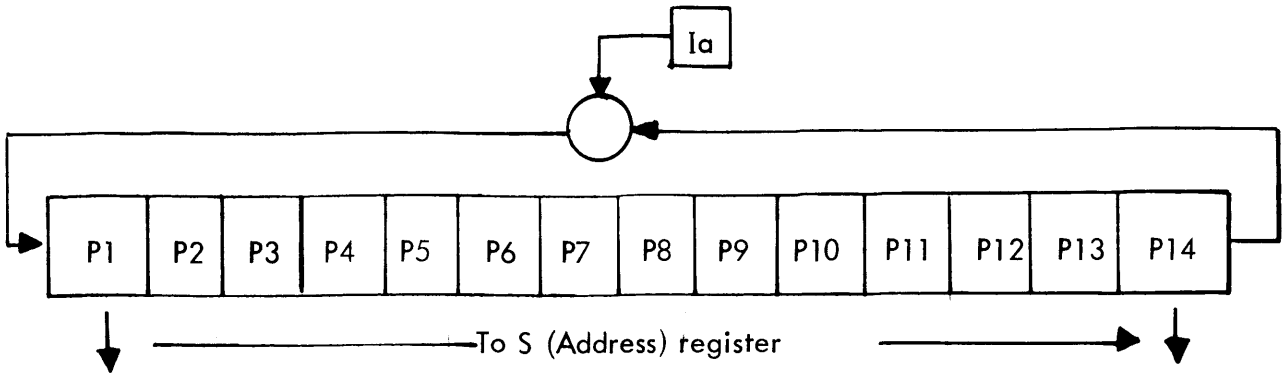


When writing into these recirculating registers a "not recirculate" term is enabled for the register involved, i.e., Anr (A register not recirculated). Enable signals are also required for the repeater inputs, i.e., Ag (A register enabled).

P REGISTER

Function:

The P register is the program counter. It contains the location from which the instruction was taken. It is increased by one, just preceding the time when the memory must be addressed for the next instruction. Since it must transfer its contents to the S (address) register in parallel, in order to keep the memory going at full rate, it is composed of 14 repeater flip-flops. These flip-flops are the repeater-type, since P is basically a shifting register. It is incremented by ring-shifting right for one complete revolution, using the Ia flip-flop as a carry in a half adder.



Implementation:

14 repeater flip-flops

Control Terms:

The P register is enabled by Pg. The P register recirculates during pulse times T23 through T10 and for phases Ø4, Ø5, Ø6 and Ø7, and increments by one during phases Ø4, Ø5 and Ø6, and sometimes during phase Ø7.

$$Pg = Q2\ \overline{Ts}\ Go\ F1 - - - + - - -$$

For certain instruction codes incrementing will occur during phase Ø7. Incrementing is performed using the Ia (Indirect Address) flip-flop as a carry in a half adder.

$$sIa = T24\ F1\ \overline{F3}\ (\overline{Ti}) + \overline{C2}\ \overline{C5}\ \overline{C8}\ (\overline{C3} + \overline{C4})\ Ø0\ \overline{Ia}\ T24\ Go + T24\ Ø7\ Sk + - - -$$

$$rIa = \overline{P14}\ F1\ \overline{T24} - - - + T0\ F1 + - - -$$

The sum is implemented on setting P1.

$$sP1 = \overline{Ia}\ P14\ F1 - - - + Ia\ \overline{P14}\ F1 - - - + - - -$$

The P register is always loaded serially. The new instruction address is transferred to the S (Address) register at pulse time T9 of each End cycle.

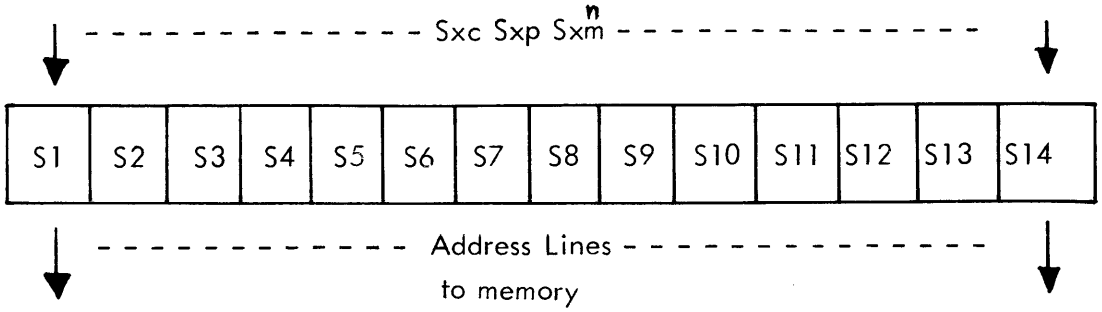
$$Sxp = T9\ \overline{Int}\ End + - - -$$

S REGISTER

Function:

The S register holds the address of the memory location to be accessed. It must remain in a static state when memory is actively reading or writing. The S register receives its address information in parallel from the P register, the C register (the address field of the instruction is in the C register), or from the interrupt address lines.

The S register does not shift and only receives information.



Implementation:

14 RS flip-flops

Control Terms:

The S register is reset or cleared by Sc just prior to receiving the address of the next instruction (T10 End), or the address of the operand (T10 Ø0).

$$Sc = T10\ (End + Ø0)$$

At T9 the S register receives the address of the next instruction from the P register.

$$Sxp = T9\ End + Ø0\ \text{Branch Instructions}$$

During phase Ø0 the S register receives the address of the operand from the C register which now contains the instruction.

$$Sxc = T9\ Ø0\ \overline{P0}\ [Ia + 03 + 02]$$

During the "interrupt" operation the S register receives its address from external address lines designated "N", where "Int" is the Interrupt flip-flop.

$$Sxn = T9\ Int\ \overline{Ts}$$

During certain shifting and arithmetic operations the upper end (S9 - S14) of the S register is used as a shift counter. The basic equations for the S flip-flops are:

$$sS1 = C0\ Sxc + P1\ Sxp$$
$$rS1 = Sc$$

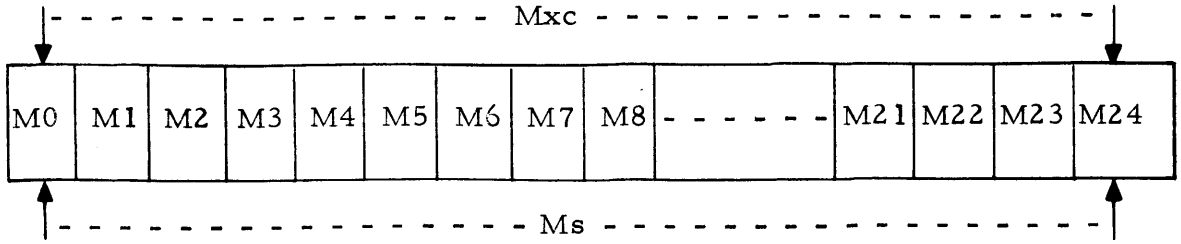
$$sS5 = C4\ Sxc + P5\ Sxp + N5\ Sxn$$
$$rS5 = Sc + - - -$$

$$sS14 = C13\ Sxc + P14\ Sxp + N14\ Sxn$$
$$rS14 = Sc + - - -$$

M REGISTER

Function:

The M register is the Read-Write register for the core memory. It receives the information from memory during a read process and holds the information for storing in memory during a write process.



Implementation:

25 RS flip-flops

Control Terms:

The M register is cleared or reset by Mc just prior to receiving information to be stored or just prior to receiving information from memory. Mc also acts as the pulse which signals memory to start a read-write cycle.

Prior to writing new data:

$$Mc = Tp\ P0 + Tp\ \emptyset 4 + Tp\ \overline{02}\ \overline{03}\ 05\ \emptyset 0\ \overline{Ia}$$
$$+ - - - -$$

Prior to reading:

$$Mc = Q1\ \overline{Q2}\ Q3\ \overline{Q4}\ Q5\ (F1 + \overline{F3} + Ts)\ (Tsm + \overline{Ts}) + - - -$$

Note that the above pulse time is T8 and phases are $\emptyset 0$, $\emptyset 2$, $\emptyset 4$, $\emptyset 5$, $\emptyset 6$, $\emptyset 7$, excluding phases $\emptyset 1$ and $\emptyset 3$.

The contents of the C register are transferred to M for storage at T24, when writing new data into memory.

$$Mxc = T24\ P0$$

In this case, P0 is time-shared and remembers that the data in C is to be stored.

The equations for the M flip-flops are:

$$sM0 = Md0\ Ms + Mxc\ C0$$
$$rM0 = Mc$$
$$\vdots$$
$$sM24 = Md24\ Ms + Mxc\ C24$$
$$rM24 = Mc$$

Ms is the strobe pulse which loads the M register from memory, and Md0 - Md24 represent the data signals from memory.

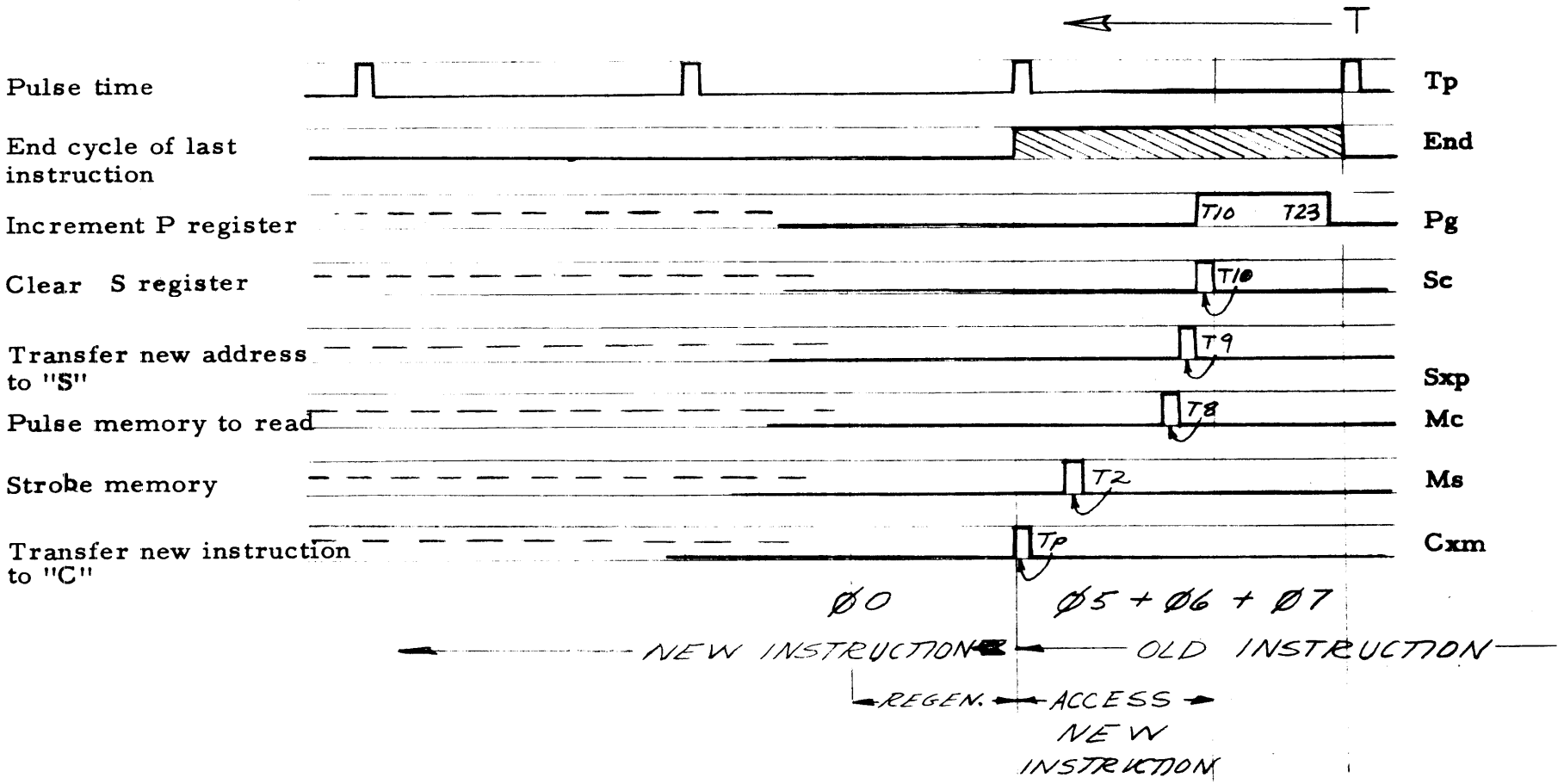
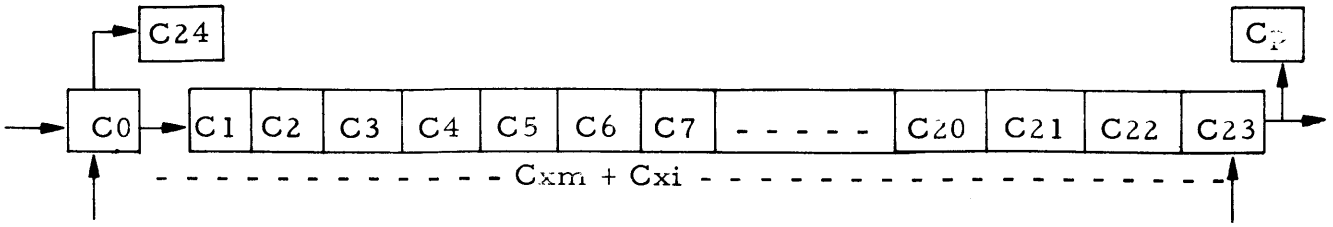


FIGURE 6
ACCESSING AN INSTRUCTION

C REGISTER

Function:

The C register acts as a central transfer station for all information going to and from memory. During certain arithmetic operations C acts as an arithmetic register.



Implementation:

24 repeater and 2 RS flip-flops.

Control Terms:

The C register receives information in parallel from the M register. The next instruction is transferred to the C register at Tp time of an End cycle.

$$Cxm = Tp\ End\ Go + - - -$$

Operands are transferred to C from M during phase Ø0 with the exception of instructions 40, 41, 43, 45, which are branch, skip or register change instructions.

$$Cxm = Tp\ \overline{P0}\ Ø0\quad \overline{[Ø0\ \overline{Ia}\ \overline{02}\ \overline{03}\ 01]}\quad + - - -$$

For instruction 33 the C register receives its information in parallel from external signals.

$$Cxi = Ø2\ 02\ 06$$

The C register is shifted to the right by a term designated Cs. Cs shifts to the right during phases Ø0, Ø1, Ø4 and Ø6; it also shifts to the right during exchange of registers and time-share, which are discussed in their respective sections.

$$Cs = \overline{Tp}\ \overline{T24}\ F1\ \overline{F3} + \overline{Tp}\ \overline{T24}\ \overline{F1}\ \overline{F2} + - - -$$

Since repeater flip-flops are used in the C register, it is "enabled" to receive information by a term designated Cg.

$$Cg = Cs + Cxm + Cxi\ Q1 + - - -$$

Some of the basic equations for the C register flip-flops are:

$$sC0 = Cxm\ M0 + Cxi\ Cd0 + \text{(Kc0)}$$
$$+ - - - - - + - - - - -$$

(NOTE: Other C0 inputs are described in Instruction explanations.)

$$sC1 = Cs\ C0 + Cxm\ M1 + Cxi\ Cd1 + (Kc1)$$

$$\vdots$$

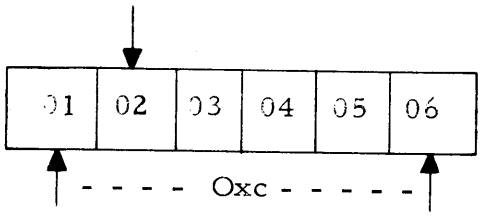
$$sC23 = Cs\ C22 + Cxm\ M23 + Cxi\ Cd23 + (Kc23)$$

(Kc) is the C register manual set button on the control panel.

O REGISTER

Function:

The O register usually contains the 6-bit instruction code during the execution of the instruction. This register does not shift and only receives information.



Implementation:

6 RS flip-flops.

Control Terms:

The O register is cleared or reset by a term designated Oc at the end of each End cycle, at the start of phase Ø7, and at the end of phase ØC for Instruction 23.

$$Oc = Tp\ End\ Go + \quad Ø7 + \overline{01}\ \overline{03}\ 05\ Tp\ \overline{Ia} + - - -$$

The O register is always reset to a (20) by setting 02.

The O register receives the instruction code from the C register at T24 of Ø0.

$$Oxc = T24\ Ø0\ Go\ \overline{Ia}\ \overline{C2}$$

In this equation C2 represents the P0 (program operator) bit, in which case the instruction code would remain a (20).

The basic equations for the O register flip-flops are:

$$s01 = Oxc\ C3 \qquad s04 = Oxc\ C6$$

$$r01 = Oc \qquad r04 = Oc$$

$$s02 = Oc \qquad s05 = Oxc\ C7$$

$$r02 = Oxc\ \overline{C4} \qquad r05 = Oc$$

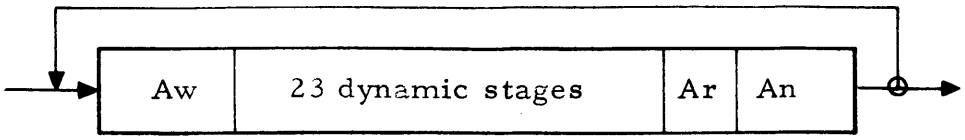
$$s03 = Oxc\ C5 \qquad s06 = Oxc\ C8$$

$$r03 = Oc \qquad r06 = Oc$$

A REGISTER

Function:

The A register is the accumulator for the computer and is employed in all arithmetic, logical and shifting operations.



Implementation:

The write flip-flop (repeater) is followed by 23 stages of dynamic delay. The read flip-flop (Ar) and the "now" flip-flop (An) are both repeaters. To maintain recirculation the Aw and Ar flip-flops are always enabled. The An flip-flop is disabled for certain instructions. The A register is exactly 26 bits in length and must recirculate to maintain internally stored information, as in a delay line.

Control Terms:

The An flip-flop is enabled by a term designated Ag, which is disabled during Instruction 64 (multiply) and shift right.

$$Ag = Rf\ 01\ \overline{T24}\ \overline{Ts}$$

The recirculation of the A register is inhibited by Anr (A not circulate). The recirculation of the A register is blocked during the execution (Ø6) of logical instructions and Add and Subtract instructions.

$$Anr = \overline{02}\ 03\ 04\ Ø6 + - - -$$

It is inhibited during the loading of A from memory (76).

$$Anr = 01\ 02\ 03\ 04\ \overline{06}\ Ø6 + - - -$$

It is inhibited during the register change (46).

$$Anr = \overline{02}\ \overline{03}\ 04\ \overline{Ts} + - - -$$

It is also inhibited during the Multiply step (64), Divide step (65), Right Shift (66) and Left Shift (67).

$$Anr = Rf\ 01\ \overline{T24}\ \overline{Ts} + F2\ \overline{03}\ 04\ 06\ \overline{Ts} + - - -$$

For instructions 64 and 66, Rf is time-shared and designates that the A B register pair will shift to the right.

The basic recirculation equations are shown below. Additional input terms are discussed under Instructions.

$$sAw = \overline{Anr}\ An\ \overline{Tp} + - - -$$

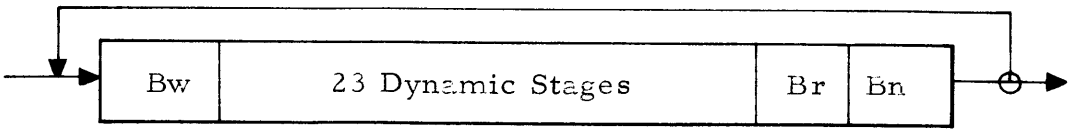
$$sAn = Ar\ \overline{[- - - -]} + - - -$$

Ar is fed from the last dynamic stage of the A register.

B REGISTER

Function:

The B register acts as an extension of the least significant end of the A register for double precision operations and for shift operations. The B register holds operands for the multiply (64) and divide instructions (65).



Implementation:

The B register is implemented in the same manner as the A register.

Control Terms:

Bw (write) and Br (read) are always enabled to maintain recirculation and Bn (now) is enabled by the same term that enables An in the A register.

$$Ag = Bg = \overline{Rf\ 01\ \overline{T24}\ \overline{Ts}}$$

The recirculation of the B register is inhibited by Bnr (B not recirculate). The contents of the B register are inhibited during the loading of B from memory (75).

$$Bnr = 01\ 02\ 03\ 04\ \overline{05}\ \emptyset 6 + - - -$$

It is inhibited during the register change instruction (46).

$$Bnr = \overline{02}\ \overline{03}\ 04\ \overline{Ts} + - - -$$

It is also inhibited during shift (66, 67) and Multiply, Divide (64, 65) instructions.

$$Bnr = Rf\ 01\ \overline{T24}\ \overline{Ts} + F2\ \overline{03}\ 04\ 06\ \overline{Ts} + - - -$$

The recirculation equations for the B register are shown below. Additional input terms are described under Instructions.

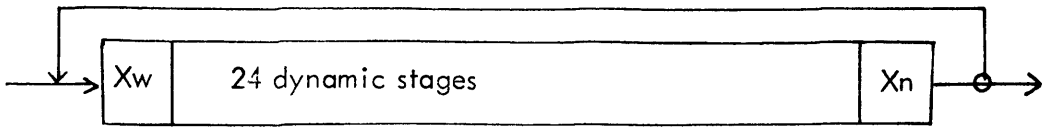
$$sBw = \overline{Bnr}\ Bn + - - -$$

$$sBn = Br$$

X REGISTER

Function:

The X register is the Index register. The least significant 14 bits will be added to the address of the instruction before execution, if there is an index tag in position T1 of the instruction. Indexing takes place during phase $\emptyset 0$ and is discussed in detail under Indexing.



Implementation:

The Xw flip-flop (repeater) is followed by 24 stages of dynamic delay and the last dynamic stage feeds the Xn (now) repeater.

Control Signals:

The Xw and Xn repeaters are always enabled to maintain recirculation. Recirculation of the X register is inhibited by Xnr (X not recirculate) during certain instructions.

Recirculation is inhibited during the loading of X from memory (71),

$$Xnr = 01\ 02\ 03\ \overline{04}\ \overline{05}\ 06\ \emptyset 6 + - - -$$

Increment X and branch (41),

$$Xnr = \emptyset 0\ 01\ \overline{02}\ \overline{03}\ \overline{05}\ Ia + - - -$$

and during normalize (67) and copy effective address to X (77),

$$Xnr = \emptyset 3\ 06\ S2 + 01\ 02\ 03\ 04\ 05\ 06\ \overline{Ia}\ \emptyset 0\ Q2$$

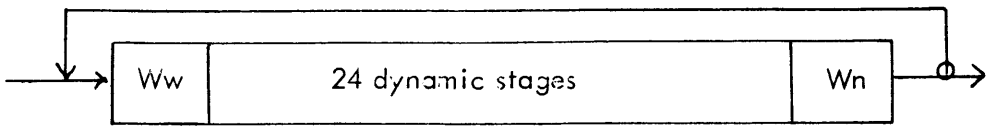
S2 is time-shared during normalize. The basic recirculation term for the X register is:

$$sXw = \overline{Xnr}\ Xn + - - -$$

WORD ASSEMBLY REGISTER (WAR)

Function:

The WAR is part of the Buffer System and is described in detail in the W Buffer section. Its basic function is to hold information on input and output.



Implementation:

The WAR is implemented in the same manner as the X register.

Control Terms:

Refer to the W Buffer Section.

MEMORY CONTROL

The computer generates the basic timing signals used by the core memory.

The M (memory) register receives the information in parallel from the sense amplifiers during reading and controls the Z inhibit lines to the core memory during writing.

The memory cycle is out of phase with the computer cycle. Reading is started at T7 and the word read is strobed into the M register at T2, so that the word is ready for the computer to use before Tp. The memory write cycle is performed during the first part of the computer cycle, from T21 through T11.

The address that controls the memory is usually set up in the computer S register. This register is cleared at T10 time and loaded in parallel at T9 time from the C or P registers. At T8 the Mc signal clears the M register and starts the memory cycle. The memory is cycled every computer cycle, except when the computer is in phase $\phi 1$ or $\phi 3$ for shift instructions and except during the second cycle of a time-share interlace operation.

$$Mc = Q1\ \overline{Q2}\ Q3\ \overline{Q4}\ Q5\ (F1 + \overline{F3} + Ts)\ (Tsm + \overline{Ts})$$

where:

$$Q1\ \overline{Q2}\ Q3\ \overline{Q4}\ Q5 = T8$$

This signal (Mc) clears the M register. It also sets Mg, the flip-flop that gates all timing signals to the memory.

$$sMg = Mc\ Q1\ (Me)\ \overline{St}$$
$$rMg = Q1\ Q2\ Q3\ \overline{Q4}\ Mg = T11$$

The (Me) signal is a signal from the automatic "start-up, shut-down" relays and it is normally true.

The read timing signal is true from T7 through T0.

$$Mrt = Mg\ \overline{Q2}\ (\overline{Q3} + Q4)$$

The memory is strobed into the M register at T2 by Ms.

$$Ms = Mg(T5 - T0)\ \ Q1\ \overline{Q4}\ \overline{Q5}$$
$$sM0 = Ms\ Md0 + - - -$$
$$\vdots$$
$$sM24 = Ms\ Md24 + - - -$$

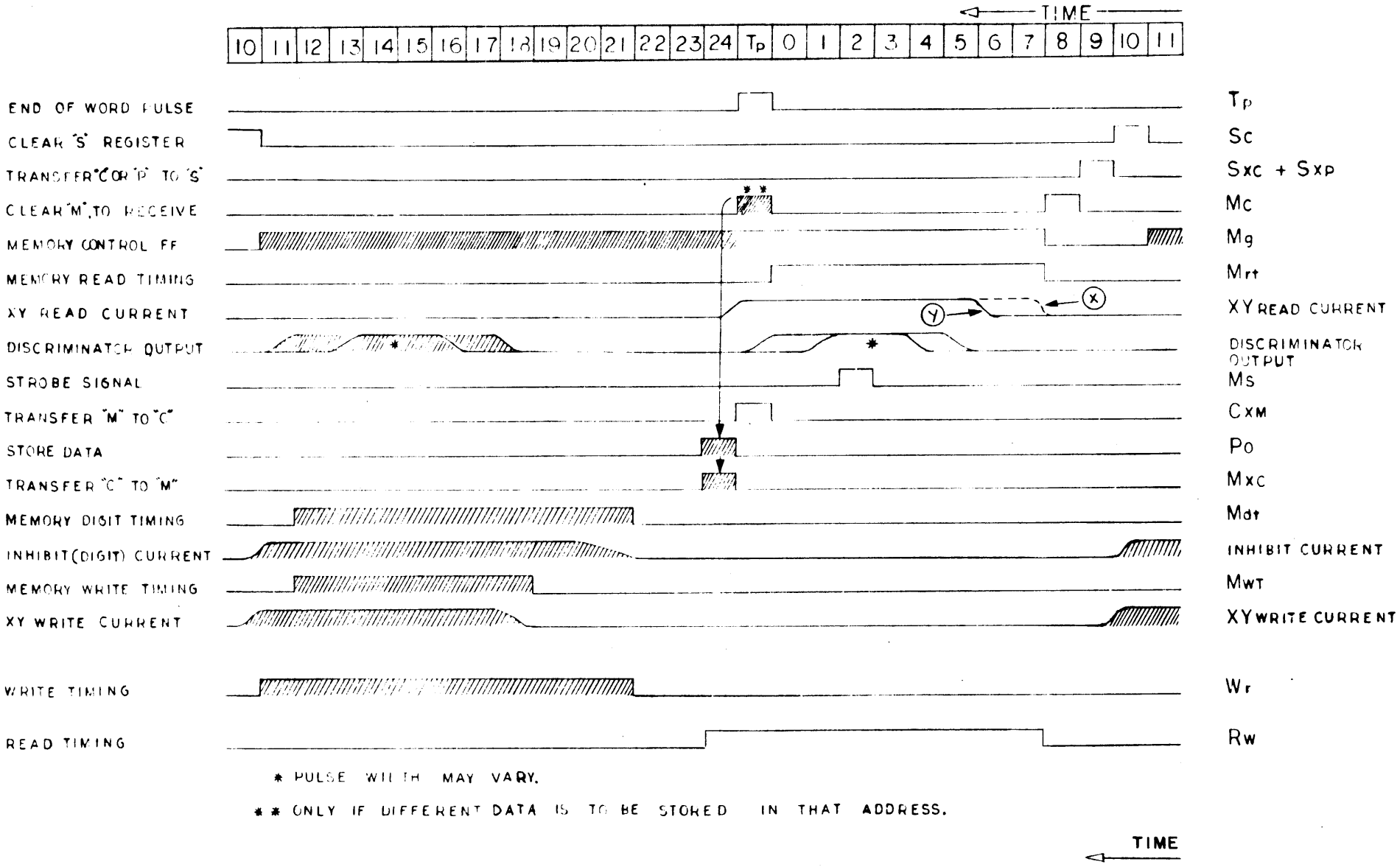


FIGURE 7

COMPUTER MEMORY CYCLE

If the M (memory) register is to be copied in parallel to the C register, it is done at Tp by Cxm. The contents of the M register are then regenerated (written back) into memory. If new information is to be stored in memory, then the M register is cleared at Tp by Mc.

This sets the P0 flip-flop, if it was not already set. P0 then remembers to transfer the contents of C to M at the following T24 time for storage.

$$sP0 = Mc\ Tp$$

$$rP0 = T24\ P0$$

$$Mxc = T24\ P0$$

The write timing signal to the memory is true from T18 through T12.

$$Mwt = Mg\ Q2\ (Q1\ Mdt + \overline{Q3}\ Q5)$$

The two terms in the OR gate overlap in time so that there is no cross-over voltage spike.

The digit timing pulse supplied to the memory is true from T21 through T12.

$$Mdt = Mg\ Q2\ (\overline{Q3} + Q4)$$

Two signals control the direction of the current through the cores, as the current must be reversed between the "Read" and "Write" periods.

$$Rw = \overline{Q2}\ Mg$$

$$Wr = (Q2\ Mg)\ (Mdt + Q1)$$

ADDER

The adder is a full adder composed of the Xz (augend) input, the Yz (addend) input and the Cz carry flip-flop. It is completely serial and there is no "logical" delay through the adder.

The Yz input always delivers the word from memory. The adder is used in many instructions and during indexing. The basic equations are shown below. The adder is discussed in additional detail under Instructions and Indexing.

$$\begin{aligned} \text{Add} = {} & Xz\ Yz\ Cz + Xz\ \overline{Yz}\ \overline{Cz} \\ & + \overline{Xz}\ Yz\ \overline{Cz} + \overline{Xz}\ \overline{Yz}\ Cz \end{aligned}$$

$$\begin{aligned} \overline{\text{Add}} = {} & \overline{Xz}\ \overline{Yz}\ \overline{Cz} + \overline{Xz}\ Yz\ Cz \\ & + Xz\ \overline{Yz}\ Cz + Xz\ Yz\ \overline{Cz} \end{aligned}$$

Only the $\overline{\text{Add}}$ is implemented and the Add is formed by negation.

INDEX OPERATION

All instructions begin at Ø0 T24 with the instruction in the C register. During phase Ø0 the C register is shifted right, through the adder, and back to itself.

Enable C register:

$$Cs = \overline{Tp}\ \overline{T24}\ \overline{F1}\ \overline{F2} + - - -$$

As it is shifted right, through the adder, it enters through Yz.

$$Yz = C23\ \overline{F1}\ \overline{F3}\ \overline{Ts} + - - -$$

The carry flip-flop Cz starts in a reset condition.

$$rCz = Ø0\ \overline{Ix} + - - -$$

If there is an index bit in C1 at Ø0 T24, the Index flip-flop will be set.

$$sIx = T24\ Ø0\ C1\ Go + - - -$$

It is reset at the end of phase Ø0.

$$rIx = Tp + - - -$$

During the addition the Index register (X) is selected by Xz (augend input), if Ix is set.

$$Xz = Xn\ \overline{F1}\ \overline{F3}\ Ix + - - -$$

If the Index flip-flop Ix is not set, then zero will be added to the C register and the address will remain the same.

$$sC0 = Ø0\ Add\ Q2\ \overline{P0}\ \overline{Ø0\ \overline{Ia}\ \overline{02}\ \overline{03}\ 01} + - - -$$

Note that the addition only takes place from T23 through T10. The last term is to inhibit indexing in this manner during branch instructions 41 and 43. Branch instructions (01, 41, and 43) index by running the adder directly into P1.

$$sP1 = Ø0\ \overline{Ia}\ \overline{02}\ \overline{03}\ \ Add$$

$$Pg = (Kr)\ Q2\ \overline{Ts}\ Go\ \ Ø0\ \overline{Ia}\ \overline{02}\ \overline{03}$$

At T9 the address has shifted completely around so that it is in the upper half of the C register and, at this time, it is transferred in parallel to the S register in order to access the operand.

$$Sxc = T9\ Ø0\ \overline{P0}\ (- - - + 03 + 02) + - - -$$

INDIRECT ADDRESS OPERATION

All instructions begin at Ø0 T24 with the complete instruction in the C register. At the end of pulse time T24 the 6-bit instruction code is transferred to the instruction register.

$$Oxc = T24\ Ø0\ Go\ \overline{Ia}\ \overline{C2}$$

If the instruction contains an indirect address bit in C9 during T24 Ø0, the indirect address flip-flop, Ia, is set.

$$sIa = T24\ \overline{Ia}\ Go\ Ø0\ \overline{C2}\ C9 + - - -$$

C2 is the program operator bit which inhibits indirect addressing.

An indirect address bit programmed with a single-cycle instruction (except BRU and EXU) has no effect, since the Ia flip-flop is set in any event to increment the P register (Program Counter). The indirect addressing in this case is ignored.) For any other instruction the Ia flip-flop inhibits the phase count and allows an additional cycle of phase Ø0.

Indexing may occur on the first address during phase Ø0 and at T9 the address (of the "indirect instruction") is transferred to the S register for accessing. At pulse time Tp the indirect instruction is transferred from M to the C register.

At T24 of this second cycle of phase Ø0 the transfer of the instruction code to the O register is inhibited by Ia.

$$Oxc = Ø0\ T24\ \overline{Ia}\ \overline{C2}\ Go$$

Ia is reset at this second T24 unless another indirect address bit is present. In this case it is not reset and another cycle of indirect address takes place.

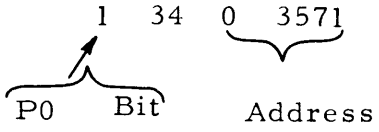
$$rIa = T24\ Ø0\ \overline{C9}\ Ia\ Go\ + - - -$$

C9 represents the new indirect address bit. When Ia is finally reset at some T24, the real operand (effective address) is obtained from memory and the instruction is executed.

Indexing will take place on any indirect address cycle where an index bit is present.

PROGRAM OPERATOR

When an instruction is executed that has a "program operator" bit, the computer stores the contents of the P register in memory word zero and transfers control to the address specified by the seven-bit instruction code, including the program operator bit itself. For example, an instruction,



will cause a transfer to word 134 after storing the contents of the P register (location of above instruction) in word zero. The contents of the overflow flip-flop are stored in the sign position of word zero and the overflow flip-flop is reset. An indirect address bit is also stored in word zero. The addressed subroutine can then indirect address word zero, which, in turn, can indirect address the location of the above instruction, which, in turn, will address the operand in 03571.

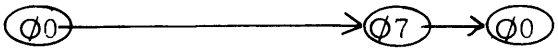
At time Ø0 T24 (start of the instruction) the P0 (program operator) flip-flop is set, if C2 in the C register contains a "program operator" bit.

$$sP0 = C2\ (Ø0\ \overline{Ia}\ T24\ Go) + - - -$$

It is reset at the start of the next cycle.

$$rP0 = T24\ P0 + - - -$$

The operation takes two cycle times. The phasing is shown below with one cycle time in each phase.



During Ø0 Q2 the P register is shifted into C for eventual storage in word zero.

$$sC0 = (P0\ Ø0)\ Q2\ P14 + (P0\ Ø0)\ T9 + (P0\ Ø0)\ Of\ T0 + - - -$$

The second term above inserts an indirect address bit in word zero so that the true operand of the original instruction can be easily accessed indirectly. The third term inserts the contents of the Of (overflow) flip-flop in the sign position,

$$rOf = Ø0\ P0\ T0 + - - -$$

and the overflow flip-flop is reset.

The O register is not set at Ø0 T24 as usual, because the transfer is inhibited.

$$Oxc = Ø0\ T24\ \overline{Ia}\ \overline{C2}\ Go$$

The O register remains reset in the 20 (NOP) configuration. As P is shifting into C, the 7-bit instruction code, which is now the transfer address in C, is shifted into the P register.

$$sP1 = C8\ P0\ Ø0\ \overline{Q1}\ Q2 + - - -$$

This is followed by 7 zeros into the P register.

$$Pg = (Kr)\ Q2\ \overline{Ts}\ Go\ P0\ Ø0 + - - -$$

At Ø0 Tp the memory register is cleared and memory is pulsed for storage,

$$Mc = Tp\ P0 + - - -$$

and at T24 Ø7 the word constructed in C is transferred to the M register for storage and P0 is reset.

$$Mxc = T24\ P0$$

$$rP0 = T24\ P0 + - - -$$

The S register is cleared at T10 Ø0, as usual, for receiving the address of the operand,

$$Sc = T10\ \emptyset 0$$

But P0 inhibits the usual C to S transfer.

$$Sxc = T9\ \emptyset 0\ \overline{P0}\ (Ia + 03 + 02)$$

Therefore, the address remains zero. At T_P Ø0 the phase changes to phase Ø7.

$$sF1 = sF2 = sF3 = T_P\ \overline{Ts}\ P0 + - - -$$

During phase Ø7 word zero is stored, P is transferred to S at T_P and the instruction from this new address is accessed.

Note that address positions 100 through 177 must always be reserved for program operator instructions and that these instructions, in turn, must be BRU instructions to transfer to other starting locations outside the reserved field.

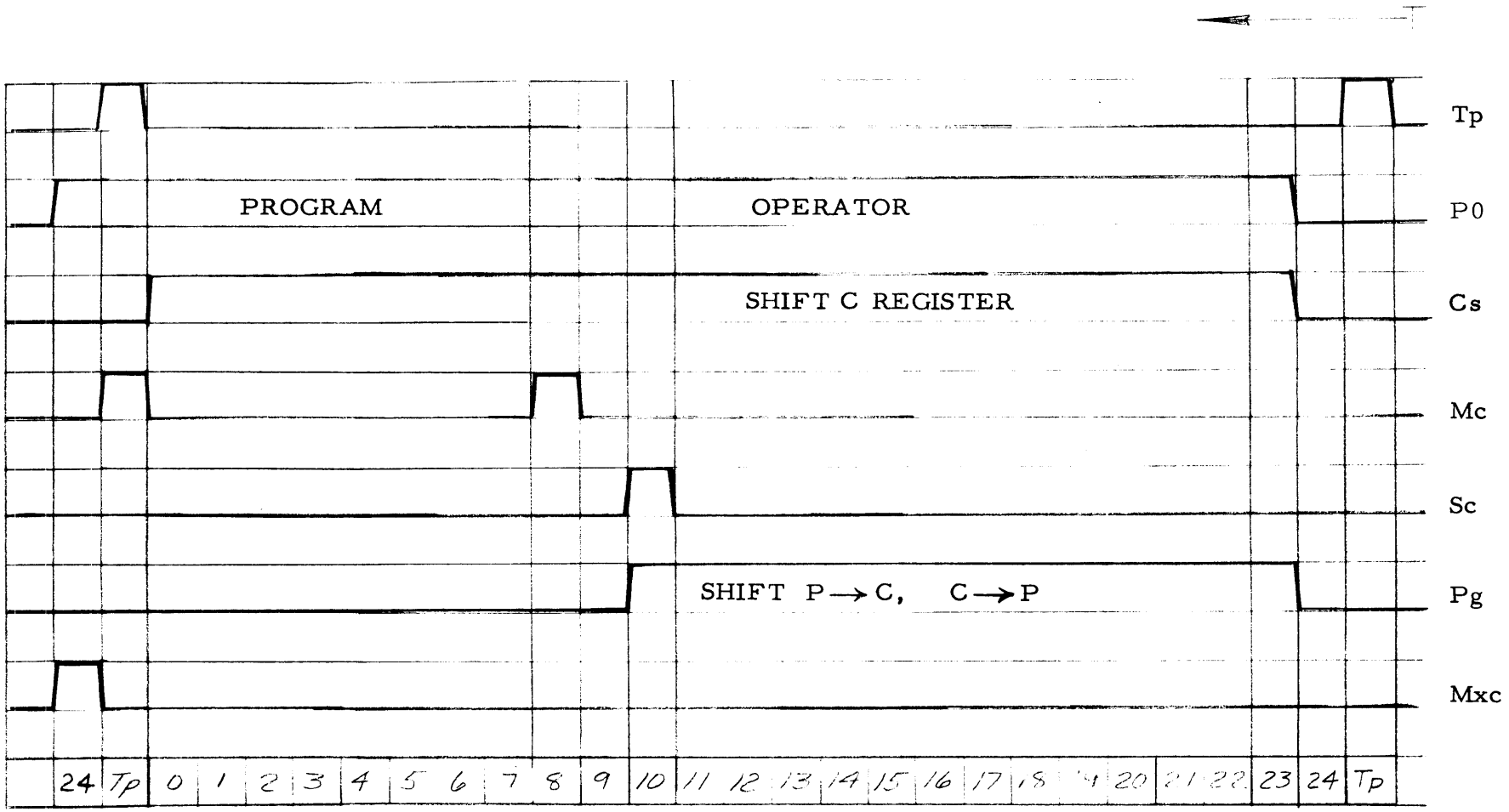


FIGURE 8

PROGRAM OPERATOR

INTERRUPT

The computer can be interrupted by an external signal through the priority interrupt logic which causes the computer to access an address assigned to the interrupting channel. The location addressed usually contains a 43 (BRM) instruction, which stores the location of the present instruction and branches to the interrupt subroutine.

Interrupt channels 3 and 4 are designated (I3) and (I4), with (I3) being of higher priority. Interrupt channels can be used in additional groups of 16. When interrupt line (I4) requests an interrupt, a flip-flop designated Is4 is set to establish the request.

$$sIs4 = (\underline{T22 - T17})\ (I4)$$

The underlined term is used as a clocking term.

If there are no higher priority requests present, then an "interrupt request" is sent to the computer,

$$Ir = - - - + Is4\ \overline{Is3}\ \overline{Ip4}\ \overline{Is1}\ \overline{Is2} + - - -$$

Note that a request will not be sent, if Is3 (a higher priority) is true, or priorities higher than Is3 are true. The computer will set the interrupt flip-flop (Int) at T10 of the first end cycle, if a request is present.

$$sInt = T10\ End\ Ir$$

The time elapsed between the request signal Ir and the actual interrupt Int varies according to the length of time necessary to complete the instruction that the computer is performing when the request is made.

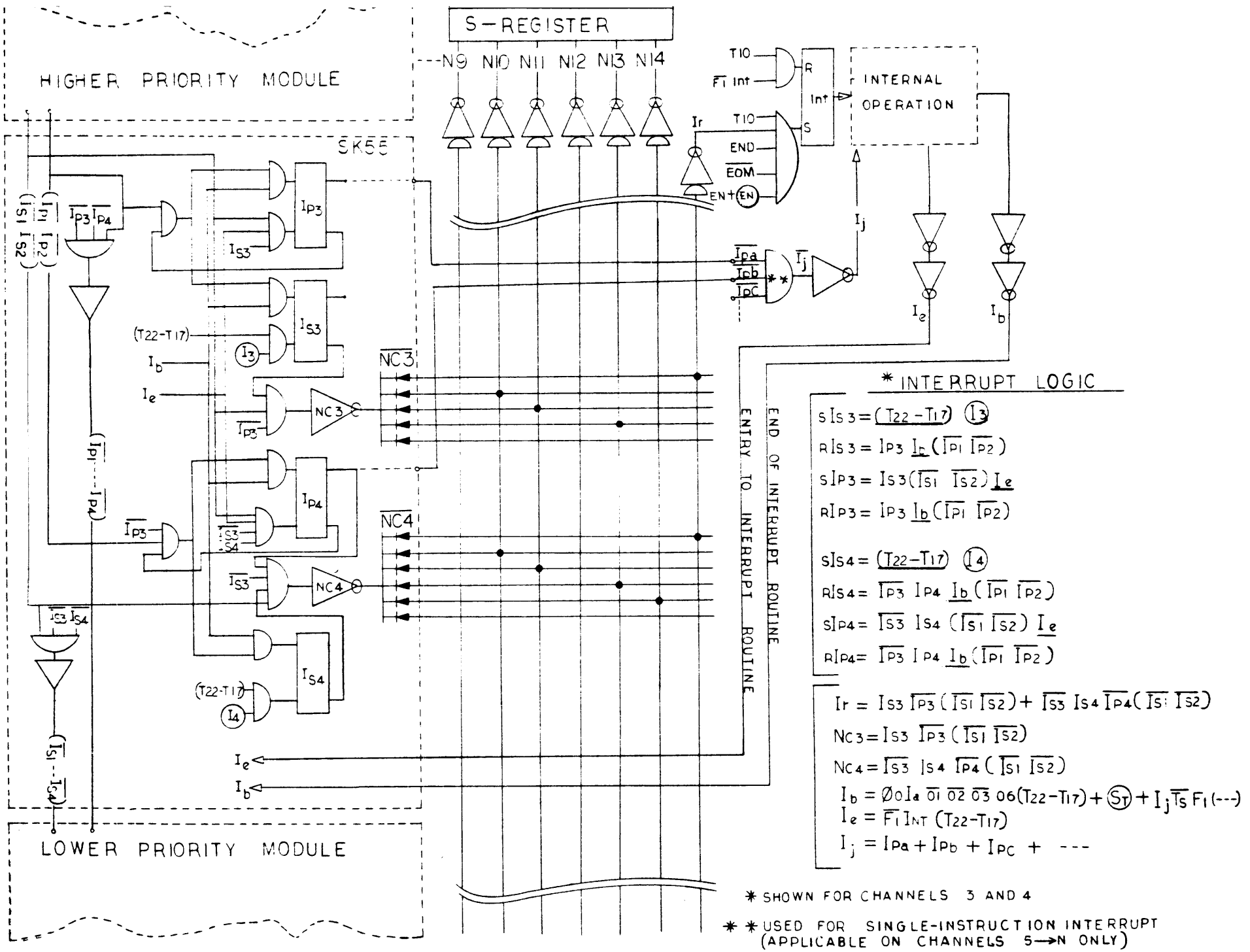
If the computer was halted from a Halt instruction (not a parity error), and the Control switch is in the "Run" position, an interrupt request will allow the computer to run.

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

Ht is a Halt control flip-flop and Cp is the parity flip-flop.

When the Int (Interrupt) flip-flop is on, the transfer from P to S which usually takes place is inhibited.

$$Sxp = T9\ End\ \overline{Int} + - - -$$



* INTERRUPT LOGIC

$sIs3 = \underline{(T_{22}-T_{17})}\ (I_3)$

$RIs3 = I_{P3}\ \underline{I_b}\ (\overline{I_{P1}}\ \overline{I_{P2}})$

$sIP3 = I_{S3}(\overline{I_{S1}}\ \overline{I_{S2}})\ \underline{I_e}$

$RIP3 = I_{P3}\ \underline{I_b}\ (\overline{I_{P1}}\ \overline{I_{P2}})$

$sIs4 = \underline{(T_{22}-T_{17})}\ (I_4)$

$RIs4 = \overline{I_{P3}}\ I_{P4}\ \underline{I_b}\ (\overline{I_{P1}}\ \overline{I_{P2}})$

$sIP4 = \overline{I_{S3}}\ I_{S4}\ (\overline{I_{S1}}\ \overline{I_{S2}})\ \underline{I_e}$

$RIP4 = \overline{I_{P3}}\ I_{P4}\ \underline{I_b}\ (\overline{I_{P1}}\ \overline{I_{P2}})$

$I_r = I_{S3}\ \overline{I_{P3}}\ (\overline{I_{S1}}\ \overline{I_{S2}}) + \overline{I_{S3}}\ I_{S4}\ \overline{I_{P4}}\ (\overline{I_{S1}}\ \overline{I_{S2}})$

$NC3 = I_{S3}\ \overline{I_{P3}}\ (\overline{I_{S1}}\ \overline{I_{S2}})$

$NC4 = \overline{I_{S3}}\ I_{S4}\ \overline{I_{P4}}\ (\overline{I_{S1}}\ \overline{I_{S2}})$

$I_b = \emptyset 0 I_a\ \overline{01}\ \overline{02}\ \overline{03}\ 06(T_{22}-T_{17}) + (S_T) + I_j\ \overline{I_S}\ F_1(---)$

$I_e = \overline{F_1}\ I_{NT}\ (T_{22}-T_{17})$

$I_j = I_{Pa} + I_{Pb} + I_{Pc} + \cdots$

* SHOWN FOR CHANNELS 3 AND 4

** USED FOR SINGLE-INSTRUCTION INTERRUPT (APPLICABLE ON CHANNELS 5→N ONLY)

PRIORITY INTERRUPT SYSTEM

FIGURE 9

Instead, ten address lines labeled N5 through N14, controlled by the priority interrupt system, are transferred to the least significant bits of S to address a pre-assigned subroutine.

$$Sxn = T9\ Int\ \overline{Ts}$$

$$sS14 = Sxn\ N14$$ (S5-S13 are set in a similar way.)

During the first cycle after the last "End" an interrupt "acknowledgment" signal (Int $\overline{F1}$) is sent back to the priority logic and the Int flip-flop is reset.

$$rInt = T10\ (Int\ \overline{F1})$$

$\overline{F1}$ signifies that the computer is not in an End cycle. The interrupt acknowledgment signal sets a flip-flop labeled Ip4 (Ip3 for the higher channel) which, in turn, cancels the request and signifies that the computer is now in interrupt subroutine "4"

$$sIp4 = Is4\ \overline{Is3}\ \underline{Ie}\ \overline{Is1}\ \overline{Is2}$$

$$Ir = - - - + \overline{Is3}\ Is4\ \overline{Ip4}\ \overline{Is1}\ \overline{Is2}$$

If a higher priority interrupt ((I3)) occurs during the execution of interrupt subroutine "4", it will interrupt accordingly. Or if (I3) requests an Interrupt before interrupt (I4) is acknowledged, then interrupt (I3) will be acknowledged instead and interrupt subroutine "4" will not begin until interrupt subroutine "3" is completed.

$$Ir = Is3\ \overline{Ip3}\ (\overline{Is1}\ \overline{Is2}) + - - -$$

The end of an interrupt subroutine is signaled by a term labeled Ib, which is enabled by an 01 (BRU) instruction with indirect addressing.

$$Ib = (\emptyset 0\ Ia\ \overline{01}\ \overline{02}\ \overline{03}\ 06)\ (T22 - T17)$$

This term will reset only the highest priority interrupt channel as the others are either in a request state or were interrupted in the execution of their routine.

$$rIs3 = Ip3\ \underline{Ib}\ \overline{Ip1}\ \overline{Ip2} + \overline{St}\ \boxed{dc}$$

$$rIp3 = Ip3\ \underline{Ib}\ \overline{Ip1}\ \overline{Ip2} + \overline{St}\ \boxed{dc}$$

$$rIs4 = \overline{Ip3}\ Ip4\ \underline{Ib}\ \overline{Ip1}\ \overline{Ip2} + \overline{St}\ \boxed{dc}$$

$$rIp4 = \overline{Ip3}\ Ip4\ \underline{Ib}\ \overline{Ip1}\ \overline{Ip2} + \overline{St}\ \boxed{dc}$$

If the end cycle that causes Int to be set is terminating a skip instruction and a skip occurs, the phase is changed to phase $\emptyset 7$ to increment P before the interrupt instruction is performed. In this case a higher priority interrupt request could occur during phase $\emptyset 7$, but the N lines are transferred again during phase $\emptyset 7$ so that the higher priority subroutine would be entered. This is the reason for $\overline{F1}$. The computer cannot acknowledge the lower priority program because it will be entering a higher priority program instead. Study the interrupt timing diagrams for these relationships.

If Ts is set after Int is set and if a higher priority is set during the Ts time, the 43 instruction from the first interrupt will be executed before the higher priority interrupt begins because the acknowledgment of the interrupt is not qualified by $\overline{Ts}$,

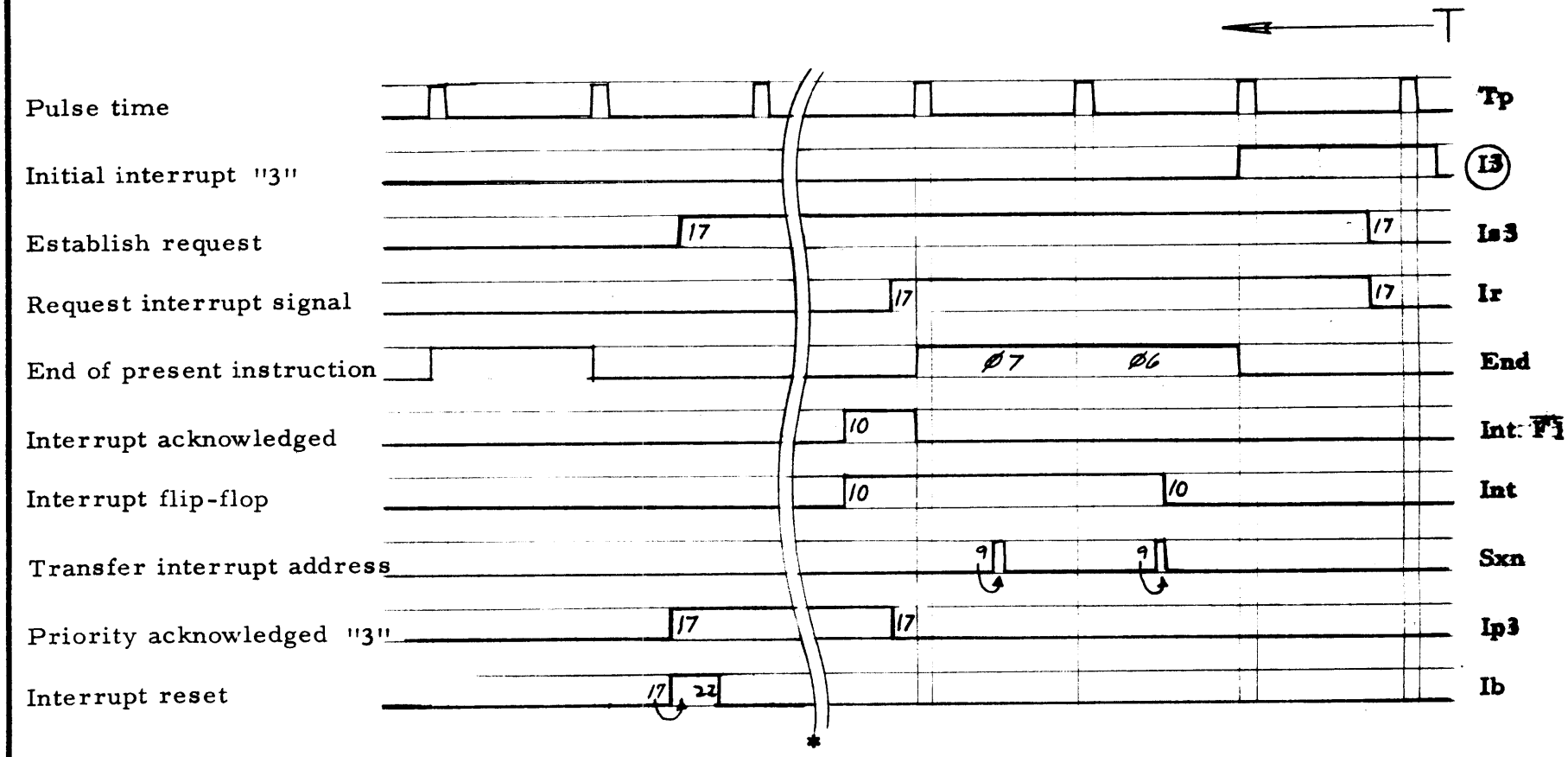
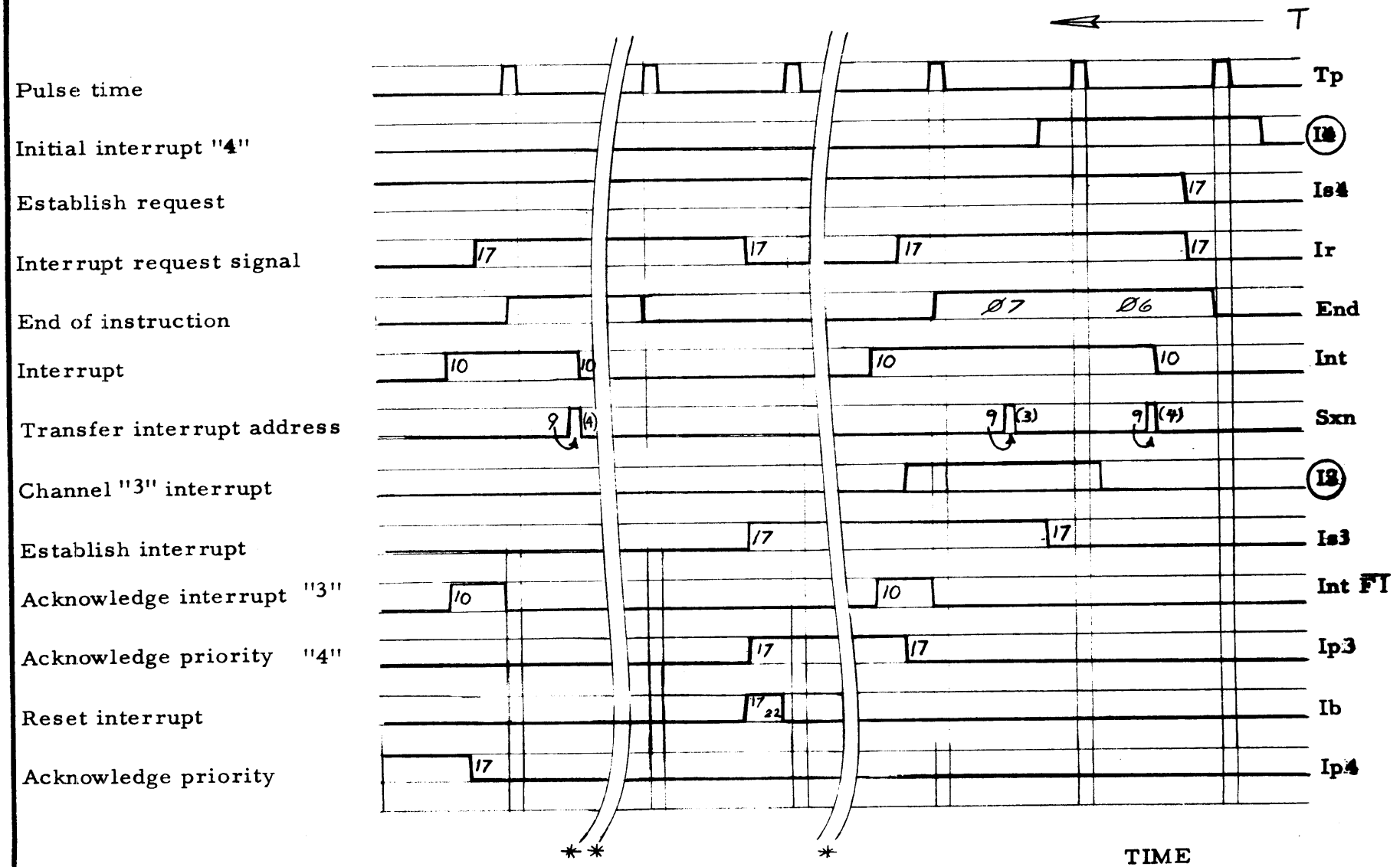


FIGURE 10

INTERRUPT TIMING A

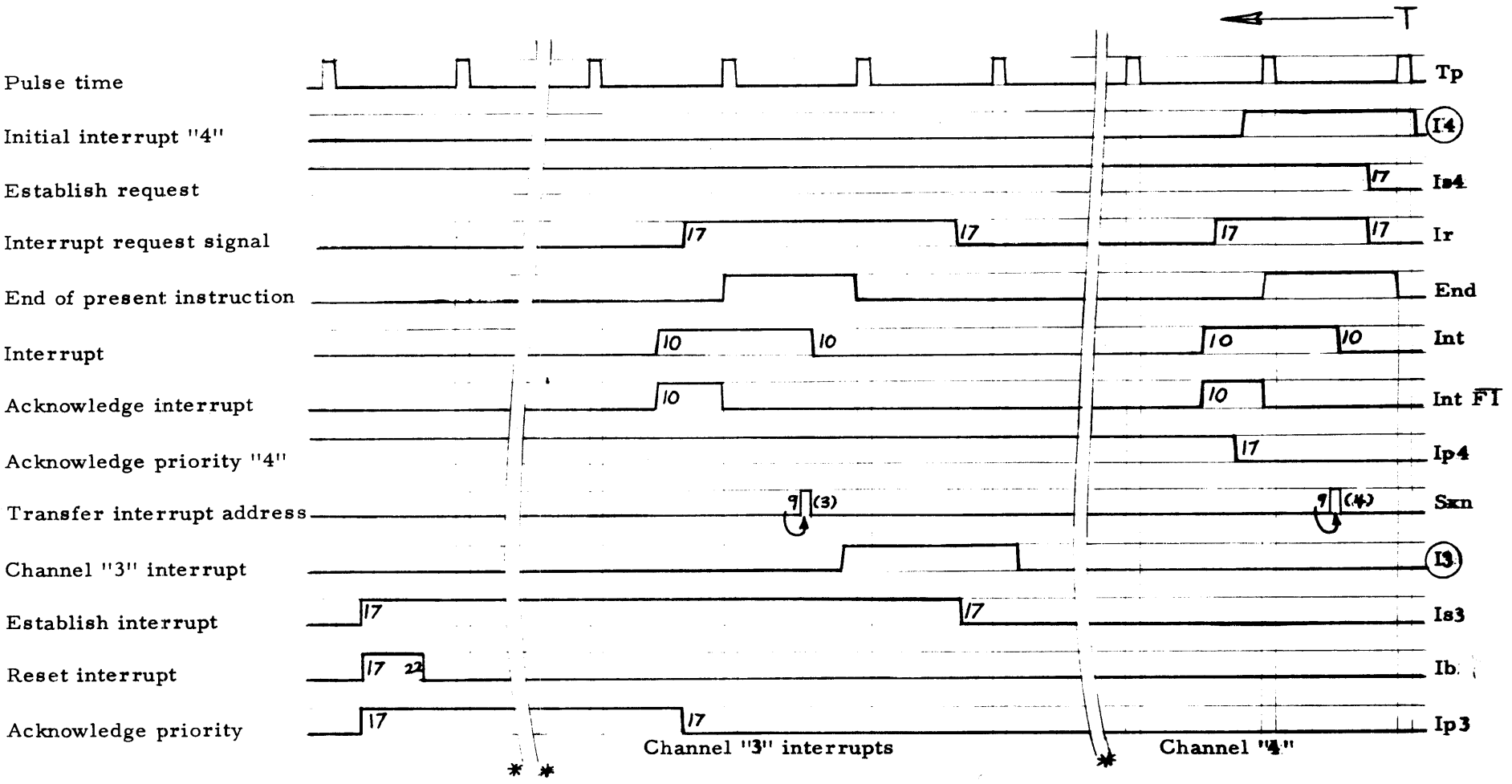
Channel (13)



* Execution of interrupt program "3"
** Execution of any program of higher priority than "3"

FIGURE 11

INTERRUPT TIMING B
"4" interrupted by "3" before "4" is acknowledged.



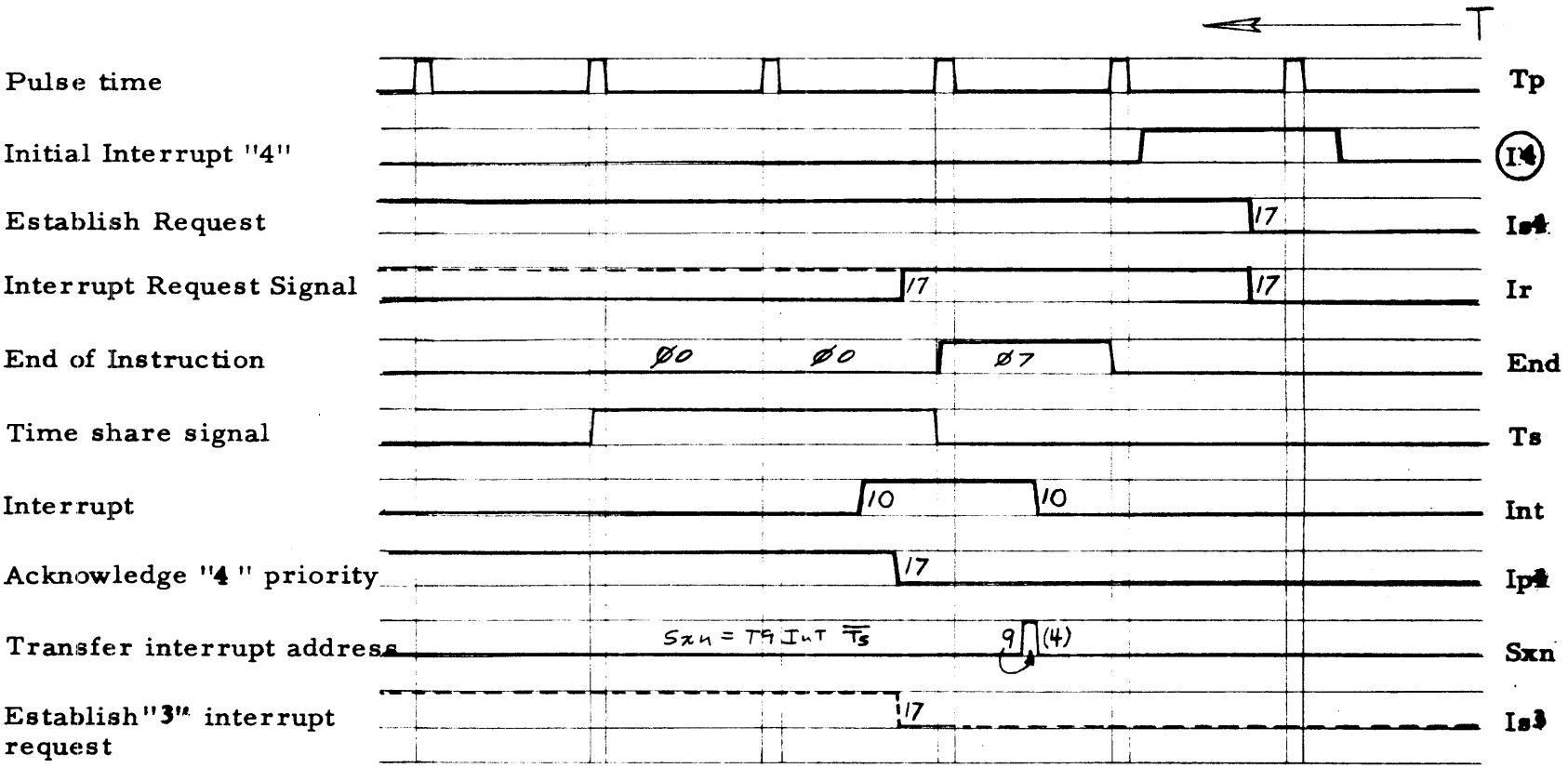
* Partial execution of interrupt program "4".

* * Execution of interrupt program "3"

FIGURE 12

INTERRUPT TIMING C

"4" interrupted by "3" after "4" is acknowledged.



Note: The lower priority address ((I4)) will be executed before the higher priority address ((I3)).

TIME

FIGURE 13

INTERRUPT TIMING D

Channels (I4) and (I3) in combination with (Ts) Time-Share.

$$Ie = Int\ \overline{F1}\ (T22 - T17)$$

and the higher priority interrupt will require an end cycle to interrupt and the first end cycle will be the end cycle of the 43 instruction in the lower priority program. Refer to the interrupt timing diagram showing Ts.

When any interrupt subroutine is completed, Ib, which is caused by an indirect branch, resets the interrupt channel flip-flops, and the P register is then set to the value it had when the interrupt occurred. If an intervening priority channel is waiting, it will not be recognized until the end cycle of the indirect transfer instruction, by which time P has the proper contents to be stored by the intervening priority channel. Any of the external interrupt channels can be modified for execution of a single instruction (the P count is maintained). This is accomplished by generating an external term Ij, which represents any external signal designated as a single-instruction interrupt,

$$Ij = Ipa + Ipb + Ipc + - - -$$

where a, b, c, represent the numbers of the desired single-instruction, interrupt channels.

Ij is used to block the incrementing of the P register by disabling the setting of Ia during phases Ø4 or Ø6.

$$sIa = T24\ F1\ \overline{F3}\ \overline{Ij} + - - -$$

Ij is also used to immediately reset the interrupt system.

$$Ib = Ij\ \overline{Ts}\ F1\ (T22 - T17) + - - -$$

Ij is implemented as follows:

$$\overline{Ij} = \overline{Ipa}\ \overline{Ipb}\ \overline{Ipc} - - -$$

$$Ij = \overline{\overline{Ij}}$$

TIME-SHARE (MEMORY INTERLACE)

With the optional W or Y Buffer time-share interlace equipment the memory can be automatically shared between the W and/or Y Buffer and the computer. Whenever the buffer has received a full word from a peripheral unit, it can momentarily stop the computer and insert the word into the memory without disrupting the program, and without any program control. Similarly, if either W or Y is being used for output, two memory cycles are taken from the computer to reload the buffer whenever the buffer is empty.

The W or Y buffer logic issues a Tsw or a Tsy (time-share request), respectively, to request use of the memory. At the next Tp time the Ts (time-share) flip-flop is set and the computer idles regardless of what it was doing.

$$Tsw = \overline{Wf}\ W0\ \overline{Wh}\ (Iw)$$

$$Tsy = \overline{Yf}\ Y0\ \overline{Yh}\ (Iy)$$

$$sTs = (Tsw + (Tsy))\ Tp$$

Almost all control gates in the computer are disabled when Ts is set. The computer idles wherever it is for the two cycle times that Ts is on, and it resumes operation from that state when Ts is reset. A, B and X registers (dynamic) must re-circulate.

Tsw, if true, does not go false until Ts Tp; therefore, Ts is always on for two cycles.

$$rTs = \overline{Tsw}\ \overline{Tsy}\ Tp\ \overline{Tsm}\ \ (\overline{Kf})$$

Wp is set if the W Buffer is to use the memory; it is reset if the Y Buffer is to use the memory.

$$sWp = (\overline{Go} + Tsw)\ \overline{Tsy}\ \overline{Tsm}\ Tp$$

$$rWp = (Go\ \overline{Tsw} + (Tsy))\ \overline{Tsm}\ Tp + - - -$$

If both Tsw and Tsy are true at Tp, then Wp will be reset at Tp and the Y Buffer will get the first two Ts cycles. In this case Ts stays set for four cycles, two cycles for each of the two buffers. Wp will be complemented at the end of the second cycle. When Ts is set, almost all terms in the computer are disabled. Most of them are disabled since $\overline{Ts}$ is ANDed into the expressions for the phases. For example, phase $\emptyset 3 = \overline{F1}\ F2\ F3\ \overline{Ts}$. There are exceptions. If P0 is on at Tp when Ts is being set, it causes the Mxc to write in memory at the next T24 regardless of Ts. But except for recirculating the shift registers, A, B and X, the computer basically does not change state while Ts is set. The phase counter is set in the phase that will be executed when Ts is reset. During Ts, the C register and the W register exchange serially (if $\overline{Wp}$, C and Y exchange).

$$Cs = Ts\ (\overline{T24}\ \overline{Tp})$$

$$sC0 = Ts\ Wn\ Wp\ \overline{Tp}$$

$$+ Ts\ (Yn)\ \overline{Wp}\ \overline{Tp} + - - -$$

At Ts T10 the Tsm flip-flop is set for one cycle only.

$$sTsm = Ts\ T10\ \overline{Tsm}$$

$$rTsm = T10\ Tsm + \overline{Ts}$$

Tsm is used to modify the memory control. The memory is pulsed at the first Ts T8, but not at the second Ts T8. Tsm controls this,

$$Mc = Q1\ \overline{Q2}\ Q3\ \overline{Q4}\ Q5\ \ (F1 + \overline{F3} + Ts\ )\ (\ Tsm + \overline{Ts}) + - - -$$

and the lines which control the addressing of the memory are not gated from the S register but instead are gated from address lines Iw10 through Iw23 (or Iy10 through Iy23), which are generated in the interlace units,

$$\overline{L1} = \overline{S1\ \overline{Tsm} + (Iw10)\ (Tsm\ Wp) + (Iy10)\ (Tsm\ \overline{Wp})}$$

etc. This ensures that the previous word being regenerated during the first part

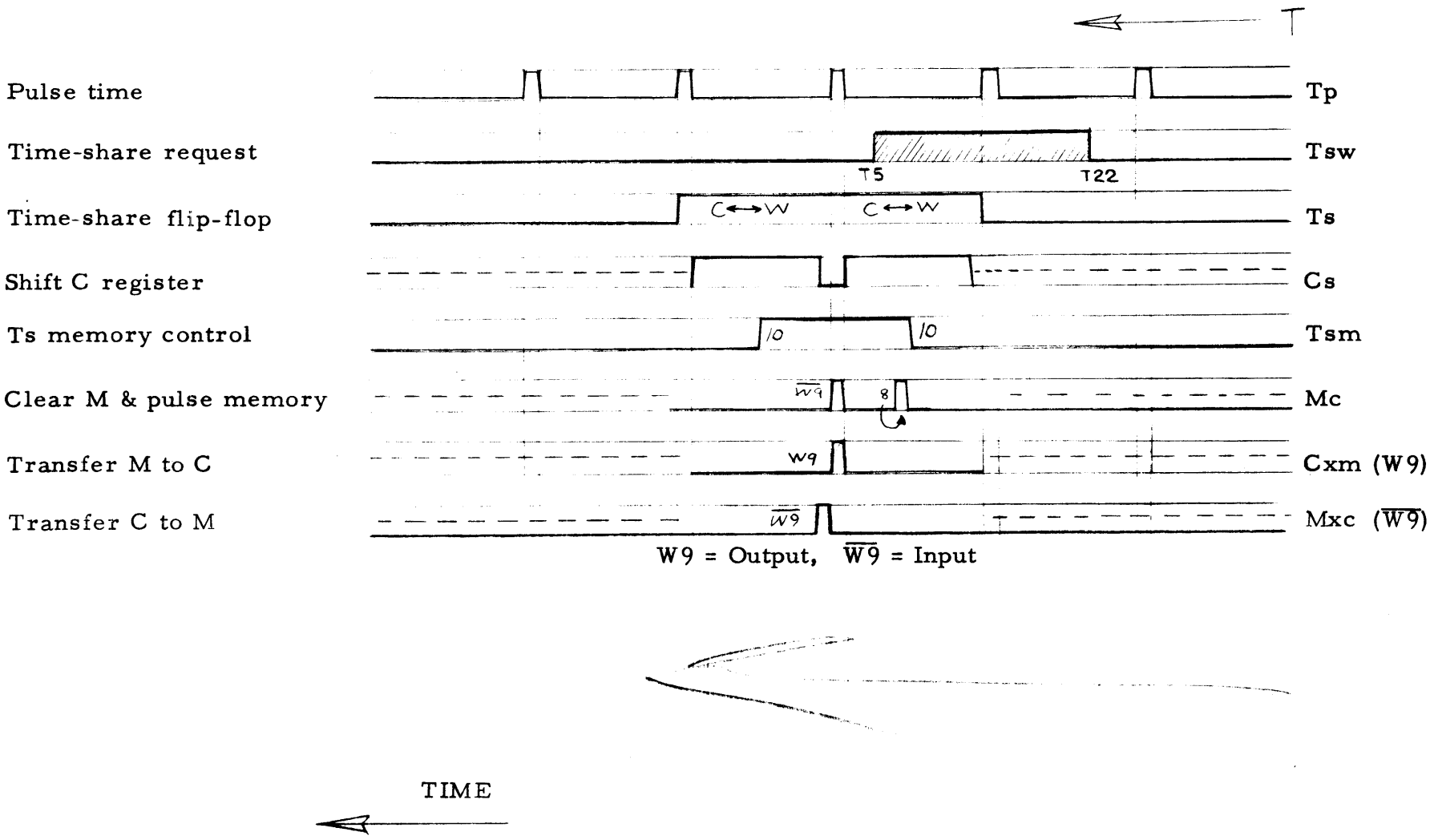


FIGURE 14

INTERLACE TIMING

(TIME-SHARE)

of Ts is controlled by the S register. At the first Tp Ts the word addressed by the buffer has been read from memory. If the buffer is being used for output, W9, then M is transferred to C.

$$Cxm = Tp\ Tsm\ Wp\ W9 + Tp\ Tsm\ \overline{Wp}\ \textcircled{Y9}$$

Here Tsm is used to specify the first Tp Ts and not the second. If the buffer using the interlace is being used for input, $\overline{W9}$, the memory register is cleared at this Tp and P0 is set to remember to load M from C at T24.

$$Mc = Tp\ Ts\ \overline{W9}\ (Tsm\ Wp) + Tp\ Ts\ \textcircled{\overline{Y9}}\ (Tsm\ \overline{Wp})$$

$$sP0 = Mc\ Tp$$

$$rP0 = (P0\ T24)$$

$$Mxc = (P0\ T24)$$

During the second cycle of Ts the buffer register and the C register again exchange serially. At the second Tp Ts the C register again contains the number it originally contained when Ts was just set and the buffer register, if used for output, contains the requested word from memory.

Ts is reset at this second Ts Tp and the computer continues. Refer to the "Time-Share" timing diagram.

The buffer waits two cycle times after the Tsw signal is given before it precesses the next character in or out, unless both buffers are interlacing. In this case, the waiting time is four cycles.

Interlacing can take place regardless of the state of the Go and Ht (control) flip-flops, regardless of the position of the Control switch, and regardless of the position of the Register Display switch.

CONTROL SWITCH

The Control switch is a three-position switch on the control panel of the computer labeled "Run-Idle-Step."

In the "Idle" position, the computer idles by inhibiting the advance of the P register, and by repetitively executing a "NOP" instruction.

In the "Run" position, the computer executes instructions as required until: a Halt instruction is executed, the switch is placed in the Idle position, or a parity error occurs with the Parity Override switch in the Halt position.

In the "Step" position, the computer executes the instruction in the C register, reads the next instruction and halts. The switch locks in the Run or Idle positions, but must be held in the Step position.

The Control switch controls two flip-flops designated Go and Ht, which, in turn, control the state of the computer.

$\overline{Go}\ \overline{Ht}$ = Halt, Ready to run

Go $\overline{Ht}$ = Run

Go Ht = Run, but the computer will halt at completion of the Instruction

$\overline{Go}$ Ht = Halt, the computer will not run until the switch is returned to Idle and back to Step or Run

Refer to the control switch diagram for the following discussion.

The two signals supplied by the Control switch are (Ks), which is true when the switch is in the Step position, and (Kg), which is true when the switch is in the Run position. Both of these signals are false when the switch is in the Idle position.

If the switch is put in Run, the Go flip-flop is set at the next Tp.

$$sGo = Tp\ (Kg)\ \overline{Ht}\ - - - + - - -$$

The Ht flip-flop is not set to stop the computer unless: a Halt instruction is executed, the switch is placed in Idle, or a parity error occurs with the Parity Override switch in the Halt position or while the Control switch is in the Step position.

$$sHt = (\overline{Kg})\ Go\ \ T0\ End + T0\ \ \emptyset 5\ \overline{01}\ \overline{02}\ \overline{05}\ \overline{Int}$$
$$+ Cp\ Tp\ (Kp)\ \ \overline{Ts}\ \ + - - -$$

During a Halt instruction Ht is set at T0 and Go is then reset at Tp.

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

The Ht flip-flop remains set until the switch is returned to Idle.

If the switch is set to Idle when the computer is running, Ht is set at T0 just before the end of the instruction is being performed,

$$sHt = (\overline{Kg})\ Go\ \ T0\ \ End + - - -$$

and Go is reset at the next clock time, Tp, stopping the computer.

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

Then Ht is reset, since Go is in the reset state.

$$rHt = (\overline{Ks})(\overline{Kg})\ \overline{Go}\ \overline{Cp}\ T24 + - - -$$

If the switch is depressed to the Step position, Go is set at the next Tp time and the instruction set up in the C register is performed.

$$sGo = Tp\ (Ks)\ \overline{Ht}\ \ - - - + - - -$$

At T0 of the last cycle of the instruction, Ht is set and at Tp, Go is reset, and the computer idles. As long as the switch is held in the Step position, Ht stays set preventing Go from being set again. When the switch is released, Ht is reset;

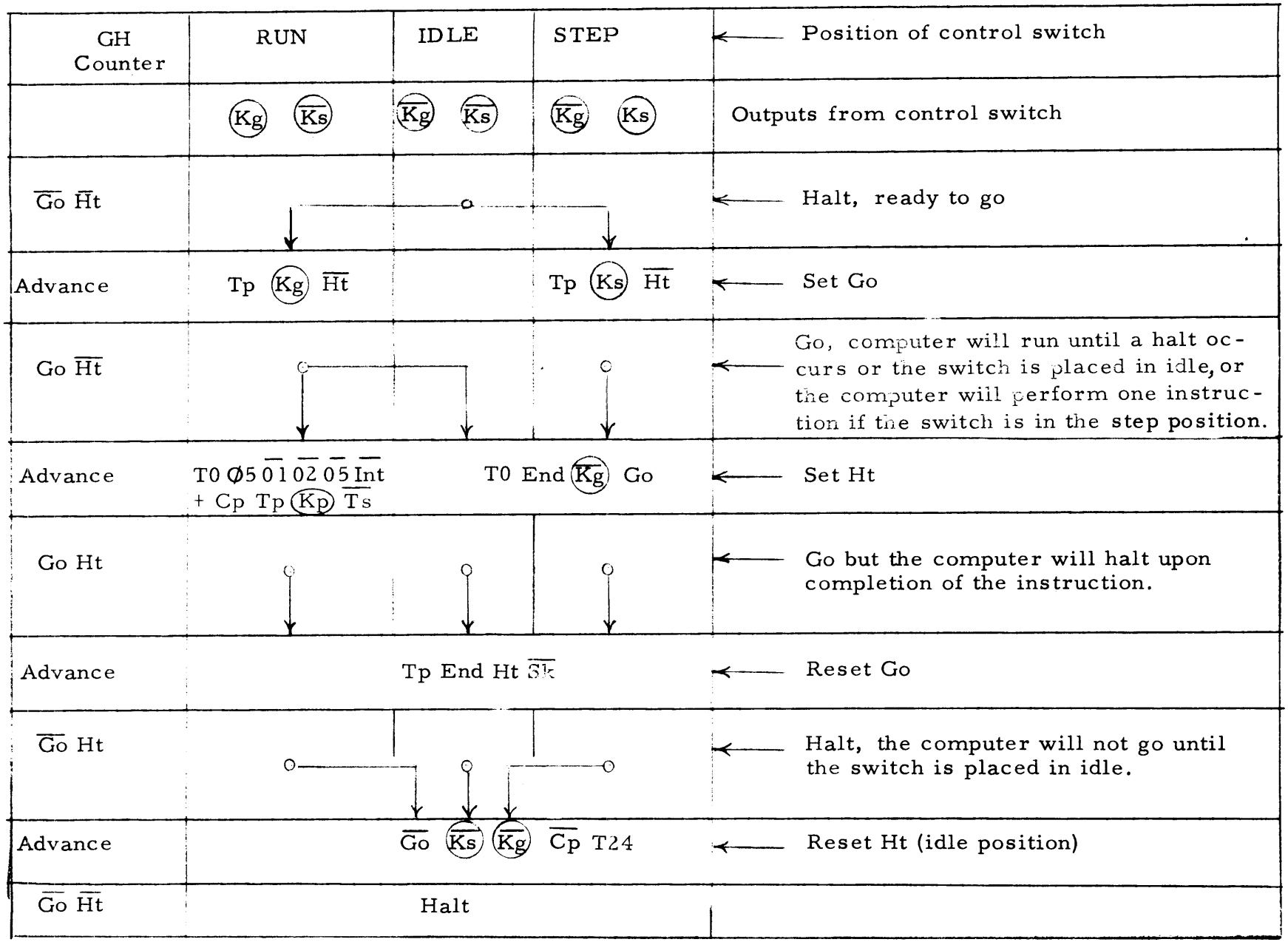


FIGURE 15

CONTROL SWITCH

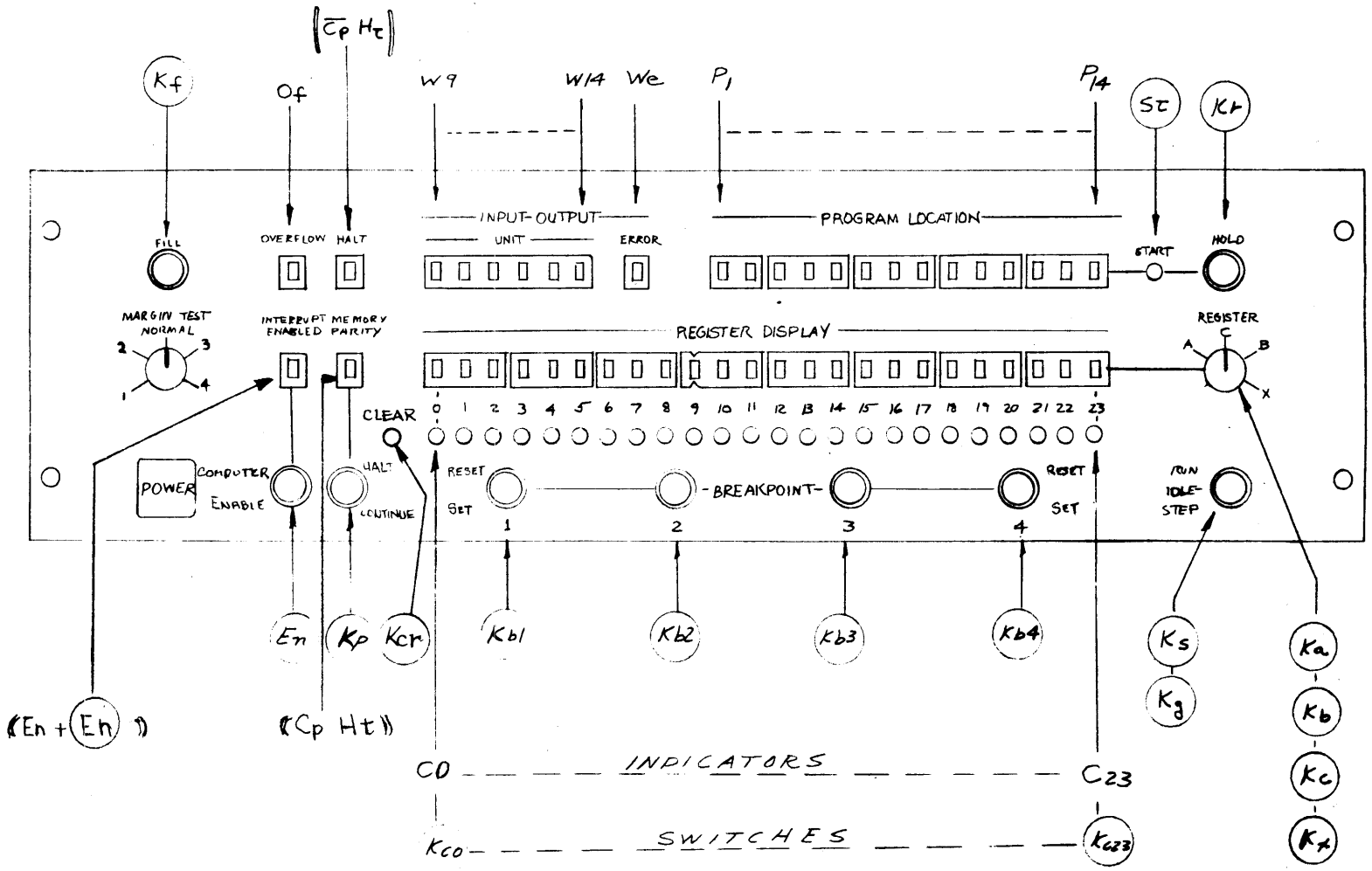


FIGURE 16

CONTROL PANEL

but now Go has lost its (Ks) input requirement.

The Go flip-flop controls whether the computer executes instructions or not. It does so by inhibiting Oxc, which sets up the new operation code. Therefore, a NOP (20) instruction is set up in the O register and is performed repetitively.

$$Oxc = \emptyset 0 \ \ Go \ \ T24 \ \ \overline{Ia} \ \ \overline{C2} \ + - - -$$

$$Oc \ = \ Tp \ \ End + - - -$$

The P register does not shift while Go is reset so that it cannot be incremented and can be displayed on the control panel.

$$Pg = - - - \ Q2 \ \overline{Ts} \ Go + - - -$$

The transfer from the M register to the C register is inhibited so that the word set up in C is not destroyed.

$$Cxm = Tp \ End \ Go + - - -$$

C does not shift in the execution of a NOP instruction regardless of Go because repetitive NOP executions hold the phase counter in phase $\emptyset 5$ where C does not shift.

REGISTER SWITCH

When Go and Ht are both reset (IDLE), it is possible to display the A, B or X registers in place of the C register. This is accomplished by interchanging the contents of the register to be displayed and the C register. The contents must, of course, be switched back before computation is resumed.

Two flip-flops, Dc and Su, in the computer, "remember" which of these registers is being displayed in the C register.

$$\overline{Dc} \ \overline{Su} \ = \text{C Register}$$

$$\overline{Dc} \ \ Su \ = \text{B Register}$$

$$Dc \ \ \overline{Su} \ = \text{A Register}$$

$$Dc \ \ Su \ = \text{X Register}$$

When the computer first reaches the $\overline{Go} \ \overline{Ht}$ configuration, Dc and Su are both reset indicating that all registers are home (the contents are in their original registers).

$$rDc = Tp \ End \ Go + - - -$$

$$rSu = Tp \ End \ Go + - - -$$

If the switch is turned to the "A" position, then at the next T24 time a flip-flop designated Ex (exchange) is set.

$$sEx \ = - - - \ \overline{Ts} \ T24 \ \overline{Go} \ \overline{Ht} \ \overline{Dc} \ \ (Ka') + - - -$$

Ex controls the timing of the actual interchange and is set for one cycle time.

$$rEx = T0\ F1$$

While Ex is set and the switch is in the "A" position, the C and A registers exchange contents.

$$sAw = Anr\ Ex\ C23 + - - -$$
$$Anr = Ex\ (Ka) + - - -$$

The C register is enabled by Ex.

$$Cs = Ex + - - -$$
$$sC0 = Ex\ (Ka)\ An + - - -$$

At pulse time T0 the flip-flops are set to Dc $\overline{Su}$ indicating that the contents of the A register are in the C register.

$$sDc = Ex\ T0\ \overline{Dc}\ (Ka) + - - -$$

Now if the switch is rotated, it must first pass through the (Kc) position. Ex is set the next T24 time after the switch reaches the C position,

$$sEx = T24\ \overline{Ts}\ \overline{Go}\ \overline{Ht}\ Dc\ (Kc')\ (\overline{Kf}) + - - -$$
$$rEx = T0\ F1$$

and again A and C exchange

$$sAw = Anr\ Ex\ C23 + - - -$$
$$Anr = Ex\ Dc\ \overline{Su} + - - -$$
$$sC0 = Ex\ Dc\ \overline{Su}\ An$$

At Ex T0 the flip-flops change back to the $\overline{Dc}\ \overline{Su}$ configuration.

$$rDc = Ex\ T0\ Dc + - - -$$

There are two sets of contacts on the Register switch. One set is "break before make" and the other set is "make before break." The "break before make" contacts are indicated by the ' (prime) and are used to set Ex. The "make before break" contacts are used to control the exchanges. This ensures that the exchanges will not be interrupted by bouncing contacts, since the "exchange" contacts are well made before Ex is set. If the Register switch is turned to the (Kb) position, Ex is set at T24,

$$sEx = T24\ \overline{Ts}\ \overline{Go}\ \overline{Ht}\ \overline{Su}\ (Kb') + - - -$$
$$rEx = T0\ F1$$

the B and C registers interchange,

$$sBw = Bnr\ Ex\ C23 + - - -$$
$$Bnr = Ex\ (Kb)$$
$$sC0 = Ex\ (Kb)\ Bn + - - -$$

and at T0 the flip-flops are set in the $\overline{Dc}$ Su configuration:

$$sSu = Ex\ T0\ \overline{Su}\ (Kb) + - - -$$

If the switch is advanced further to the (Kx) position, it first advances through another (Kc') position, which causes the registers to first return home and the Dc and Su flip-flops to be reset to the $\overline{Dc}\ \overline{Su}$ configuration. If the Register switch is not at the C position when the Control switch is placed into Step or Run, Ex comes on to send the registers home.

$$sEx = T24\ \overline{Ts}\ \overline{Go}\ \overline{Ht}\ (Dc + Su)\ ((Ks) + (Kg)) + - - -$$

$$rEx = T0$$

For example, if in the (Kx) position, the C and X registers interchange,

$$sXw = Xnr\ Ex\ C23 + - - -$$
$$Xnr = Ex\ Dc\ Su + - - -$$
$$sC0 = Ex\ Dc\ Su\ Xn + - - -$$

and at T0 the flip-flops will be set in the $\overline{Dc}\ \overline{Su}$ configuration,

$$rDc = Ex\ T0\ Dc + - - -$$
$$rSu = Ex\ T0\ Su + - - -$$

and the Go flip-flop is not set to start computation until the registers are home.

$$sGo = \overline{Dc}\ \overline{Su}\ Tp\ ((Kg)\ \overline{Ht} + (Ks)\ \overline{Ht})$$

The Register switch can, of course, be turned safely while the computer is running, since Dc and Su are not concerned with the register position at this time and Ex will be reset.

During the Step operation the C register is displayed when the Control switch is held down because Ht remains set. When the control switch is released, Ht is reset and the Register switch may cause an exchange. If the computer halts due to a Halt instruction or a parity error, an exchange cannot occur until the Control switch is returned to Idle, or in the case of a parity error, until the Parity Override switch is put in the Continue position.

MANUAL CONTROL SWITCHES

There are several manual switches on the front panel for controlling the computer. The Control and Register switches have just been discussed in the preceding pages, and the Fill and Start switches are discussed on the following pages. The others are described below.

Hold:

The Hold switch, (Kr), inhibits the incrementing of the P register. It does so by disabling Pg which enables the P register.

$$Pg = (Kr)\ Q2\ \overline{Ts}\ Go - - -$$

Parity Override:

The Parity Override switch, (Kp), when it is in the "Continue" position, disables the gate which causes the computer to halt if there is a memory parity error.

$$sHt = Cp\ Tp\ Go\ (Kp)\ \overline{Ts} + - - -$$

If the switch is in the Halt position, it does not allow Cp to be reset after a parity error.

$$rCp = (\overline{Kp})\ Ht + - - -$$

Clear:

The Clear switch, (Kcr), clears the C register by causing it to shift to the right; zeros are shifted in at the left.

$$Cs = (Kcr) + - - -$$

C Load Switches:

The C register Load switches ((Kc0) ⟶ (Kc23)) set the individual C register flip-flops.

$$sC0 = (Kco) + - - -$$

$$sC23 = (Kc23) + - - -$$

Due to the auxiliary DC inputs on the repeater flip-flops, the C register need not be enabled for this load operation.

Breakpoint Switches:

The Breakpoint switches are four program switches on the control panel, whose state can be tested for by the computer with the SKS instruction.

$$\begin{aligned} Sks = {} & C10\ \overline{C11}\ C15\ (\overline{Kb1}) \\ & + C10\ \overline{C11}\ C16\ (\overline{Kb2}) \\ & + C10\ \overline{C11}\ C17\ (\overline{Kb3}) \\ & + C10\ \overline{C11}\ C18\ (\overline{Kb4}) \end{aligned}$$

Enable Switch:

The Enable switch, (En), forces the enabling of the interrupt system, regardless of the state of the Enable flip-flop, En.

$$sInt = T10\ Ir\ End\ (En + (En)) + - - -$$

START AND FILL SWITCHES

When the computer is first turned on, certain flip-flops must be cleared or set in specified configurations to enable the start of operation.

The Start switch, (St), on the front panel performs this function.

This switch clears the Overflow, the Memory Parity, Skip, and the Enable flip-flop.

$$rOf = (St) + - - -$$
$$rCp = (St) + - - -$$
$$rSk = (St) + - - -$$
$$rEn = (St) + - - -$$

The C register is cleared making ready for a Halt instruction (00).

$$Cs = (St)(\overline{Tp}\ \overline{T24}) + - - -$$

The Start switch also clears the W Buffer and the P register, Go, Dc and Su are all reset and Ht is set.

$$Wc = (St)(T0 - T5) + - - -$$
$$Pg = (St) + - - -$$
$$rGo = (St) + - - -$$
$$sHt = (St) + - - -$$
$$rDc = (St) + - - -$$
$$rSu = (St) + - - -$$

If there is a program still in the computer, then a BRU to the start of the program should be set up manually in the C register, and the Control switch placed in the Run position. Then the BRU in the C register is executed and the computer is in operation.

If there is no program in memory, then the Control switch should be placed in the Run position and the Halt instruction in the C register will be executed. The computer halts in the $\overline{Go}$ Ht configuration. The Ex flip-flop does not cause the exchange of any registers regardless of the Register switch because Ht is on. Then the Fill switch, (Kf), is pushed to force the reading of a bootstrap program. The Fill switch, (Kf), sets up the W Buffer to read at four characters per word from photo-reader No. 1. It also forces a WIM00002 instruction into the C register, and it turns off the Ht flip-flop.

$$sC4 = (Kf) + - - -$$
$$sC5 = (Kf) + - - -$$
$$sC7 = (Kf) + - - -$$
$$sC22 = (Ki) + - - -$$
$$rHt = (Kf) + - - -$$

The Fill switch also forces a (-7) into the X register.

$$sXw = (Kf)\ \overline{T21}\ \overline{T22} + - - -$$

$$Xnr = (Kf)\ (T21 + T22) + - - -$$

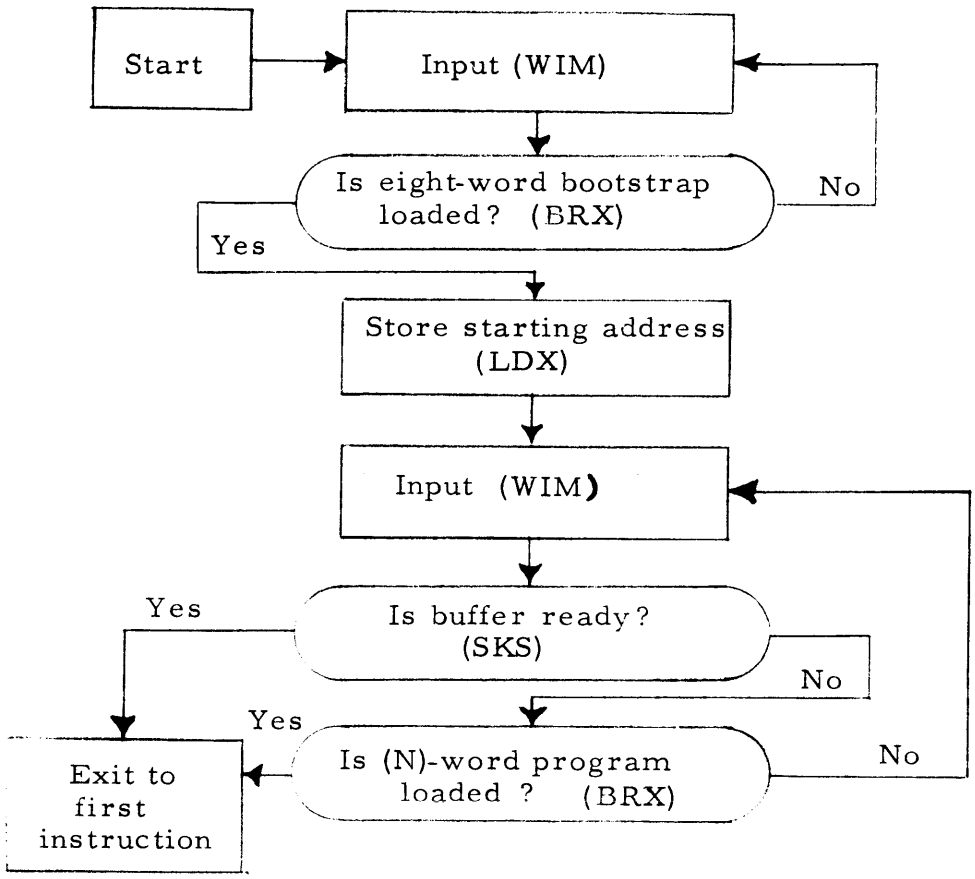
This places zeros in positions T21 and T22 and ones in all other positions.

When the Fill switch is released, the Go flip-flop usually turns on at the next Tp time. The computer executes the WIM instruction (input from the W Buffer) when the first word from the tape is ready in the buffer. This word is inserted in memory location 0002. The program counter was cleared with the Start button, it counted once in executing the Halt instruction, and the next instruction is picked up from address 0002. This word is the one just read in from the tape.

The first eight words on the tape should be the following:

Location in Memory	Instruction
000 2	2 WIM00012
000 3	BRX00002
000 4	LDX00011
000 5	2 WIM00000
000 6	SKS21000 (Buffer Ready)
000 7	BRX00005
001 0	(First Instruction)
001 1	(Starting Address with Ia Tag)

A flow chart of the program is shown below:



While the Fill switch is held down, Ht is turned off, but neither Go nor Ex will go until the Fill switch is released.

$$sGo = \overline{Dc}\ \overline{Su}\ \overline{Ts}\ Tp\ \overline{Ht}\ (\text{- - -})\ \overline{Kf}$$

$$sEx = \overline{Ts}\ \overline{Kf}\ T24\ \overline{Go}\ \overline{Ht}\ (\text{- - -}$$

If, when the switch is released, Go comes on at Tp time, then Ex will be innibited by Go.. If Ex is set first (at T24), then an exchange will occur if the Register switch is not at C.. However, Go will not be set until a second exchange returns the register contents home, since the setting of Go requires ($\overline{Dc}\ \overline{Su}$), and Ex will be set the second time because the Control switch is in the Run position (Kg).

A detailed description of each instruction is included in this material. An intimate knowledge of all preceding descriptions and the Reference Manual is assumed. For additional information on phase changes refer to the phase counter.

SKIP AND BRANCH INSTRUCTIONS

Instruction 01 (BRU)

Phasing: (Ø0) ⟶ (Ø0)

During phase Ø0 the address field in the C register is shifted through the adder for possible indexing and into the P register, and the next instruction is accessed from this address. The P register is enabled by Pg,

$$Pg = (Kr)\ Q2\ \overline{Ts}\ Go\ Ø0\ \overline{Ia}\ \overline{02}\ \overline{03} + - - -$$

The new address is received by P1,

$$sP1 = Ø0\ \overline{Ia}\ \overline{02}\ \overline{03}\ Add + - - -$$

and Xz and Yz perform the indexing, if any.

$$Xz = Xn\ Ix\ \overline{F1}\ \overline{F3}\ \overline{Ts} + - - -$$

$$Yz = C23\ \overline{F1}\ \overline{F3} + - - -$$

Instruction 40 (SKS)

Phasing: (Ø0) —T24→ (Ø5) —skip→ (Ø7) ⟶ (Ø0)

Instruction 40 advances directly to phase Ø5 at Ø0 T24. During phase Ø5 the Sk (Skip) flip-flop will be set if Sks is true.

$$sSk = Ø5\ 01\ \overline{04}\ T0\ Sks + - - -$$

Sks will represent one of thirteen internal signals, or one of N external signals which have been specified by the instruction address.

```
Sks = + - - -
      + C10 C11   C19  Ye
      + C10 C11   C20  We
      + C10 C11   C21  En
      + C10 C11   C22  En
      + C10 C11   C23  Of
      + C10 C11   Sio
      + C10 C11   Ssc
      + - - - -
      + - - - -
```

Sio and Ssc represent external signals used for additional skip inputs. If Sk is set at T0, then the phase counter is advanced to phase Ø7 to increment P again and perform the skip.

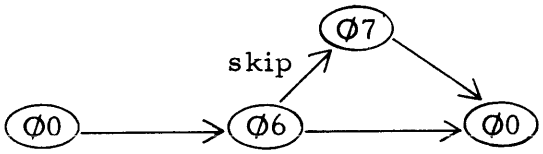
```
sF2 = Tp Ts Sk - - - - + - - - -
sIa = T24 Ø7 Sk
rSk = Ø7
```

If Sk was not set, the phase counter will reset to phase Ø0.

```
rF1 = Tp End Sk
rF3 = Tp End Sk
```

Instructions 53, 70, 72 and 73

Phasing:



During phase Ø0 of the above instructions the operand is accessed from memory in the usual manner, and then the phase counter is advanced to phase Ø6.

During phase Ø6 the Sk flip-flop will determine if a skip is to occur.

Instruction 53 (SKN) will set the Sk flip-flop at T0 if the operand was negative.

```
sSk = 01 02 03 04 05 06 Ø6  T0  C23  + - - -
```

Note that the sign of the operand has been shifted to C23, by T0 time.

Instructions 70 (SKM) and 72(SKA) preset the Sk flip-flop at T24 time.

```
sSk = 01 03 04 06 Ø6  T24 + - - -
```

Instruction 70 will reset Sk if the contents of A and C (operand) differ when B is a "one".

$$rSk = 01\ 03\ \overline{04}\ \overline{05}\ \emptyset 6\ \overline{Tp}\ \overline{T24}\ Bn\ (\overline{An}\ C23 + An\ \overline{C23})\ + - - -$$

If Sk is left in a set condition, then the skip will occur.

Instruction 72 will reset Sk if the operand and (A) compare "ones".

$$rSk = 01\ 03\ \overline{04}\ 05\ \overline{06}\ \emptyset 6\ \overline{Tp}\ \overline{T24}\ C23\ An + - - -$$

Instruction 73, (SKG) will set Sk if A has a "1" bit and C has a "0" in the corresponding bit position; Sk will reset for the opposite condition. If 2 bits in corresponding bit positions in the registers compare, Sk remains in its current state. Since the sign bits have negative weights, the reverse logic applies at T0; and since the least significant bits arrive first, the final state of Sk is determined by which register contains the larger number.

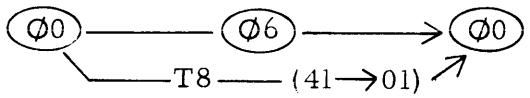
$$sSk = 01\ 02\ 03\ \overline{04}\ 05\ 06\ \emptyset 6\ \overline{Tp}\ \overline{T24}\ \overline{T0}\ An\ \overline{C23}$$
$$+ 01\ 03\ \overline{04}\ 05\ 06\ \emptyset 6\ T0\ C23\ \overline{An} + - - -$$

$$rSk = 01\ 03\ \overline{04}\ 05\ 06\ \emptyset 6\ \overline{Tp}\ \overline{T24}\ \overline{T0}\ \overline{An}\ C23$$
$$+ 01\ 03\ \overline{04}\ 05\ 06\ \emptyset 6\ T0\ An\ \overline{C23} + - - -$$

and a skip will or will not occur accordingly.

Instruction 41 (BRX)

Phasing:



In phase Ø0 the address portion of the instruction is shifted into the P register through the adder as indexing may simultaneously take place.

$$sP1 = \overline{02}\ \overline{03}\ \emptyset 0\ \overline{Ia}\ Add\ + - - -$$

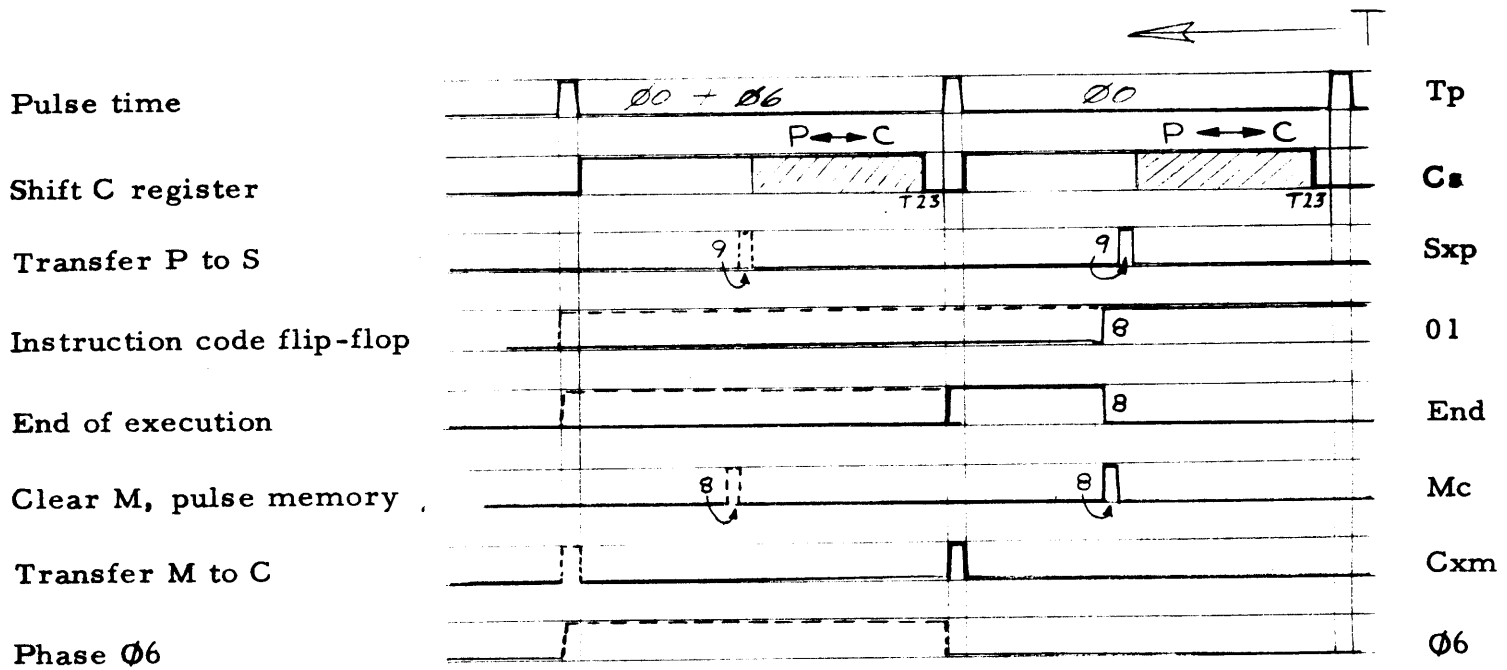
The previous number in P must be saved in case a transfer does not take place; therefore, it is shifted into the C register.

$$sC0 = \overline{02}\ \overline{03}\ \emptyset 0\ \overline{Ia}\ P14\ Q2$$

The X register is incremented during phase Ø0 with $\overline{Sk}$ being used as a carry in a half adder. Sk will come in to Ø0 in a reset state.

$$sSk = 01\ \overline{05}\ \ \overline{02}\ \overline{03}\ \emptyset 0\ \overline{Ia}\ \ \overline{Xn}\ \ \overline{Tp}\ \overline{T24}\ \ \ \overline{T0} + - - -$$
$$rSk = \emptyset 0\ T0 + - - -$$

$$sXw = 01\ \overline{05}\ \ \overline{02}\ \overline{03}\ \emptyset 0\ \overline{Ia}\ \ (\overline{Xn}\ \overline{Sk} + Xn\ Sk)\ + - - -$$
$$Xnr = \emptyset 0\ \overline{Ia}\ \overline{02}\ \overline{03}\ 01\ \overline{05} + - - -$$



Note: Dashed lines indicate newly formed X9 bit is not a "one".

TIME

FIGURE 17
TIMING: INSTRUCTION 41 (BRX)

Example:

Xn 1 0 1 0 0 1 1

Sk 1 1 1 1 0 0 0

sXw 1 0 1 0 1 0 0

At T9 the P register is transferred to S to access the instruction for branching. If the newly formed X9 bit is a "one", the branch will occur. The Xw flip-flop is gated at T8 to reset the 01 instruction code flip-flop. This changes the operation code from 41 to 01,

$$r01 = 01\ \overline{02}\ \overline{03}\ \overline{05}\ \emptyset 0\ \overline{Ia}\ Xw\ Mc\ Q1 + Oc$$

where Mc is the pulse to the memory and occurs at T8.

If 01 is reset, the phase Ø0 cycle becomes an End cycle since,

$$End = \emptyset 5 + \emptyset 6 + \emptyset 7 + \emptyset 0\ \overline{Ia}\ \overline{01}\ \overline{02}\ \overline{03}$$

However, it becomes an End cycle too late for a possible interrupt operation. The X register also starts to recirculate (stops incrementing) and the carry into X normally does not go beyond X8. There is no carry beyond X8 unless X becomes positive, at which time the carry can extend to X0.

If 01 is not reset T8 Ø0, then the phase changes to Ø6 at Tp and the instruction (41) continues operation.

$$sF1 = Tp\ \overline{Ia}\ \emptyset 0\ 01\ (\overline{04} + \overline{05}) + - - -$$

$$sF2 = Tp\ \overline{Ia}\ \emptyset 0\ 01\ \overline{02} + - - -$$

The C register is not altered at Tp; the word from memory is simply regenerated without being used.

During phase Ø6 the original contents of P which were shifted to C must be shifted back to P and be incremented.

$$sIa = T24\ F1\ \overline{F3}\ \overline{Ij} + - - -$$

$$sP1 = 01\ \overline{02}\ \overline{04}\ \overline{05}\ 06\ \emptyset 6\ (C23\ \overline{Ia} + \overline{C23}\ Ia)$$

$$rIa = \overline{C23}\ \emptyset 6\ 01\ \overline{02}\ \overline{04}\ \overline{05}\ 06\ \overline{T24} + - - -$$

At T9 this shift is completed and P is transferred in parallel to S to access the next instruction in sequence.

Instruction 43 (BRM)

Phasing: (Ø0) ⟶ (Ø6) ⟶ (Ø0)

During phase Ø0 the contents of the P register are shifted to the C register for eventual storage,

$$sC0 = \emptyset 0\ \overline{Ia}\ \overline{02}\ \overline{03}\ P14\ Q2 + - - -$$

and the contents of the overflow flip-flop are shifted into C.

$$sC0 = \emptyset 0\ \overline{Ia}\ \overline{02}\ \overline{03}\ Of\ T0 + - - -$$

$$Cs = \overline{Tp}\ \overline{T24}\ \overline{F1}\ \overline{F2} + - - -$$

The address portion of the instruction is shifted into the P register.

$$Pg = (Kr)\ Q2\ \overline{Ts}\ Go\ \emptyset 0\ \overline{Ia}\ \overline{02}\ \overline{03} + - - -$$
$$sP1 = \emptyset 0\ \overline{Ia}\ \overline{02}\ \overline{03}\ Add + - - -$$

There may be indexing taking place. At T9 the new address in the P register is transferred to the S register to set up the address for storage.

$$Sxp = T9\ \overline{02}\ \overline{03}\ \overline{Ia}\ \emptyset 0 + - - -$$

At Tp of $\emptyset 0$ the M register is cleared and at T24 of $\emptyset 6$ the contents of the C register will be transferred to M for storage in core memory.

$$Mc = Tp\ \emptyset 0\ \overline{Ia}\ \overline{02}\ \overline{03}\ 05 + - - -$$
$$sP0 = Mc\ Tp + - - -$$
$$Mxc = P0\ T24$$

During phase $\emptyset 6$ storage will take place, P will be incremented, and the next instruction accessed.

Instruction 51 (BRR)

Phasing: $(\emptyset 0) \longrightarrow (\emptyset 6) \longrightarrow (\emptyset 0)$

In phase $\emptyset 0$ the contents of the memory location are accessed in the usual manner. The C register shifts through the adder around to itself for possible indexing. At T10 the S register is cleared; at T9 the upper part of the C register is transferred in parallel to S (Sxc); and at $\emptyset 0$ Tp the M register is transferred to C, (Cxm) and the phase changes to phase $\emptyset 6$.

During phase $\emptyset 6$ the operand, in C, shifts to the P register. The Ia flip-flop is preset and increments the contents of C as it shifts to P.

$$sIa = T24\ F1\ \overline{F3} + - - -$$
$$rIa = \overline{C23}\ \emptyset 6\ 01\ \overline{02}\ \overline{04}\ \overline{05}\ 06\ \overline{T24} + T0\ F1 + - - -$$
$$sP1 = \emptyset 6\ 01\ \overline{02}\ \overline{04}\ \overline{05}\ 06\ (C23\ \overline{Ia} + \overline{C23}\ Ia)$$

The contents of C0 are stored in the overflow flip-flop (true side only).

$$sOf = (\emptyset 6\ 01\ \overline{02}\ 03\ \overline{04}\ \overline{05}\ 06)\ \ C0\ T24 + - - -$$

At T10 the S register is cleared; at T9 $\emptyset 6$ the new contents of P are transferred to S to access the memory for the next instruction. At Tp $\emptyset 6$ the next instruction is brought from memory and this instruction ends. The phase counter is reset to phase $\emptyset 0$.

$$Cxm = Tp\ End\ Go + - - -$$
$$rF1 = rF2 = rF3 = Tp\ End\ \overline{Sk}$$

MEMORY DECREMENT INSTRUCTION

Instruction 60 (MDE)

Phasing: $\emptyset 0 \longrightarrow \emptyset 4 \longrightarrow \emptyset 7 \longrightarrow \emptyset 0$

During phase $\emptyset 0$ the operand to be decremented will be accessed from memory in the usual manner and indexing or indirect addressing may occur.

The phase counter is then advanced to $\emptyset 4$ and C is shifted to the right one cycle time.

$$Cs = \overline{Tp}\ \overline{T24}\ F1\ \overline{F3} + - - -$$

The decrementing is performed by serially inverting the operand up to and including the first one and then passing the rest unchanged.

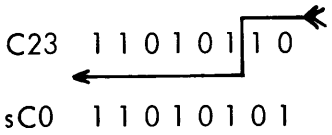
$$sCz = O1\ \emptyset 4\ T24 + - - -$$

$$Yz = F1\ \overline{C23}\ \overline{O6}\ \overline{Rf} + - - -$$

$$rCz = \overline{T24}\ \overline{Xz}\ \overline{Yz}\ \overline{Su} + - - -$$

$$sC0 = \emptyset 4\ O1\ (\overline{C23}\ Cz + C23\ \overline{Cz}) + - - -$$

Example:



An overflow can occur during this arithmetic operation if octal 4000 0000 is decremented.

$$sOf = \emptyset 4\ O1\ T0\ \overline{Yz}\ Cz + - - -$$

During $\emptyset 7$ the result is transferred from C to M by Mxc,

$$Mxc = T24\ PO$$

$$sPO = Tp\ Mc + - - -$$

$$Mc = To\ \emptyset 4 + - - -$$

and is stored in memory.

MEMORY INCREMENT INSTRUCTION

Instruction 61 (MIN)

Phasing: Ø0 → Ø4 → Ø7 → Ø0

During phase Ø0 the operand to be incremented will be accessed from memory in the usual manner and indexing or indirect addressing may occur.

The phase counter is then advanced to phase Ø4 and C is shifted to the right one cycle time.

$$Cs = \overline{Tp}\ \overline{T24}\ F1\ \overline{F3} + - - -$$

The incrementing is performed by using Cz (carry flip-flop) in a half adder and the sum is returned to Co,

$$sC0 = Ø4\ 01\ (\overline{C23}\ Cz + C23\ \overline{Cz}) + - - -$$

$$sCz = Ø4\ 01\ T24 + - - -$$

$$rCz = Tp + \overline{T24}\ \overline{Su}\ \overline{Xz}\ \overline{Yz} + - - -$$

where

$$Yz = F1\ C23\ 06\ \overline{Rf} + - - -$$

If the instruction code is (61), then the Yz input will represent C23 and Cz will be reset the first time C23 is false. Therefore, the operand is incremented by (+ 1)

Example:

C23	1 1 0 1 0 1 1 1
Cz	0 0 0 0 1 1 1 1
sC0	1 1 0 1 1 0 0 0

An overflow can occur during this arithmetic operation.

$$sOf = 01\ Ø4\ T0\ \overline{Yz}\ Cz + - - -$$

During phase Ø7 the operand is transferred from C to M by Mxc,

$$Mxc = T24\ F0$$

and is stored in memory.

STORE INSTRUCTIONS

Instructions 35 (STA), 36 (STB), 37 (STX)

Phasing: Ø0 → Ø6 (T24) → Ø4 → Ø7 → Ø0

During phase Ø0 the address is shifted through the adder for possible indexing, and

at T9 is transferred (Sxc) to the S register to address the store location.

The phase counter advances to phase Ø6 for one pulse time and is reset to phase Ø4.

During phase Ø4 the information to be stored is shifted into the C register.

$$Cs = \overline{Tp}\ \overline{T24}\ F1\ \overline{F3} + - - -$$

Instructions 35, 36, and 37 store the contents of the A, B and X registers, respectively.

$$sC0 = Ø4\ \overline{01}\ 04\ 05\ 06\ Xn$$
$$+ Ø4\ \overline{01}\ 04\ \overline{05}\ An$$
$$+ Ø4\ \overline{01}\ 04\ \overline{06}\ Bn$$

At the completion of phase Ø4 the counter advances to phase Ø7 and the information will be transferred to the M register and stored.

LOAD INSTRUCTIONS

Instructions 71, 75, 76, and 77

Phasing: (Ø0) ⟶ (Ø6) ⟶ (Ø0)

During phase Ø0 the operand is accessed in the usual fashion and transferred to the C register at Tp. During phase Ø6 the operand is shifted into the designated registers depending upon the instruction code.

$$Cs = \overline{Tp}\ \overline{T24}\ F1\ \overline{F3} + - - -$$

For Instruction 71 (LDX):

$$sXw = Xnr\ 01\ 03\ \overline{04}\ Ø6\ C23 + - - -$$
$$Xnr = 01\ 03\ \overline{04}\ Ø6\ 02\ 05\ 06 + - - -$$

For Instruction 75 (LDB):

$$sBw = Bnr\ 03\ C23 + - - -$$
$$Bnr = 01\ 02\ 03\ 04\ \overline{05}\ Ø6$$

For Instruction 76 (LDA):

$$sAw = Anr\ 01\ 02\ 03\ 04\ C23$$
$$Anr = 01\ 02\ 03\ 04\ \overline{06}\ Ø6 + - - -$$

Instruction 77 (EAX) causes the effective address of the instruction to be shifted through the adder and into the X register during phase Ø0.

$$sXw = Xnr\ 01\ 02\ 03\ 04\ Add + - - -$$
$$Xnr = 01\ 02\ 03\ 04\ 05\ 06\ \overline{Ia}\ Ø0\ Q2$$

The input to the adder is as usual during phase Ø0.

During phase Ø6 the P register is incremented, and at Tp the phase counter is reset to phase Ø0.

INPUT INSTRUCTIONS

Instruction 30 (YIM) and 32 (WIM)

Phasing: $\phi 0 \rightarrow \phi 2 \rightarrow \underset{T24}{\phi 6} \rightarrow \phi 4 \rightarrow \phi 7 \rightarrow \phi 0$

During phase $\phi 0$ the operand is accessed in the normal fashion and transferred to the C register at Tp (although it will not be used). At Tp $\phi 0$ $\overline{Ia}$ the F2 flip-flop is set for either phase $\phi 2$ or phase $\phi 6$.

$$sF2 = Tp\ \overline{Ia}\ \phi 0\ 03$$

If the Buffer (W or Y) is ready to receive data, the "Ready" flip-flop Rf is set either during phase $\phi 0$ or phase $\phi 2$.

$$sRf = \overline{Ts}\ \overline{01}\ 03\ \overline{04}\ \overline{Ia}\ \overline{F1}\ \overline{F3}\ \left[\overline{Wf}\ (W0 + \overline{W9})\ 05\ \overline{06} + \overline{Yf\ (Y0 + \overline{Y9})}\ \overline{05}\right]$$

$$rRf = Tp\ End\ \overline{Sk}\ \overline{Ts} + - - -$$

The W and Y terms are buffer terms which signal that the respective buffer is ready to transfer a new word to memory. The F1 flip-flop is set forming phase $\phi 6$ only if Rf is set.

$$sF1 = \overline{Ts}\ Tp\ \overline{01}\ 03\ \overline{Ia}\ \overline{F1}\ \overline{F3}\ Rf\ \overline{04} + - - -$$

At $\phi 6$ T24 the F2 flip-flop is reset and the phase changes to phase $\phi 4$. During phase $\phi 4$ the buffer is shifted into the C register.

$$Cs = F1\ \overline{F3}\ \overline{Tp}\ \overline{T24} + - - -$$

$$sC0 = \phi 4\ \overline{01}\ \overline{04}\ \overline{05}\ \overline{06}\ (Yn) + \phi 4\ \overline{01}\ \overline{04}\ 05\ \overline{06}\ Wn$$

The address in the S register is not altered and the word in memory is interrogated again. Mc clears the memory register (M) at Tp,

$$Mc = Tp\ \phi 4 + - - -$$

and P0 remembers to write in the memory.

$$sP0 = Mc\ Tp$$

$$rP0 = P0\ T24$$

$$Mxc = P0\ T24$$

During phase $\phi 4$ the P register is incremented. At $\phi 4$ Tp the phase counter changes to phase $\phi 7$. During phase $\phi 7$ the data is generated in the memory and the next instruction is accessed.

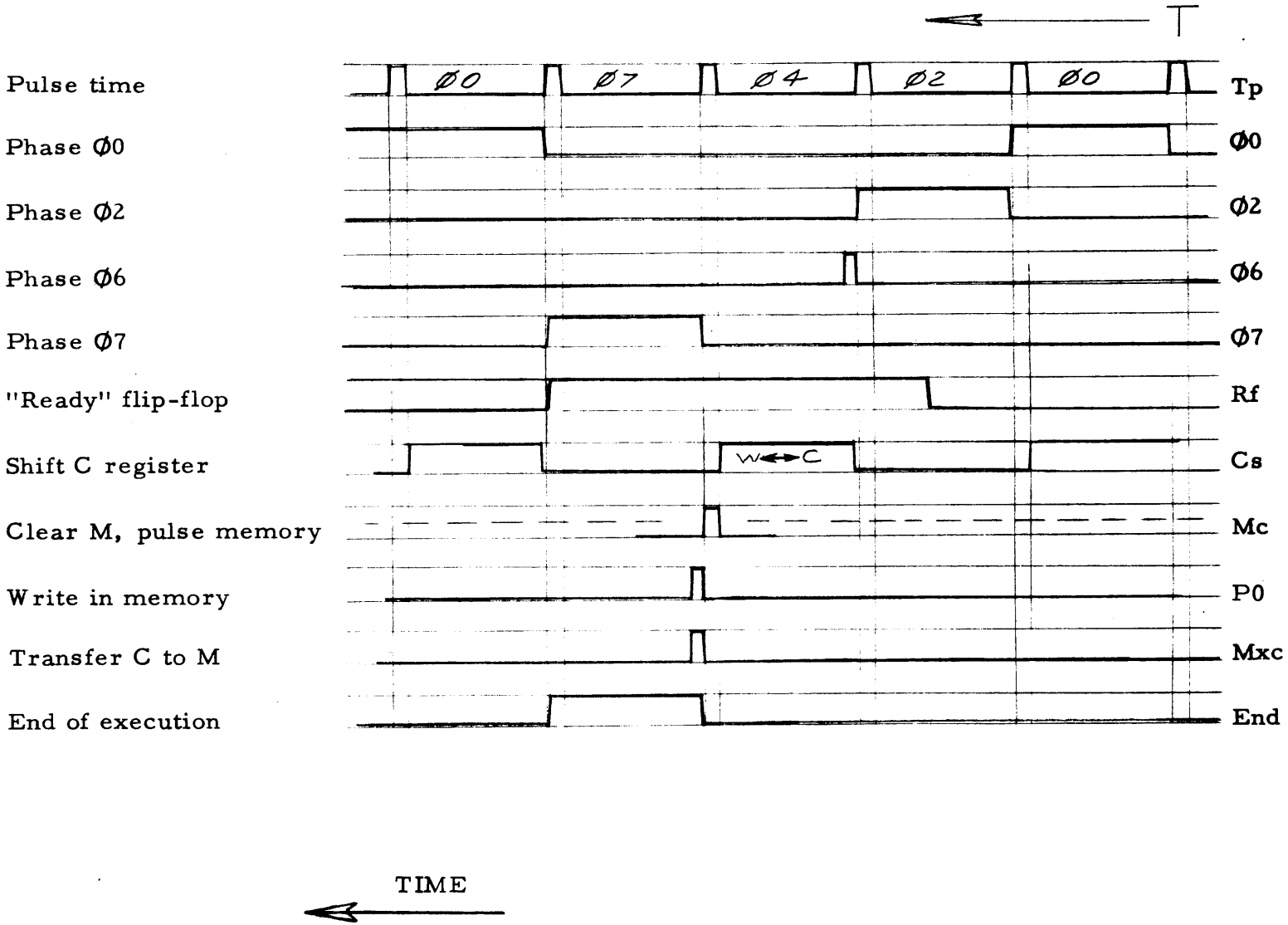


FIGURE 18
TIMING: INSTRUCTION 30, 32

Instruction 33 (PIN)

Phasing: Ø0 → Ø2 → Ø6 → Ø4 → Ø7

T24

During phase Ø0 the storage address is transferred to the S register in the usual fashion.

The phase counter then advances to phase Ø2, as this phase cannot be by-passed on instruction code 33. This allows at least one cycle time for the parallel input transfer to occur. The ready flip-flop Rf is not set until phase Ø2, where Rt is the appropriate external ready signal for input.

$$sRf = \overline{01}\ 03\ \overline{Ia}\ \overline{F1}\ \overline{F3}\ \overline{Ts}\ Ø2\ 06\ Rt\ Q2\ \overline{04} + - - -$$

During phase Ø2 a signal designated Pin is generated to indicate that the input lines are being strobed.

$$Pin = Cxi\ Q1$$

F1 will not be set to advance to phase Ø6 until the Rf flip-flop is set.

$$sF1 = \overline{Ts}\ Tp\ \overline{01}\ 03\ \overline{Ia}\ \overline{F1}\ \overline{F3}\ Rf\ \overline{04} + - - -$$

The external information is transferred in parallel to the C register via lines Cd0, Cd1, - - - - Cd23. These are gated by:

$$Cxi = 02\ 06\ Ø2$$

$$Cg = Cxi\ Q1 + - - -$$

During phase O4 the C register shifts around to itself, parity is generated and a completion signal, Rti, is also generated.

$$Cs = \overline{Tp}\ \overline{T24}\ F1\ \overline{F3} + - - -$$

$$Cg = Cs + - - -$$

$$sC0 = Ø4\ \overline{01}\ \overline{04}\ 06\ C23 + - - -$$

$$sC24 = \overline{T24}\ \overline{T23}\ C0\ \overline{C24}$$

$$rC24 = T23 + \overline{T24}\ \overline{T23}\ C0\ C24 + (St)$$

$$Rti = Ø4\ \overline{01}\ \overline{04}\ 06$$

During phase Ø4 the P register is incremented and during phase Ø7 the information is generated in memory.

OUTPUT INSTRUCTIONS

Instructions 10 (MIY) and 12 (MIW)

Phasing: Ø0 → Ø2 → Ø6 → Ø0

During phase Ø0, the operand is accessed in the normal fashion and transferred to the C register. At Tp Ø0 Ia, the F2 flip-flop is set for either phase Ø2 or phase Ø6.

$$sF2 = Tp\ \overline{Ia}\ Ø0\ 03 + - - -$$

If the buffer is ready to receive a new word, the ready flip-flop is set during either phase Ø0 or Ø2.

$$sRf = \overline{Ts}\ \overline{C1}\ 03\ \overline{04}\ \overline{Ia}\ \overline{F1}\ \overline{F3}\ \left[\overline{Wf}\ (W0 + \overline{W9})\ 05\ \overline{06} + \overline{(\overline{Yf}\ (Y0 + \overline{Y9})}\ \overline{05}\right] + - - -$$

$$rRf = Tp\ End\ \overline{Sk}\ \overline{Ts} + - - -$$

Therefore, if the buffer is ready before the instruction is given, phase Ø2 is not reached and the phase counter goes directly to phase Ø6. If the buffer is not ready, the computer remains in phase Ø2 until Rf is set by the buffer. In phase Ø2 the C register does not shift.

During phase Ø6 the C register is shifted into the W or Y Buffer Word Assembly registers. At Tp the phase counter is reset to phase Ø0.

Instruction 13 (POT)

Phasing: Ø0 → Ø2 → Ø6 → Ø0

During phase Ø0 the operand is accessed in the normal fashion and transferred to the C register at Tp. During instruction 13 phase Ø2 cannot be by-passed, even if the external buffer is ready. This allows at least one cycle time for the parallel output transfer to occur. The ready flip-flop is not set until phase Ø2.

$$sRf = \overline{01}\ 03\ \overline{04}\ \overline{F1}\ \overline{F3}\quad Ø2\ 06\ Rt\ Q2\quad \overline{Ia}\ \overline{Ts} + - - -$$

$$rRf = Tp\ End\ \overline{Sk} + - - -$$

Rt is the appropriate external ready signal for output.

During phase Ø2 two signals are generated for external use. Pot 1 indicates that the information is being presented. Pot 2 can be used as a "strobe" signal for setting external flip-flops.

$$Pot\ 1 = \overline{F1}\ F2\ \overline{F3}\ \overline{Ts}\ \overline{02}\ 06$$

$$Pot\ 2 = Ø2\ \overline{02}\ 06\ Q1$$

F1 will not be set for advancing to phase Ø6 until the Rf flip-flop is set.

$$sF1 = \overline{Ts}\ Tp\ \overline{01}\ 03\ \overline{Ia}\ \overline{F1}\ \overline{F3} + - - -$$

Memory parity for the output data is checked during Ø6.

$$sCp = \overline{Cp}\ C23\ \overline{Ht}\ \overline{Tp}\ \overline{T24}\ (03\ Ø6 + - - -) + - - -$$

$$rCp = Cp\ C23\ \overline{Ht}\ \overline{Tp}\ \overline{T24}\ (03\ Ø6 + - - -) + - - -$$

During phase Ø6 the P register is incremented and at Tp the phase counter is reset to phase Ø0.

CONTROL INSTRUCTIONS

Instruction 23 (EXU)

Phasing: Ø0 → Ø0

During phase Ø0 the address of the instruction is shifted through the adder for possible indexing and at T9 transferred (Sxc) to the S register to access the instruction to be executed.

At Tp the O register is reset.

$$Oc = \overline{01}\ \overline{03}\ 05\ Tp\ \overline{Ia} + - - -$$

The P register is not affected and will be incremented by the accessed instruction in the normal fashion. The phase counter remains in phase Ø0.

Instructions 00, 02, and 20

Phasing: (Ø0) ⟶ (Ø5) ⟶ (Ø0)
T24

The phase counter remains in phase Ø0 for only one pulse time and then advances to phase Ø5. The C register does not shift during phase Ø5. Instruction 00 (HLT) sets the Ht flip-flop during phase Ø5 to initiate a halt,

$$sHt = Ø5\ T0\ \overline{01}\ \overline{02}\ \overline{05}\ \overline{Int} + - - -$$

and at Tp Ø5 the computer will halt.

$$rGo = Tp\ End\ \overline{Sk}\ \ Ht + - - -$$
$$End = F1\ F3\ \overline{Ts} + - - -$$

Instruction 02 (EOM) activates Eom signals to external devices and/or to the W Buffer during phase Ø5.

$$Eom = Ø5\ \overline{01}\ 05$$
$$Sys = Ø5\ \overline{01}\ 05\ \overline{C9}\ C10\ C11\ Q1$$
$$Ioc = Ø5\ \overline{01}\ 05\ \overline{C10}\ C11\ Q1$$
$$Buc = Ø5\ \overline{01}\ 05\ \overline{C10}\ \overline{C11}$$

where:

Sys = Systems communications

Ioc = Input/output control

Buc = Buffer control

Instruction 20 (NOP) does not cause any special operation to occur in phase Ø5 and all registers recirculate as normal.

The P register is incremented during phase Ø5 and the next instruction is accessed.

LOGICAL INSTRUCTIONS

Instructions 14, 16, and 17

Phasing: (Ø0) ⟶ (Ø6) ⟶ (Ø0)

During phase Ø0 the operand is accessed in the normal fashion and at Tp the phase counter is advanced to phase Ø6.

During phase Ø6 the C register will be shifted right and logical operations performed as it enters the A register.

$$Cs = \overline{Tp}\ \overline{T24}\ F1\ \overline{F3} + - - -$$

Instruction 14 (ETR) causes the contents of the A register and the operand to be "ANDED" together.

$$sAw = Anr\ \overline{01}\ 04\ \overline{06}\ An\ C23 + - - -$$
$$Anr = \overline{02}\ 03\ 04\ \emptyset 6 + - - -$$

Instruction 16 (MRG) causes the contents of the A register and the operand to be "inclusively OR'd" together.

$$\begin{aligned} sAw &= Anr\ \overline{01}\ 04\ 05\ An\ \overline{C23} \\ &+ Anr\ \overline{01}\ 04\ 05\ \overline{An}\ C23 \\ &+ Anr\ \overline{01}\ 04\ \overline{06}\ An\ C23 \\ &+ - - - \end{aligned}$$

Instruction 17 (EOR) causes the contents of the A register and the operand to be "exclusively OR'd" together.

$$\begin{aligned} sAw &= Anr\ \overline{01}\ 04\ 05\ An\ \overline{C23} \\ &+ Anr\ \overline{01}\ 04\ 05\ \overline{An}\ C23 \\ &+ - - - \end{aligned}$$

At $\emptyset 6$ Tp the phase counter is reset to phase $\emptyset 0$.

REGISTER CHANGE INSTRUCTIONS

Instruction 46 (XAB)

Phasing: $\underset{T24}{\emptyset 0} \longrightarrow \emptyset 5 \longrightarrow \emptyset 0$

The phase counter advances to phase $\emptyset 5$ after one pulse time. During phase $\emptyset 5$ certain logical transfers will take place between the A and B registers under the control of address bits C10 and C11.

$$sAw = \overline{02}\ \overline{03}\ 04\ \overline{Ts}\ Bn\ \overline{C10} + - - -$$
$$Anr = \overline{02}\ \overline{03}\ 04\ \overline{Ts} + - - -$$

$$sBw = \overline{02}\ \overline{03}\ 04\ \overline{Ts}\ An\ \overline{C11} + - - -$$
$$Bnr = \overline{02}\ \overline{03}\ 04\ \overline{Ts} + - - -$$

$\overline{C10}\ \overline{C11}$ (XAB): The contents of the A and B registers will interchange.

$\overline{C10}\ C11$ (BAC): The contents of B will be transferred to A and B will be cleared.

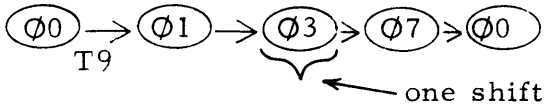
$C10\ \overline{C11}$ (ABC): The contents of A will be transferred to B and A will be cleared.

C10 C11 (CLR): Both A and B will be cleared.

C does not shift, the P register is incremented and a new instruction is accessed.

SHIFT INSTRUCTIONS

Instruction 66

Phasing: 

During phase Ø0 indexing may proceed as usual but only from T23 to T16 (8 bit times). The Ix flip-flop is reset at T16 so that indexing will not modify the instruction modifiers originally in bits C10 and C11. The instruction modifier (C10) determines if the operation is to be a right shift or a right cycle.

66 2 00XX (Right Cycle) RCY

66 0 00XX (Right shift) RSH

Bit C10 will ultimately be transferred to S1. The carry flip-flop is also forced off when Ix is off to prevent a carry into this bit position.

$$rIx = Ø0\ \overline{Ia}\ 02\ \overline{03}\ 04\ \ 05\ Q1\ \overline{Ts} + - - -$$

$$rCz = Ø0\ \overline{Ix} + - - -$$

The adder continues to feed C0 but after T15 its output must be the same as C23. At T10 the S register is cleared as usual during phase Ø0.

$$Sc = T10\ Ø0 + - - -$$

At T9 the C register is transferred to the S register,

$$Sxc = T9\ Ø0\ \overline{P0}\ \left[Ia + 03 + 02\right] + - - -$$

and at the same time the phase is changed to phase Ø1.

$$sF3 = T9\ 05\ \overline{Ia}\ Ø0\ 02\ \overline{03}\ 04\ + - - -$$

The memory is not pulsed at T8 by Mc because of phase Ø1. Therefore, the S register does not have to remain constant and it will count down the number of shifts required.

$$rS9 = \emptyset 3\ T24\ \overline{S10}\ \overline{S11}\ \overline{S12}\ \overline{S13}\ \overline{S14} + - - -$$

$$sS10 = \emptyset 3\ T24\ \overline{S10}\ \overline{S11}\ \overline{S12}\ \overline{S13}\ \overline{S14} + - - -$$

$$rS10 = \emptyset 3\ T24\ S10\ \overline{S11}\ \overline{S12}\ \overline{S13}\ \overline{S14} + - - -$$

$$sS11 = \emptyset 3\ T24)\ \overline{S11}\ \overline{S12}\ \overline{S13}\ \overline{S14} + - - -$$

$$rS11 = \emptyset 3\ T24\ S11\ \overline{S12}\ \overline{S13}\ \overline{S14} + - - -$$

$$sS12 = \emptyset 3\ T24\ \overline{S12}\ \overline{S13}\ \overline{S14} + - - -$$

$$rS12 = \emptyset 3\ T24\ S12\ \overline{S13}\ \overline{S14} + - - -$$

$$sS13 = \emptyset 3\ T24\ \overline{S13}\ \overline{S14} + - - -$$

$$rS13 = \emptyset 3\ T24\ S13\ \overline{S14} + - - -$$

$$sS14 = \emptyset 3\ T24\ \overline{S14} + - - -$$

$$rS14 = \emptyset 3\ T24\ S14 + - - -$$

At $\emptyset 1$ Tp the phase is changed to phase $\emptyset 3$ (unless the required shift count was zero).

$$sF2 = \emptyset 1\ Tp$$

The Rf flip-flop is time-shared and controls the right shift. It is set by $\emptyset 3\ T24\ \overline{0\upsilon}$.

$$sAw = Anr\ Rf\ 01\ \overline{F1}\ Ar\ \overline{T0}\ \overline{Tp}$$
$$+ Anr\ Rf\ 01\ 05\ \overline{T0}\ \overline{S1}\ Aw$$
$$+ Anr\ Rf\ 01\ 05\ T0\ S1\ Bn + - - -$$
$$Anr = Rf\ 01\ \overline{Ts}\ \overline{T24} - - -$$

The first term shifts A by by-passing the An flip-flop, feeding Aw directly from Ar. The second term extends the sign of A for right shift ($\overline{S1}$). The third term inserts the least significant bit of B, stored in Bn, into the sign positions of A for right cycle (S1).

An holds the least significant bit of A and Bn holds the least significant bit of B during the shift. They hold these bits because their respective enable terms are false.

$$Ag = Bg = \overline{Rf} + \overline{01} + T24 + Ts$$

The T24 term lets in each new least significant bit every cycle of the shift. Similarly,

$$sBw = Bnr\ \ Rf\ 01\ \ Br\ \overline{T0}$$
$$+ Bnr\ \ Rf\ 01\ \ An\ T0 + - - -$$

The least significant bit of A always shifts to the most significant bit of B. The S register S9 through S14 is counted down one at each Ø3 T24 time. The Sk flip-flop will be set when the count reaches zero.

$$sSk = \overline{F1}\ F3\left[\overline{S9}\ \overline{S10}\ \overline{S11}\ \overline{S12}\ \overline{S13}\ \overline{S14}\right]\ \overline{Ts} + - - -$$

The Sk flip-flop forces phase Ø7 and causes the Ia flip-flop to be set at T24 Ø7 so that the P register will be incremented. The P register is transferred to S at Ø7 T9 to access the next instruction.

Instruction 67

Phasing: Ø0 (T9) → Ø1 → Ø3 (one shift per cycle) → Ø7 → Ø0

During phase Ø0 indexing may proceed as usual but only from T23 to T16 (8 bit times). The Ix flip-flop is reset at T16 so that indexing will not modify the instruction modifiers originally in bits C10 and C11.

The instruction modifier C10 determines if the instruction is to be a left shift or a left cycle.

67 2 00XX (Left cycle) LCY

67 0 00XX (Left shift) LSH

67 1 00XX (Normalize) NOD

Bit position C11 determines if the instruction is to be a "normalize." Bit positions C10 and C11 will eventually be transferred to S1 and S2 of the S register.

At T16 the index flip-flop is reset, and at T15 the carry flip-flop is reset.

$$rIx = Ø0\ \overline{Ia}\ 02\ \overline{03}\ 04\ 05\ Q1\ \ \overline{Ts} + - - -$$
$$rCz = Ø0\ \overline{Ix} + - - -$$

At T10 Ø0 the S register is cleared; at T9 Ø0 the C register containing the modified shift count is transferred to the S register. Also, at T9 Ø0 the phase counter is advanced to phase Ø1.

$$sF3 = T9\ 05\ \ Ø0\ \overline{Ia}\ 02\ \overline{03}\ 04 + - - -$$

This phase inhibits the memory pulse, Mc, normally at T8, so that the memory

is not cycled and the S register is free to count, since it does not have to control the memory. At Ø1 Tp the phase counter steps to phase Ø3.

$$sF2 = Ø1\ Tp + - - -$$

The most significant bit of the B register is picked up by Dc at each Tp.

$$sDc = \overline{03}\ 04\ 06\ \overline{F1}\ \overline{Ts}\ Tp\ Bw + - - -$$
$$rDc = \overline{03}\ 04\ 06\ F2\ \overline{Ts}\ Tp\ \overline{Bw} + - - -$$

The Dc flip-flop is then used as an additional delay to shift the A register left.

$$sDc = \overline{03}\ 04\ 06\ F2\ \overline{Ts}\ An\ \overline{Tp}\ \overline{T24} + - - -$$
$$rDc = \overline{03}\ 04\ 06\ F2\ \overline{Ts}\ \overline{An}\ \overline{Tp}\ \overline{T24} + Tp\ End\ Go + - - -$$
$$sAw = Ø3\ 06\ Dc + - - -$$
$$Anr = \overline{03}\ 04\ 06\ F2\ \overline{Ts} + - - -$$

The P0 flip-flop is borrowed and used as the additional delay in shifting the B register left. P0 must pick up the most significant bit of A for the left cycle instruction. However P0 cannot be set at Tp since it also controls the C to M transfer at T24. Therefore the Ia flip-flop is borrowed and picks up the most significant bit of the A register at Tp and transfers it to P0 at T24.

$$sIa = \overline{F1}\ F3\ Aw\ Tp + - - -$$
$$rIa = T24\ Ø3 + - - -$$
$$sP0 = \overline{03}\ 04\ 06\ F2\ \overline{Ts}\ 05\ Ia\ S1\ T24 + - - -$$
$$rP0 = F2\ T0 + - - -$$

Here S1 is the modifier that causes left cycle rather than left shift. If S1 is false, then zeros are shifted into B. P0 then causes B to shift left.

$$sP0 = \overline{03}\ 04\ 06\ F2\ \overline{Ts}\ \overline{T0}\ \overline{Tp}\ \overline{T24}\ Bn + - - -$$
$$rP0 = \overline{03}\ 04\ 06\ F2\ \overline{Ts})\ \overline{T0}\ \overline{Tp}\ \overline{T24}\ \overline{Bn} + - - -$$

$$sBw = \overline{03}\ 04\ 06\ F2\ \overline{Ts}\ P0 + - - -$$
$$Bnr = F2\ \overline{03}\ 04\ 06\ \overline{Ts}$$

At each Ø3 T24 the S register is counted down. When it reaches zero the Sk flip-flop is set to force phase Ø7.

$$sSk = \overline{F1}\ F3\ \left[\overline{S9}\ \overline{S10}\ \overline{S11}\ \overline{S12}\ \overline{S13}\ \overline{S14}\right] + - - -$$
$$sF1 = sF2 = sF3 = Tp\ \overline{Ts}\ Sk$$

In phase Ø7 the P register is incremented and the next instruction is accessed. The O register is cleared to (20) at T24 Ø7 so that the above shift gates are inoperative. All registers recirculate during phase Ø7. The left shift can cause an overflow. This occurs when the sign of A changes, that is, when the new and old signs of A are unlike.

$$sOf = Ø3\ 06\ \overline{S1}\ T0\ An\ \overline{Dc}$$
$$+\ Ø3\ 06\ \overline{S1}\ T0\ \overline{An}\ Dc + - - -$$

The S1 term inhibits the setting of overflow during a left cycle operation. The normalize instruction is only slightly different. The Sk flip-flop is set before the shift

count reaches zero, provided that the two most significant bits of A are different.

$$sSk = [\overline{F1}\ F3\ \overline{Ts}\ 06\ S2\ T0]\ Aw\ \overline{An}$$
$$+[\overline{F1}\ F3\ \overline{Ts}\ 06\ S2\ T0]\ \overline{Aw}\ An + - - -$$

These gates set Sk, if the old sign bit is different from the new A1 bit, except during phase Ø1 when it compares the present two most significant bits. However, in phase Ø3 the old sign bit must be the same as the new sign bit because otherwise it would not be snifting.

The normalize instruction also causes the X register to be decremented, one for eacn left shift.

$$sXw = Xnr\ Ø3\ \overline{Ix}\ Xn$$
$$+\ Xnr\ Ø3\ Ix\ \overline{Xn} + - - -$$
$$Xnr = Ø3\ 06\ S2 + - - -$$
$$sIx = Ø3\ T24 + - - -$$
$$rIx = Ø3\ \overline{T24}\ Xn + Tp + - - -$$

S2 (normalize) causes the X register to be decremented. The Ix flip-flop is borrowed and used as a carry for the half adder feeding Xw.

ARITHMETIC INSTRUCTIONS

Instructions 54 (SUB) and 55 (ADD)

Phasing: (Ø0) ⟶ (Ø6) ⟶ (Ø0)

During phase Ø0 the operand is accessed in the normal fashion and at Tp the phase counter is advanced to phase Ø6. During phase Ø6 the contents of the C register are shifted through the adder and are added or subtracted from the contents of the A register.

$$Cs = F1\ \overline{F3}\ \overline{Tp}\ \overline{T24} + - - -$$
$$sAw = 01\ \overline{02}\ 03\ 04\ Ø6\ Add + - - -$$
$$Anr = \overline{02}\ 03\ 04\ Ø6 + - - -$$

Instruction 54 (SUB)

$$Xz = An\ F1\ 03 + - - -$$
$$Yz = F1\ \overline{C23}\ \overline{06}\ \overline{Rf} + - - -$$
$$sCz = Ø6\ T24\ \overline{06}\ \overline{Rf} + F1\ \overline{T24}\ \overline{Tp}\ Xz\ Yz\ \overline{Cz}\ \overline{Su} + - - -$$
$$rCz = \overline{T24}\ \overline{Xz}\ \overline{Yz}\ \overline{Su} + Tp + - - -$$

The Cz flip-flop is preset to complete the two's complement.

Instruction 55 (ADD)

$$Xz = An\ F1\ 03 + - - -$$
$$Yz = F1\ C23\ 06\ \overline{Rf} + - - -$$

At Tp Ø6 the phase counter is reset to phase Ø0. The overflow flip-flop will be set if the result of the arithmetic operation exceeds the allowable number range.

$$sOf = 01\ \overline{02}\ 03\ 04\ \phi 6\ T0\ (\overline{Xz}\ \overline{Yz}\ Cz + Xz\ Yz\ \overline{Cz})$$

The state of Of can then be tested with an SKS instruction.

Instruction 64 (MUS)

Phasing: $\phi 0 \longrightarrow \phi 6 \longrightarrow \phi 0$

During phase $\phi 0$ the operand (multiplicand) is accessed in the normal fashion. Also during phase $\phi 0$ the sign of the A register is extended two bits to the left forming a 26-bit word in A. During phase $\phi 0$ the A-B registers shift right one position. During phase $\phi 6$ the contents of C (multiplicand) are added or subtracted to the 25-bit word in A according to the three least significant bits in B (multiplier) at the beginning of the instruction. As this arithmetic is being done the result in A-B is shifted right another position. Since these tnree "Operator" bits appear too late during phase $\phi 6$ to properly begin the arithmetic, they are pre-examined during phase $\phi 0$.

MULTIPLIER BIT TABLE

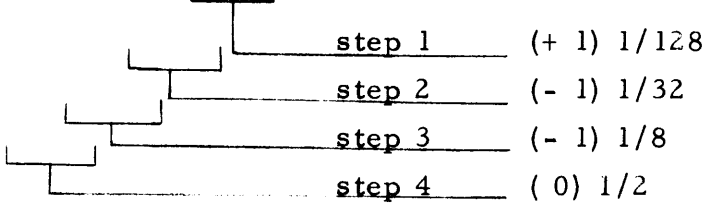
B Register			
B21	B22	B23	Operation
0	0	0	+ 0
0	0	1	A) + 2 (M)
0	1	0	A) + 2 (M)
0	1	1	A) + 4 (M)
1	0	0	A) - 4 (M)
1	0	1	A) - 2 (M)
1	1	0	A) - 2 (M)
1	1	1	- 0

The following heuristic proof is given to justify the use of the multiplier bit table shown above.

Approaching the computer as a fixed point, fractional device, let us evaluate 2 fractional numbers following the rules of the table, remembering that the register shifts one bit at a time.

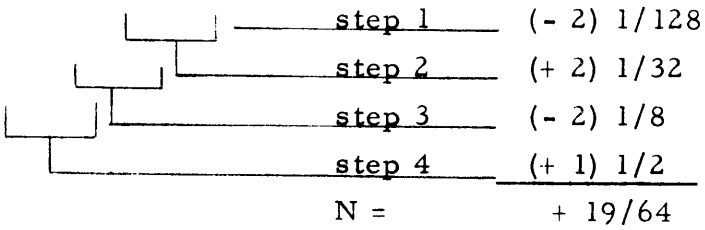
Example: - 19/128

N = 1.1 1 0 1 1 0 1 0



Example: + 19/64

N = 0.0 1 0 0 1 1 0 0



The Su flip-flop is used to detect a subtract condition and picks up bit B21. The Dc flip-flop is used to detect if bits B22 and B23 are alike. From the Multiplier Bit Table it can be seen that if these two bits are unlike, one times the multiplicand is to be used; if they are alike, then either twice the multiplicand is to be used or zero is to be used.

If Dc is reset, then addition or subtraction must start immediately when the least significant bits in A reach Ar during phase Ø6. This is at Ø6 T24. If Dc is set indicating B22 and B23 were originally alike, then one more clock time is available before arithmetic is to begin, since twice the multiplicand is to be used. If all three of the bits in B were alike, then zero is to be added to A. The P0 flip-flop is set at Ø6 T24 when this occurs and P0 gates off the input to the adder.

The A-B registers are shifted right during both cycle times. During phase Ø6 arithmetic is done simultaneously with the right shift. The Rf flip-flop controls right shifts.

$$sRf = \overline{C2}\ \overline{C5}\ C6\ \overline{C7}\ \overline{C8}\ \overline{C9}\ \ Ø0\ T24\ \overline{Ia}\ Go$$
$$+\ \overline{03}\ 04\ \overline{05}\ \overline{06}\ \overline{C9}\ \ Ø0\ T24\ Go\ \ \overline{Ts} + - - -$$

$$rRf = Tp\ End + - - -$$

The first set term above is for the case of direct addressing. The second term sets Rf only if the operand, or multiplicand, is indirectly addressed. In either case, then, Rf is set for two cycle times, phase Ø0 and Ø6.

During phase Ø0 the shift is simple:

$$sAw = Anr\ \ Rf\ 01\ \ \overline{F1}\ Ar\ \overline{T0}\ \overline{Tp}$$
$$+\ \ Rf\ 01\ \ Ø0\ \overline{Of}\ Aw\ (T0 + Tp)$$
$$Anr = \ Rf\ 01\ \ \overline{Ts}\ \overline{T24} + - - -$$

Note that the sign of A is extended two bit positions in A if the overflow flip-flop Of is reset; otherwise, zeros are extended.

$$sBw = \ Rf\ 01\ \ Br\ \overline{T0}\ Bnr$$
$$+\ \ Rf\ 01\ \ An\ T0$$

$$Bnr = \ Rf\ 01\ \ \overline{Ts}\ \overline{T24} + - - -$$

An holds the least significant bit of A and the second term above inserts that bit into B.

During phase Ø0 Dc is set if the two least significant bits of B were alike.

$$sDc = \ Rf\ 01\ Ø0\ \ \ T23\ (Br\ Bn + \overline{Br}\ \overline{Bn}) + - - -$$

$$rDc = Tp\ End\ Go + - - -$$

This will control the input to the adder. If the least significant two bits of B are alike, twice the multiplicand should feed the adder (if arithmetic is to be done at all). If the two bits are different, the multiplicand will be used directly and either addition or subtraction will definitely occur.

The Su flip-flop picks up the third least significant bit of B during phase Ø0. If it is set, it will control the carry flip-flop Cz as a "borrow" for subtraction.

$$sSu = Rf\ 01\ \emptyset 0\ \ Br\ T22 + - - -$$

$$rSu = Tp\ End\ Go$$

$$sCz = F1\ \overline{T24}\ Yz\ \ Xz\ \overline{Su}$$
$$+\ F1\ \overline{T24}\ Yz\ \ \overline{Xz}\ Su + - - -$$

$$rCz = \overline{T24}\ \overline{Xz}\ \overline{Yz}\ \overline{Su}$$
$$+ F1\ Xz\ \overline{Yz}\ Su + - - -$$

At T24 $\emptyset 6$, An receives Ar, if Dc, so that the bit can be transferred to the most significant position of B. If $\overline{Dc}$, the sum of C23 and Ar goes to An to be delivered to the most significant position of B.

$$Ag = T24 + \overline{Rf} + \overline{01} + Ts$$

$$sAn = \overline{Ar}\ \overline{Dc}\ C23\ T24\ \emptyset 6\ Rf\ 01$$
$$+\ Ar\ \left[\overline{\overline{Dc}\ C23\ T24\ \emptyset 6\ Rf\ 01}\right] + - - -$$

The P0 flip-flop is set at $\emptyset 6$ T24 to stop arithmetic if the three least significant bits of B were originally alike. Dc set indicates that the two least significant bits were alike. Br at T24 has the second bit and Su has picked up the third bit:

$$sP0 = Rf\ 01\ \emptyset 6\ T24\ (Br\ Su + \overline{Br}\ \overline{Su}) + - - -$$

$$rP0 = F2\ T0 + - - -$$

$\overline{P0}$ is gated to the adder input so that zeros will be added to A if all three bits in B were alike.

$$Yz = \emptyset 6\ Rf\ 01\ \ \overline{Dc}\ C22 + - - -$$
$$+\ \emptyset 6\ Rf\ 01\ \ Dc\ \overline{P0}\ C23 + - - -$$

$$Xz = Ar\ \emptyset 6\ Rf\ 01$$

Note that C22 feeds the adder if $\overline{Dc}$ is present; C23 feeds the adder if Dc is present. C23 presents twice the multiplicand whereas C22 presents one times the multiplicand. The timing is such that the first bit in A to be added is in Ar at T24. Remember that the first bit of the result is stored in An to be deposited later in the B register The sign of the multiplicand is extended.

$$sC0 = \overline{03}\ 04\ \overline{05}\ \emptyset 6\ C0\ \overline{Tp}\ \overline{Ts} + - - -$$

The carry flip-flop may be set at T24 if Dc,

$$sCz = Rf\ 01\ \emptyset 6\ \ T24\ \overline{Dc}\ C23\ \ (\overline{Su}\ Ar + Su\ \overline{Ar}) + - - -$$

and afterwards Cz is controlled by Su for addition or subtraction. The adder feeds Aw.

$$sAw = 01\ 02\ \overline{03}\ 04\ \overline{05}\ \ \emptyset 6\ Add + - - -$$

During phase $\emptyset 6$, Ia adds one to the P register and the next instruction is accessed as usual.

The overflow flip-flop is set in one special case. This case is when two times - 1 is subtracted from A when A is already zero. This + 2 result overflows even the 25-bit register of A, (25 bits before the last shift.)

$$sOf = Rf\ 01\ \emptyset 6\ T0\ Su\ Dc\ \overline{P0}\ \overline{Cz}\ \overline{Ar}\ C23 + - - -$$

In this term Su Dc $\overline{P0}$ indicates that subtraction of twice the multiplicand is taking place, that is, the bits in B were 100. $\overline{Cz}$ $\overline{Ar}$ C23 gives the adder output a "one", yet C23 $\overline{Ar}$ will turn on the carry, Cz, for the next clock time. If the signs of A and C were to be further extended, the next bit from the adder would be a zero and hence, the overflow.

When the overflow is set, the next multiply step should extend zeros to the left of A, not ones. This is why $\overline{Of}$ is present in the gate that extends the sign of A. If there are no further multiply step instructions, then Of being set indicates that the multiplication performed was - 1 x - 1 = + 1 and the overflow remains set.

After the bits in A are extended as zeros during the multiply step instruction, the overflow flip-flop is reset.

$$rOf = Rf\ 01\ \emptyset 6\ T24 + - - -$$

Instruction 65 (DIS)

Phasing: (Ø0)→(Ø6)→(Ø0)

The function of the instruction is to shift A-B left one bit position. The complement of the sign of A is inserted in the least significant bit position of B. If the signs of A and (M) are alike, (M) is subtracted from the shifted A register. If the signs are different, (M) is added to the shifted A register.

During phase Ø0 the operand is accessed as usual. The only different activity performed is that the Dc flip-flop, which is used in phase Ø6 to shift A left, picks up the most significant bit of B.

$$sDc = \overline{03}\ 04\ 06\ \overline{Ts}\ Tp\ Bw\ \overline{F1} + - - -$$

At Ø6 T24 the Su flip-flop is set for subtraction if the signs of C and A are alike:

$$sSu = F2\ \overline{Ts}\ \overline{03}\ 04\ 06\ T24\ (\overline{C0}\ \overline{Aw} + C0\ Aw) + - - -$$

$$rSu = Tp\ End\ Go$$

The sign of A is in Aw at T24 because the sign of A is always extended when A is recirculated.

$$sAw = \overline{Anr}\ An\ \overline{Tp}$$
$$+ \overline{Anr}\ Aw\ Tp + - - -$$

During phase Ø6 An feeds Dc to shift A left and Dc feeds the adder. C23 feeds the second input to the adder. Su controls the carry flip-flop, Cz, if subtraction is necessary.

$$sDc = F2\ \overline{03}\ 04\ 06\ An\ \overline{T24} + - - -$$

$$rDc = F2\ \overline{03}\ 04\ 06\ \overline{An}\ \overline{T24} + - - -$$

$$Xz = F2\ \overline{03}\ 04\ 06\ Dc + - - -$$

$$Yz = F1\ C23\ 06\ \overline{Rf} + - - -$$

$$sAw = 01\ 02\ \overline{03}\ 04\ \overline{05}\ \emptyset 6\ Add + - - -$$

The B register is shifted by P0. Since the quotient bit to enter the least significant position of B is the complement of the sign of A, P0 is set by $\overline{Aw}$ at 24.

$$sP0 = F2\ \overline{03}\ 04\ 06\ \overline{05}\ \overline{Aw}\ T24$$
$$+ F2\ \overline{03}\ 04\ 06\ Bn\ \overline{T24}\ \overline{T0}\ \overline{Tp} + - - -$$
$$rP0 = F2\ \overline{03}\ 04\ 06\ \overline{Bn}\ \overline{T24}$$
$$+ F2\ T0$$
$$+ P0\ T24 + - - -$$
$$sBw = F2\ \overline{03}\ 04\ 06\ P0 + - - -$$

During phase Ø4 the P register is incremented and the next instruction is accessed as usual.

MEMORY PARITY

MEMORY PARITY GENERATION

The memory has a capacity for 25-bit words. One of these bits is used for parity. The total number of ones in any word in memory, including the parity bit, must be even. If a word read out of memory has an odd number of ones, then normally the computer will halt and the memory parity indicator on the control panel will be lit. Parity is automatically generated on words stored in the memory.

The C24 flip-flop generates parity on words entering the C register serially. It counts ones in the C0 flip-flop as the entering bits step through; it counts from T22 to Tp.

$$sC24 = (\overline{T24}\ \overline{T23})\ C0\ \overline{C24}$$

$$rC24 = (\overline{T24}\ \overline{T23})\ C0\ C24 + T23 + (St)$$

The C24 flip-flop is transferred to M24 with the rest of the C register when Mxc is actuated. This is at T24 time. Even on the parallel input (PIN) instruction, C24 generates parity by shifting the contents of C serially in phase Ø4.

MEMORY PARITY CHECKING

The Cp flip-flop checks parity on words read from memory. It counts ones as they step out of the C register at C23. It is preset by the parity bit from memory. If, after it has counted all the ones, it is left set, it will cause Ht to be set and the computer will halt.

When the Control switch is in the Idle position and Go and Ht are both reset, which is normal, Cp is reset every T24 time. It may be set at Tp for one pulse time, if the parity bit in memory M24 is set, but Cp will be reset again at T24.

When the Control switch is moved and Go is set, Cp will be properly preset at the same time Go is set.

$$sCp = M24\ Tp\ \overline{Ht}\ \overline{Ts} + - - -$$

$$rCp = \overline{Go}\ \overline{Ht}\ T24 + - - -$$

During any phase that parity is not being checked it will be turned off at Q1 time.

$$rCp = \overline{(\overline{F1}\ \overline{F2}\ \overline{Ts}\ \overline{Wp} + 01\ Ø4 + 02\ Ø6 + 03\ Ø6)}\ \overline{Ts}\ \overline{Ht}\ Q1$$

Cp is preset by the memory parity bit and then is complemented by every "one" stepping out of the C register at C23 during certain phases.

$$sCp = \overline{Cp}\ C23\ \overline{Ht}\ (T23 - T0)\ (\overline{F1}\ \overline{F2}\ \overline{Ts}\ \overline{Wp} + 01\ \emptyset 4 + 02\ \emptyset 6 + 03\ \emptyset 6)$$

$$rCp = Cp\ C23\ \overline{Ht}\ (T23 - T0)\ (\overline{F1}\ \overline{F2}\ \overline{Ts}\ \overline{Wp} + 01\ \emptyset 4 + 02\ \emptyset 6 + 03\ \emptyset 6)$$

The above term gives the phases when parity is checked. If at Tp the Cp flip-flop is set, an odd number of ones must have come from memory. If so, an error is indicated and Ht is set to halt the computer,

$$sHt = Cp\ Tp\ \overline{Ts}\ (Kp) + - - -$$

where (Kp) is from the Parity switch on the control panel and is true in the halt position. When both Cp and Ht are set, they lock each other set. The gate Cp Ht controls the parity error light on the control panel.

When Cp and Ht are set, the Control switch is inoperative because Ht will not reset. The computer idles because Go will reset at the completion of the instruction. The computer is inoperative until the Parity switch is placed in the continue position and the Control switch is returned to Idle.

$$rCp = (\overline{Kp})\ Ht + - - -$$

$$rHt = (\overline{Ks})\ (\overline{Kg})\ \overline{Go}\ \overline{Cp}\ T24 + - - -$$

The Cp flip-flop will not trigger when Ht is set until the parity switch is placed in Continue.

Parity is not checked on certain instructions from the memory.

These are the instructions that use phase $\emptyset 5$ (00, 02, 20, 40, 46). Other operation codes that are not used also cause $\emptyset 5$ and they are not parity-checked (04, 06, 22, 24, 26, 42, 44). The reason is that the C register cannot shift during phase $\emptyset 5$.

Words from the memory delivered to the W or Y Buffer during a time-share interlace are not parity-checked. $\overline{Ts}$ inhibits the check.

When the computer is locked up with a parity error (Cp Ht), an interrupt request will not re-start the computer.

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

The first instruction executed when starting or stepping will not be checked because this instruction may have been manually generated by the operator, using the control panel. This is accomplished by inhibiting Cp when Wp is on.

$$sCp = - - - (\emptyset 0\ \overline{Wp}\ \overline{Ts} + - - -)\ \overline{Cp}\ C23$$

$$rCp = - - - (\emptyset 0\ \overline{Wp}\ \overline{Ts} + - - -)\ Cp\ C23$$

Wp will always be turned on during the idle phase,

$$sWp = (\overline{Go} + Tsw)\ \overline{Tsy}\ \overline{Tsm}\ Tp$$

and it will be turned off at Tp time at the end of the first phase $\emptyset 0$.

$$rWp = (Go\ \overline{Tsw} + (Tsy))\ \overline{Tsm}\ Tp + - - -$$

IMPLEMENTATION

The algebraic equations discussed in preceding sections are implemented in the computer using diode-transistor logic (DTL).

The diode-transistor circuits are all silicon and are divided into several module types. Each of these modules uses printed circuit techniques on glass epoxy boards. The output signal levels for the circuits are:

one = true output = + 9.5v to + 6.5v

zero = false output = + 0.6v to 0.0v

The supply voltages to all circuits are +25, +8, 0v, and -25 v. The following descriptions are of individual circuit types and do not represent the contents of any actual module.

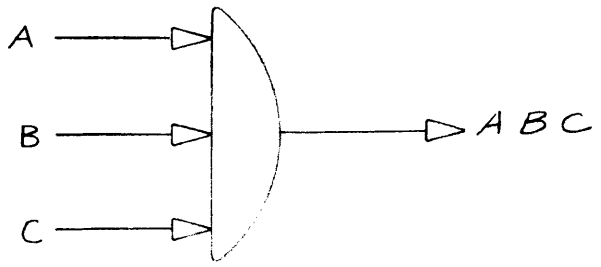
GATES

AND Gates

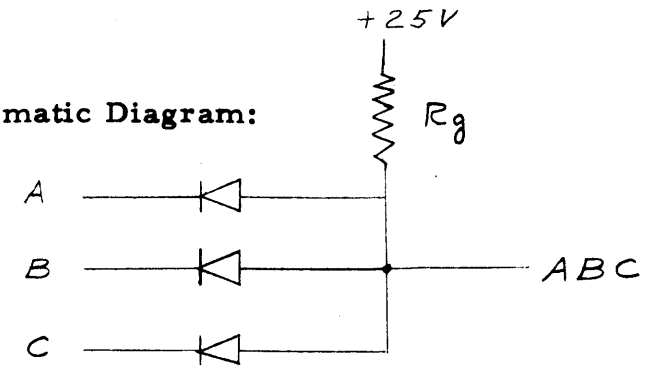
The algebraic expression ABC represents an "AND" expression and is implemented by the use of an AND gate composed of diodes and resistors.

The function of an AND gate is that it will yield a high output (true) only if all of its inputs are true.

Logic Diagram:



Schematic Diagram:

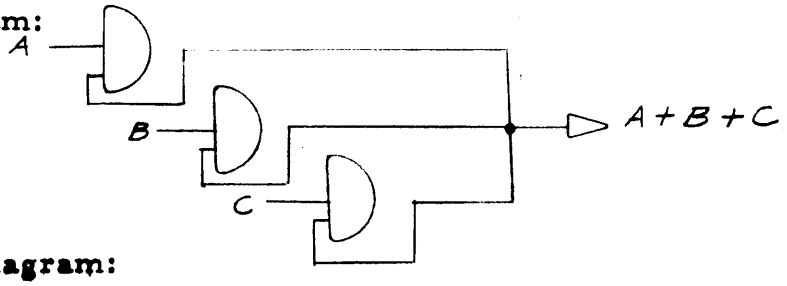


The AND gate operates as follows: If any one of the inputs is false (0 volts), it must absorb the full load of Rg and the output point (ABC) will fall to 0 volts (except for the forward drop at the diode); when all three inputs are true (approximately + 8v), the output point will be at the potential of the most negative of the inputs.

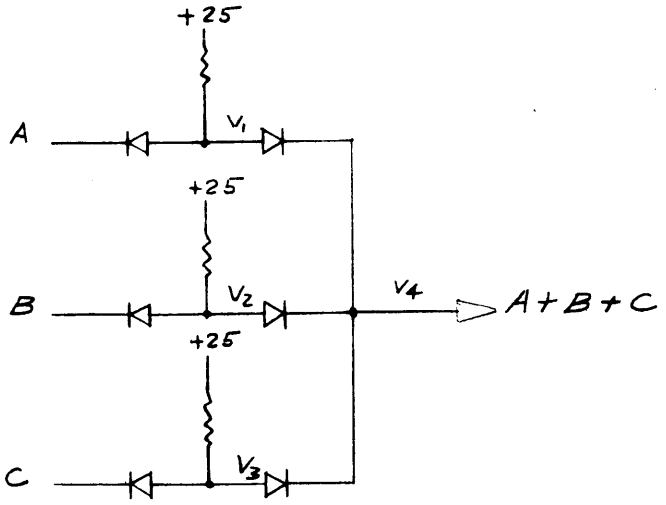
OR Gates

The algebraic expression A + B + C represents an OR expression and is implemented through the use of an OR gate composed of diodes and resistors. The OR gate shown below generated the logical OR function of several input signals. The input signals may be provided by inverters, buffers, or flip-flops.

Logic Diagram:



Schematic Diagram:

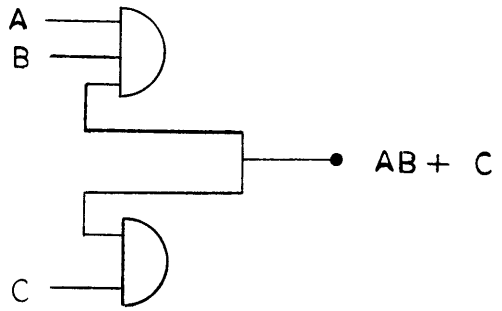


If any one of the input terms is true (+8v), its output point, V1, V2, or V3, will be at + 8 and will hold the output V4 true, and the other two OR diodes will be back-biased unless a similar condition exists.

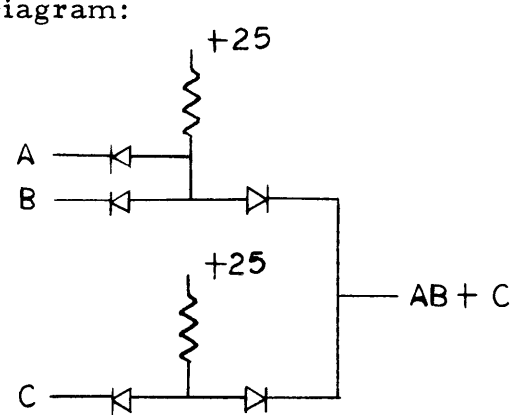
AND-OR Gate

AND-OR expressions such as AB + C can be implemented in the following manner:

Logic Diagram:



Schematic Diagram:



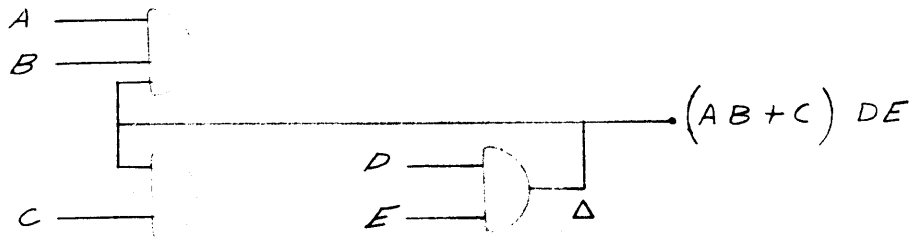
This circuit operates essentially the same as the previously described OR gate with additional diodes on the input to form AND gates. Or it can be said that all OR gates must be fed from AND gates, though they may have a single term, as in the previous case.

An OR gate can be formed by connecting one input terminal of each of several AND gates. The resulting OR output must be connected to the input terminal of an Inverter, Buffer, or a flip-flop.

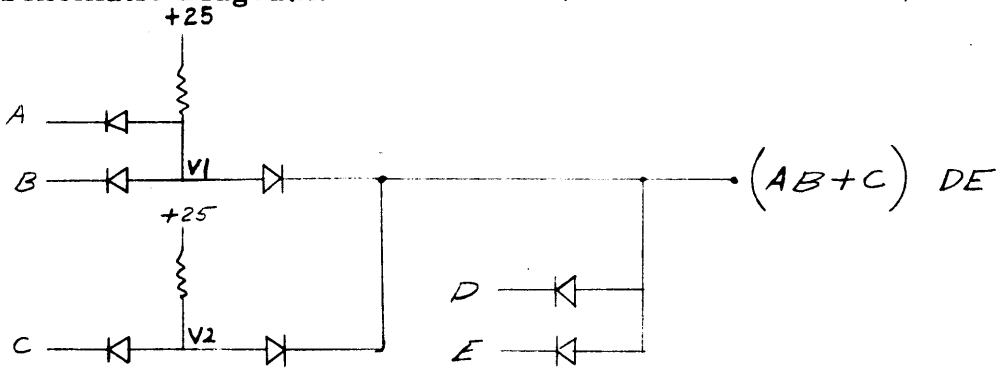
AND-OR-AND Gate

An algebraic AND-OR-AND expression, such as (AB + C) DE, can be implemented in the following manner:

Logic Diagram:



Schematic Diagram: (△means no resistor)



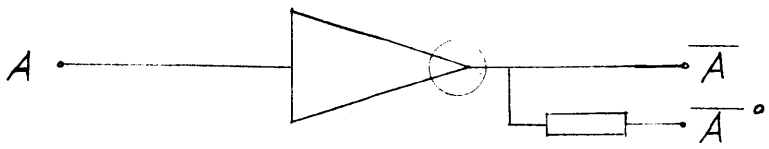
Note that if D or E goes false they will absorb the current load of both Rg resistors and the output will be false as well as points V1 and V2.

INVERTERS, BUFFER AMPLIFIERS, AND FLIP-FLOPS

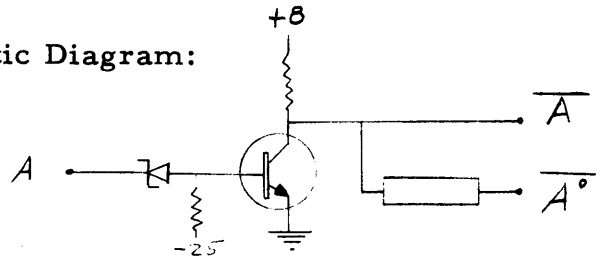
Inverters

An inverter circuit generates the inverse of the output of any gate. The input circuit has a Zener diode to provide a stable input threshold of approximately + 3 volts. This reduces noise problems as it essentially provides noise rejection. This Zener also clamps input signals to approximately + 3 to + 6 volts through the base emitter junction of the transistor.

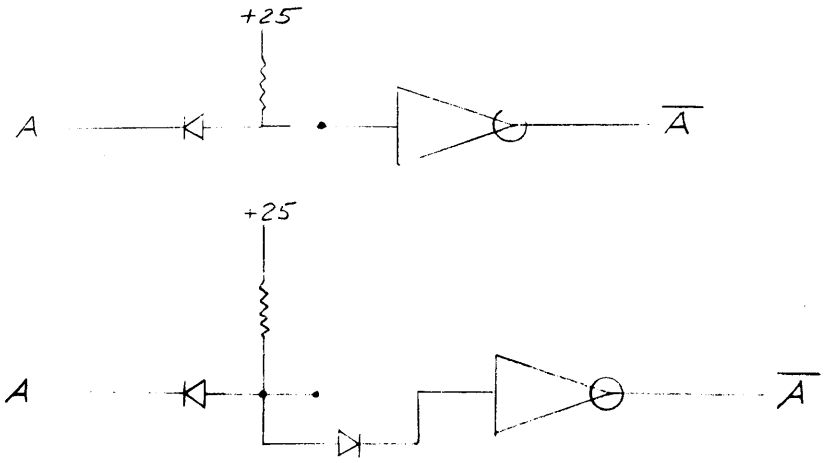
Logic Diagram:



Schematic Diagram:



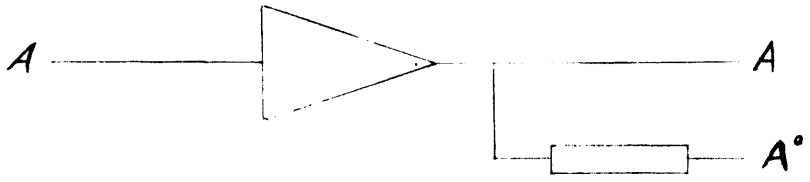
To invert a single input signal an AND gate or an AND-OR gate coupling circuit is used with an inverter and must be fed from isolated circuits.. The $\overline{A}^{o}$ output uses a high loss magnetic element to reduce ringing on long connecting lines.



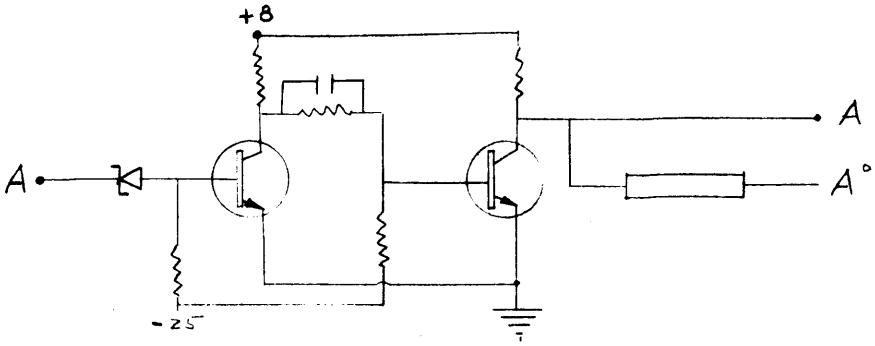
Buffer Amplifiers

A buffer amplifier generates a buffered signal from the output of any gate. The input circuit has a Zener diode to provide a stable input threshold of approximately + 3 volts. The rules for coupling to the input are the same as those for the inverter.

Logic Diagram:

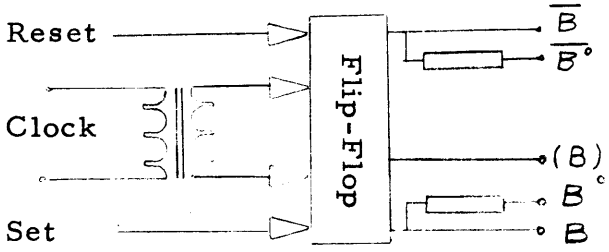


Schematic Diagram:



Flip-Flops

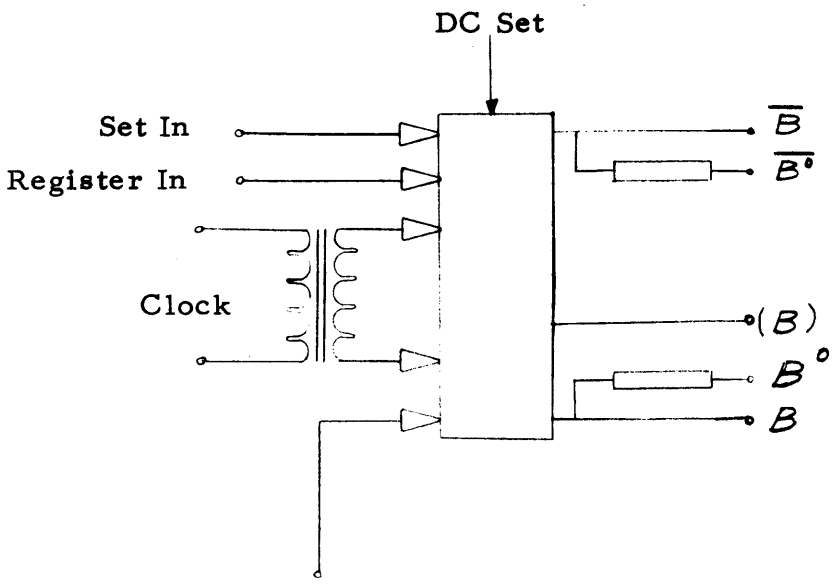
Two basic types of flip-flops are used. The first is a standard, set-reset flip-flop shown below. All flip-flops are clocked on input and clock is always transformer-coupled. The clock signal is approximately 100 nanoseconds wide (positive signal) and the flip-flop will trigger on the trailing edge.



The B^{o} and $\overline{B}^{o}$ outputs use special high loss magnetic elements to reduce line ringing and are used for outputs where the wire length is considerable.

The (B) output is used for additional source power and can be coupled directly to B or to a separate load resistor and used independently.

The second type of flip-flop is the repeater. This flip-flop cannot set or reset if the "hold" input is false. If the hold input is true the flip-flop will reset if there is no true input present on the set side. This flip-flop has similar outputs to the Rs-type but also has a DC set input and a special input from the dynamic register card.



REPEATER FLIP-FLOP

CONDENSED COMPUTER LOGIC EQUATIONS

A REGISTER

sAw	= $\overline{\text{Anr}}$ An $\overline{\text{Tp}}$
(65)	+ $\overline{\text{Anr}}$ Aw Tp
(76)	+ Anr 01 02 03 04 C23 05
(16, 17)	+ Anr $\overline{01}$ 04 05 An $\overline{\text{C23}}$
(16, 17)	+ Anr $\overline{01}$ 04 05 $\overline{\text{An}}$ C23
(16, 14)	+ Anr $\overline{01}$ 04 $\overline{06}$ An C23
(64, 66)	+ Anr Rf 01 $\overline{\text{F1}}$ Ar $\overline{\text{T0}}$ $\overline{\text{Tp}}$
(66)	+ Anr Rf 01 05 T0 $\overline{\text{S1}}$ Aw
(66)	+ Anr Rf 01 05 T0 S1 Bn
(46)	+ $\overline{02}$ $\overline{03}$ 04 $\overline{\text{Ts}}$ Bn $\overline{\text{C10}}$
	+ Anr Ex C23
(54, 55)	+ 01 $\overline{02}$ 03 04 Ø6 Add
(64, 65)	+ $\overline{\text{Ts}}$ $\overline{03}$ 04 $\overline{05}$ Ø6 Add
(64)	+ Rf 01 Ø0 $\overline{0f}$ Aw (T0 + Tp)
(67)	+ Ø3 06 Dc

sAn	= $\overline{\text{Ar}}$ $\overline{\text{Dc}}$ C23 T24 Ø6 Rf 01
	+ Ar $\overline{\overline{\text{Dc}}\ \text{C23 T24 Ø6 Rf 01}}$

An enable = Ag = $\overline{\text{Rf}}$ + $\overline{01}$ + T24 + Ts

Anr	=
(14-17, 54-57)	+ $\overline{02}$ 03 04 Ø6
(76, 74)	+ 01 02 03 04 $\overline{06}$ Ø6
(46)	+ $\overline{02}$ $\overline{03}$ $\overline{\text{Ts}}$ 04
	+ Ex (Ka)
	+ Ex Dc $\overline{\text{Su}}$
(64, 66)	+ Rf 01 $\overline{\text{Ts}}$ $\overline{\text{T24}}$
(65, 67)	+ F2 $\overline{03}$ 04 06 $\overline{\text{Ts}}$

B REGISTER

sBw	=
	+ $\overline{\text{Bnr}}$ Bn
(75, 74)	+ Bnr 03 C23
(46)	+ $\overline{02}$ $\overline{03}$ 04 $\overline{\text{Ts}}$ An $\overline{\text{C11}}$
	+ Bnr Ex C23
(64, 66)	+ Bnr Rf 01 Br $\overline{\text{T0}}$
(64, 66)	+ 01 Rf $\overline{\text{T24}}$ $\overline{\text{Ts}}$ T0 An
(65, 67)	+ F2 $\overline{03}$ 04 06 $\overline{\text{Ts}}$ P0

sBn = Br

Bn enable = Ag= $\overline{\text{Rf}}$ + $\overline{01}$ + T24 + Ts

Bnr	=
(75, 74)	+ 01 02 03 04 $\overline{05}$ Ø6
(46)	+ $\overline{02}$ $\overline{03}$ 04 $\overline{\text{Ts}}$
	+ Ex $\overline{\text{Dc}}$ Su
	+ Ex (Kb)
(64, 66)	+ Rf 01 $\overline{\text{Ts}}$ $\overline{\text{T24}}$
(65, 67)	+ F2 $\overline{03}$ 04 06 $\overline{\text{Ts}}$

C REGISTER

sC0 =

(30) + Ø4 $\overline{01}$ $\overline{04}$ $\overline{05}$ (Yn)

(37) + Ø4 $\overline{01}$ 04 05 06 Xn

(32) + Ø4 $\overline{01}$ $\overline{04}$ 05 $\overline{06}$ Wn

(33) + Ø4 $\overline{01}$ $\overline{04}$ 06 C23

(35) + Ø4 $\overline{01}$ 04 $\overline{05}$ An

(36) + Ø4 $\overline{01}$ 04 $\overline{06}$ Bn

(60, 61) + Ø4 01 C23 $\overline{Cz}$

(60, 61) + Ø4 01 $\overline{C23}$ Cz

(01, 41, 43) + Ø0 $\overline{Ia}$ $\overline{02}$ $\overline{03}$ P14 Q2

(01, 41, 43) + Ø0 $\overline{Ia}$ $\overline{02}$ $\overline{03}$ 0f T0

+ P0 Ø0 Q2 P14

+ P0 Ø0 0f T0

+ P0 Ø0 T9

(64, 65) + $\overline{03}$ 04 $\overline{05}$ Ø6 C0 $\overline{Tp}$ $\overline{Ts}$

+ Ex Dc $\overline{Su}$ An

+ Ex $\overline{Dc}$ Su Bn

+ Ex Dc Su Xn

+ Ex (Ka) An

+ Ex (Kb) Bn

+ Ex (Kx) Xn

+ $\overline{P0}$ Ø0 $\overline{Ø0\ Ia\ 02\ 03\ 01}$ Add Q2

+ Cxm M0

+ Cxi Cd0

+ (Kc0)

+ Ts Wn Wp $\overline{Tp}$

+ Ts (Yn) $\overline{Wp}$ $\overline{Tp}$

C REGISTER (continued)

sC1 = Cs C0
+ Cxm M1
+ Cxi Cd1
+ (Kc1)

similarly C2 - C23

Also sC4, sC5, sC7, sC22 with (Kf)

C register enable = Cg

Cg =
+ Cs
+ Cxm
+ Cxi Q1
+ (Kf) Ht

Shift C

Cs = $\overline{Tp}\ \overline{T24}$ Ex
(Ø4, Ø6) + $\overline{Tp}\ \overline{T24}$ F1 $\overline{F3}$
(Ø0, Ø1) + $\overline{Tp}\ \overline{T24}\ \overline{F1}\ \overline{F2}$
+ Ts $\overline{Tp}\ \overline{T24}$
+ (St) $\overline{Tp}\ \overline{T24}$
+ (Kcr) $\overline{Tp}\ \overline{T24}$

Transfer Memory to C

Cxm' =
+ $\overline{P0}\ \overline{Ø0\ \overline{Ia}\ \overline{02}\ \overline{03}\ 01\ Ø0}$
+ Tsm Wp W9
+ Tsm $\overline{Wp}$ (Y9)
+ End Go
+ $\overline{Ht}$ Int $\overline{Su}$ $\overline{Dc}$ $\overline{Ts}$ $\overline{C_p}$

Cxm = Cxm' 1p

Transfer Inputs to C

$Cxi = \emptyset 2\ 06\ 02$

Parity Generation

$$sC24 = \overline{T24}\ \overline{T23}\ C0\ \overline{C24}$$

$$rC24 = T23 + \overline{T24}\ \overline{T23}\ C0\ C24 + (St)$$

Parity Check

$$sCp = M24\ Tp\ \overline{Ht}\ \overline{Ts} + \overline{Cp}\ [C23\ \overline{Ht}\ \overline{Tp}\ \overline{T24}\ (\overline{F1}\ \overline{F2}\ \overline{Ts}\ \overline{Wp} + 01\ \emptyset 4 + 02\ \emptyset 6 + 03\ \emptyset 6)]$$

$$rCp = \overline{Go}\ \overline{Ht}\ T24 + Cp\ [C23\ \overline{Ht}\ \overline{Tp}\ \overline{T24}\ (\overline{F1}\ \overline{F2}\ \overline{Ts}\ \overline{Wp} + 01\ \emptyset 4 + 02\ \emptyset 6 + 03\ \emptyset 6)] + \overline{(\overline{F1}\ \overline{F2}\ \overline{Ts}\ \overline{Wp} + 01\ \emptyset 4 + 02\ \emptyset 6 + 03\ \emptyset 6)}\ \overline{Ts}\ \overline{Ht}\ Q1 + (St) + (\overline{Kp})\ Ht$$

Parity error light = Cp Ht

Halt light = $\overline{Cp}$ Ht

Carry

```
sCz         =
            + Xz Yz Cz F1 Q2
(54)        + Ø6 T24 06 Rf
(60,61)     + T24 Ø4 01
            + F1 T24 Tp Su Xz Yz Cz
            + F1 T24 Tp Yz Xz Su
(64)        + Rf 01 Ø6 T24 Dc C23 Su Ar
(64)        + Rf 01 Ø6 T24 Dc C23 Su Ar

rCz         =
            + Ø0 Ix
            + F1 Xz Yz Cz
            + T24 Xz Yz Su
            + F1 Xz Yz Su T24
            + Tp
```

Shifts A left - Delays C in MUS

```
sDc         = Rf 01 Ø0 T23 Br Bn
            + Rf 01 Ø0 T23 Br Bn
            + F2 Ts 03 04 06 An T24 Tp
            + 03 04 06 Tp Bw Ts F1
            + Ex T0 Dc (Ka)
            + Ex T0 Dc (Kx)

rDc         =
(65, 67)    + F2 Ts 03 04 06 An T24 Tp
(65, 67)    + F2 Ts 03 04 06 Tp Bw
            + Tp End Go
            + Ex T0 Dc
            + (St)
```

Enable

sEn = Eom C10 $\overline{C11}$ C22 T0

rEn = Eom C10 $\overline{C11}$ C21

+ (St)

End

End =

(Ø5, Ø7) + F1 F3 $\overline{Ts}$

(Ø6, Ø7) + F1 F2 $\overline{Ts}$

(01) + Ø0 $\overline{Ia}$ $\overline{02}$ $\overline{03}$ $\overline{01}$ $\overline{Ts}$

Eom Signals

Eom = Ø5 $\overline{01}$ 05

System Communications

(Sys) = Ø5 $\overline{01}$ 05 $\overline{C9}$ C10 C11 Q1

Input/Output Control

Ioc = Ø5 $\overline{01}$ 05 $\overline{C10}$ C11 Q1

Buffer Control

Buc = Ø5 $\overline{01}$ 05 $\overline{C10}$ $\overline{C11}$

Exchange C and A, B or X

sEx = $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ Dc Ks
+ $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ Dc Kg
+ $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ Dc Kc'
+ $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ Su Ks
+ $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ Su Kg
+ $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ Su Kc'
+ $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ Dc Kb'
+ $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ Su Ka'
+ $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ $\overline{Dc}$ Ka'
+ $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ $\overline{Dc}$ Kx'
+ $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ $\overline{Su}$ Kx'
+ $\overline{Ts}$ T24 $\overline{Go}$ $\overline{Ht}$ $\overline{Kf}$ $\overline{Su}$ Kb'

rEx = F1 T0

Phase Flip-Flops

sF1 (60, 61, 64, 65, 50-55, 70-75, 41, 43) = Tp $\overline{Ts}$ (Sk + P0 + Ø4)
+ Tp $\overline{Ia}$ Ø0 01 $\overline{04}$
+ Tp $\overline{Ia}$ Ø0 01 $\overline{05}$

(14-17, 34-37, 76, 77)
+ Tp $\overline{Ia}$ Ø0 03 04
+ T24 Go $\overline{Ia}$ Ø0 $\overline{C5}$ $\overline{C8}$ $\overline{C2}$ ($\overline{C3}$ + $\overline{C4}$)
+ (T24 $\overline{Go}$)
+ Tp $\overline{Ts}$ $\overline{01}$ 03 $\overline{04}$ $\overline{Ia}$ $\overline{F1}$ $\overline{F3}$ Rf

rF1 = Tp End $\overline{Sk}$

Phase Flip-Flops (continued)

sF2 =

+ Tp $\overline{Ts}$ (Sk + P0 + Ø4)

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

(1-, 3-, 5-, 7-) + Tp $\overline{Ia}$ Ø0 03

+ Ø1 Tp

(64, 65) + $\overline{Ts}$ $\overline{03}$ 04 $\overline{05}$ Tp Ø0 $\overline{Ia}$

rF2 = Tp End $\overline{Sk}$

+ Ø6 $\overline{01}$ 02 03

sF3 =

+ Tp $\overline{Ts}$ (Sk + P0 + Ø4)

+ T24 Go $\overline{Ia}$ Ø0 $\overline{C5}$ $\overline{C8}$ $\overline{C2}$ ($\overline{C3}$ + $\overline{C4}$)

+ T24 $\overline{Go}$

(66, 67) + T9 05 $\overline{Ia}$ Ø0 02 $\overline{03}$ 04 $\overline{Ts}$

rF3 =

+ Tp End $\overline{Sk}$

Phases

Ø0 = $\overline{F1}$ $\overline{F2}$ $\overline{F3}$ $\overline{Ts}$

Ø1 = $\overline{F1}$ $\overline{F2}$ F3 $\overline{Ts}$

Ø2 = $\overline{F1}$ F2 $\overline{F3}$ $\overline{Ts}$

Ø3 = $\overline{F1}$ F2 F3 $\overline{Ts}$

Ø4 = F1 $\overline{F2}$ $\overline{F3}$ $\overline{Ts}$

Ø5 = F1 $\overline{F2}$ F3 $\overline{Ts}$

Ø6 = F1 F2 $\overline{F3}$ $\overline{Ts}$

Ø7 = F1 F2 F3 $\overline{Ts}$

Control Flip-Flops

sGo = $\overline{Dc}\ \overline{Su}\ \overline{Ts}\ Tp\ \overline{Ht}\ (\ (Ks) + (Kg)(Kf)\ )\ Mg$

rGo = Tp End $\overline{Sk}$ Go Ht
+ (St)

sHt =

(external halt) + $\overline{Ts}$ (H) T0 $\overline{Int}$

(00) + Ø5 T0 $\overline{01}$ $\overline{02}$ $\overline{05}$ $\overline{Int}$
+ (Kg) Go T0 End
+ Cp Tp (Kp) $\overline{Ts}$
+ (St)

rHt = (Ks)(Kg) $\overline{Go}$ $\overline{Cp}$ T24
+ (Kg) Int $\overline{Cp}$
+ (Kf) $\overline{Wp}$

Indirect Addressing

sIa =
+ Ø0 $\overline{Ia}$ T24 Go $\overline{C2}$ C9
+ T24 Ø7 Sk

(Ø4, Ø6) + T24 F1 $\overline{F3}$ (Ij)
+ T24 Go Ø0 $\overline{Ia}$ $\overline{C2}$ $\overline{C5}$ $\overline{C8}$ ($\overline{C3}$ + $\overline{C4}$)

(67) + $\overline{F1}$ F3 $\overline{Ts}$ Aw Tp

rIa =
+ T24 $\overline{C9}$ Ø0 Ia Go

(41, 51) + $\overline{C23}$ Ø6 01 $\overline{02}$ $\overline{04}$ $\overline{05}$ 06 $\overline{T24}$

($\overline{Ø0}$) + T0 F1
+ Ø3 T24
+ $\overline{P14}$ F1 Ø6 01 $\overline{02}$ $\overline{04}$ $\overline{05}$ 06 $\overline{T24}$

Index Flip-Flop

sIx =

+ Ø0 T24 Go C1

+ T24 Ø3

rIx =

+ Ø3 $\overline{T24}$ Xn

(66, 67) + Ø0 $\overline{Ia}$ 02 $\overline{Ts}$ $\overline{03}$ 04 05 Q1

+ Tp

Interrupt

sInt = End T10 Ir (En + (En)) $\overline{Eom}$ + End T10 Is

rInt = (St) + T10 $\overline{F1}$ Int

Ir = $\overline{(\overline{Ir})}$

Is = $\overline{(\overline{Is})}$

Indirect Transfer Gate for External Use on Priority Interrupt

Ib = Ø0 Ia $\overline{01}$ $\overline{02}$ $\overline{03}$ 06 (T22 - T17)

+ (St) (T22 - T17)

+ Ij $\overline{Ts}$ F1 (T22 - T17)

Interrupt Subroutine Entry

Ie = Int $\overline{F1}$ (T22 - T17)

Address Lines to Memory

$\overline{L1}$ = $\overline{S1\ \overline{Tsm} + (Iw10)\ Tsm\ Wp + (Iy\ 10)\ Tsm\ \overline{Wp}}$

$\overline{L2}$ = $\overline{S2\ \overline{Tsm} + (Iw11)\ Tsm\ Wp + (Iy\ 11)\ Tsm\ \overline{Wp}}$

⋮

$\overline{L14}$ = $\overline{S14\ \overline{Tsm} + (Iw23)\ Tsm\ Wp + (Iy23)\ Tsm\ \overline{Wp}}$

Also provide $\overline{L3}$ through $\overline{L13}$

Clear M

Mc =

+ Tp P0
+ Tp Ø4
(43) + Tp $\overline{02}$ $\overline{03}$ Ø0 $\overline{Ia}$ 05
+ Tp Tsm Wp $\overline{W9}$ Ts
+ Tp Tsm $\overline{Wp}$ $\overline{Y9}$ Ts
+ Q1 $\overline{Q2}$ Q3 $\overline{Q4}$ Q5 (F1 + $\overline{F3}$ + Ts) (Tsm + $\overline{Ts}$)

Transfer C to M

Mxc = T24 P0

Memory Pulse

sMg = Mc Q1 (Me) $\overline{St}$

rMg = Q1 Q2 Q3 $\overline{Q4}$ Mg

Read Timing

Mrt = Mg $\overline{Q2}$ ($\overline{Q3}$ + Q4)

Write Timing

Mwt = Mg Q2 (Mdt Q1 + $\overline{Q3}$ Q5)

Digit Timing

Mdt = Mg Q2 ($\overline{Q3}$ + Q4)

Rw = Mg $\overline{Q2}$

Wr = Mg Q2 (Mdt + Q1)

Memory Strobe

Ms = Mg (T5 - T0) $\overline{Q4}$ $\overline{Q5}$ Q1

Memory Register Logic (25 flip-flops)

sM0 = Mxc C0
+ Ms Md0
rM0 = Mc

sM1 = Mxc C1
+ Ms Md1
rM1 = Mc

sM24 = Mxc C24
+ Ms Md24
rM24 = Mc

O REGISTER

Clear O

Oc = Tp End
+ Ø7
(23) + $\overline{01}$ $\overline{03}$ 05 Tp $\overline{Ia}$
+ St

Transfer C to O

Oxc = Ø0 Go T24 $\overline{Ia}$ $\overline{C2}$

sO1 = Oxc C3
rO1 = Oc
(41) + 01 $\overline{05}$ $\overline{02}$ $\overline{03}$ Ø0 $\overline{Ia}$ Xw Mc Q1

O REGISTER (continued)

sO2 = Oc
rO2 = Oxc $\overline{C4}$

sO3 = Oxc C5
rO3 = Oc

sO4 = Oxc C6
rO4 = Oc

sO5 = Oxc C7
rO5 = Oc

sO6 = Oxc C8
rO6 = Oc

Overflow Flip-Flop

sOf =
(54, 55) + 01 $\overline{02}$ 03 04 Ø6 T0 $\overline{Xz}$ $\overline{Yz}$ Cz
(54, 55) + 01 $\overline{02}$ 03 04 Ø6 T0 Xz Yz $\overline{Cz}$
(67) + Ø3 06 $\overline{S1}$ T0 An $\overline{Dc}$
(67) + Ø3 06 $\overline{S1}$ T0 $\overline{An}$ Dc
(64) + Rf 01 Ø6 T0 $\overline{Cz}$ C23 $\overline{Ar}$ Su Dc $\overline{P0}$
(51) + Ø6 01 $\overline{02}$ $\overline{04}$ $\overline{05}$ 06 03 C0 T24
(61) + Ø4 01 T0 $\overline{Yz}$ Cz

rOf =
(40) + Ø5 01 $\overline{04}$ T0 C10 $\overline{C11}$ C23
+ (St)
+ Eom C10 $\overline{C11}$ C23
(64) + Rf 01 Ø6 T24
+ P0 Ø0 Of T0

Program Operator

```
sP0         = Ø0 Ia T24 Go C2
            + Mc Tp
(64)        + Rf 01 Ø6 T24 Br Su
(64)        + Rf 01 Ø6 T24 Br Su
(65,67)     + F2 Ts 03 04 06 Bn T24 T0 Tp
(67)        + F2 Ts 03 04 06 05 Ia S1 T24
(65)        + F2 Ts 03 04 06 05 Aw T24

rP0         =
            + (St)
            + P0 T24
            + F2 T0
(65,67)     +  F2 Ts 03 04 06  Bn T24
```

P Register enable = Pg

```
Pg          = (Kr) Q2 Ts Go F1
            + (Kr) Q2 Ts Go P0 Ø0
(01,41,43)  + (Kr) Q2 Ts Go Ø0 Ia 02 03
            + (St)

sP1         = P0 Ø0 Q1 Q2 C8
(01,41,43)  + Ø0 Ia 02 03 Add
(41,51)     + Ø6 01 02 04 05 06 C23 Ia Ts
(41,51)     + Ø6 01 02 04 05 06 C23 Ia
            + Ia P14 F1 Ø6 01 02 04 05 06 Go
            + Ia P14 F1 Ø6 01 02 04 05 06

sP2         = P1
sP3         = P2
 |
 |
 |
 |
sP14        = P13
```

PIN, POT

Pin = Cxi Q1

Pot 1 = $\overline{F1}$ F2 $\overline{F3}$ $\overline{Ts}$ $\overline{02}$ 06

Pot 2 = Ø2 06 $\overline{02}$ Q1

Computer to computer

Ø4 $\overline{01}$ $\overline{04}$ 06 = Rti

Pulse Counter

sQ1 = $\overline{Q3}$ Q5 Q6

rQ1 = $\overline{Q3}$ $\overline{Q4}$ $\overline{Q5}$

sQ2 = $\overline{Q1}$ $\overline{Q5}$ $\overline{Q6}$

rQ2 = Q1 $\overline{Q4}$ $\overline{Q5}$

sQ3 = $\overline{Q3}$ Q4 $\overline{Q5}$ $\overline{Q6}$ + T0

rQ3 = Q3 Q4 $\overline{Q5}$ $\overline{Q6}$

sQ4 = $\overline{Q4}$ Q5 $\overline{Q6}$ (Q1 + Q3)

rQ4 = Q4 Q5 $\overline{Q6}$

sQ5 = $\overline{Q5}$ Q6

rQ5 = Q5 Q6 (Q1 + Q3)

sQ6 = $\overline{Q6}$

rQ6 = Q6

Pulse Timing

(T9) = Q1 $\overline{Q2}$ Q3 $\overline{Q4}$ $\overline{Q5}$

(T0) = $\overline{Q1}$ $\overline{Q2}$ $\overline{Q3}$ Q5

(Tp) = $\overline{Q1}$ $\overline{Q2}$ Q3 Q5

(T24) = $\overline{Q1}$ $\overline{Q2}$ Q3 $\overline{Q5}$

(T23) = $\overline{Q1}$ Q2 $\overline{Q4}$ $\overline{Q5}$

(T22-T17) = $\overline{Q1}$ Q2 $\overline{T23}$

($\overline{Tp}$ $\overline{T24}$) = Q1 + Q2 + $\overline{Q3}$

(T21 + T22) = $\overline{Q1}$ Q2 Q3 Q5

(T10) = Q1 Q2 $\overline{Q4}$ $\overline{Q5}$

(T5 - T0) = $\overline{Q2}$ $\overline{Q3}$

T22 = (T21 - T22) $\overline{Q4}$

T2 = (T5 - T0) Q1 $\overline{Q4}$ $\overline{Q5}$

Ready Flip-Flop

sRf =

(12, 32) + $\overline{01}$ 03 $\overline{Ia}$ $\overline{04}$ $\overline{F1}$ $\overline{F3}$ $\overline{Ts}$ $\overline{Wf}$ (W0 + $\overline{W9}$) 05 $\overline{06}$

(10, 30) + $\overline{01}$ 03 $\overline{Ia}$ $\overline{04}$ $\overline{F1}$ $\overline{F3}$ $\overline{Ts}$ $\overline{Yf}$ (Y0 + $\overline{Y9}$) $\overline{05}$

(13, 33) + $\overline{01}$ 03 $\overline{Ia}$ $\overline{04}$ $\overline{F1}$ $\overline{F3}$ $\overline{Ts}$ Q2 Rt Ø2 06

(64) + $\overline{C2}$ $\overline{C5}$ C6 $\overline{C7}$ $\overline{C8}$ $\overline{C9}$ Ø0 Go T24 $\overline{Ia}$

(66) + Ø3 $\overline{06}$ T24

(64) + $\overline{03}$ 04 $\overline{05}$ $\overline{06}$ $\overline{C9}$ Ø0 T24 Go $\overline{Ts}$

rRf =

+ Tp Ø3

+ Tp End $\overline{Sk}$ $\overline{Ts}$

Clear S

Sc = End T10

+ Ø0 T10

Transfer C to S

$Sxc = T9\ \emptyset 0\ \overline{P0}\ (Ia + 03 + 02 + \overline{\emptyset 0})$

Transfer P to S

$Sxp = T9\ End\ \overline{Int}$
$+ \overline{02}\ \overline{03}\ \overline{Ia}\ \emptyset 0\ T9\ \overline{Int}$

Interrupt N to S

$Sxn = T9\ Int\ \overline{Ts}$

S REGISTER

$sS1 =$
$+ C0\ Sxc$
$+ P1\ Sxp$
$rS1 = Sc$

$sS2 =$
$+ C1\ Sxc$
$+ P2\ Sxp$
$rS2 =$
$+ Sc$

$sS3 =$
$+ C2\ Sxc$
$+ P3\ Sxp$
$rS3 = Sc$

$sS4 = C3\ Sxc$
$+ P4\ Sxp$
$rS4 = Sc$

S REGISTER (continued)

$$
\begin{aligned}
sS5 &= N5\ Sxn \\
&+ C4\ Sxc \\
&+ P5\ Sxp \\
rS5 &= Sc
\end{aligned}
$$

$$
\begin{aligned}
sS6 &= N6\ Sxn \\
&+ C5\ Sxc \\
&+ P6\ Sxp \\
rS6 &= Sc
\end{aligned}
$$

$$
\begin{aligned}
sS7 &= N7\ Sxn \\
&+ C6\ Sxc \\
&+ P7\ Sxp \\
rS7 &= \\
&+ Sc
\end{aligned}
$$

$$
\begin{aligned}
sS8 &= N8\ Sxn \\
&+ C7\ Sxc \\
&+ P8\ Sxp \\
rS8 &= Sc
\end{aligned}
$$

$$
\begin{aligned}
sS9 &= \\
&+ C8\ Sxc \\
&+ P9\ Sxp \\
&+ N9\ Sxn \\
rS9 &= \\
&+ Sc \\
&+ \emptyset 3\ T24\ \overline{S10}\ \overline{S11}\ \overline{S12}\ \overline{S13}\ \overline{S14}
\end{aligned}
$$

S REGISTER (continued)

$$
\begin{aligned}
sS10 &= \\
&+ C9\ Sxc \\
&+ P10\ Sxp \\
&+ N10\ Sxn \\
&+ \emptyset 3\ T24\ \overline{S10}\ \overline{S11}\ \overline{S12}\ \overline{S13}\ \overline{S14} \\
rS10 &= \\
&+ Sc \\
&+ \emptyset 3\ T24\ S10\ \overline{S11}\ \overline{S12}\ \overline{S13}\ \overline{S14}
\end{aligned}
$$

$$
\begin{aligned}
sS11 &= \\
&+ C10\ Sxc \\
&+ P11\ Sxp \\
&+ N11\ Sxn \\
&+ \emptyset 3\ T24\ \overline{S11}\ \overline{S12}\ \overline{S13}\ \overline{S14} \\
rS11 &= \\
&+ Sc \\
&+ \emptyset 3\ T24\ S11\ \overline{S12}\ \overline{S13}\ \overline{S14}
\end{aligned}
$$

$$
\begin{aligned}
sS12 &= \\
&+ C11\ Sxc \\
&+ P12\ Sxp \\
&+ N12\ Sxn \\
&+ \emptyset 3\ T24\ \overline{S12}\ \overline{S13}\ \overline{S14} \\
rS12 &= \\
&+ Sc \\
&+ \emptyset 3\ T24\ S12\ \overline{S13}\ \overline{S14}
\end{aligned}
$$

$$
\begin{aligned}
sS13 &= \\
&+ C12\ Sxc \\
&+ P13\ Sxp \\
&+ N13\ Sxn \\
&+ \emptyset 3\ T24\ \overline{S13}\ \overline{S14} \\
rS13 &= \\
&+ Sc \\
&+ \emptyset 3\ T24\ S13\ \overline{S14}
\end{aligned}
$$

S REGISTER (continued)

$$sS14 = N14\ Sxn + C13\ Sxc + P14\ Sxp + \emptyset 3\ T24\ \overline{S14}$$

$$rS14 = + Sc + \emptyset 3\ T24\ S14$$

SCOPE SIGNALS

	Color
An	Red
Bn	Yellow
C23	White
Xn	Natural
Ecw	Black
Ts	Yellow
Ø0	Orange
Tp End $\overline{Sk}$ Go M0	Red ← Note: This signal provides a scope "sync" on any instruction executed which has a "1" in bit position zero.
T0	Brown
Gnd	Black

Skip Flip-Flop

sSk	=
(70, 72)	+ 01 03 $\overline{04}$ Ø6 $\overline{06}$ T24
(73)	+ 01 03 $\overline{04}$ Ø6 05 06 02 $\overline{Tp}$ $\overline{T24}$ $\overline{T0}$ An $\overline{C23}$
(73)	+ 01 03 $\overline{04}$ Ø6) 05 06 T0 C23 $\overline{An}$
(53)	+ 01 03 $\overline{04}$ Ø6 05 06 T0 C23 $\overline{02}$
(40)	+ Ø5 01 $\overline{04}$ T0 Sks
(66, 67)	+ $\overline{F1}$ F3 $\overline{Ts}$ $\overline{S9}$ $\overline{S10}$ $\overline{S11}$ $\overline{S12}$ $\overline{S13}$ $\overline{S14}$
(67)	+ $\overline{F1}$ F3 $\overline{Ts}$ 06 S2 T0 Aw $\overline{An}$
(67)	+ $\overline{F1}$ F3 $\overline{Ts}$ 06 S2 T0 $\overline{Aw}$ An
(41)	+ 01 $\overline{05}$ $\overline{02}$ $\overline{03}$ Ø0 $\overline{Ia}$ $\overline{Tp}$ $\overline{T24}$ $\overline{T0}$ $\overline{Xn}$

rSk	=
(72)	+ 01 03 $\overline{04}$ Ø6 05 $\overline{06}$ $\overline{Tp}$ $\overline{T24}$ C23 An
(73)	+ 01 03 $\overline{04}$ Ø6 05 06 $\overline{An}$ C23 $\overline{Tp}$ $\overline{T24}$ $\overline{T0}$
(73)	+ 01 03 $\overline{04}$ Ø6 05 06 An $\overline{C23}$ T0
	+ Ø0 T0
	+ Ø7
(70)	+ Ø6 01 03 $\overline{04}$ $\overline{05}$ ($\overline{Tp}$ $\overline{T24}$) Bn $\overline{An}$ C23
(70)	+ Ø6 01 03 $\overline{04}$ $\overline{05}$ ($\overline{Tp}$ $\overline{T24}$) Bn An $\overline{C23}$
	+ (St)

SKS

Skip if:

Sks = C10 $\overline{C11}$ C13 $\overline{Y9}\ \overline{Y10}\ \overline{Y11}\ \overline{Y12}\ \overline{Y13}\ \overline{Y14}$

+ C10 $\overline{C11}$ C14 $\overline{W9}\ \overline{W10}\ \overline{W11}\ \overline{W12}\ \overline{W13}\ \overline{W14}$

+ C10 $\overline{C11}$ C15 $\overline{Kb1}$

+ C10 $\overline{C11}$ C16 $\overline{Kb2}$

+ C10 $\overline{C11}$ C17 $\overline{Kb3}$

+ C10 $\overline{C11}$ C18 $\overline{Kb4}$

+ C10 $\overline{C11}$ C19 $\overline{Ye}$

+ C10 $\overline{C11}$ C20 $\overline{We}$

+ C10 $\overline{C11}$ C21 (En + $\overline{En}$)

+ C10 $\overline{C11}$ C22 $\overline{En}$

+ C10 $\overline{C11}$ C23 $\overline{Of}$

+ $\overline{C10}$ C11 Sio

+ C10 C11 Ssc

Sio = Inverter fed externally

Ssc = Inverter fed externally

Su (Used in Exchange, Mup and Div Step)

sSu =
(64) + Rf 01 Ø0 Br (T22 + T21) $\overline{Q4}$
(65) + F2 $\overline{03}$ 04 06 $\overline{Ts}$ T24 C0 Aw
(65) + F2 $\overline{03}$ 04 06 $\overline{Ts}$ T24 $\overline{C0}$ $\overline{Aw}$
+ Ex T0 $\overline{Su}$ (Kb)
+ Ex T0 $\overline{Su}$ (Kx)
rSu = Tp End Go
+ Ex T0 Su
+ (St)

Buffer Select Flip-Flop

sWp = ($\overline{Go}$ + Tsw) $\overline{Tsy}$ $\overline{Tsm}$ Tp
rWp = (Go $\overline{Tsw}$ + (Tsy)) $\overline{Tsm}$ Tp + (Kf)

Time-Share Flip-Flop

sTs = (Tsw + (Tsy)) Tp
rTs = $\overline{Tsw}$ $\overline{Tsy}$ Tp $\overline{Tsm}$ ($\overline{Kf}$)

Time-Share Memory

sTsm = Ts T10 $\overline{Tsm}$
rTsm = $\overline{Ts}$ + T10 Tsm

$\overline{01}$ 03 $\overline{04}$ $\overline{06}$ F1 $\overline{F3}$ $\overline{Ts}$ 05 to W buffer
$\overline{01}$ 03 $\overline{04}$ $\overline{06}$ F1 F3 $\overline{Ts}$ $\overline{05}$ to Y buffer

X REGISTER

sXw	=
	+ $\overline{Xnr}$ Xn
(71)	+ Xnr 01 03 $\overline{04}$ Ø6 C23
(41)	+ 01 $\overline{05}$ $\overline{02}$ $\overline{03}$ Ø0 $\overline{Ia}$ $\overline{Xn}$ $\overline{Sk}$
(41)	+ 01 $\overline{05}$ $\overline{02}$ $\overline{03}$ Ø0 $\overline{Ia}$ Xn Sk
(67)	+ Xnr Ø3 $\overline{Ix}$ Xn
(67)	+ Xnr Ø3 Ix $\overline{Xn}$
	+ Xnr Ex C23
(77)	+ Xnr 01 02 03 04 Add
	+ (Kf) $\overline{T21}$ $\overline{T22}$

Xnr	=
(71)	+ 01 03 $\overline{04}$ Ø6 02 $\overline{05}$ 06
(41)	+ Ø0 $\overline{Ia}$ $\overline{02}$ $\overline{03}$ 01 $\overline{05}$
(67)	+ Ø3 06 S2
	+ Ex Dc Su
	+ Ex (Kx)
(77)	+ 01 02 03 04 05 06 $\overline{Ia}$ Ø0 Q2
	+ (Kf) (T21 + T22)

Adder

Xz	= Xn Ix $\overline{F1}$ $\overline{F3}$ $\overline{Ts}$
(54-57)	+ An F1 03
(64)	+ Ar Rf 01 Ø6
(65)	+ Dc F2 $\overline{Ts}$ $\overline{03}$ 04 06

Yz	=
	+ C23 $\overline{F1}$ $\overline{F3}$ $\overline{Ts}$
(+, -,)	+ F1 C23 06 $\overline{Rf}$
	+ F1 $\overline{C23}$ $\overline{06}$ $\overline{Rf}$
(64)	+ Ø6 Rf 01 $\overline{Dc}$ C22
(64)	+ Ø6 Rf 01 Dc C23 $\overline{P0}$

Add = $\overline{Xz\ Yz\ \overline{Cz} + Xz\ \overline{Yz}\ Cz + \overline{Xz}\ Yz\ Cz + \overline{Xz}\ \overline{Yz}\ \overline{Cz}}$

Priority Interrupt

$$sIs1 = (\underline{T22 - T17})\ \text{(I1)}$$

$$rIs1 = Ip1\ \underline{Ib} + \overline{St}\ [dc]$$

$$sIp1 = Is1\ \underline{Ie}$$

$$rIp1 = Ip1\ \underline{Ib} + \overline{St}\ [dc]$$

$$sIs2 = (\underline{T22 - T17})\ \text{(I2)}$$

$$rIs2 = \overline{Ip1}\ Ip2\ \underline{Ib} + \overline{St}\ [dc]$$

$$sIp2 = \overline{Is1}\ Is2\ \underline{Ie}$$

$$rIp2 = \overline{Ip1}\ Ip2\ \underline{Ib} + \overline{St}\ [dc]$$

$$\text{(Ir)} = Is1\ \overline{Ip1} + \overline{Is1}\ Is2\ \overline{Ip2} + \overline{Is1}\ \overline{Is2}\ Is3\ \overline{Ip3} + - - -$$

SECTION II

W BUFFER AND INPUT/OUTPUT OPERATIONS

INTRODUCTION

This section describes the theory of operation of the W Buffer and Input/Output Equipment. It is intended for engineering, maintenance and training use. For proper use of this document, the reader should be familiar with the SDS Reference Manual for this computer.

The operation of the optional Y Buffer is almost identical to that of the W Buffer.

GENERAL OPERATION

The Input/Output or W Buffer contains a full-word register, a six-bit character register, six-bit unit address register and the associated control circuitry needed to transfer full words to and from memory. The W Buffer communicates directly with external devices using a 6-bit character. Parity detection and generation is automatically performed in the buffer, but is not part of the computer word.

Input and output operations using the buffer are basically performed in the following manner. An EOM instruction sets the input or output address to designate the particular I/O device, input or output status, and the character count (number of characters per word). WIM and MIW instructions are then used to transfer information between memory (via the C register) and the Word Assembly Register (WAR). When the WAR has been filled on input (or emptied on output), an interrupt can occur to cause the computer to unload (WIM) or fill (MIW) the WAR unless the interrupt has been disabled; in case the interrupt has been disabled, the WIM or MIW instruction will be executed prior to need and held until the buffer is ready for execution.

When an EOM0XXXX instruction to start an input or output process is executed, the buffer unit address register and character counter is set up from the C register. The registers are first cleared by Wc,

$$Wc = Buc\ \overline{C17}\ (T22 - T17) + - - -$$
$$Buc = \emptyset 5\ \overline{01}\ 05\ \overline{C10}\ \overline{C11}$$

and then set from the C register by Ws,

$$Ws = Buc\ \overline{C17}\ (T5 - T0)$$

where $\overline{C17}$ designates the W Buffer.

sW14 = Ws C23
rW14 = Wc
⋮
sW10 = Ws C19
rW10 = Wc

} Unit address code

sW9 = Ws C18
rW9 = Wc

} W9 = 1 for Output
W9 = 0 for Input

sW8 = Ws C16 + - - -
rW8 = Wc (T22-T17) + - - -
sW7 = Ws C15 + - - -
rW7 = Wc (T22-T17) + - - -

} Character count

W7	W8	Character/word
1	1	4
1	0	3
0	1	2
0	0	1

Refer to Drawing 19.

At the end of the start EOM0XXXX instruction, W4 is set on for two pulse times to gate the storing of the character count into the WAR.

$$sW4 = Ws\ T0 \qquad + - - -$$

$$rW4 = W4\ T24 + - - -$$

$$sWw = W4\ Tp\ W8 + W4\ T24\ W7$$
$$+ - - - + (Tp + T24)\ \overline{W4}\ Wn + \overline{W4}\ \overline{Wx}\ Wn$$

This allows W8 and W7 to act as a character counter during input or output and be reloaded from the WAR each time that a completed input word is stored from the WAR, or a new output word is loaded into the WAR by a Wx signal.

$$sW8 = - - - + Wx\ T24\ Ww + - - -$$

$$sW7 = - - - + Wx\ T24\ \overline{W4}\ Wn + - - -$$

Both for shifting an input character from the single character register into the WAR, and for shifting an output character from the WAR into the SCR, W4 is set for (T23-T0) of one machine cycle to gate a one-character precession in the WAR, by causing the data in the WAR to recirculate through the SCR.

$$sR1 = W4\ Wn\ \overline{Wx}\ \overline{Tp}\ \overline{T24} + - - -$$

$$rR1 = W4\ \overline{Wn}\ \overline{Wx} + - - -$$

$$sR2 = W4\ R1 + - - -$$

$$rR2 = W4\ \overline{R1} + - - -$$

$$\vdots$$

$$sR6 = W4\ R5 + - - -$$

$$rR6 = W4\ \overline{R5} + - - -$$

$$sWw = W4\ R6 + - - -+ \overline{W4}\ Wn\ \overline{Wx}$$

Wf is set when a WIM or MIW instruction is executed. Wf then allows precessions of the WAR until the character counter W7 W8 indicates that the last character is being precessed and Wf is reset.

The signaling to the computer for a WIM or MIW instruction is conditioned by $\overline{Wf}$ (- - - -), indicating that the WAR is full or empty.

$$sWf = Wx\ (T5 - T0)\ \overline{W4}$$

$$rWf = \overline{W8}\ \overline{W7}\ W4\ (T22 - T17)$$

The character count at T24 and Tp in Wn is not altered by this precession. At the end of each precession of the WAR, the character count in W8 and W7 is decremented until W8 W7 = 00.

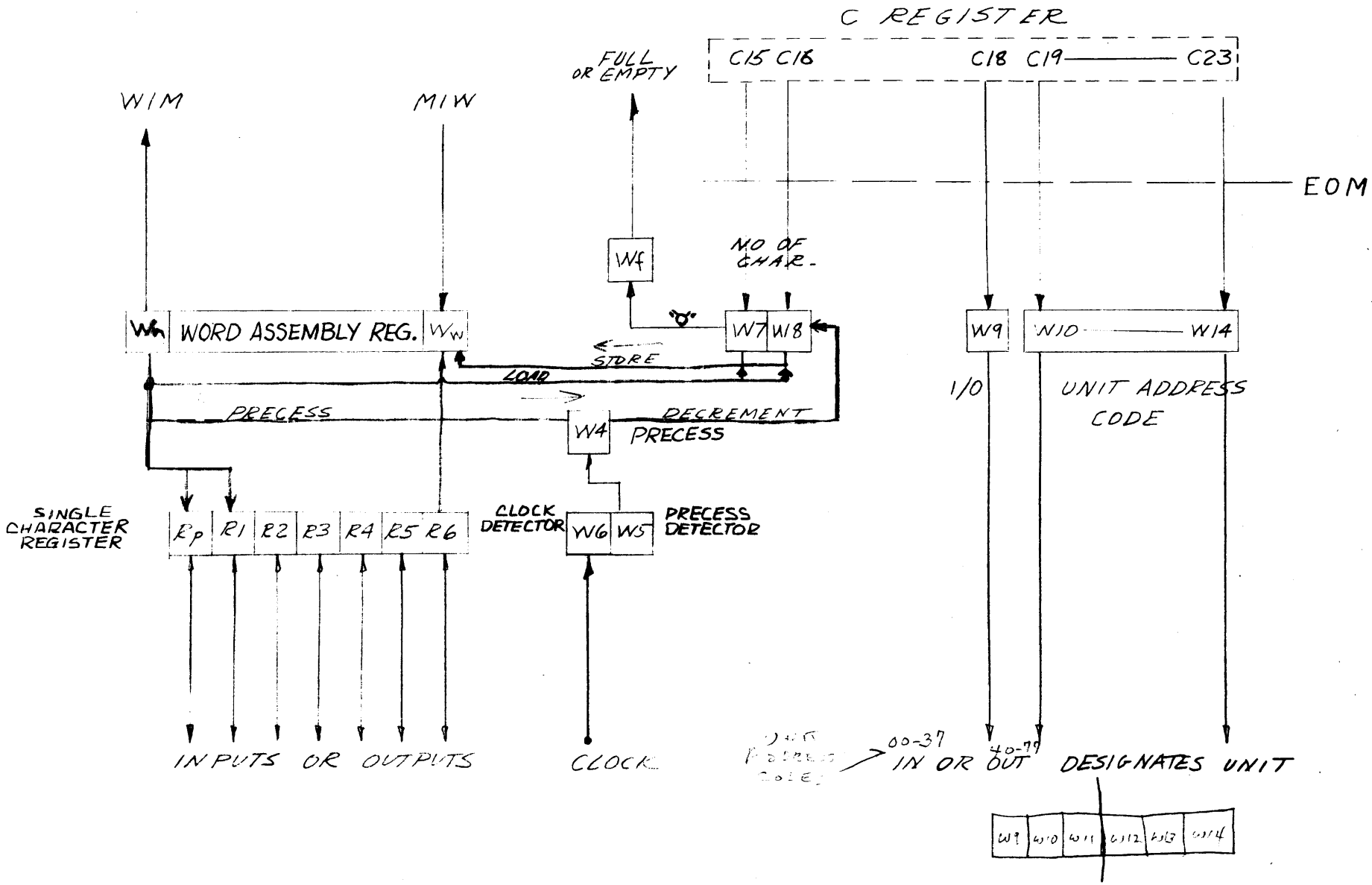


FIGURE 19

W BUFFER

INFORMATION FLOW

$$sW8 = - - - + W7\ \overline{W8}\ W4\ T0 + - - -$$

$$rW8 = - - - - - - + W8\ W4\ T0$$

$$rW7 = - - - + W7\ \overline{W8}\ W4\ T0$$

The setting of W4 is interlocked with W6 and W5 to cause a precession of the WAR after each input or output clock signal, Ecw. The setting of W4 is also interlocked with Wf to prevent a precession when the WAR is full on input or empty on output. The clock Ecw is detected by W6 as follows.

$$sW6 = \overline{W5}\ Ecw\ (T22 - T17)$$

$$rW6 = W5\ T0 + Wc$$

The precess condition is then detected by W5.

$$sW5 = \overline{W5}\ W6\ \overline{Ecw}\ T0 + - - -$$

$$rW5 = W4\ T0 + Wc$$

The actual precessing is then controlled by W4.

$$sW4 = - - - + W5\ Wf\ T24 + - - -$$

$$rW4 = - - - + W4\ T0$$

Wh and W0 provide an interlock on interrupt signaling in connection with the stopping of an input or output process, as will be indicated later.

The foregoing discussion applies to both input and output processes. Particular features of the input and output processes are discussed separately in the following material.

INPUT PROCESS ($\overline{W9}$)

After an EOM0XXXX instruction starts an input process, Wf is set to prepare the W Buffer to precess the first input character into the WAR.

$$sWf = - - - + Wc\ \overline{Wh}$$

The S.C.R. is cleared between input clock signals by $\overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4}$ and the input character signals Zw1, Zw2 . . . . Zw6 and Zwp, are gated into the character buffer during input clock signals by $\overline{W9}\ W6\ \overline{W5}$.

$$sR1 = - - - + \overline{W9}\ W6\ \overline{W5}\ Zw1$$

$$rR1 = - - - + \overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4} + - - -$$

$$\vdots$$

$$sR6 = - - - + \overline{W9}\ W6\ \overline{W5}\ Zw6$$

$$rR6 = - - - + \overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4} + - - -$$

$$sRp = \overline{W9}\ W6\ \overline{W5}\ Zwp + - - -$$

$$rRp = \overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4} + - - -$$

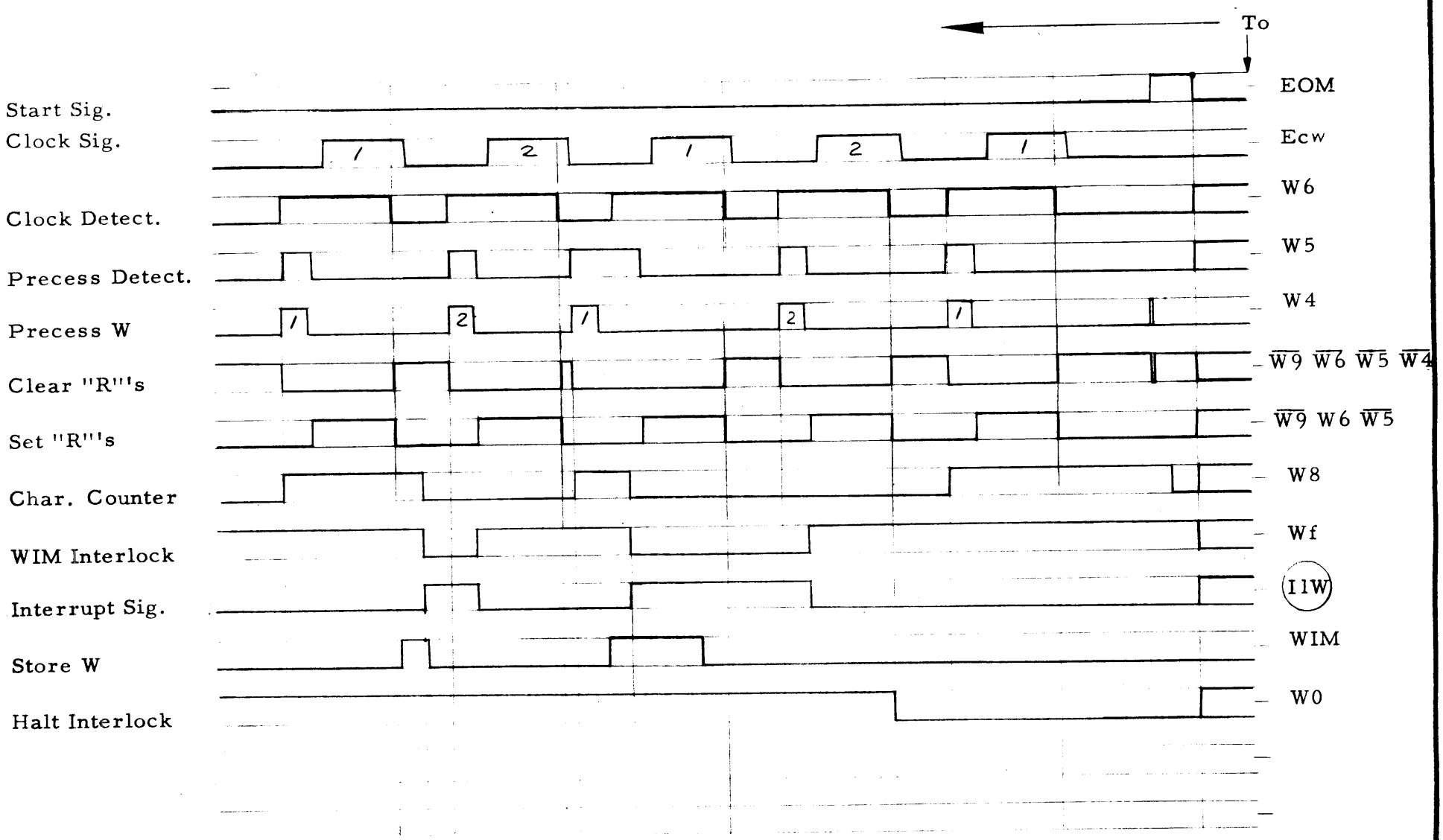


FIGURE 20

INPUT TIMING CHART

(2 Char. /Word)

The Input Timing Chart indicates the basic flow of the input process. The execution of the first WIM instruction is shown occurring late to illustrate how the S.C.R. accepts and holds one more character after the WAR is full. When a WIM instruction is executed late, the Error Detector flip-flop is set and, if desired, this condition can be tested by an SKS instruction. The Error Detector flip-flop (We) is reset by each EOM0XXXX instruction.

$$sWe = W0\ \overline{W6}\ W5\ Ecw\ Tp + - - -$$

$$rWe = Wc\ \overline{Wh}$$

The Error Detector flip-flop is also set if an input character has an even parity.

The Rp flip-flop is used to check the character parity as each character is shifted into the WAR.

$$sRp = - - - + \overline{W9}\ W4\ \overline{Rp}\ Ww\ (T22 - T17) + - - -$$

$$rRp = - - - + \overline{W9}\ W4\ Rp\ Ww\ (T22 - T17) + - - -$$

$$sWe = - - - + \overline{W9}\ W4\ \overline{Rp}\ (T5 - T0) + - - -$$

Each time a WIM instruction is executed, the C and Word Assy registers are interchanged.

$$sC0 = \emptyset 4\ \overline{01}\ \overline{04}\ 05\ \overline{06}\ Wn$$

$$Wx = (\overline{01}\ 03\ \overline{04}\ 05\ \overline{06}\ F1\ \overline{F3})\ \overline{Ts} + - - -$$

$$sWw = W4\ R6 + (\overline{T24}\ \overline{Tp})\ \overline{W4}\ Wx\ C23 + - - - + \overline{W4}\ Wn\ \overline{Wx}$$

$$sR1 = W4\ Wn\ \overline{Wx} + - - -$$

$$rR1 = W4\ \overline{Wn}\ \overline{Wx} + - - -$$

$$sR6 = W4\ R5 + - - -$$

$$rR6 = W4\ \overline{R5} + - - -$$

Synchronizing W6 and W5 with T0 and T22-T17 allows the input clock to be ambiguous for one machine cycle both while going true and while going false. The input clock must be unambiguously true for almost one machine cycle and unambiguously false for at least one machine cycle (f max = 62.5 Kc). Refer to Figure 21. The input character signals must rise while the clock signal is up and fall before the rise of the next clock signal.

A photo-reader input process is terminated by detecting tape gap following the block of input data. Since the photo-reader must be able to read tape leader until the start of a block, the W0 flip-flop inhibits the sprocket clock signals until it is set by the first or second character.

$$sW0 = \overline{W9}\ W6\ \overline{W8} + - - -$$

$$rW0 = Wc + - - -$$

The input clock signal is derived from the tape characters, and after W0 is set, by the sprocket signal. SP

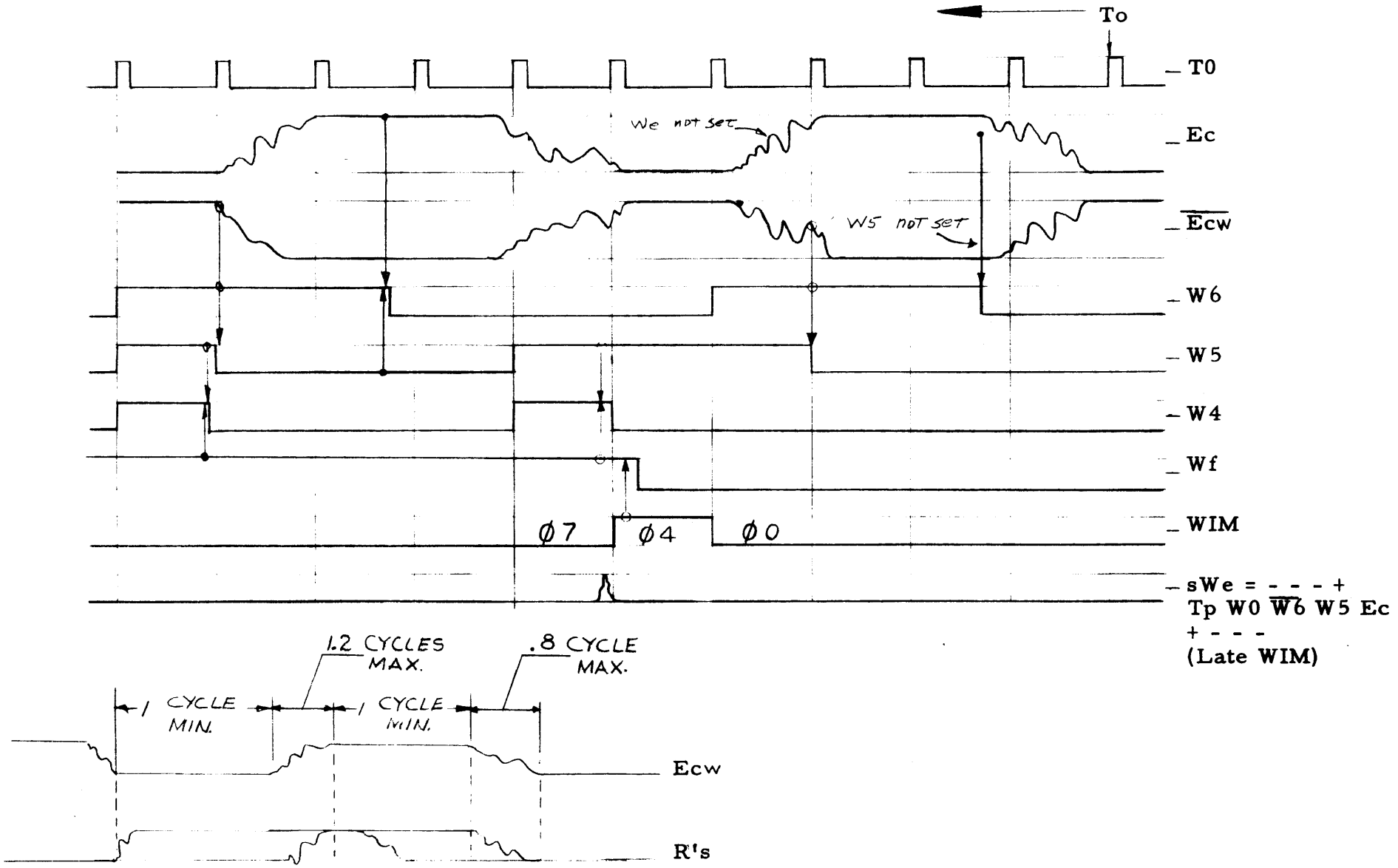


FIGURE 21

INPUT SIGNAL CHARACTERISTICS

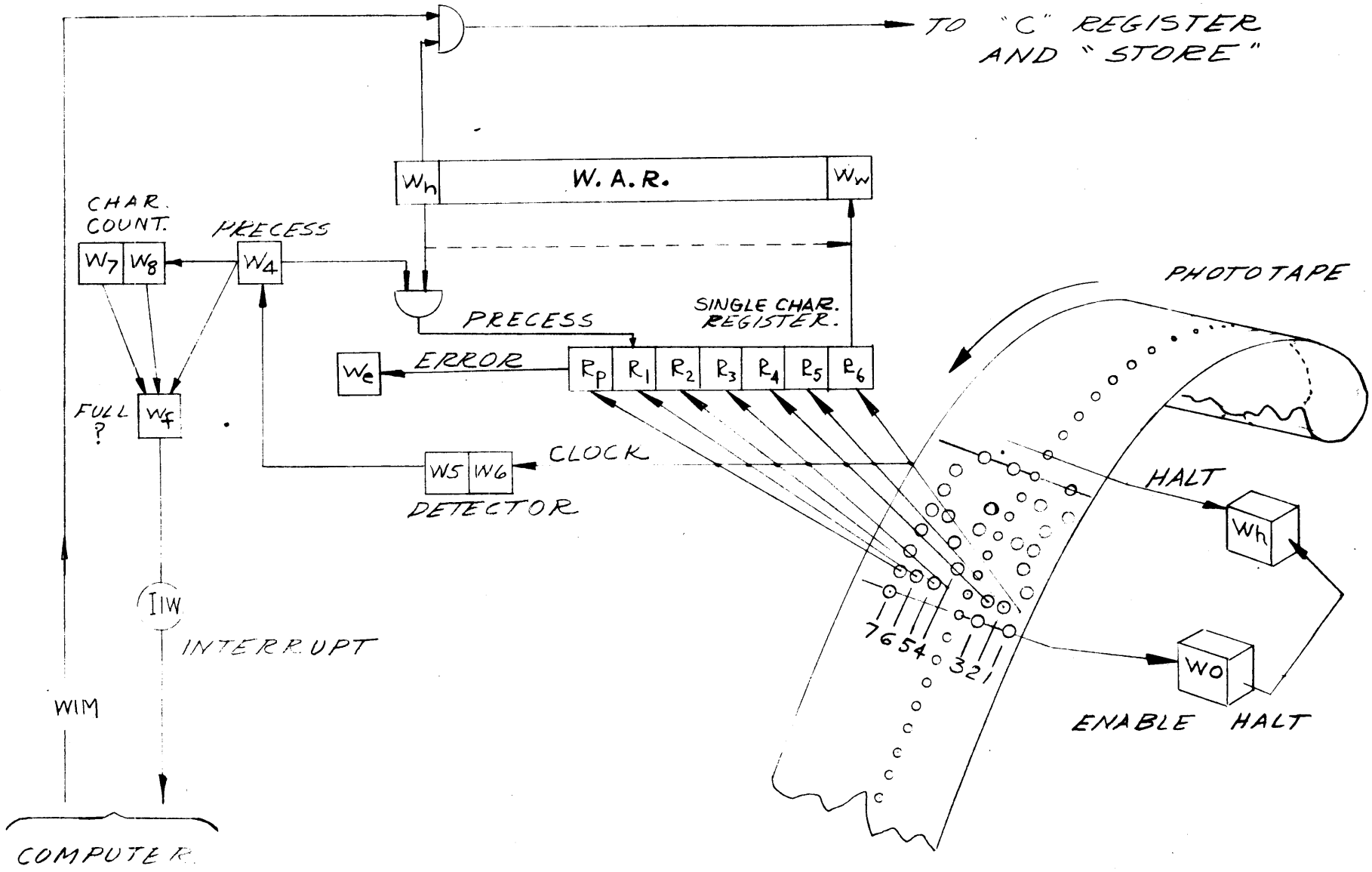


FIGURE 22

INFORMATION FLOW DIAGRAM

PHOTOTAPE

$$Ecw = (Zw1 + Zw2 + Zw3 + Zw4 + Zw5 + Zw6 + Zwp + W0)\ Sp\ Re$$

$$Zw1 = (Ch\ 6\ amplifier)\ Re$$
$$Zw2 = (Ch\ 5\ amplifier)\ Re$$
$$\vdots$$
$$Zwp = (Ch\ 7\ amplifier)\ Re$$

The first, all-zeros character gated into the S.C.R. is detected by the halt detector, Wh, at the same time that W4 is set.

$$sWh = \overline{W9}\ \overline{W10}\ \overline{W11}\ W12\ \overline{W13}\ (\overline{R1}\ \overline{R2}\ \overline{R3}\ \overline{R4}\ \overline{R5}\ \overline{R6}\ \overline{Rp})\ W5\ T24$$
$$+ - - -$$
$$rWh = Wc$$
$$sW4 = - - - + W5\ Wf\ T24$$
$$rW4 = - - - + W4\ T0$$

With Wh set, parity errors are no longer used to set We.

$$sWe = - - - + \overline{W9}\ W4\ \overline{Rp}\ (T5\text{-}T0)\ \overline{Wh} + - - -$$

W4 is set for successive cycles until the character counter reads 00 when Wf is reset, and the W Buffer is reset.

$$sW4 = - - - + Wh\ T24$$
$$rW4 = - - - + W4\ T0$$

$$sWf = - - - + Wc\ \overline{Wh}$$
$$rWf = \overline{W8}\ \overline{W7}\ W4\ (T22\text{-}T17) + - - -$$

$$Wc = + - - - Wh\ \overline{Wf}\ T0 + - - -$$

The tape reader brake is signaled.

$$reader\ \ 1\ brake \leftarrow \overline{\overline{W9}\ \overline{W10}\ \overline{W11}\ W12\ \overline{W13}\ \overline{W14}\ \overline{Kf}}$$

This termination process fills in the final input word with all-zero characters. The Photo-tape Termination Timing Charts show the flow of this termination process. A halt interrupt signal is generated during the $\overline{Wf}$ Wh cycle to call for a final WIM instruction and to signal the termination of the input process.

$$(12w) = \overline{Wf}\ Wh\ (- - -)$$

The final WIM instruction in the Halt subroutine will store the data even though the last input may not have a complete complement of characters. The Magnetic Tape Input Timing Chart shows the flow of this input process. As with photo-reader input a final WIM instruction is executed after the halt interrupt to store any remaining characters in the WAR. The magnetic tape unit generates a halt signal with a time delay triggered by the tape gap signal and W0. The halt detector is then triggered by this halt signal.

$$sWh = - - - + Whs\ T24 \qquad + - - -$$

The magnetic tape-handler is also stopped directly from the Whs signal.

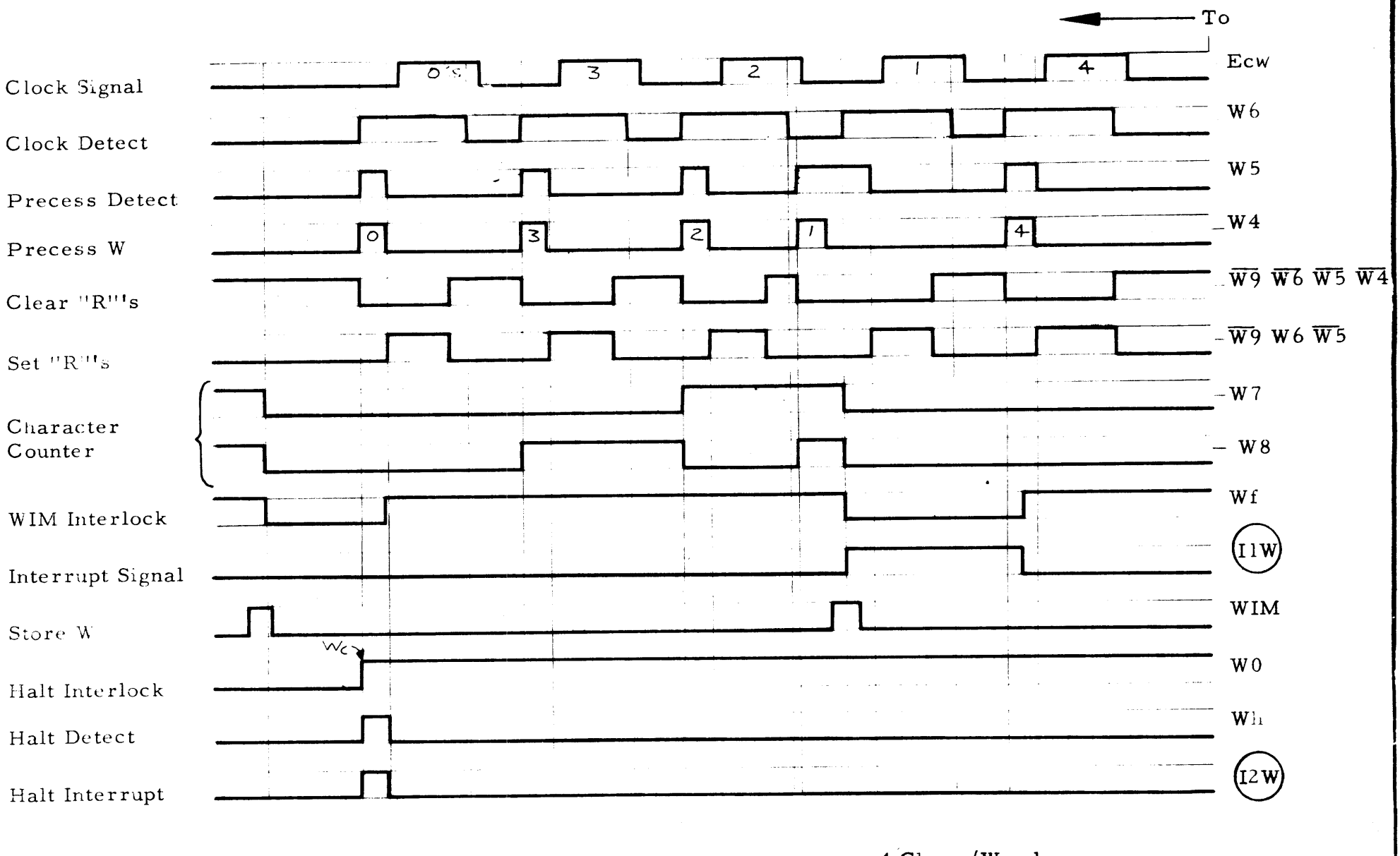


FIGURE 23

TERMINATION TIMING A

PHOTOTAPE INPUT

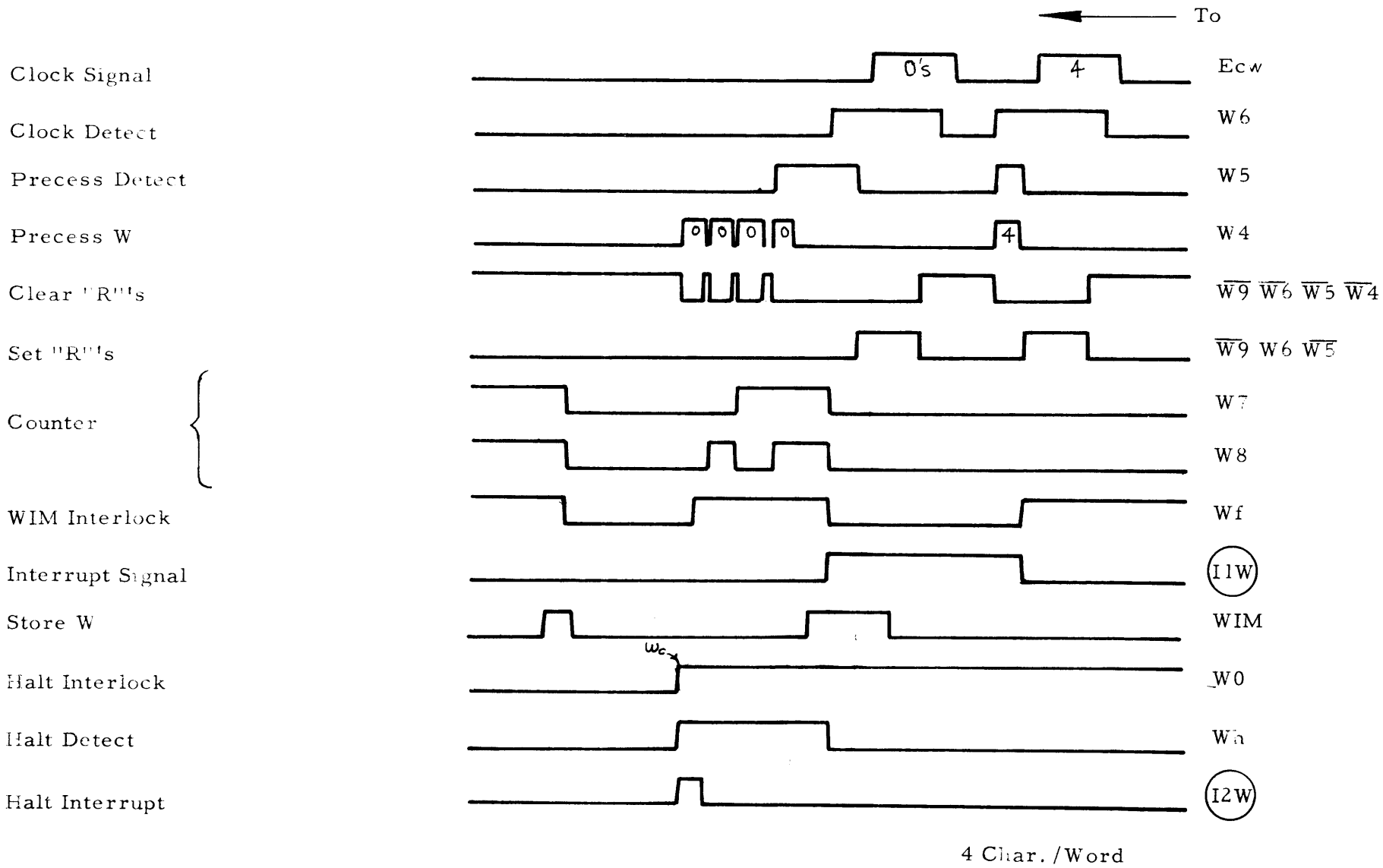


FIGURE 24

TERMINATION TIMING B

PHOTOTAPE INPUT

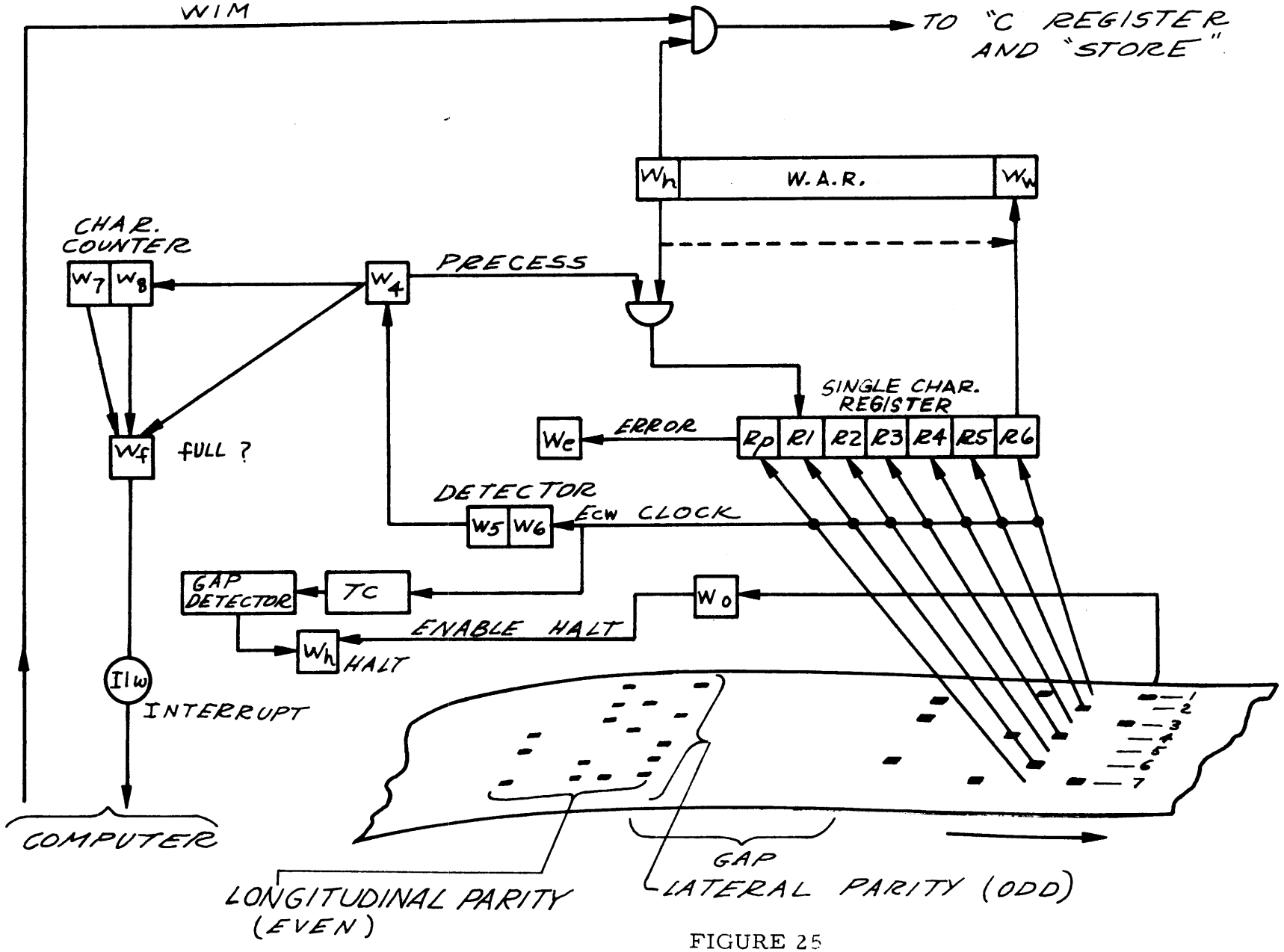


FIGURE 25
INFORMATION FLOW DIAGRAM
MAGNETIC TAPE

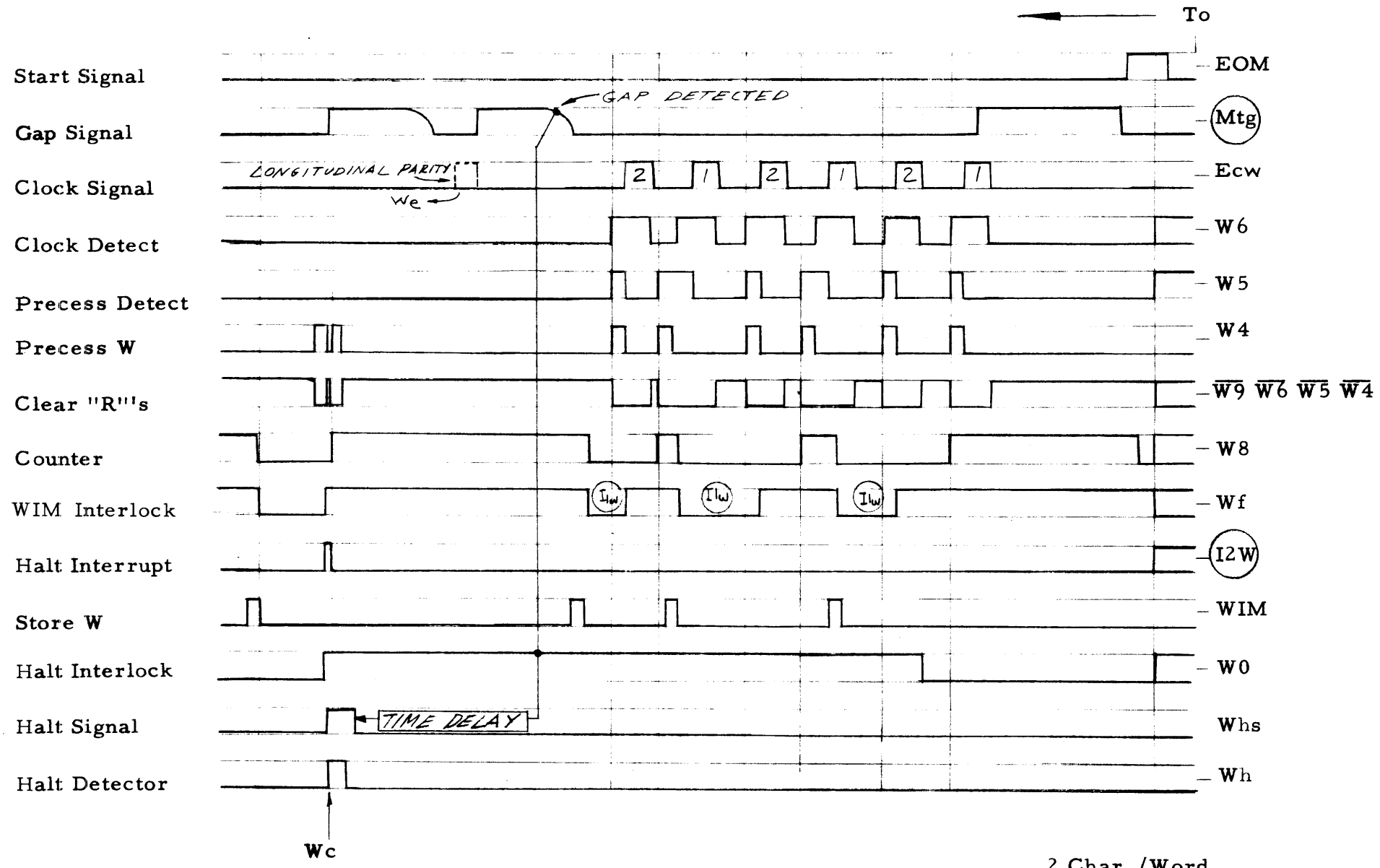


FIGURE 26

INPUT TIMING

MAGNETIC TAPE

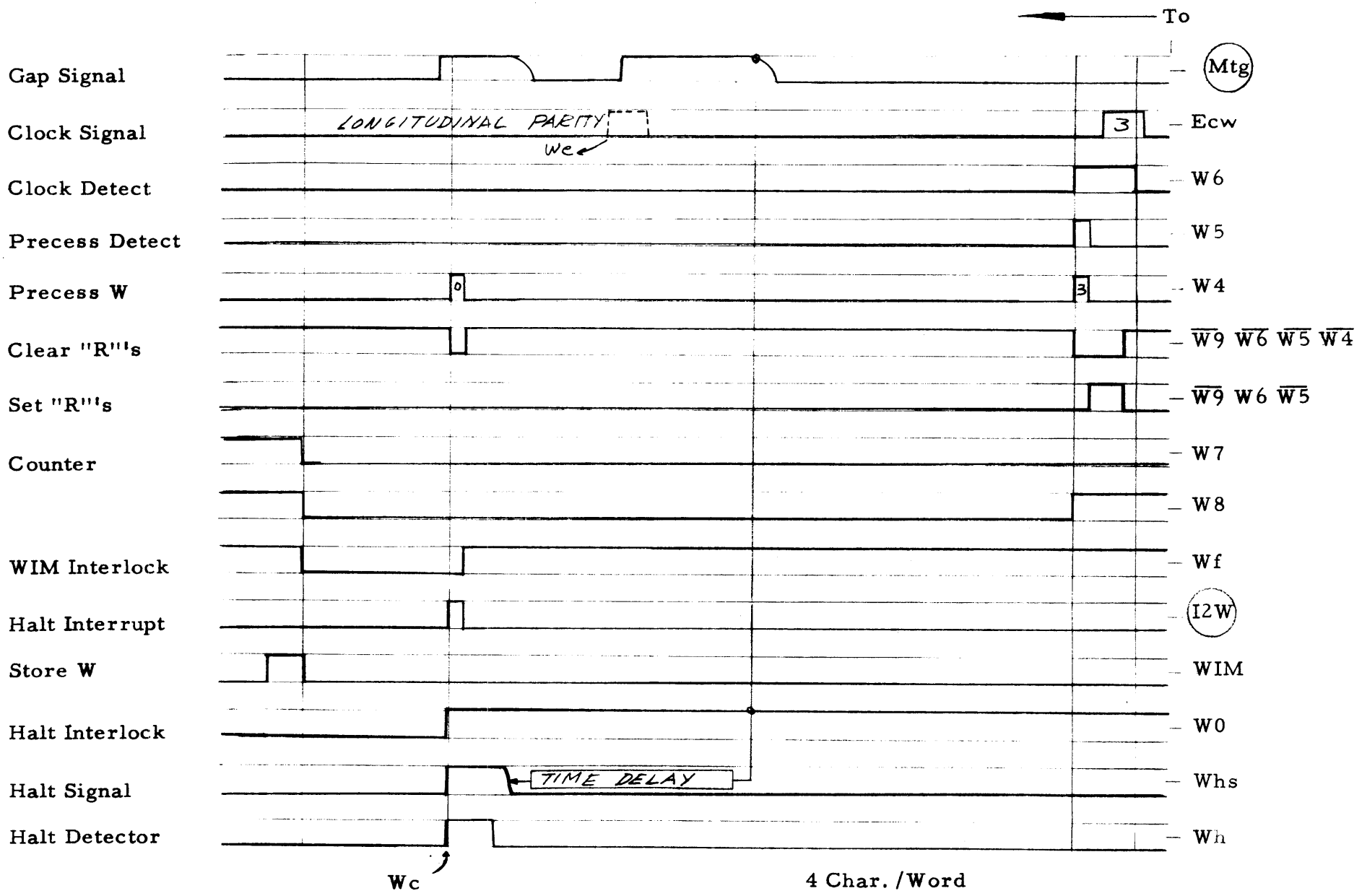


FIGURE 27
INPUT TERMINATION TIMING
MAGNETIC TAPE

The tape unit may also check the longitudinal parity and generate an error signal, $\overline{Wes}$, if there is an error. This signal is used to set We.

$$sWe = - - - + Wes$$

The tape unit blocks the longitudinal parity character from generating an input clock signal.

OUTPUT PROCESS (W9)

When an EOM0XXXX instruction is executed to start an output process, an interrupt can be immediately signaled to call on the computer to load the WAR with the first output word.

$$rWf = - - - + Ws\ W9 + - - - \qquad sW0 = - - - + Ws\ W9$$

$$I1w = \overline{Wf}\ W0\ \overline{Wh}\ (- - - - -) + - - -$$

Each MIW instruction or time-share operation loads the WAR from C register.

$$sWw = W4\ R6 + (\overline{T24}\ \overline{Tp})\ \overline{W4}\ Wx\ C23 + - - - \overline{W4}\ Wn\ \overline{Wx}$$

$$sR1 = W4\ Wn\ \overline{Wx}\ (\overline{Tp}\ \overline{T24}) + - - - + - - -$$

$$rR1 = W4\ \overline{Wn}\ \overline{Wx} + - - - + - - -$$

$$sR6 = W4\ R5 + - - -$$

$$rR6 = W4\ \overline{R5} + - - -$$

$$Wx = (\overline{01}\ 03\ \overline{04}\ 05\ \overline{06}\ F1\ \overline{F3}) + - - -$$

Each time W4 is set to precess an output character from the WAR into the SCR, Rp is used to generate an odd parity output bit.

$$sRp = - - - + W9\ W4\ \overline{Rp}\ Wn\ (T5 - T0)\ \overline{Wx} + W9\ W4\ (T22 - T17)$$
$$+ W9\ W4\ \overline{Rp}\ C23\ (T5 - T0)\ Wx$$

$$rRp = - - - + W9\ W4\ Rp\ Wn\ (T5 - T0)\ \overline{Wx} + W9\ W4\ Rp\ C23\ (T5 - T0)\ Wx$$
$$+ - - -$$

Figure 29, Output Timing Chart 1, indicates the basic flow of the output process. The execution of the second MIW instruction is shown occurring late, to illustrate how the SCR is cleared when a new output character is not available. If a new output character is not available in time, the error detector is set as with input.

$$sWe = W0\ \overline{W6}\ W5\ Ec\ Tp + - - -$$

$$rWe = Wc\ \overline{Wh}$$

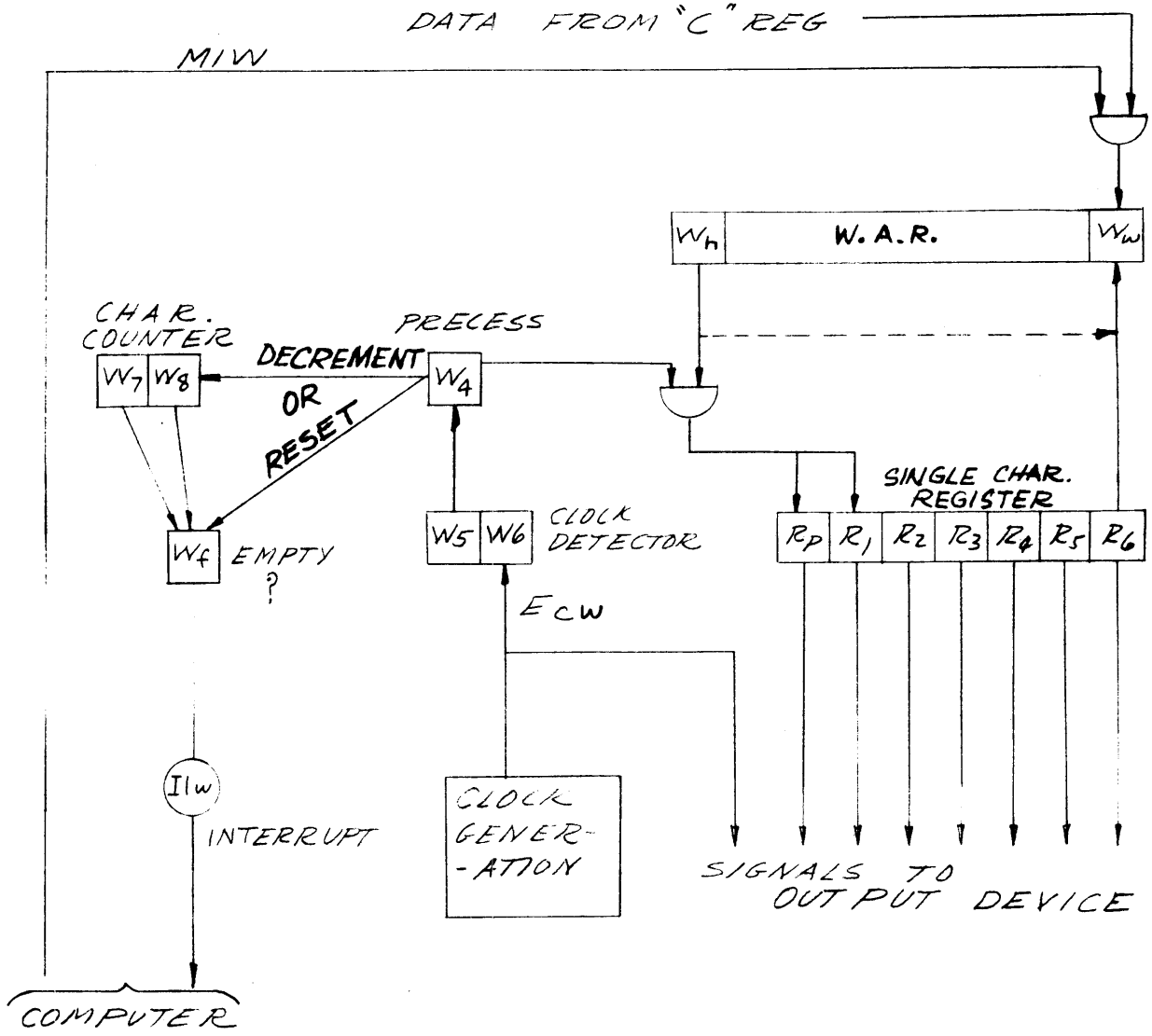


FIGURE 28

FLOW DIAGRAM

OUTPUT PROCESS

As with input, the SCR is cleared at the start of output by $\overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4}$. The first output character is, therefore, all zeros for the first clock signal, which is appropriate for a leader, or a gap in tape punching. For some forms of output, such as leaderless punching or typing, the first output character should be in the character buffer before the first clock signal. For this type of output, an EOM02XXX start instruction with a one-bit in C13 is used. This code bit is used to set W5 which then causes the first loading of the WAR to be followed directly by precession of the first output character into the SCR.

$$sW5 = - - - + Ws\ C13\ C18$$

$$rW5 = - - - + W4\ T0 + Wc$$

$$sWf = Wx\ (T5\text{-}T0) + - - -$$

$$sW4 = W5\ Wf\ T24 + - - -$$

$$rW4 = W4\ T0 + - - -$$

Figure 30, Output Timing Chart 2, illustrates the flow of this process.

On output, Ecw is generated by various oscillator-type circuitry instead of from the characters. This circuitry is described in a separate publication.

To terminate an output process, the MIW instruction to load the last output word into the WAR is followed by an EOM 14000 instruction to reset W0 (EOM 14100 instruction for the Y Buffer).

$$rW0 = - - - + Ioc\ C12\ \overline{C17}\ \overline{C20} + - - -$$

Resetting W0 results in the following: Further normal interrupt signals are blocked;

$$(Ilw) = \overline{Wf}\ W0\ \overline{Wh}\ (- - - -)$$

further late load W error signals are blocked;

$$sWe = W0\ \overline{W6}\ W5\ Ecw\ Tp + - - -$$

and Wf is not set again after it is reset as the last output character is precessed into the SCR.

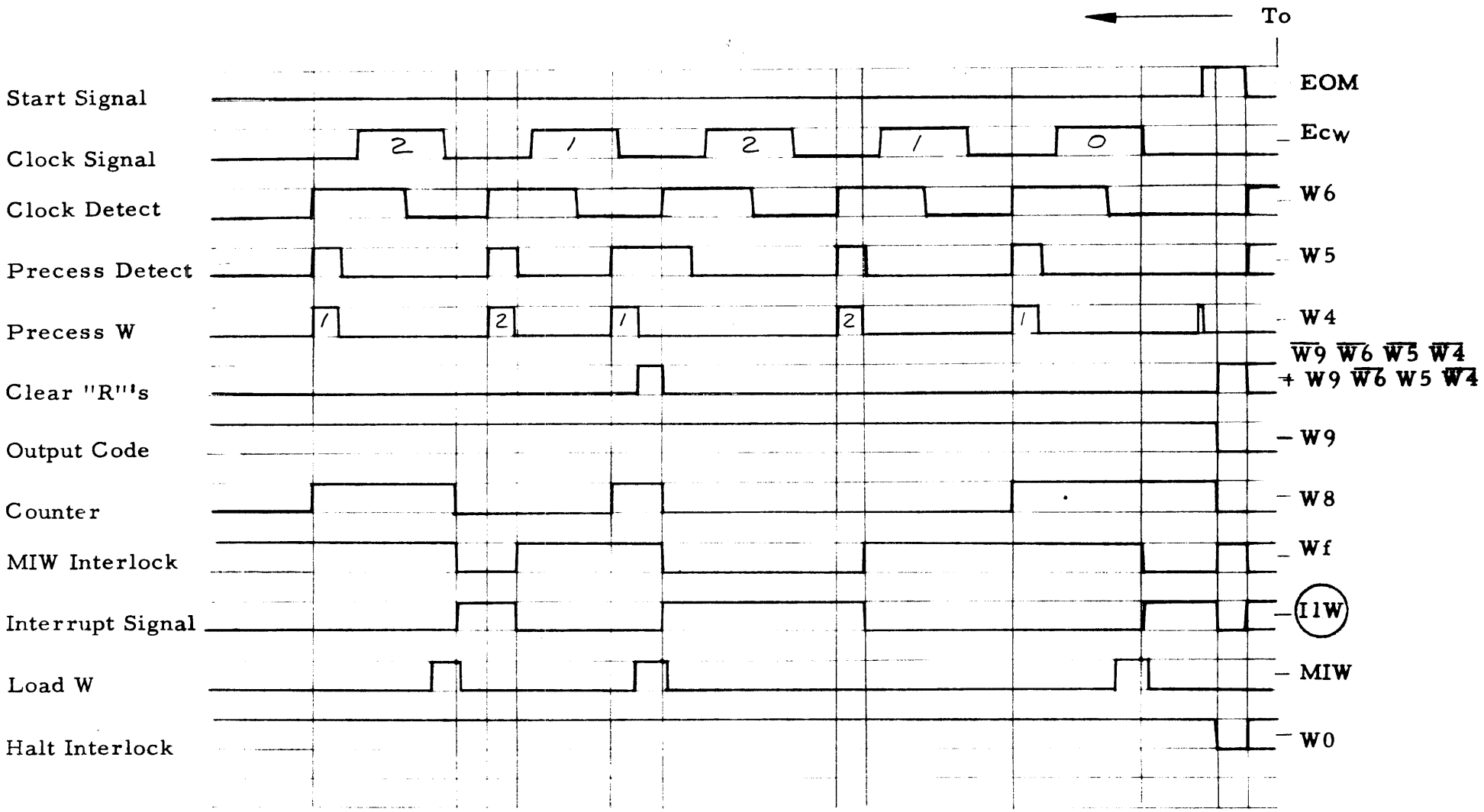
$$sWf = Wx\ (T5\text{-}T0) + - - -$$

$$rWf = \overline{W8}\ \overline{W7}\ W4\ (T22\text{-}T17) + - - -$$

Preventing the setting of Wf blocks W4 from being set after the last output character is precessed into the SCR.

$$sW4 = W5\ Wf\ T24 + - - -$$

Preventing W4 from being set produces the state, $\overline{W0}\ \overline{W4}\ W5\ \overline{W6}$, after the last output character has been processed by the output clock.

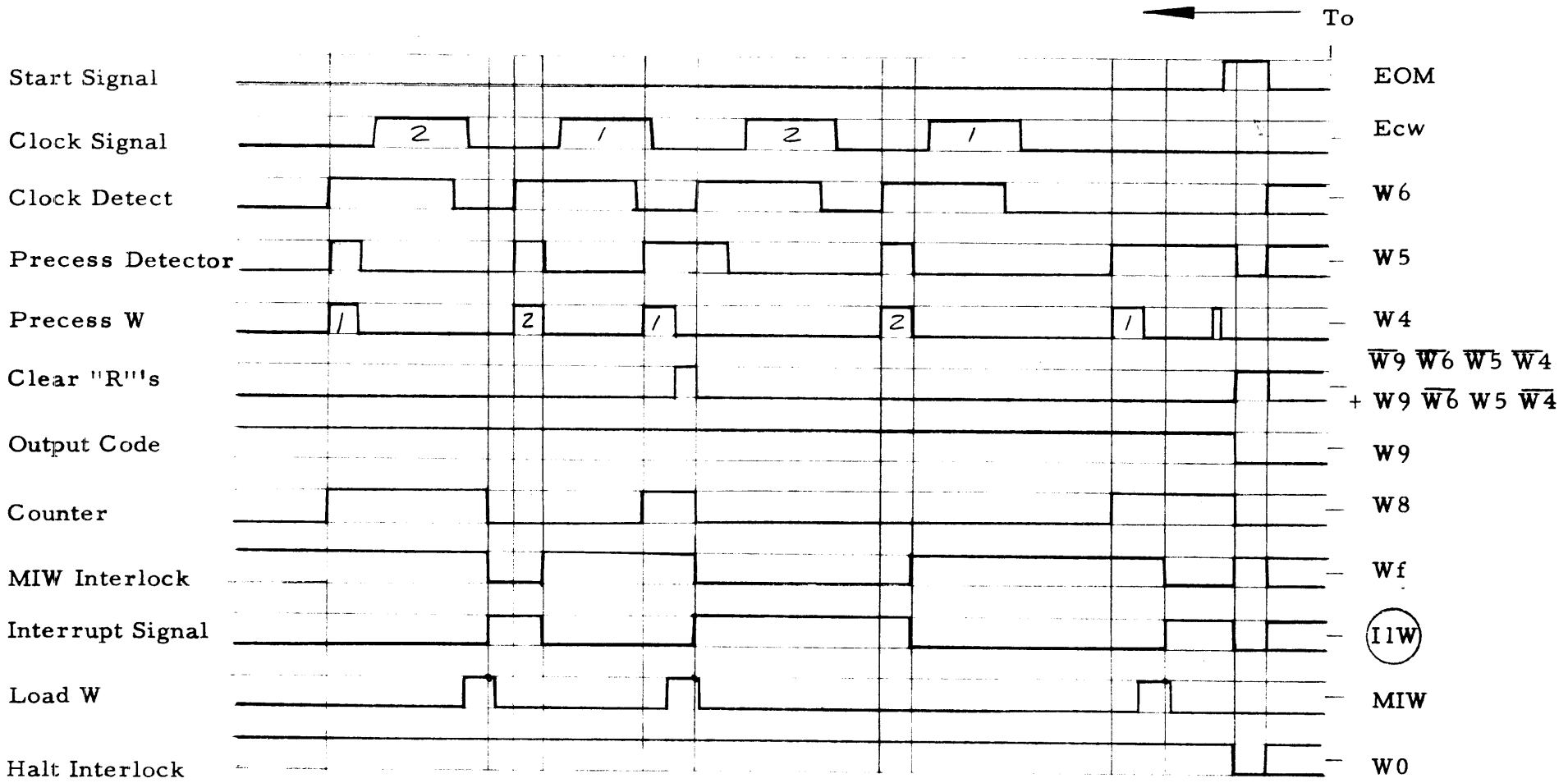


2 Char. /Word starting with Gap or Leader

TIME

FIGURE 29

OUTPUT TIMING CHART 1



2 Char./Word starting with Leaderless punching or Typing

TIME

FIGURE 30

OUTPUT TIMING CHART 2

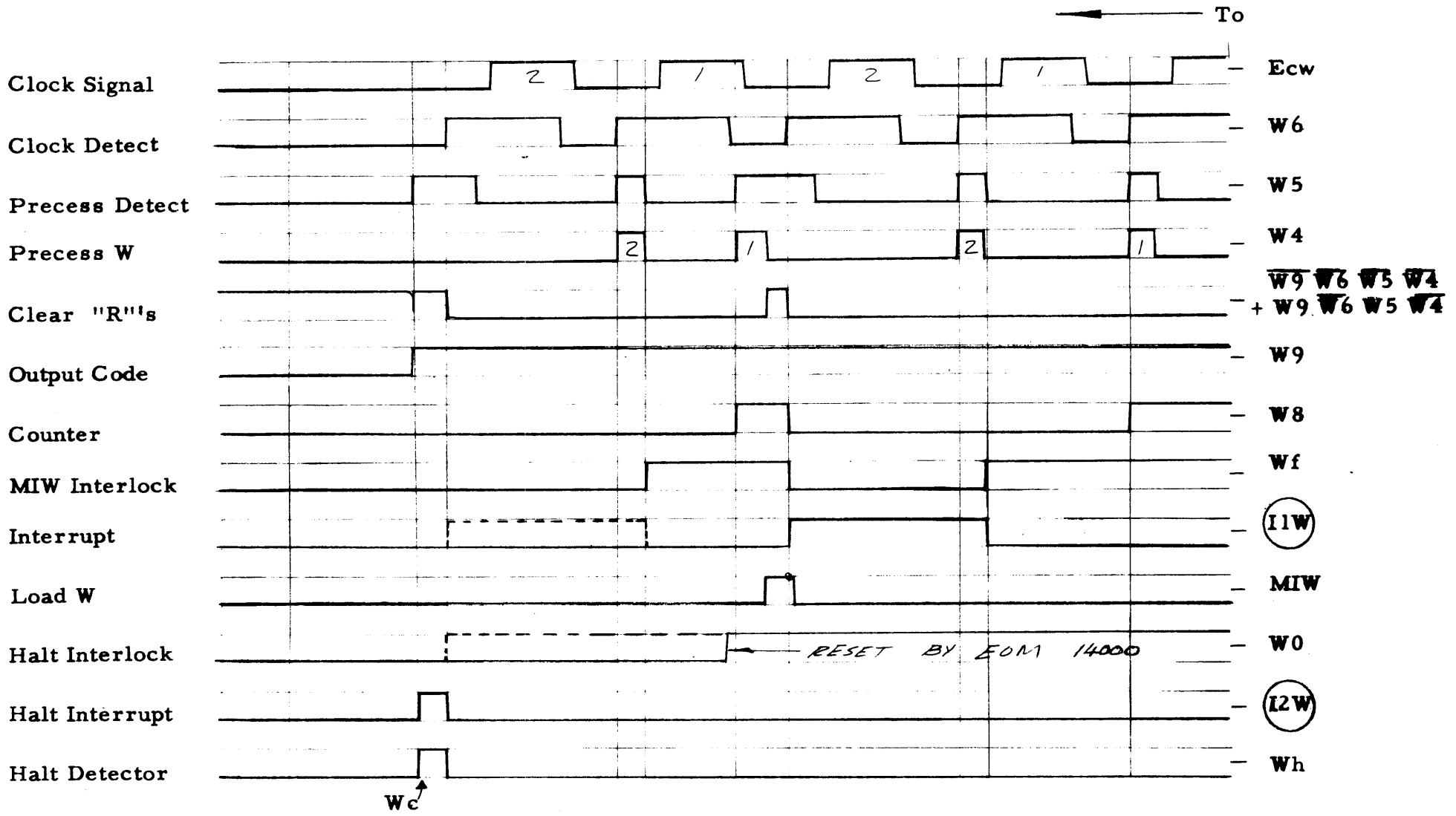


FIGURE 31

OUTPUT TERMINATION TIMING

(Except Magnetic Tape)

FIGURE 32

OUTPUT TERMINATION TIMING

MAGNETIC TAPE

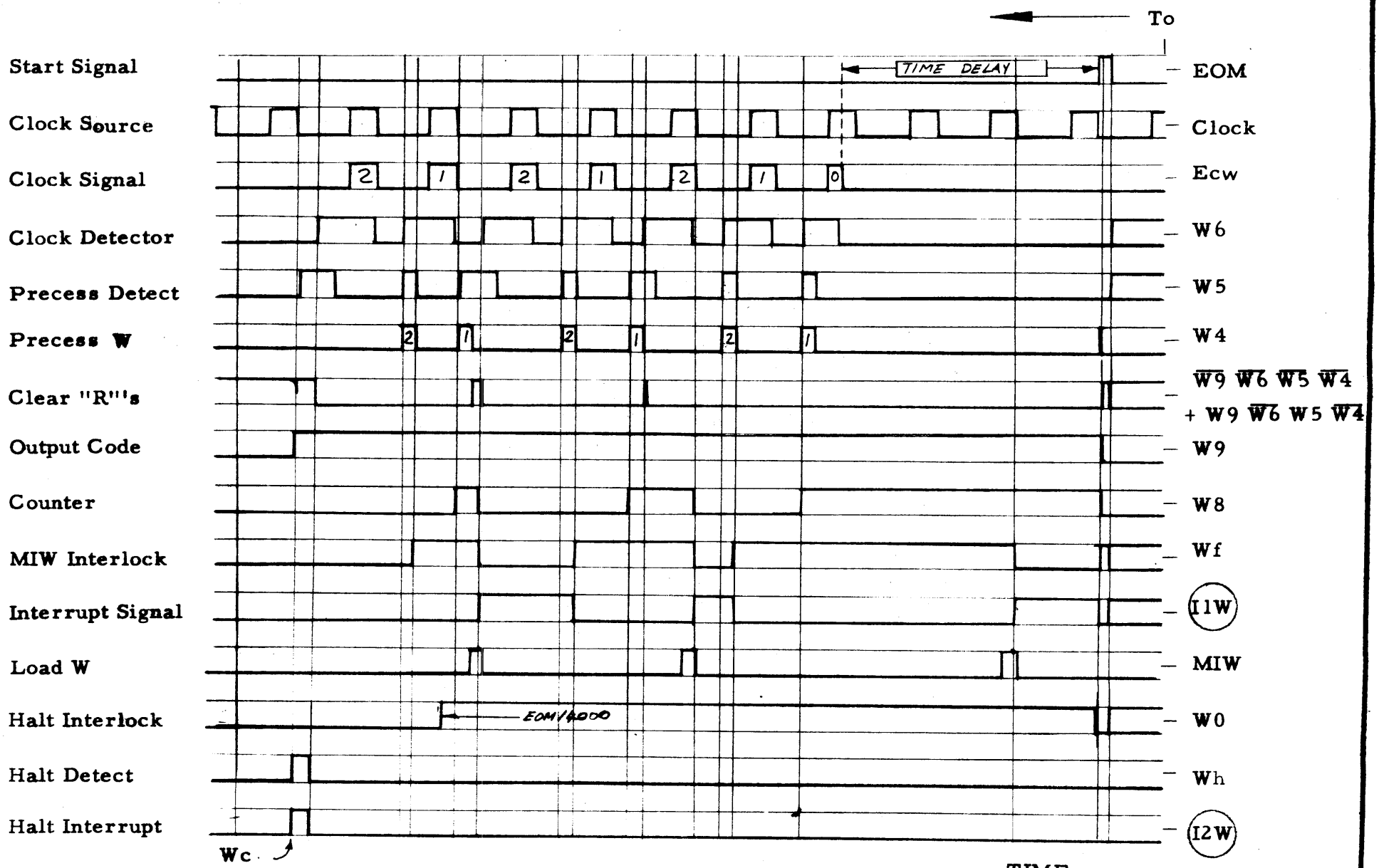


FIGURE 33

OUTPUT TIMING CHART

PUNCH

$$sW6 = \overline{W5}\ Ecw\ (T22-T17)$$

$$rW6 = W5\ T0 + Wc$$

$$sW5 = \overline{W5}\ W6\ \overline{Ecw}\ T0 + - - -$$

$$rW5 = W4\ T0 + Wc$$

The state, $\overline{W4}\ W5\ \overline{W6}$, is used to clear the SCR.

$$rR1 = - - - + W9\ \overline{W4}\ W5\ \overline{W6}$$

$$rR6 = - - - + W9\ \overline{W4}\ W5\ \overline{W6}$$

$$rRp = - - - + W9\ \overline{W4}\ W5\ \overline{W6}$$

While the state, $\overline{W0}\ W5\ \overline{W6}$, is used to set the halt detector for outputs other than magnetic tape outputs,

$$sWh = - - + W9\ \overline{W11}\ \overline{W0}\ W5\ \overline{W6}\ T24 + - - -$$

and to signal a magnetic tape unit that the last output character has been processed.

$$\text{Magnetic tape unit} \leftarrow \overline{W0}\ W5\ \overline{W6}$$

A magnetic tape unit, after a suitable delay, provides a halt signal, Whs, to set Wh.

$$sWh = - - - + Whs\ T24 + - - -$$

Regardless of the method of setting Wh to terminate an output process, the halt interrupt signal is generated in the cycle that Wh is set.

$$(I2w) = \overline{Wf}\ Wh\ (En + (En)')$$

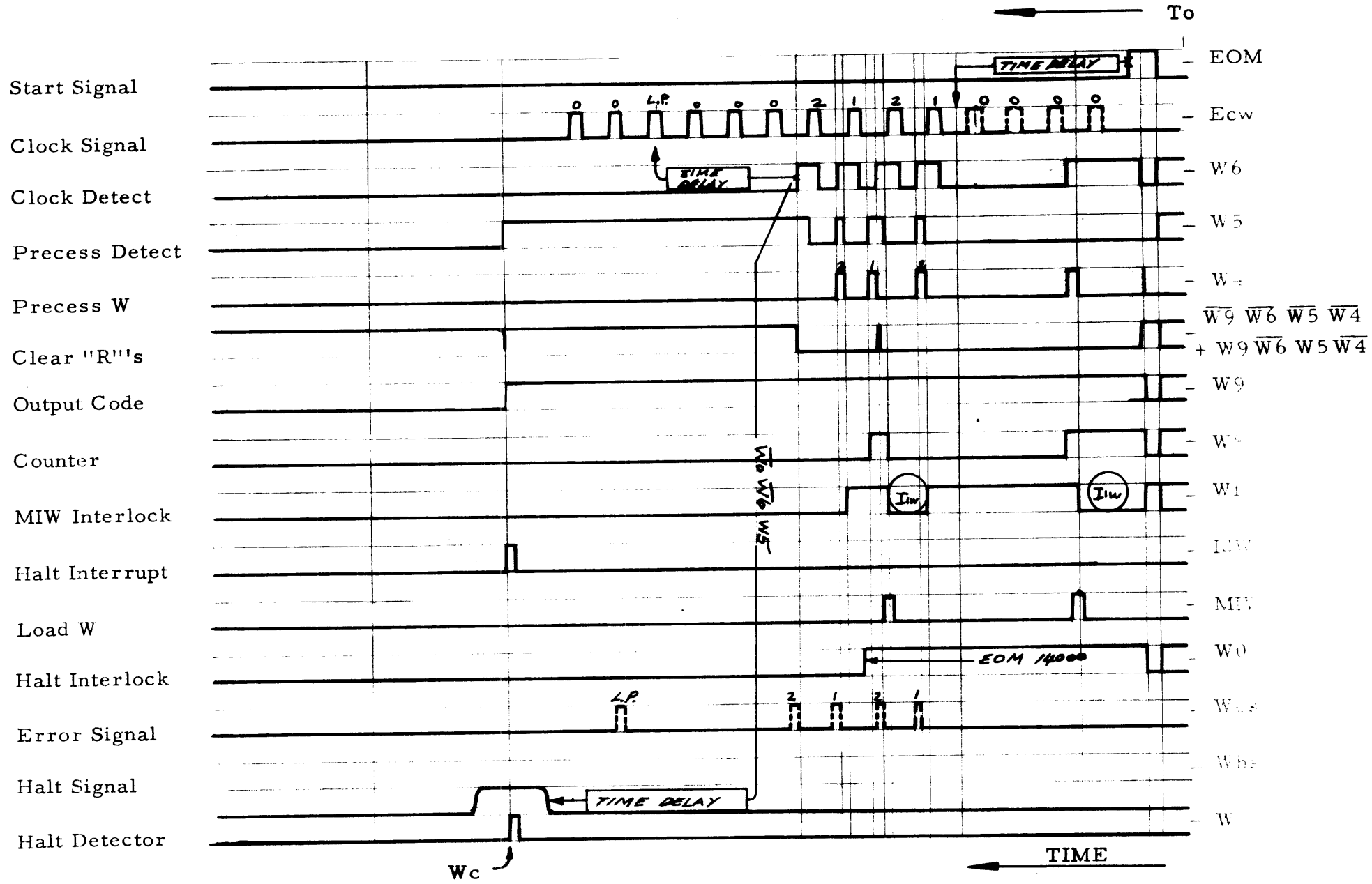
$$rWh = Wc$$

$$Wc = - - - + Wh\ \overline{Wf}\ T0 + - - -$$

The Output Termination Timing Charts indicate the flow of the output termination processes.

The Punch Output Timing Chart shows the flow of this complete output process. A time delay triggered by the EOM0 XX4X start instruction causes tape leader to be punched first while inhibiting output clock signals. An all-zeros character is also punched for the first output clock signal. After the last output character is processed, a halt interrupt signal is generated.

The Magnetic Tape Output Timing Chart shows the flow of this output process.



2 Char. /Word

FIGURE 34
OUTPUT TIMING CHART
MAGNETIC TAPE

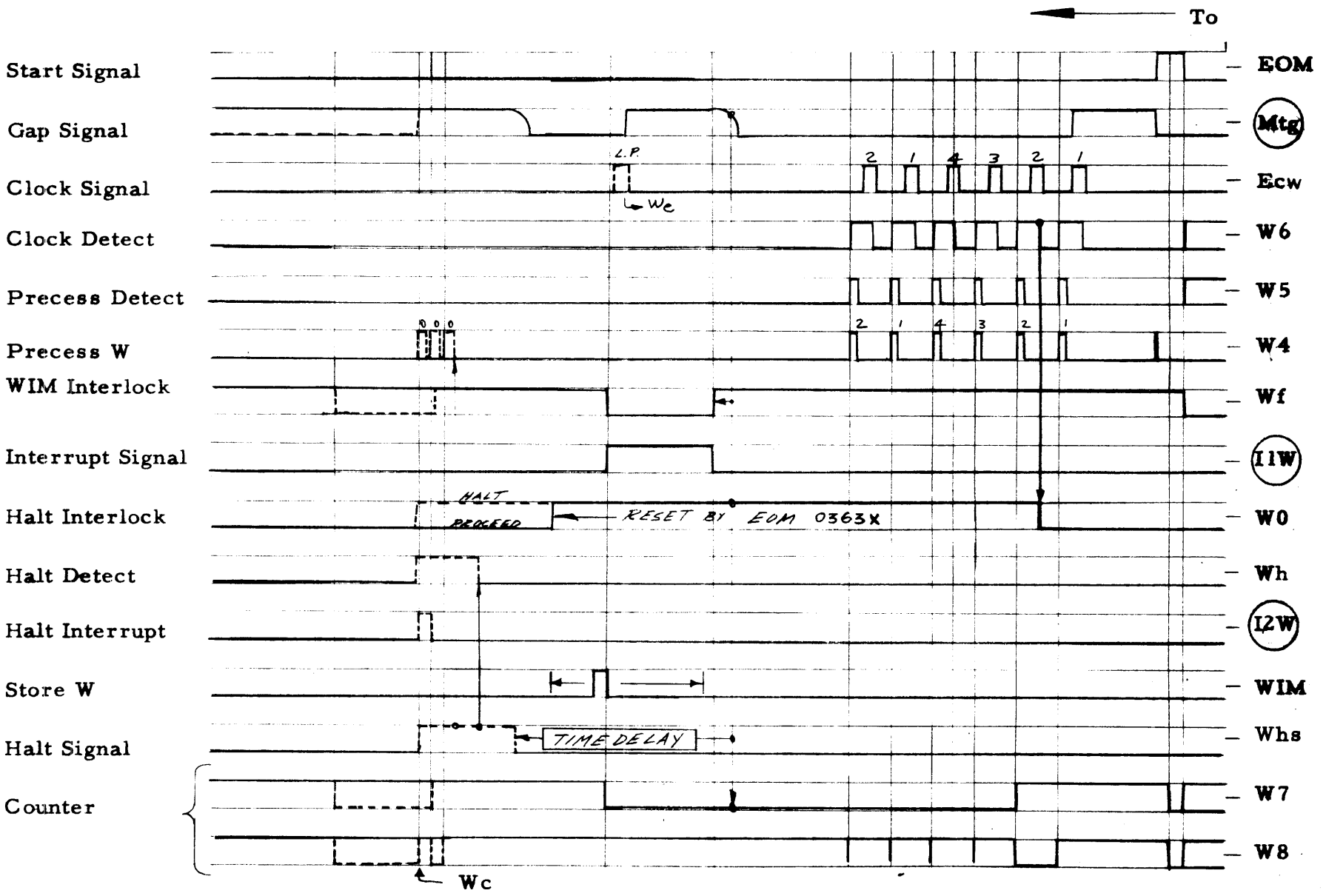


FIGURE 35
FORWARD SCAN TIMING
MAGNETIC TAPE

A time delay triggered by the EOM02X5X start instruction causes a tape gap to be recorded first while inhibiting the output clock signals. After the last output character is recorded, W9 $\overline{W4}$ $\overline{W5}$ $\overline{W6}$ clears the SCR and $\overline{W0}$ $\overline{W6}$ W5 signals the tape unit to count three clock signals and record the longitudinal parity character, and triggers a second time delay. This delay causes a gap to be recorded after the data block. After the gap is completed, the tape unit generates a $\overline{Whs}$ signal to halt the output process. Each character parity and the longitudinal parity of the characters reproduced at the read head are checked by the tape unit and an error signal, $\overline{Wes}$, is generated to set We for any detected errors.

$$sWe = - - - + Wes$$

SCAN ($\overline{W9}$ W10 W11)

A forward scan of magnetic tape data blocks is started with an EOM0363X instruction. The process is similar to the magnetic tape reading process except that the character counter is blocked from reaching 00 again after W0 is set by $\overline{W9}$ W6 $\overline{W8}$.

$$sW8 = - - - + \overline{W7}\ \overline{W9}\ W10\ W11\ \overline{Wh}$$

This prevents the normal interrupt signals by keeping Wf from being reset.

$$rWf = \overline{W8}\ \overline{W7}\ W4\ (T22\text{-}T17) + - - -$$

$$\overline{I1W} = \overline{Wf}\ W0\ \overline{Wh}\ (- - - - -)$$

This allows each input character to precess into the WAR without WIM instructions for setting Wf. When the end of a data block is reached, an interrupt signal is generated as Wf is reset.

$$rWf = - - - + \overline{W9}\ W10\ W11\ W0\ \overline{Mtg}\ \overline{W7}\ (T22\text{-}T17)$$

$$\overline{I1w} = \overline{Wf}\ W0\ \overline{Wh}\ (- - - - -)$$

This interrupt signal calls on the computer to execute a WIM instruction to store the last four characters of the data block from the WAR. Executing the WIM instruction sets W7 and Wf.

$$sWf = - - - + Wx\ (T5 - T0)\ W4$$

$$sW7 = - - - + Wx\ T24\ \overline{W4}\ Wn$$

Based on the last four data characters or a block counting program, the computer can reset W0 with another EOM0363X instruction to cause the scan process to continue without a pause through the gap and the next record. If W0 is not reset, the scan process will be terminated by a $\overline{Whs}$ signal from the tape unit in a manner similar to the manner for terminating magnetic tape input. When the scan process is allowed to terminate, a halt interrupt signal is generated.

$$\overline{I2w} = \overline{Wf}\ Wh\ (- - - - -)$$

During the scan process the character parity is checked by the Rp flip-flop and the longitudinal parity is checked by the tape unit (only if W0 is reset after the longitudinal parity character). Any error during the entire scan process will set We as in the magnetic tape input process.

A reverse scan of magnetic tape data blocks is started with an EOM 0763X instruction The process is similar to a forward magnetic tape scan, except that a WIM instruction at the end of a data block will store the first four data characters in reverse order, and that the longitudinal parity is not properly checked and We may be extraneously set.

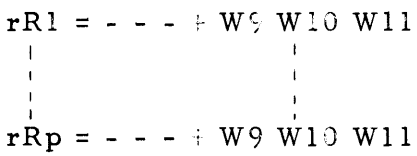
In both the Magnetic Tape Forward and Reverse Scan Timing Charts, the first interrupt signal is followed by a WIM instruction to store the last four characters read. The halt interrupt signal informs the computer that the gap has been reached. Wf is shown reset early on termination by $\overline{W9}$ W10 W11 W0 (Mtg) $\overline{W7}$ (T22-T17) rather than by the normal $\overline{W8}$ $\overline{W7}$ W4 (T22-T17) term. This allows Wh T24 to set W4 for only three 0's precessions rather than four before Wh $\overline{Wf}$ T0 resets the W Buffer. This has no significance because the WAR is cleared by the preceding WIM instruction.

If, after $\overline{W9}$ W10 W11 W0 (Mtg) $\overline{W7}$ (T22-T17) resets Wf to generate an interrupt signal, the WIM instruction is executed after Whs has set Wh; the last four characters read will be stored and a halt interrupt signal will still be generated. This is shown in the Magnetic Tape Forward Scan Timing Chart (late WIM instruction).

If, after $\overline{W9}$ W10 W11 W0 (Mtg) $\overline{W7}$ (T22-T17) resets Wf to generate an interrupt signal, the WIM instruction is executed before Whs can set Wh but an EOM 0363X instruction to continue the scan is executed after Whs has set Wh, a halt interrupt signal will be generated.

ERASE (W9 W10 W11)

A forward data block erase is started with an EOM 01X7X instruction. The erase process is similar to the magnetic tape output process except the SCR is forced to hold all zeros.



The process flows exactly as with magnetic tape output. Interrupt signals call on the computer to load W each time the WAR is empty. Executing an EOM 14000 instruction after an MIW instruction starts the output process termination. The tape unit generates a (Whs) signal after a gap is recorded to reset the W Buffer and generate a halt interrupt. The read head provides error signals, (Wes) for longitudinal or character parity errors.

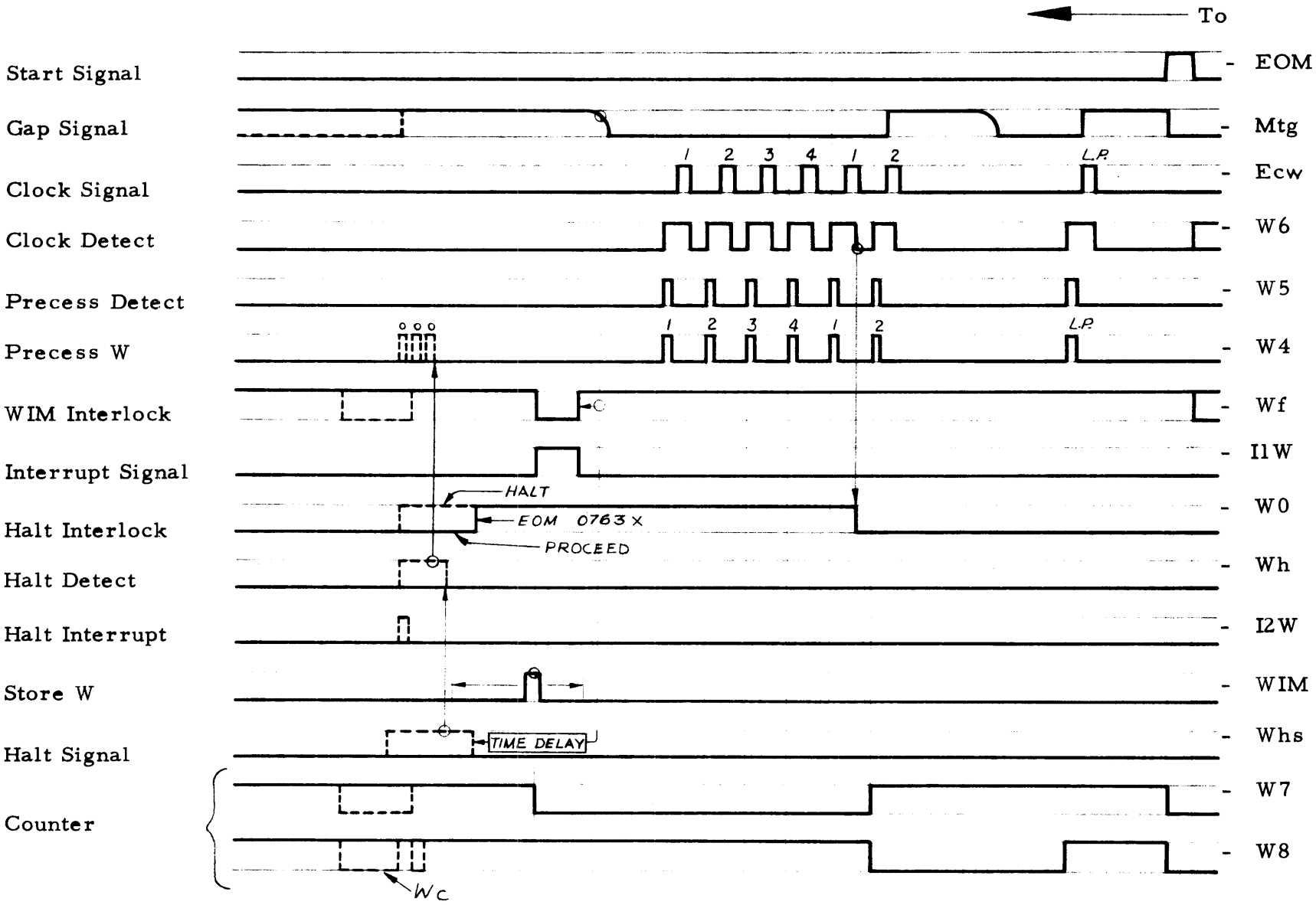


FIGURE 36
REVERSE SCAN TIMING
MAGNETIC TAPE

FILL

Photo-Reader Input 1 is used to fill the computer. First, the Start pushbutton is pressed to make (St) true. (St) makes Wc true to reset the W Buffer.

$$Wc = - - - + (St)\,(T5\text{-}T0)$$

$$rW14 = Wc$$
$$\vdots$$
$$rW9 = Wc$$
$$rW6 = - - - + Wc$$
$$rW5 = - - - + Wc$$
$$sWf = - - - + Wc\ \overline{Wh}$$
$$rWh = Wc$$
$$rWe = Wc$$
$$rW0 = Wc + - - -$$

(St) sets the character code to 11 for four characters per word and loads this code into the WAR.

$$sW8 = - - - + (St)$$
$$rW8 = Wc\ (T22\text{-}T17) + - - -$$

$$sW7 = - - - + (St)$$
$$rW7 = Wc\ (T22\text{-}T17) + - - -$$

$$sW4 = - - - + (St)\ T0$$
$$rW4 = - - - + W4\ T24$$
$$rIw = - - - + (St)$$
$$sWw = W4\ Tp\ W8 + W4\ T24\ W7 + - - - (T24 + Tp)\ \overline{W4}\ Wn$$

The fill input process is started by depressing the Fill switch, making (Kf) true. Kf is used to set W12.

$$sW12 = - - - + (Kf)$$

The input process proceeds and is terminated normally, when the Fill switch is released.

$$Re = \overline{W9}\ \overline{W10}\ \overline{W11}\ W12\ \overline{W13}\ \overline{W14}\ \overline{Kf}$$

TIME-SHARE INTERLACE

The interlace system is optional equipment. One can be installed to work in conjunction with the W Buffer and another can be independently installed to work in conjunction with the Y Buffer. No more than two interlace systems can be attached to each 910 or 920 Computer.

The interlace system is "enabled" by an Ioc or Buc instruction that contains a "one" in bit position 9. Therefore, the interlace system can be enabled with the same instruction that sets up the W Buffer (BUC), or it can be enabled without disturbing the W Buffer (Ioc). Either instruction clears the entire interlace system and then sets the enable flip-flop, Ew. (An underlined term represents the "clock"; the trigger occurs when the "clock" falls to zero volts.)

Clear:

$$Iwc = Eom\ C9\ \overline{C10}\ \overline{C17}\ Q2 + Wh + (St)$$

Enable Flip-Flop:

$$sEw = Eom\ C9\ \overline{C10}\ \overline{C17}\ \overline{Ew}\ \underline{Q1}$$

$$rEw = Iwc + \underline{Pot\ 1}\ Ew$$

$$Pot\ 1 = \overline{F1}\ F2\ \overline{F3}\ \overline{Ts}\ \overline{02}\ 06$$

With Ew set the computer can preset the interlace register with the starting address and the word count for the input or output process. If the word count is to be greater than 1023, the two most significant bits of the word count register must be preset. This must be done after the system is enabled and before the POT instruction is executed. An Ioc instruction, without bit 9, is given.

$$sIwb = Iwc$$

$$rIwb = Ioc\ C22\ Ew$$

$$sIwa = Iwc$$

$$rIwa = Ioc\ C23\ Ew$$

Note that these two most significant bits of the word count are set by the "clear" pulse and will remain set unless the Ioc instruction resets them. The flip-flops in the rest of the word counter register (Iw0 through Iw9) are reset by the clear pulse and are set by zeros in the C register during the POT instruction. After the POT instruction the word counter register contains the "one's complement" of the desired count. The counter can then count up, not down, and generate the termination signal when all of the flip-flops in the register contain "ones".

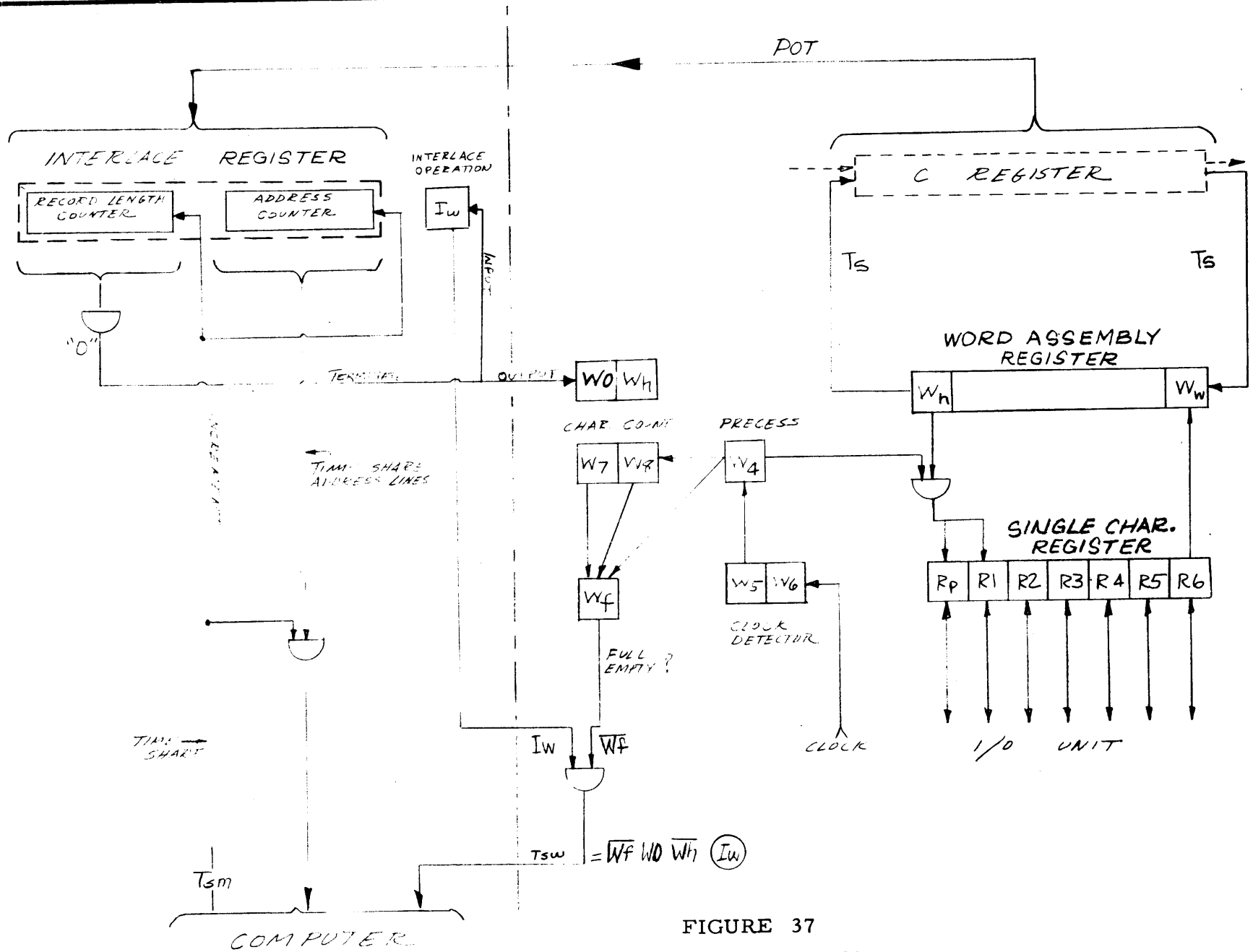


FIGURE 37

INFORMATION FLOW

INTERLACE OPERATION

The POT instruction sets the interlace counter registers and it also resets Ew and sets Iw.

$$rEw = (\underline{Pot\ 1})\ Ew + - - -$$

$$sIw = (\underline{Pot\ 1})\ Ew\ \overline{Iw}$$

Ew furnishes the required "ready" signal for the POT instruction. The Iw and Ew flip-flops inhibit W buffer interrupts and Iw allows the buffer to issue a time-share request signal to the computer whenever the buffer needs memory access. Iw indicates that the interlace is in operation.

(Interlace) $Tsw = \overline{Wf}\ W0\ \overline{Wh}\ (Iw)$

(Interrupt) $Ilw = \overline{Wf}\ W0\ \overline{Wh}\ (En + (En))(\overline{Ew}\ \overline{Iw})$

The POT instruction sets the interlace counters:

$$Iws = (Pot\ 2)\ Ew$$

$$sIw0 = Iws\ \overline{C0} + \underline{Iw\ 1}\ \overline{Iw0}$$

$$rIw0 = Iwc + \underline{Iw\ 1}\ Iw0$$

etc.

(Iw0 "counts" by the falling of the previous stage Iw1)

The address portion of the interlace register (Iw10 through Iw23) is reset by the clear pulse and is set by "ones" from the C register. It too counts up rather than down.

For example:

$$sIw20 = Iws\ C20 + \underline{Iw21}\ \overline{Iw20}$$
$$rIw20 = Iwc + \underline{Iw21}\ Iw20$$

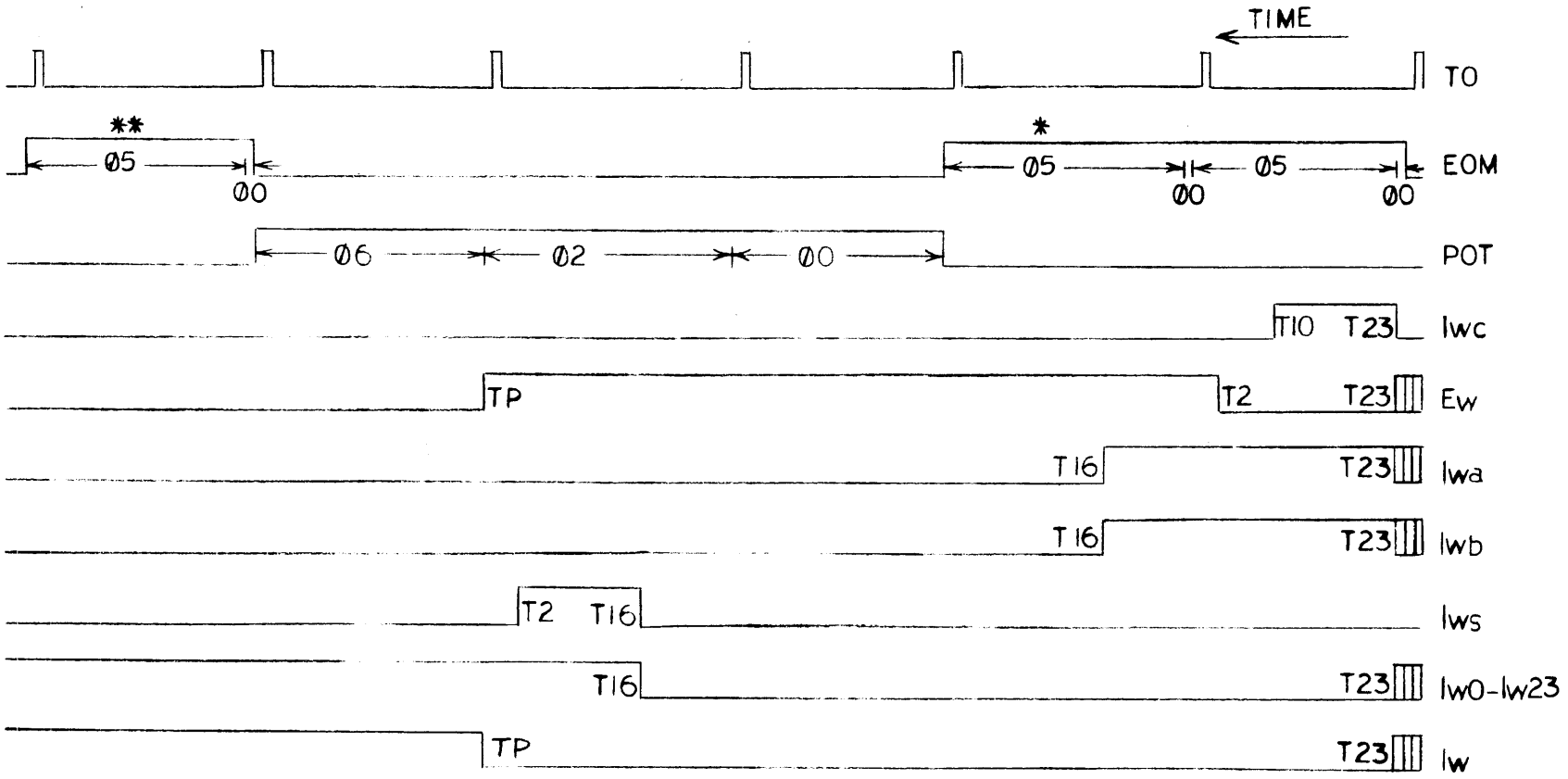
The LSB of both the word counter and the address counter are counted by (Wp Tsm).

$$sIw9 = Iws\ \overline{C9} + \underline{(Wp\ Tsm)}\ \overline{Iw9}$$
$$rIw9 = Iwc + \underline{(Wp\ Tsm)}\ Iw9$$

$$sIw23 = Iws\ C23 + \underline{(Wp\ Tsm)}\ \overline{Iw23}$$
$$rIw23 = Iwc + \underline{(Wp\ Tsm)}\ Iw23$$

When the word count register contains "ones" a signal, Iwf, is generated:

$$Iwf = Iw\ Iwb\ Iwa\ Iw0\ Iw1 - - - Iw9$$



* Second EOM is eliminated if not desired to reset Iwa or Iwb

** Third EOM (to activate buffer) may be combined with first EOM.

TIME ←

FIGURE 38

LOADING THE INTERLACE REGISTER

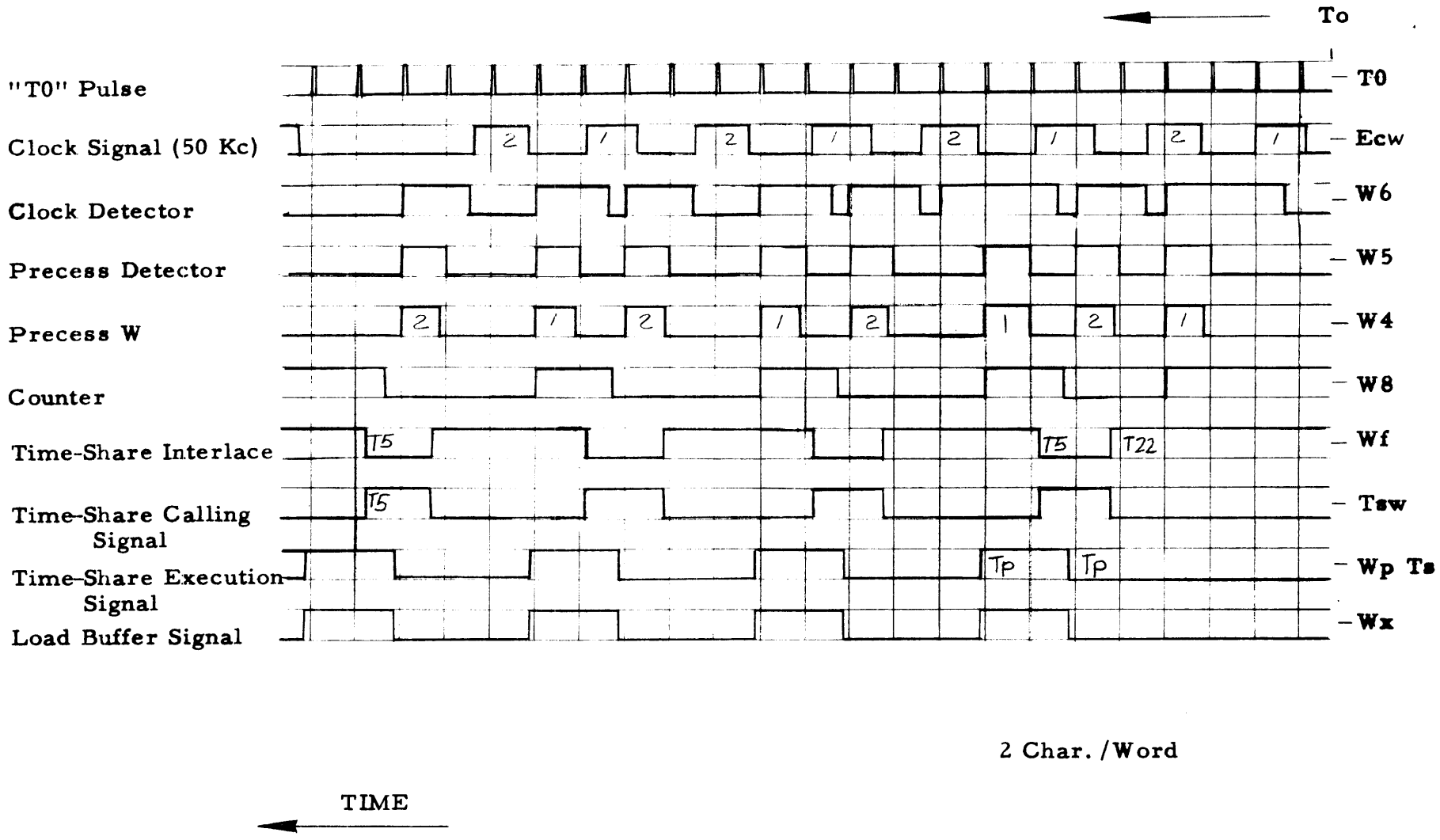


FIGURE 39

INPUT/OUTPUT TIMING

TIME-SHARE

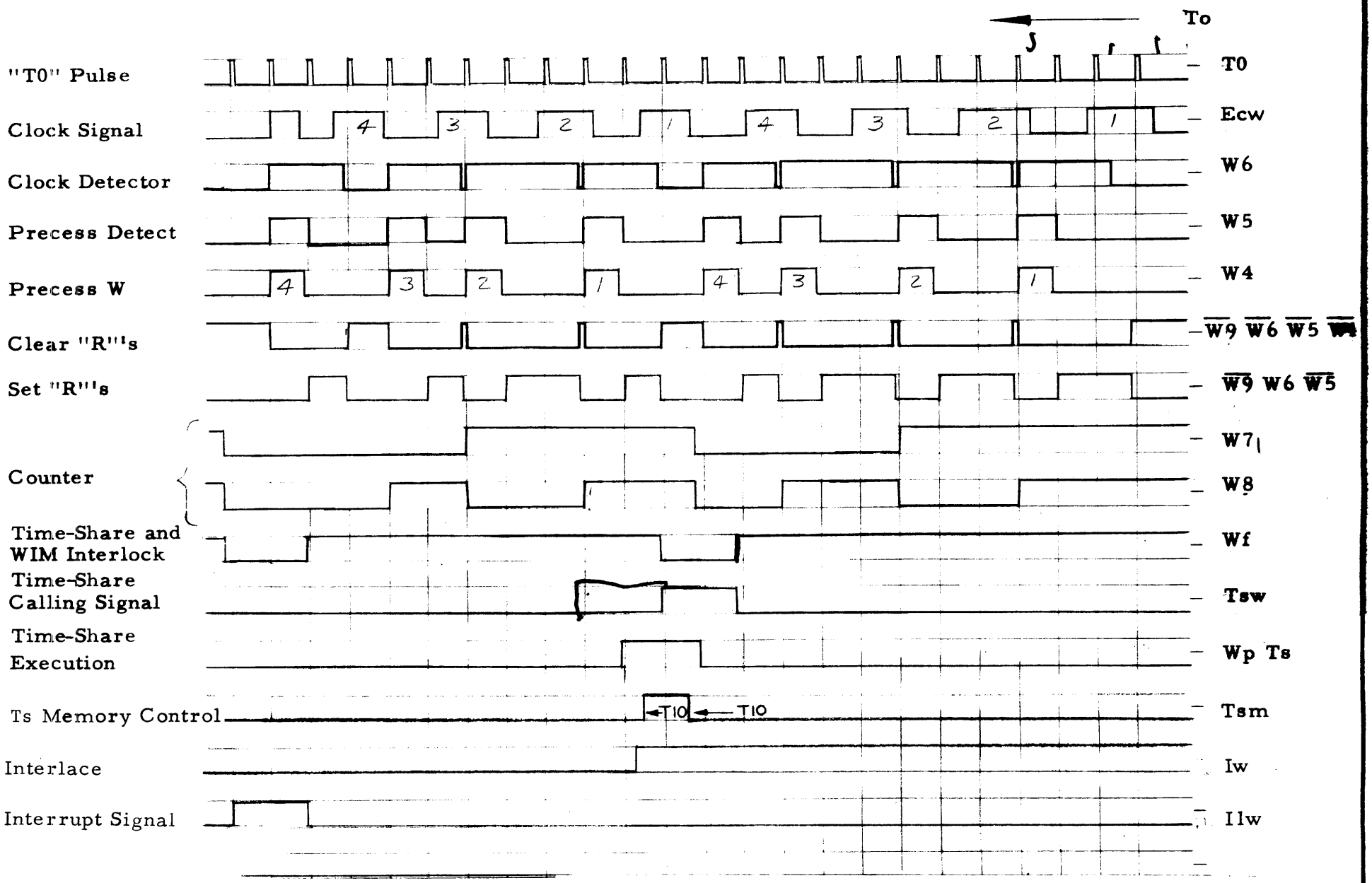


FIGURE 40
INPUT TERMINATION TIMING
TIME-SHARE

If the interlace is controlling an input process ($\overline{W9}$), the Iw flip-flop is reset so that the next time the buffer register is full an I1w interrupt (word ready interrupt) signal will be generated and the interlace register is disengaged.

$$rIw = Iwf\ Iw\ \underline{Q1}$$

If the interlace is controlling outputs, the Iwf signal will reset the W0 flip-flop in the W Buffer just before the last character in the W register is precessed to the character register. Signals W0 W5 W6 will then set Wh (except in the case of magnetic tape) after the last character has been clocked to the external device. When Wh is set, the interlace register is cleared as well as the W Buffer.

$$rW0 = (Iwf)\ W9\ (T5\text{-}T0)$$

$$Iwc = Wh + - - -$$

When the W Buffer is cleared an I2w interrupt (record complete interrupt) signal is generated by the buffer:

$$I2w = \overline{Wf}\ Wh\ (En + (En))$$

In the case of magnetic tape outputs, W0 is reset by the interlace signal Iwf. This causes the magnetic tape control to stop writing and stop the tape. After the tape has stopped the tape control unit sets Wh, which clears the W Buffer and the interlace register and allows the I2w interrupt.

An interlace system used in conjunction with the Y Buffer is exactly analogous with the W interlace. The two interlace registers are completely independent of one another.

The four interlace signals to the computer, Iw, Iwf and Iy, Iyf, must be grounded if a corresponding interlace register is not tied to the computer. This is done around the front of the interlace connector 25G if no interlace is connected (4 wires), or around the front of the "repeated" interlace connector at 45L if one interlace is connected (2 wires). If two interlace registers are connected, one plugs to the computer and the second plugs to the first (it does not matter which is W or Y). In this case, none of these four wires should be grounded.

PHOTO-READER 1

Photo-reader 1 is enabled by unit code 04.

$$Re = \overline{W9}\ \overline{W10}\ \overline{W11}\ W12\ \overline{W13}\ \overline{W14}\ \overline{Kf}$$

Note: (For photo-reader 2, $Re = \overline{W9}\ \overline{W10}\ \overline{W11}\ W12\ \overline{W13}\ W14$ for unit code 05).

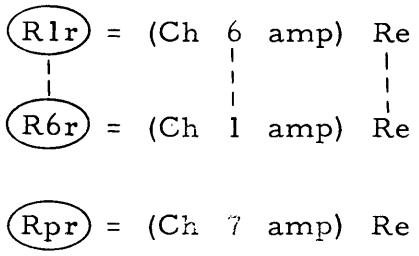
The pinch roller driver is energized by Re.

$$Rd = \overline{Rb} = Re + Kff$$

The signal from the sprocket channel amplifier is gated by Re.

$$\overline{Re\ Sp} = \overline{Re\ Psp}$$

The signals from the data channel amplifiers are gated by Re to form the character input signals.



The input clock is derived from the sprocket signal if the W Buffer is addressing the tape reader (Re). W0 is automatically set in the W Buffer during inputs after the first or second clock signal. Until W0 is set there must be a data channel punched in the tape for the Ecw clock to be sent. Beginning leader on the tape is therefore not clocked, but leader after W0 is set is clocked.

$$Ecw = (Zw1 + Zw2 + Zw3 + Zw4 + Zw5 + Zw6 + Zwp + W0)\ Sp\ Re$$

$$sW0 = \overline{W9}\ W6\ \overline{W8} + - - -$$

DICTIONARY OF BUFFER LOGIC TERMS

Buc	A control term derived from the EOM 0XXXX instruction and sent to the W and Y buffers.
Ecw	The eternal clock pulse used to detect inputs, it is generated by the internal clock in each input or output unit.
En	The enable flip-flop enables interrupts to occur in the computer.
(En)	The manual enable switch. This allows interrupts to occur.
EOM	The mnemonic code for the instruction which activates the buffer.
Eom	The execution term for an EOM instruction.
Ew	A flip-flop which enables a POT instruction to load the interlace register.
(I1)-(14)	Inputs to Is1 - Is4
(I1w)	An interrupt signal, signaling that a WIM or MIW instruction should be executed since the buffer is either full or empty.
(12w)	An interrupt signal, signaling that the input or output process should be terminated.
Ioc	Input/output control signal derived from the EOM 1XXXX instruction.
Ip1-Ip4	Interrupt program flip-flops that are true when executing the interrupt subroutine for its respective channel.
Is1-Is4	Interrupt storage flip-flops.
Iw	A flip-flop which determines if memory interlace is in operation.
Iwc	The term which clears the memory interlace registers.
(Iwf)	The signal for halting the input or output process when the record counter has been decremented to zero during memory interlace operation.
Iws	The term which sets the memory interlace register from the C register.
Iwb, Iwa, Iw0-Iw9	The record length counter in the memory interlace register.
Iw10-23	The address counter in the memory interlace register.
(Iw10) - (Iw23)	The address lines to memory from the address counter for memory interlace operation.
(Kf)	The manual fill switch for loading memory from the photo-reader.
Kff	Photo-reader tape feed switch on.
(Mtg)	The magnetic tape gap signal generated by the tape unit.

(Np)	No input parity, blocks sWe.
Rb	The brake driver signal for the photo-reader.
R1-R6	The six flip-flops which constitute the character buffer (parity excepted).
Rd	The pinch roller drive signal for the photo-reader.
Re	The enable signal for the photo-reader. This signal is enabled by the EOM unit address.
Rp	The parity flip-flop in the character buffer. An "odd" parity system is used.
($\overline{Rt}$)	The term which signals a transfer of interlace information from memory to the interlace register on a POT 0007X instruction.
Sio	An input gate (signal input/output) to the Sks gate (SKIP $\overline{M}$).
Sp	The sprocket signal for the photo-reader.
(St)	The manual start switch.
Sys	A control signal for system communication derived from the EOM 3XXXX instruction.
Ts	The time-share flip-flop which is on when the buffer has access to memory.
Tsw	The time-share calling signal indicating that the W Buffer desires access to memory.
Wc	The term which clears the unit address register, input/output state, the character counter, and the clock counter. In general, it resets the buffer and prepares it for a new operation.
We	The error flip-flop which will be set, when character parity is incorrect, when magnetic tape parity is incorrect, and when a WIM instruction is executed late. The state of the We flip-flop can optionally be tested for by the input program.
($\overline{Wes}$)	Error signal.
Wes	Inverse of ($\overline{Wes}$) ; it sets We.
Wf	The flip-flop which in its false state indicates that the W register is full or empty and in its true state allows the buffer to precess.
Wh	Halt flip-flop in W Buffer.
($\overline{Whs}$)	External halt signal to W Buffer
Whs	Inverse of ($\overline{Whs}$) ; it sets Wh.
Wn	The "now" flip-flop on the W register

W0	The flip-flop which, in general, enables a halt to occur after an input or output process has been initiated and has proceeded to the point that characters are being transferred.
Wp	The flip-flop which designates that the W Buffer is to participate in memory interlace operation.
Ws	The term enabled by an EOM 0X0XX which sets up the W Buffer from the C register.
Wx	The term which allows the W register to be loaded from the C register.
Ww	The "write" flip-flop on the W register.
W4	The flip-flop which controls the precessing of the data between the character register and the W register.
W5	The flip-flop which detects that a precess should occur.
W6	The flip-flop which detects that an input clock is present.
W7 W8	The flip-flops which constitute the character counter.
W9	The flip-flop which in its true state designates an output process and in its false state designates an input process.
W10-W14	The unit address register which designates which I/O unit is to be activated.
$\overline{Zw1}$-$\overline{Zw6}$	The 6 inputs to the 6-bit character buffer.
$\overline{Zwp}$	The external input to the parity flip-flop Rp.

W BUFFER AND INPUT/OUTPUT EQUATIONS

UNIT ADDRESS REGISTER

$sW14 = Ws\ C23$

$rW14 = Wc$

$sW13 = Ws\ C22$

$rW13 = Wc$

$sW12 = Ws\ C21 + (Kf)$

$rW12 = Wc$

$sW11 = Ws\ C20$

$rW11 = Wc$

$sW10 = Ws\ C19 + Ioc\ \overline{C17}\ \overline{W9}\ \overline{C19}\ \overline{C20}\ \overline{C21}\ \overline{C22}\ \overline{C23}\ (T5-T0)\ C12$

$rW10 = Wc$

INPUT/OUTPUT

$sW9 = Ws\ C18$

$rW9 = Wc$

CHARACTER COUNTER

$sW8 = Ws\ C16 + W7\ \overline{W8}\ W4\ T0 + Wx\ T24\ Ww$

$+ \overline{W7}\ \overline{W9}\ W10\ W11\ \overline{Wh} + (St)$

$rW8 = Wc\ (T22 - T17) + W8\ W4\ T0$

$sW7 = Ws\ C15 + Wx\ T24\ \overline{W4}\ Wn + (St)$

$rW7 = Wc\ (T22 - T17) + W7\ \overline{W8}\ W4\ T0$

CLOCK COUNTER

(Clock Detector)

$sW6 = \overline{W5}\ Ecw\ (T22 - T17)$

$rW6 = W5\ T0 + Wc$

(Precess Detector)

$sW5 = \overline{W5}\ W6\ \overline{Ecw}\ T0 + Ws\ C13\ C18$

$rW5 = W4\ T0 + Wc$

(Precess W)

$sW4 = W5\ Wf\ T24 + Ws\ T0 + Wh\ T24 + (St)\ T0$

$rW4 = W4\ T0 + W4\ T24$

COMPUTER INTERLOCK

$sWf = Wc\ \overline{Wh} + Wx\ (T5 - T0)\ \overline{W4}$

$rWf = \overline{W8}\ \overline{W7}\ W4\ (T22 - T17) + Ws\ W9 + \overline{W9}\ W10\ W11\ W0\ (Mtg)\ \overline{W7}\ (T22 - T17)$

INTERRUPT SIGNALS

$(I1w) = \overline{Wf}\ W0\ \overline{Wh}\ (En + (En))$ $(\overline{Iw})\ (\overline{Ew})$

$(I2w) = \overline{Wf}\ Wh\ (En + (En))$

TIME-SHARE CALLING SIGNAL

$Tsw = \overline{Wf}\ W0\ \overline{Wh}\ (Iw)$

HALT DETECTOR

$$sWh = Whs\ T24 + W9\ \overline{W11}\ \overline{W0}\ W5\ \overline{W6}\ T24$$

$$+ \overline{W9}\ \overline{W10}\ \overline{W11}\ W12\ \overline{W13}\ (\overline{R1}\ \overline{R2}\ \overline{R3}\ \overline{R4}\ \overline{R5}\ \overline{R6}\ \overline{Rp})\ W5\ T24$$

$$rWh = Wc$$

ERROR DETECTOR

$$sWe = \overline{W9}\ W4\ \overline{Rp}\ (T5 - T0)\ \overline{Wh}\ (Np) + W0\ \overline{W6}\ W5\ Ec\ Tp + Wes$$

$$rWe = Wc\ \overline{Wh}$$

SINGLE CHARACTER REGISTER

$$sR1 = W4\ Wn\ \overline{Wx}\ (\overline{Tp}\ \overline{T24}) + \overline{W9}\ W6\ \overline{W5}\ Zw1 + W4\ Wx\ C23$$

$$rR1 = W4\ \overline{Wn}\ \overline{Wx} + \overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4} + W9\ \overline{W4}\ W5\ \overline{W6} + W9\ W10\ W11 + W4\ Wx\ \overline{C23}$$

$$sR2 = W4\ R1 + \overline{W9}\ W6\ \overline{W5}\ Zw2$$

$$rR2 = W4\ \overline{R1} + \overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4} + W9\ \overline{W4}\ W5\ \overline{W6} + W9\ W10\ W11$$

$$sR3 = W4\ R2 + \overline{W9}\ W6\ \overline{W5}\ Zw3$$

$$rR3 = W4\ \overline{R2} + \overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4} + W9\ \overline{W4}\ W5\ \overline{W6} + W9\ W10\ W11$$

$$sR4 = W4\ R3 + \overline{W9}\ W6\ \overline{W5}\ Zw4$$

$$rR4 = W4\ \overline{R3} + \overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4} + W9\ \overline{W4}\ W5\ \overline{W6} + W9\ W10\ W11$$

$$sR5 = W4\ R4 + \overline{W9}\ W6\ \overline{W5}\ Zw5$$

$$rR5 = W4\ \overline{R4} + \overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4} + W9\ \overline{W4}\ W5\ \overline{W6} + W9\ W10\ W11$$

$$sR6 = W4\ R5 + \overline{W9}\ W6\ \overline{W5}\ Zw6$$

$$rR6 = W4\ \overline{R5} + \overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4} + W9\ \overline{W4}\ W5\ \overline{W6} + W9\ W10\ W11$$

$$sRp = \overline{W9}\ W4\ \overline{Rp}\ Ww\ (T22 - T17) + W9\ W4\ \overline{Rp}\ Wn\ (T5 - T0)\ \overline{Wx}$$
$$+ \overline{W9}\ W6\ \overline{W5}\ Zwp + W9\ W4\ (T22 - T17) + W9\ W4\ \overline{Rp}\ C23\ (T5 - T0)\ Wx$$

$$rRp = \overline{W9}\ W4\ Rp\ Ww\ (T22 - T17) + W9\ W4\ Rp\ Wn\ (T5 - T0)\ \overline{Wx}$$
$$+ \overline{W9}\ \overline{W6}\ \overline{W5}\ \overline{W4} + W9\ \overline{W4}\ W5\ \overline{W6} + W9\ W10\ W11 + W9\ W4\ Rp\ C23\ (T5 - T0)\ Wx$$

LOAD W FROM C

$$Wx = (\overline{01}\ 03\ \overline{04}\ 05\ \overline{06}\ F1\ \overline{F3})\ \overline{Ts} + Wp\ Ts$$

CLOCK SIGNAL

$$Ecw = \overline{\overline{Ecw}}$$

WORD ASSEMBLY REGISTER

$sWw = W4\ R6$

$+ W4\ Tp\ W8$

$+ W4\ T24\ W7$

$+ (\overline{T24}\ \overline{Tp})\ \overline{W4}\ Wx\ C23 + (T24 + Tp)\ \overline{W4}\ Wn$

$+ \overline{W4}\ Wn\ \overline{Wx}$

$rWw = (\overline{sWw})$

sWn = Ww delayed by 24 pulse times

$rWn = \overline{Ww}$ delayed by 24 pulse times

CLEAR AND SET SIGNALS

$Wc = Buc\ \overline{C17}\ (T22 - T17) + Wh\ \overline{Wf}\ T0 + (St)(T5 - T0)$

$Ws = Buc\ \overline{C17}\ (T5 - T0)$

W Buffer Ready $= \overline{W9}\ \overline{W10}\ \overline{W11}\ \overline{W12}\ \overline{W13}\ \overline{W14}$

WIM and MIW interlock $= \overline{Wf}\ (W0 + \overline{W9})$

HALT INTERLOCK

$sW0 = \overline{W9}\ W6\ \overline{W8} + Ws\ W9$

$rW0 = Wc + Ioc\ C12\ \overline{C17}\ \overline{C19}\ \overline{C20}\ \overline{C21}\ \overline{C22}\ \overline{C23}\ (T5 - T0) + W9\,(Iwf)\,(T5 - T0)$

MAGNETIC TAPE CONTROL SIGNALS

Stop Read Interlock = W0

Output Character Interlock $= \overline{W5}$

INTERLACE LOGIC

Clear

$$Iwc = Eom\ C9\ \overline{C10}\ \overline{C17}\ Q2 + Wh + (St)$$

Prepare

$$sEw = Eom\ C9\ \overline{C10}\ \overline{C17}\ \underline{Q1}\ \overline{Ew}$$

$$rEw = Iwc + \underline{Pot\ 1}\ Ew$$

Load Registers

$$Iws = Pot\ 2\ Ew$$

Interlace

$$sIw = \underline{Pot\ 1}\ Ew\ \overline{Iw}$$

$$rIw = Iwc + Iwf\ \underline{Q1}\ Iw$$

Finished

$$Iwf = Iw\ Iwb\ Iwa\ Iwo\ Iw1\ Iw2\ Iw3\ Iw4 - - - Iw9$$

Ready

$$(\overline{Rt}) = (\overline{Ew})$$

Word Count

$$sIwb = Iwc + \underline{Iwa}\ \overline{Iwb}$$

$$rIwb = Ioc\ C22\ Ew + \underline{Iwa}\ Iwb\ Iw$$

$$sIwa = Iwc + \underline{Iwo}\ \overline{Iwa}$$

$$rIwa = Ioc\ C23\ Ew + \underline{Iwo}\ Iwa$$

$$sIwo = Iws\ \overline{Co} + \underline{Iw1}\ \overline{Iwo}$$

$$rIwo = Iwc + \underline{Iw1}\ Iwo$$

$$\vdots$$

$$sIw9 = Iws\ \overline{C9} + (\underline{Tsm\ Wp})\overline{Iw9}$$

$$rIw9 = Iwc + (\underline{Tsm\ Wp})Iw9$$

Address

$$sIw10 = Iws\ C10 + \underline{Iw11}\ \overline{Iw10}$$

$$rIw10 = Iwc + \underline{Iw11}\ Iw10$$

$$sIw23 = Iws\ C23 + (\underline{Tsm\ Wp})\ \overline{Iw23}$$

$$rIw23 = Iwc + (\underline{Tsm\ Wp})Iw23$$

PHOTO READER 1

ENABLE SIGNAL

$$Re = \overline{W9}\ \overline{W10}\ \overline{W11}\ W12\ \overline{W13}\ \overline{W14}\ \overline{Kf}$$

PINCH ROLLER AND LAMP DRIVERS

$$Rd = \overline{Rb} = Re + Kff$$

READER SIGNALS

$$\overline{Zw1} = \overline{(Ch\ 6\ amp)\ Re}$$

$$\overline{Zw2} = \overline{(Ch\ 5\ amp)\ Re}$$

$$\overline{Zw3} = \overline{(Ch\ 4\ amp)\ Re}$$

$$\overline{Zw4} = \overline{(Ch\ 3\ amp)\ Re}$$

$$\overline{Zw5} = \overline{(Ch\ 2\ amp)\ Re}$$

$$\overline{Zw6} = \overline{(Ch\ 1\ amp)\ Re}$$

$$\overline{Zwp} = \overline{(Ch\ 7\ amp)\ Re}$$

$$\overline{Ecw} = \overline{(Zw1 + Zw2 + Zw3 + Zw4 + Zw5 + Zw6 + Zwp + W0)\ Sp\ Re}$$

BRAKE DRIVER

$$Rb = \overline{Re}\quad \overline{Kff}$$

Y BUFFER OPERATIONS

The Y Buffer is similar to the W Buffer in all respects except for the following:

(1) Bit C17 defines the Y Buffer in EOM instructions controlling buffer operations ($\overline{C17}$ defines the W Buffer).

(2) Op codes 10 and 30 are used to load and unload the Y Buffer; Op codes 12 and 32 indicate similar operations with the W Buffer.

(3) Y Buffer interrupts branch the program to different memory locations from the W Buffer.

(4) (Ts $\overline{Wp}$) causes interlace with Y whereas (Ts Wp) causes interlace with W.

(5) The Y Buffer may be extended to accommodate up to 24 bits per transmission.

Extending the Y Buffer beyond six bits plus parity requires that the R register be widened to the appropriate character size and the parity be checked and generated by a new circuit.

Rpy is still used to accept the parity bit during input; however during output a new line is used, Rpe. During input the parity check is made just prior to the precession, thus the sYe becomes:

$$\overline{Y9}\ \overline{Yh}\ (Np)\ \overline{Y4}\ Y5\ Rpe$$

Rpe is specially generated and is "true" if the character register (including Rpy) has an even number of "ones".

The equations for Rpy become:

$$sRpy = \overline{Y9}\ Y6\ \overline{Y5}\ Zyp$$
$$+ Y9\ Y4\ (T22 - T17)$$

$$rRpy = \overline{Y9}\ Y4\ Rpy\ Yw\ (T22 - T17)$$
$$+ Y9\ Y4\ Rpy\ (T5 - T0)\ \overline{Yx}$$ (This always resets Rpy on outputs when Y Buffer extension is used.)
$$+ \overline{Y9}\ \overline{Y6}\ \overline{Y5}\ \overline{Y4}$$
$$+ Y9\ \overline{Y4}\ Y5\ \overline{Y6}$$
$$+ Y9\ Y10\ Y11$$
$$+ Y9\ Y4\ Rpy\ C23\ (T5 - T0)\ Yx$$

The set term of Yw is modified to accept the least significant bit of the R register during precession time.

$$sYw = Y4\ (LSB\ of\ R) + - - -$$

The end of block term for the paper tape reader which sets Yh is modified to include two more bits and becomes:

$$sYh = \overline{Y9}\ \overline{Y10}\ \overline{Y11}\ Y12\ \overline{Y13}\ (\overline{R1y}\ \overline{R2y}\ \overline{R3y}\ \overline{R4y}\ \overline{R5y}\ \overline{R6y}\ \overline{R7y}\ \overline{R8y})\ Ys\ T24 + - - -$$

Y BUFFER EQUATIONS

UNIT ADDRESS REGISTER

$sY14 = Ys\ C23$

$rY14 = Yc$

$sY13 = Ys\ C22$

$rY13 = Yc$

$sY12 = Ys\ C21$

$rY12 = Yc$

$sY11 = Ys\ C20$

$rY11 = Yc$

$sY10 = Ys\ C19 + Ioc\ C17\ \overline{Y9}\ \overline{C19}\ \overline{C20}\ \overline{C21}\ \overline{C22}\ \overline{C23}\ (T5 - T0)\ C12$

$rY10 = Yc$

INPUT/OUTPUT

$sY9 = Ys\ C18$

$rY9 = Yc$

CHARACTER COUNTER

$sY8 = Ys\ C16 + Y7\ \overline{Y8}\ Y4\ T0 + Yx\ T24\ Yw + \overline{Y9}\ \overline{Y7}\ Y10\ Y11\ \overline{Yh} + (St)$

$rY8 = Yc\ (T22-T17) + Y8\ Y4\ T0$

$sY7 = Ys\ C15 + Yx\ T24\ \overline{Y4}\ Yn + (St)$

$rY7 = Yc\ (T22-T17) + Y7\ \overline{Y8}\ Y4\ T0$

CLOCK COUNTER

(Clock Detector)

$sY6 = \overline{Y5}\ Ecy\ (T22 - T17)$

$rY6 = Y5\ T0 + Yc$

(Precess Detector)

$sY5 = \overline{Y5}\ Y6\ \overline{Ecy}\ T0 + Ys\ C13\ C18$

$rY5 = Y4\ T0 + Yc$

(Precess Y)

$sY4 = Y5\ Yf\ T24 + Ys\ T0 + Yh\ T24 + (St)\ T0$

$rY4 = Y4\ T0 + Y4\ T24$

COMPUTER INTERLOCK

$sYf = \overline{Y4}\ Yx\ (T5 - T0) + Yc\ \overline{Yh}$

$rYf = \overline{Y8}\ \overline{Y7}\ Y4\ (T22 - T17) + Ys\ Y9 + \overline{Y9}\ Y10\ Y11\ Y0\ (Mtgy)\ \overline{Y7}\ (T22 - T17)$

INTERRUPT SIGNALS

$(I1y) = \overline{Yf}\ Y0\ \overline{Yh}\ (En + (En))$ $(\overline{Iy})\ (\overline{Ey})$

$(I2y) = \overline{Yf}\ Yh\ (En + (En))$

TIME-SHARE CALLING SIGNAL

$Tsy = \overline{Yf}\ Y0\ \overline{Yh}\ (Iy)$

HALT DETECTOR (See Note)

$$sYh = Yhs\ T24 + Y9\ \overline{Y11}\ \overline{Y0}\ Y5\ \overline{Y6}\ T24$$
$$+ \overline{Y9}\ \overline{Y10}\ \overline{Y11}\ Y12\ \overline{Y13}\ (\overline{R1Y}\ \overline{R2y}\ \overline{R3y}\ \overline{R4y}\ \overline{R5y}\ \overline{R6y}\ \overline{Rpy})\ Y5\ T24$$
$$rYh = Yc$$

ERROR DETECTOR (See Note)

$$sYe = \overline{Y9}\ Y4\ \overline{Rpy}\ (T5 - T0)\ \overline{Yh}\ (Np) + Y0\ \overline{Y6}\ Y5\ Ecy\ Tp + Yes$$
$$rYe = Yc\ \overline{Yh}$$

SINGLE CHARACTER REGISTER (See Note)

$$sR1y = Y4\ Yn\ \overline{Yx}\ (\overline{Tp}\ \overline{T24}) + \overline{Y9}\ Y6\ \overline{Y5}\ Zy1 + Y4\ Yx\ C23$$
$$rR1y = Y4\ \overline{Yn}\ \overline{Yx} + \overline{Y9}\ \overline{Y6}\ \overline{Y5}\ \overline{Y4} + Y9\ \overline{Y4}\ Y5\ \overline{Y6} + Y9\ Y10\ Y11$$
$$+ Y4\ Yx\ \overline{C23}$$
$$sR2y = Y4\ R1y + \overline{Y9}\ Y6\ \overline{Y5}\ Zy2$$
$$rR2y = Y4\ \overline{R1y} + \overline{Y9}\ \overline{Y6}\ \overline{Y5}\ \overline{Y4} + Y9\ \overline{Y4}\ Y5\ \overline{Y6} + Y9\ Y10\ Y11$$

$$sR3y = Y4\ R2y + \overline{Y9}\ Y6\ \overline{Y5}\ Zy3$$
$$rR3y = Y4\ \overline{R2y} + \overline{Y9}\ \overline{Y6}\ \overline{Y5}\ \overline{Y4} + Y9\ \overline{Y4}\ Y5\ \overline{Y6} + Y9\ Y10\ Y11$$

$$sR4y = Y4\ R3y + \overline{Y9}\ Y6\ \overline{Y5}\ Zy4$$
$$rR4y = Y4\ \overline{R3y} + \overline{Y9}\ \overline{Y6}\ \overline{Y5}\ \overline{Y4} + Y9\ \overline{Y4}\ Y5\ \overline{Y6} + Y9\ Y10\ Y11$$

$$sR5y = Y4\ R4y + \overline{Y9}\ Y6\ \overline{Y5}\ Zy5$$

$$rR5y = Y4\ \overline{R4y} + \overline{Y9}\ \overline{Y6}\ \overline{Y5}\ \overline{Y4} + Y9\ \overline{Y4}\ Y5\ \overline{Y6} + Y9\ Y10\ Y\ 11$$

$$sR6y = Y4\ R5y + \overline{Y9}\ Y6\ \overline{Y5}\ Zy6$$

$$rR6y = Y4\ \overline{R5y} + \overline{Y9}\ \overline{Y6}\ \overline{Y5}\ \overline{Y4} + Y9\ \overline{Y4}\ Y5\ \overline{Y6} + Y9\ Y10\ Y11$$

$$sRpy = \overline{Y9}\ Y4\ \overline{Rpy}\ Yw\ (T22 - T17) + Y9\ Y4\ \overline{Rpy}\ Yn\ (T5 - T0)\ \overline{Yx}$$
$$+ \overline{Y9}\ Y6\ \overline{Y5}\ Zyp + Y9\ Y4\ (T22 - T17) + Y9\ Y4\ \overline{Rpy}\ C23\ (T5 - T0)\ Yx$$

$$rRpy = \overline{Y9}\ Y4\ Rpy\ Yw\ (T22 - T17) + Y9\ Y4\ Rpy\ Yn\ (T5 - T0)\ \overline{Yx}$$
$$+ \overline{Y9}\ \overline{Y6}\ \overline{Y5}\ \overline{Y4} + Y9\ \overline{Y4}\ Y5\ \overline{Y6} + Y9\ Y10\ Y11 + Y9\ Y4\ Rpy\ C23\ (T5 - T0)\ Yx$$

LOAD Y FROM C (from computer)

$$Yx = (\overline{01}\ 03\ \overline{04}\ \overline{05}\ \overline{06}\ F1\ \overline{F3}\ )\ \overline{Ts} + \overline{Wp}\ Ts$$

CLOCK SIGNAL

$$Ecy = \overline{\overline{Ecy}}$$

WORD ASSEMBLY REGISTER (See Note)

$$sYw = Y4\ (R6y) + Y4\ Tp\ Y8 + Y4\ T24\ Y7 + (\overline{T24}\ \overline{Tp})\ \overline{Y4}\ Yx\ C23$$
$$+ (T24 + Tp)\ \overline{Y4}\ Yn + \overline{Y4}\ Yn\ \overline{Yx}$$

$$rYw = (\overline{sYw})$$

sYn = Yw delayed by 24 pulse times

rYn = $\overline{Yw}$ delayed by 24 pulse times

CLEAR AND SET SIGNALS

$$Yc = Buc\ C17\ (T22 - T17) + Yh\ \overline{Yf}\ T0 + (St)\ (T5 - T0)$$

$$Ys = Buc\ C17\ (T5 - T0)$$

$$Y\ Buffer\ Ready = \overline{Y9}\ \overline{Y10}\ \overline{Y11}\ \overline{Y12}\ \overline{Y13}\ \overline{Y14}$$

$$YIM\ and\ MIY\ interlock = \overline{Yf}\ (Y0 + \overline{Y9})$$

HALT INTERLOCK

$$sY0 = \overline{Y9}\ Y6\ \overline{Y8} + Ys\ Y9$$

$$rY0 = Yc + Ioc\ C12\ C17\ \overline{C19}\ C20\ C21\ \overline{C22}\ \overline{C23}\ (T5-T0) + Y9\ (\overline{Iyf})\ (T5-T0)$$

MAGNETIC TAPE CONTROL SIGNALS

Stop Read Interlock = Y0

Output Character Interlock = $\overline{Y5}$

Y BUFFER EXTENSION

NOTE: When the Y Buffer extension is used the following modifications are required in the Y Buffer Logic.

(1) $sYh = Yhs\ T24$
$+ \overline{Y9}\ \overline{Y11}\ \overline{Y0}\ Y5\ \overline{Y6}\ T24$
$+ \overline{Y9}\ \overline{Y10}\ \overline{Y11}\ Y12\ \overline{Y13}\ (\overline{R1y}\ \overline{R2y}\ \overline{R3y}\ \overline{R4y}\ \overline{R5y}\ \overline{R6y}\ \overline{R7y}\ \overline{R8y}\ \overline{Rpy})\ Y5\ T24$

(2) $sYe = \overline{Y9}\ \overline{Yh}\ \overline{Np}\ \overline{Y4}\ Y5\ Rpe$
$+ Y0\ \overline{Y6}\ Y5\ Ecy\ Tp$
$+ Yes$

(3) $sRpy = \overline{Y9}\ Y6\ \overline{Y5}\ Zyp$
$+ Y4\ Y9\ (T22 - T17)$

$rRpy = \overline{Y9}\ Y4\ Rpy\ Yw\ (T22 - T17)$
$+ Y9\ Y4\ Rpy\ (T5 - T0)\ \overline{Yx}$
$+ \overline{Y9}\ \overline{Y6}\ \overline{Y5}\ \overline{Y4}$
$+ Y9\ \overline{Y4}\ Y5\ \overline{Y6}$
$+ Y9\ Y10\ Y11$
$+ Y9\ Y4\ Rpy\ C23\ (T5 - T0)\ Yx$

(4) $sYw = Y4$ (Least significant bit of R register)
$+ Y4\ Tp\ Y8$
$+ Y4\ T24\ Y7$
$+ (\overline{T24}\ \overline{Tp})\ \overline{Y4}\ Yx\ C23$
$+ (T24 + Tp)\ \overline{Y4}\ Yn$
$+ \overline{Y4}\ Yn\ \overline{Yx}$

Reset R Register

$$RRy = \overline{Y9}\ \overline{Y6}\ \overline{Y5}\ \overline{Y4} + Y9\ \overline{Y4}\ Y5\ \overline{Y6} + Y9\ Y10\ Y11$$

Enable R Register

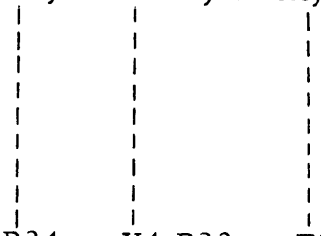
$$ERy = \overline{Y9}\ Y6\ \overline{Y5}$$

Parity

Rpe = Even number of ones in information bits.

R Register

$$sR7y = Y4\ R6y + ERy\ Zy7$$
$$rR7y = Y4\ \overline{R6y} + RRy$$



$$sR24y = Y4\ R23y + ERy\ Zy24$$
$$rR24y = Y4\ \overline{R23y} + RRy$$

SECTION III

MAGNETIC CORE MEMORY

NOTE: This section applies only to core memory units that use QK53 and HK55 type modules.

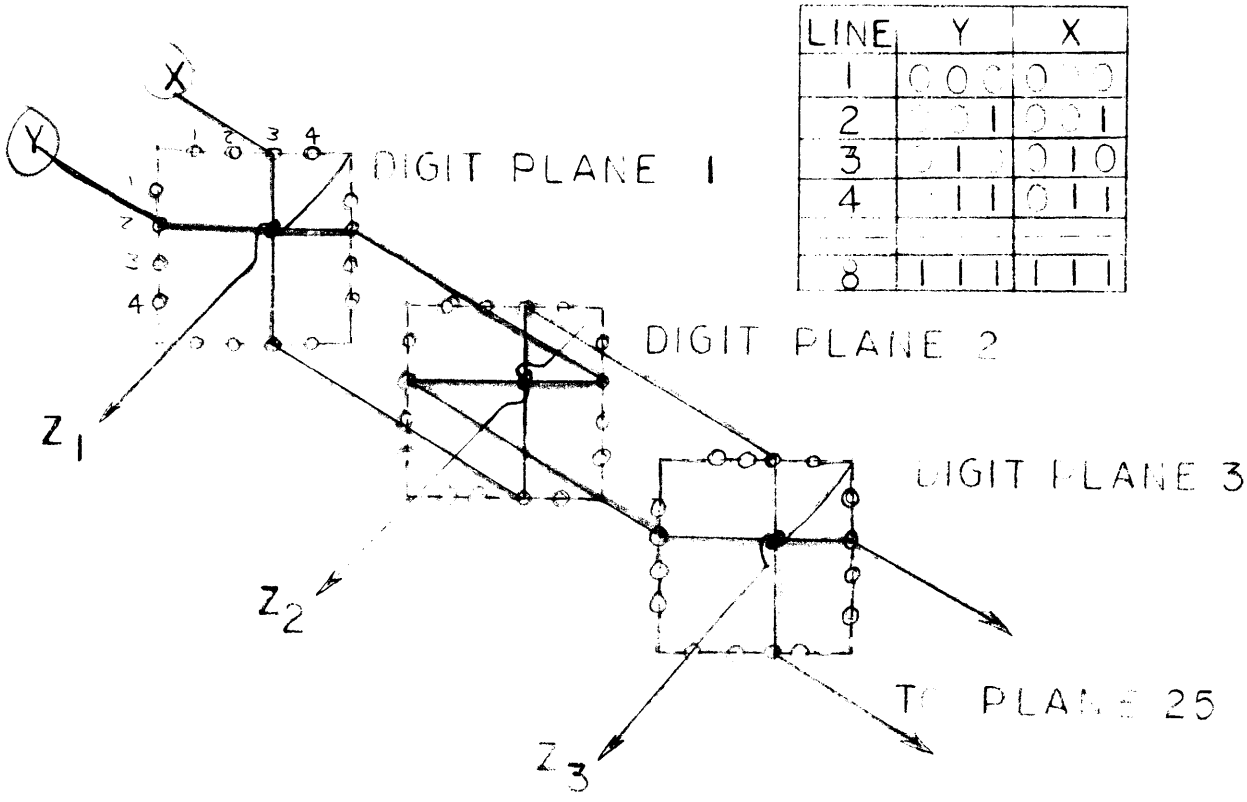
INTRODUCTION

This section explains the logical and electronic operation of the core memory in detail. Layout, wiring information, and additional circuitry information are contained in a separate manual.

The reader should be familiar with Memory Control, Page 1.39, in Section 1 of this manual.

The computer uses a parallel, random access, coincident current, magnetic core memory for permanent storage of all internal data and instructions.

Each individual bit in each data or instruction word is stored in an individual magnetic element called a core. These ferrite cores are the basic storage elements and are capable of representing one of two logical states, i.e., one-zero, true-false. The ferrite cores are physically arranged as the points of a three-dimensional coordinate system; the X and Y coordinates of a specific core represent the address of the word of which that core is a member; the Z coordinate determines which bit of the word the core represents. Thus, a single column of cores (i.e., all cores with the same X and Y coordinates) represents a single word. Through all of the cores on each Z level or digit plane there are two wires used for the reading and writing of information. Below is an example of selecting word number 12_8.



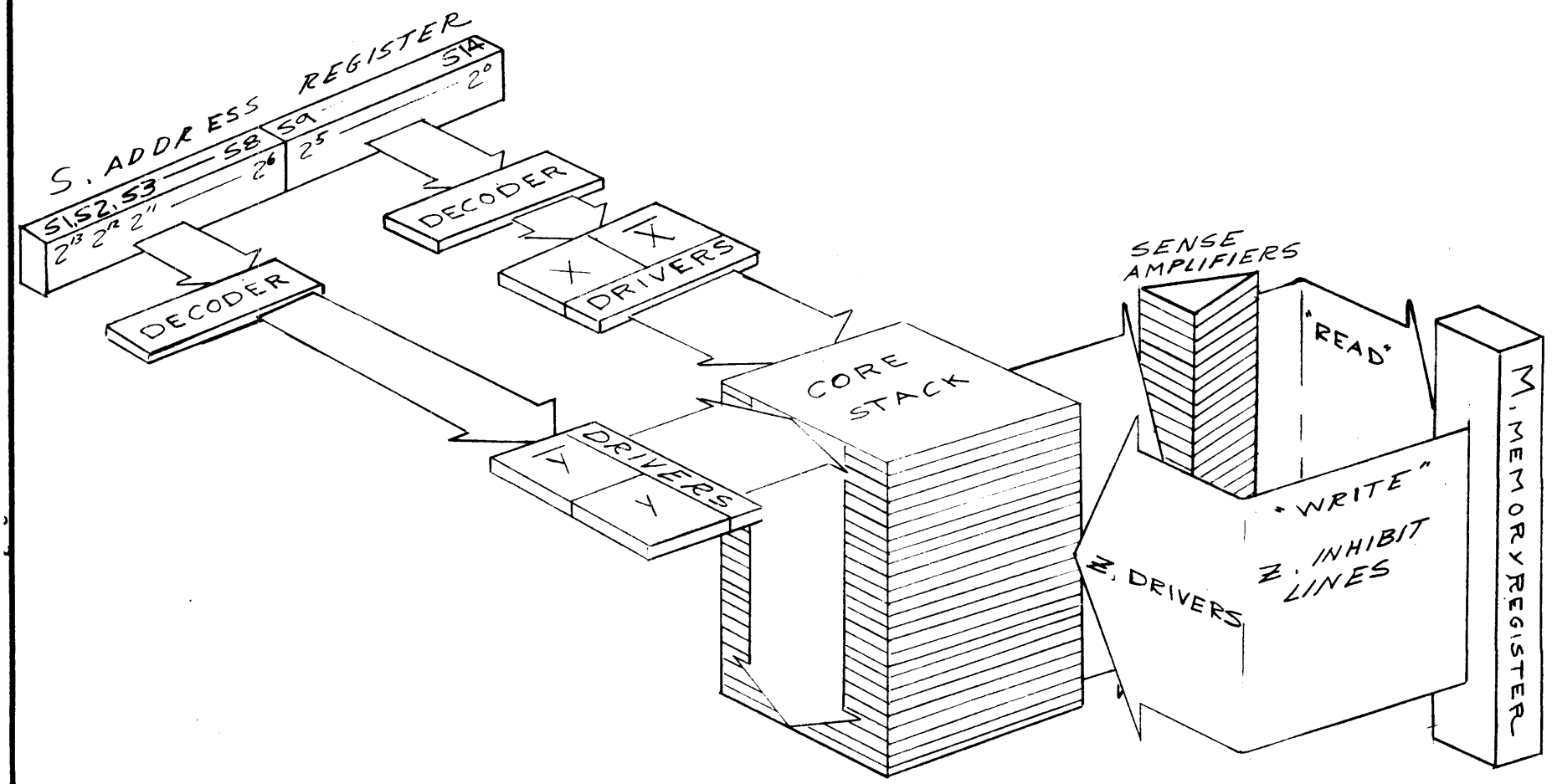


FIGURE 41.

BLOCK DIAGRAM

MAGNETIC CORE MEMORY

The Z coordinate wires are used for parallel detection of information or for indirectly writing information. The block diagram of the magnetic core memory shows the parallel outputs of the S (address) register being decoded into X and Y coordinate signals. To reduce the complexity and number of components in the address system, the X and Y coordinates are sub-divided into X, $\overline{X}$, Y, and $\overline{Y}$. These decoded address signals correspondingly drive current driver circuits whose output lines pass through the cores on the X and Y coordinates.

The 25 (25 bits per word) Z coordinate windings (hereafter called sense windings) are fed to 25 sense or detection amplifiers which detect the state of the cores in the specific word being addressed; and this information is fed in parallel to the M (memory) register for later transfer to the C register.

Note that although the sense windings pass through all cores in a specific plane, only the state of the addressed core will be detected, due to the required coincidence of the X and Y coordinates.

When writing into memory from the M register, the outputs of the $\overline{M}$ side of the M flip-flops drive 25 Z drivers that correspondingly drive a second set of Z windings, which are used to inhibit the insertion of "ones" in the respective cores requiring "zeros". The reasons for this technique will be explained later.

The reading of information from a core memory of this type is destructive in that all of the cores in the addressed word are left in a "zero" state; therefore, the information read into the M register must be regenerated (written back) into the same address position before proceeding. When regenerating or writing into a word position, the X and Y coordinates will attempt to set all cores at the coincidence points to the "one" state. Note that previously they have all been reset to the zero state by a reading process. The inhibit or Z windings will therefore inhibit the setting of cores requiring zeros, and the correct information is written into the word location. The process of reading and then immediately regenerating is called a memory cycle and takes 6 microseconds to completely execute.

MAGNETIC CORE STORAGE THEORY

The elementary storage element, the ferrite core, whose principal characteristic is a rectangular, hysteresis loop, is shown in Figure 42. H is the magnetizing force (proportional to the applied current) applied to the core, and B is the magnetic flux (proportional to the magnetic permeability of the material) produced in the core by the force H. If the force H_t is applied to the core, the flux density in the "one" direction will increase to Øm, which represents maximum flux density or saturation. When the magnetizing force H_t is removed, the flux density will decrease to the Ør value, which represents the remanent or residual flux density that (for a constant Øm) is proportional to the retentivity or remanence of the material. The core is now said to be in the "one" state.

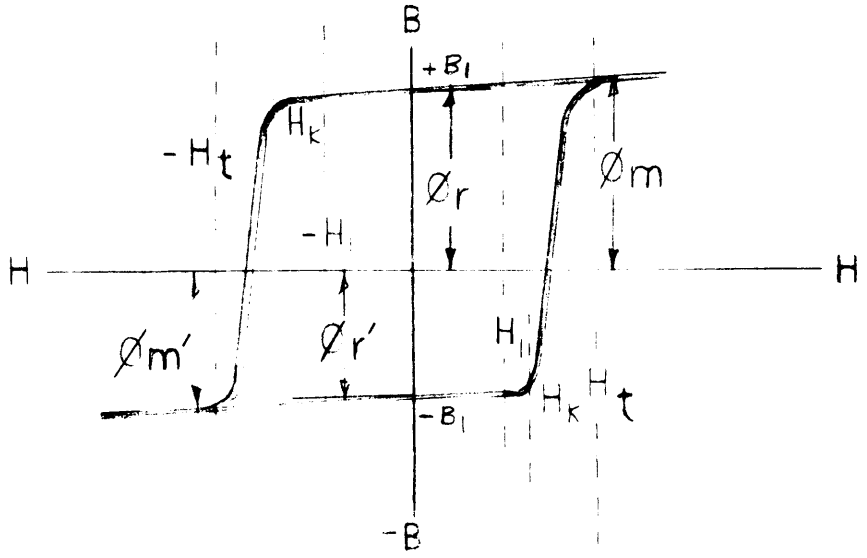


Figure 42 Hysteresis Loop

If the magnetizing force is applied in the opposite direction $-H_t$ (reverse the current), the remanent flux density Ør′ will be -B1 and the core will be in the "zero" state. Note that although the Ør′ value approximately equals the previous residual flux density value, the direction of the flux has completely reversed.

The designations of "zero" and "one" for a magnetic core are completely arbitrary and are only in relationship to the polarities of the applied signals, sense, and inhibit windings.

The high squareness ratio (Hk/Ht) of the hysteresis loop makes it possible to apply magnetizing forces which are less than that required to saturate the core in either direction, without materially changing the value of the magnetic flux density of the core. An abrupt reversal of the flux direction will take place when the magnetizing force exceeds a critical value. The reversal of the magnetic flux direction switches the core to the complement of its present state. The core will stay indefinitely in

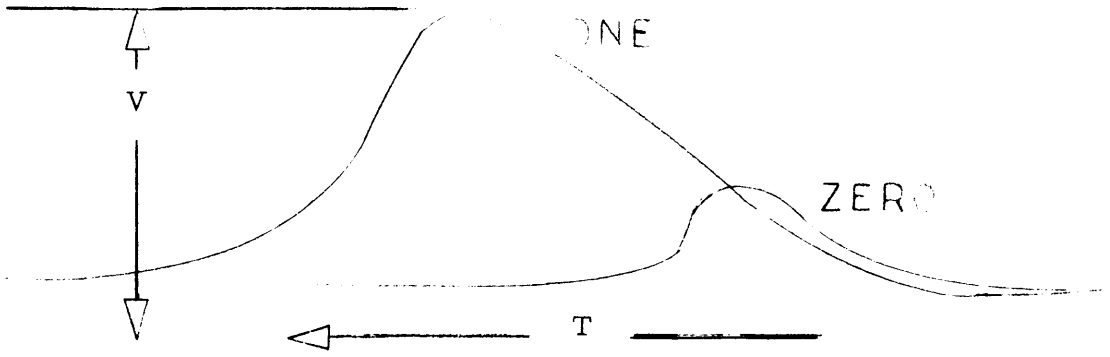
either state until switched back by a sufficient electromotive force applied in the opposite direction to the force which set it initially.

The critical values, the knee, and the squareness ratio, permit the coincident-current type of operation which required the coincidence of two partial magnetizing forces, $\frac{Ht}{2}$ or $\frac{-Ht}{2}$, each of which alone is less than the magnetizing force required to drive the core past the knee (point Hk) of the curve of the hysteresis loop shown in Figure 42. The two partial magnetizing forces, acting simultaneously, are sufficient to change the state of the core. The two partial magnetizing forces are supplied by two drive half-currents applied along X and Y drive lines, respectively. The switching action of the coincident partial magnetizing forces can be prevented by the application of an inhibit current whose value is equal, but whose direction is opposite, to either one of the write drive half-currents. The resulting inhibit magnetizing force cancels the effect of one of the partial magnetizing forces and prevents the core from switching to the other of its two stable states.

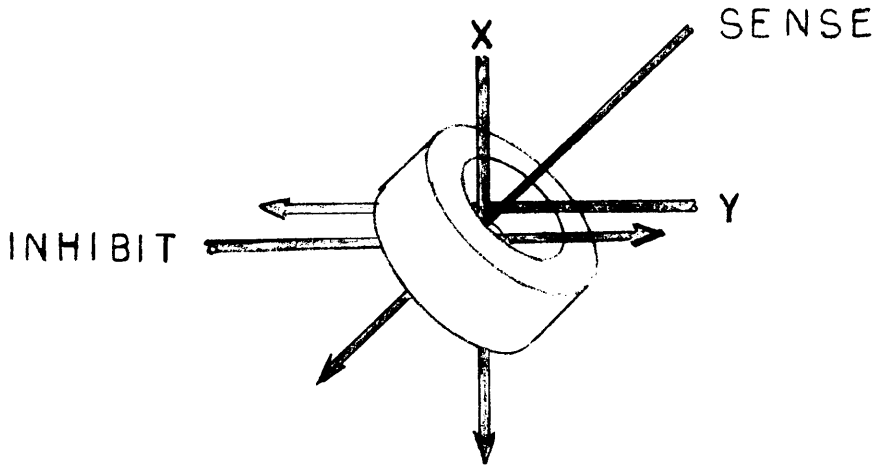
The state of a core at a given moment can be determined by the application of read drive current in the direction that would switch the core to the zero state. If the core is already in the zero state, the read drive current will drive it minutely into saturation, but no flux reversal will take place, and only a small amount of flux variation will result. If on the other hand, the core is in the "one" state, it will be switched to the "zero" state by the read drive current, the direction of the magnetic flux will be reversed, and the relative flux variation will be very large. In both cases the flux variations will appear in the form of a voltage induced in a sense winding. If the flux density is varied with time at a rate $d\Phi/dt$, then, according to Faraday's law, a voltage V is induced across the windings of magnitude:

$$V = -N \frac{d\Phi}{dt}$$

where N equals the number of turns, which in this case, would be one. The signal voltage pulse shown below permits the determination of the state that the core was in before the application of the read drive current. All magnetic cores in a matrix are linked by the X and Y drive lines, the inhibit winding, and the sense winding as shown below:



Each selected X and Y drive line carries a drive half-current that applies a partial magnetizing force to all magnetic cores linked by those lines. The core located at the intersection of the selected drive lines will receive the sum of the two partial magnetizing forces, which will be sufficient to produce the magnetic flux reversal and change the state of the core.



The X and Y drive lines are connected to the X and Y decoders and traverse all digit planes in series. During the read operation, the direction of the drive current in the matrix is such that the core at the selected address would be switched to zero. As explained above, the cores that are in the one state will produce voltage pulses in the sense winding while those that are in the zero state will produce virtually no signal. Each matrix or digit plane has its own individual sense winding, which will thus read the bit stored in that digit plane at the selected address. When the information is being written into the matrix, write or restore drive half-currents are applied to the selected X and Y drive lines in the direction opposite to that of the read drive currents, so that the core at the drive lines intersection would be switched to the one state. However, in the case of matrices in which a zero has been written, an inhibit current is applied through the inhibit winding that is individual for each matrix. The inhibit current is approximately of the same magnitude as an X or Y drive current and will prevent the core located at the intersection of the selected X and Y drive lines from being switched to one; a zero bit will remain stored in that matrix at the selected address. Each inhibit winding intersects all cores on a given digit plane.

ADDRESSING

The 14 outputs of the S register (address register) are gated into 14 address lines designated L1 through L14 as shown below:

$$\overline{L1} = \overline{S1\ \overline{Tsm} + (Iw10)\ (Tsm\ Wp) + (Iy10)\ (Tsm\ \overline{Wp})}$$

$$\overline{L14} = \overline{S14\ \overline{Tsm} + (Iw23)\ (Tsm\ Wp) + (Iy23)\ (Tsm\ \overline{Wp})}$$

$$L1 = \overline{\overline{L1}}$$

$$L14 = \overline{\overline{L14}}$$

Note that the address lines from the buffer interlace system enter at this point.

The 14 address lines are decoded into the X and Y coordinates according to the following chart:

2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14
		Y Coordinate						X Coordinate					
		Y			$\overline{Y}$			X			$\overline{X}$		

The X and Y coordinates each represent 6 bits or two octal digits. These coordinates are divided into $\overline{X}$, X, $\overline{Y}$, and Y, each of which consist of a single octal digit that can be decoded into 8 specific configurations.

The L1 and L2 positions, hereafter referred to as 2^{13} and 2^{12}, are used to define which 4096-word memory stack is being addressed.

$2^{\overline{13}}\ 2^{\overline{12}}$ = stack one, locations 00000 (0) to 07777 (4095)

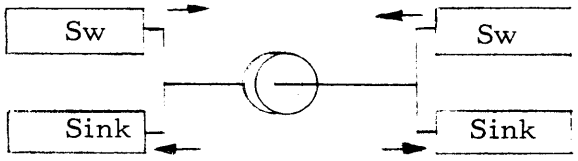
$2^{\overline{13}}\ 2^{12}$ = stack two, locations 10000 (4096) to 17777 (8191)

$2^{13}\ 2^{\overline{12}}$ = stack three, locations 20000 (8192) to 27777 (12287)

$2^{13}\ 2^{12}$ = stack four, locations 30000 (12288) to 37777 (16383)

Four decoders are thus required and are shown in Figure 43. The decoding function for a single octal digit is performed by an individual circuit module designated QK52, which has one output for each of the eight possible states.

The current path for an X or Y drive line originates at a solid state circuit which acts as a current source and will be called a "switch." The current path terminates in a solid state circuit which acts as a current sink and will be called a "sink." Any X or Y drive line requires two switches and two sinks, as the current must be reversible between the read and write operations.



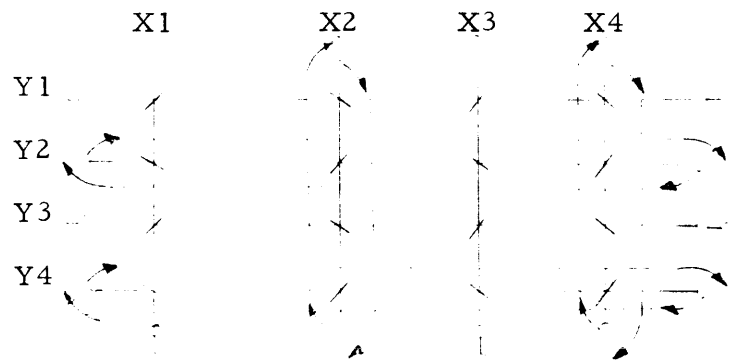
The combination of 8 $\overline{X}$ sw-sink pairs and 8 X sw-sink pairs provides selection of one of 64 X coordinate drive lines, and in a similar manner the 6 bits representing the Y coordinate selects one of 64 Y drive lines. The 4096 intersections of the 64 X drive lines and 64 Y drive lines provide selection of 4096 core storage locations.

Due to the requirement that the X and Y drive currents must reverse between the read phase and the write phase, two separate signals will be required to determine the direction of the drive current. These signals are designated Rw and Wr.

$Rw = Mg\ \overline{Q2}$

$Wr = (Mg\ Q2)\ (Mdt + Q1)$

Due to the fact that the cores alternate in their relative positions to the physical drive lines, the current must pass in the opposite direction for all even-numbered drive lines with respect to all odd-numbered drive lines. To implement this, the external connections on even-numbered lines are reversed as shown below:



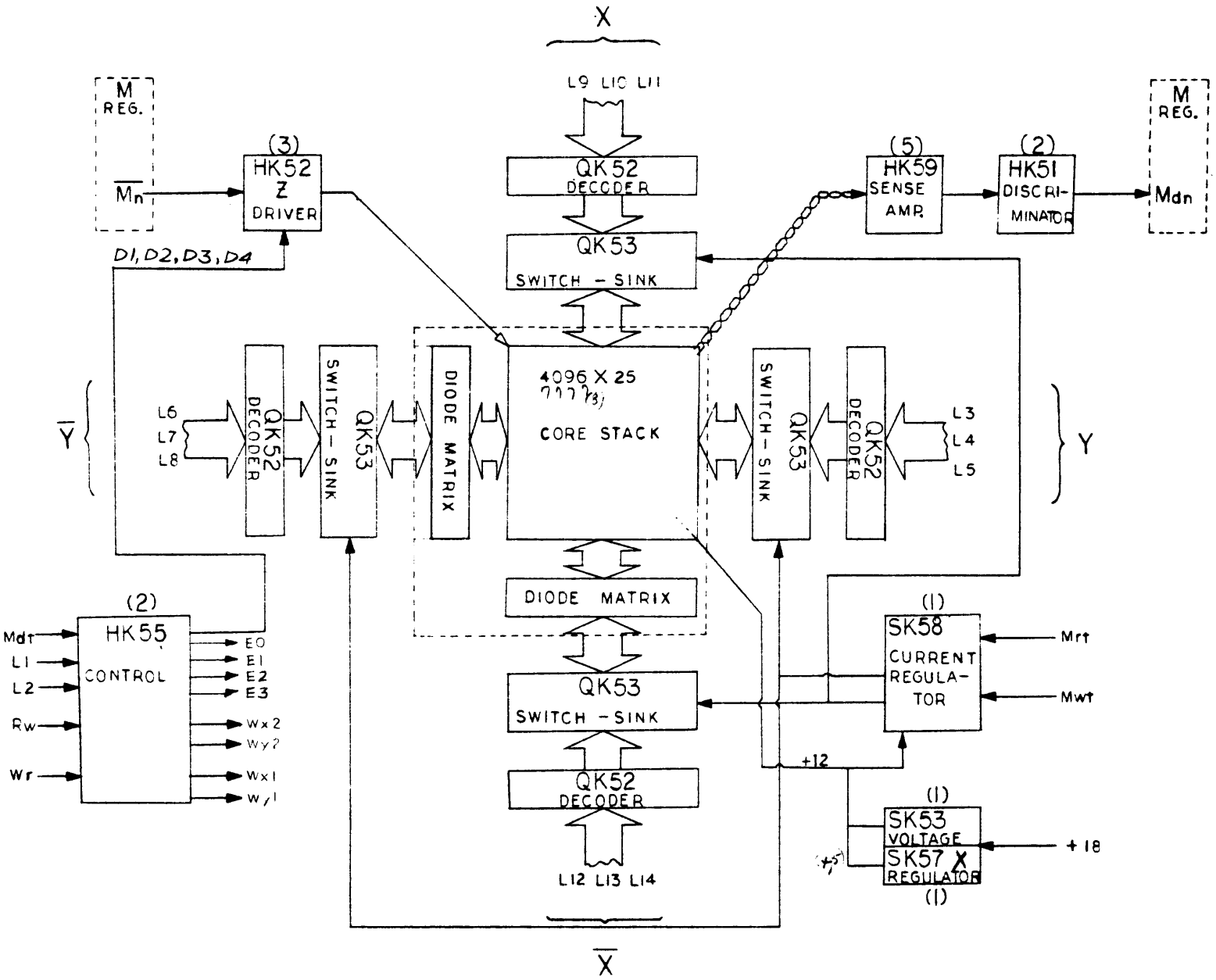


Figure 43. Memory Block Diagram

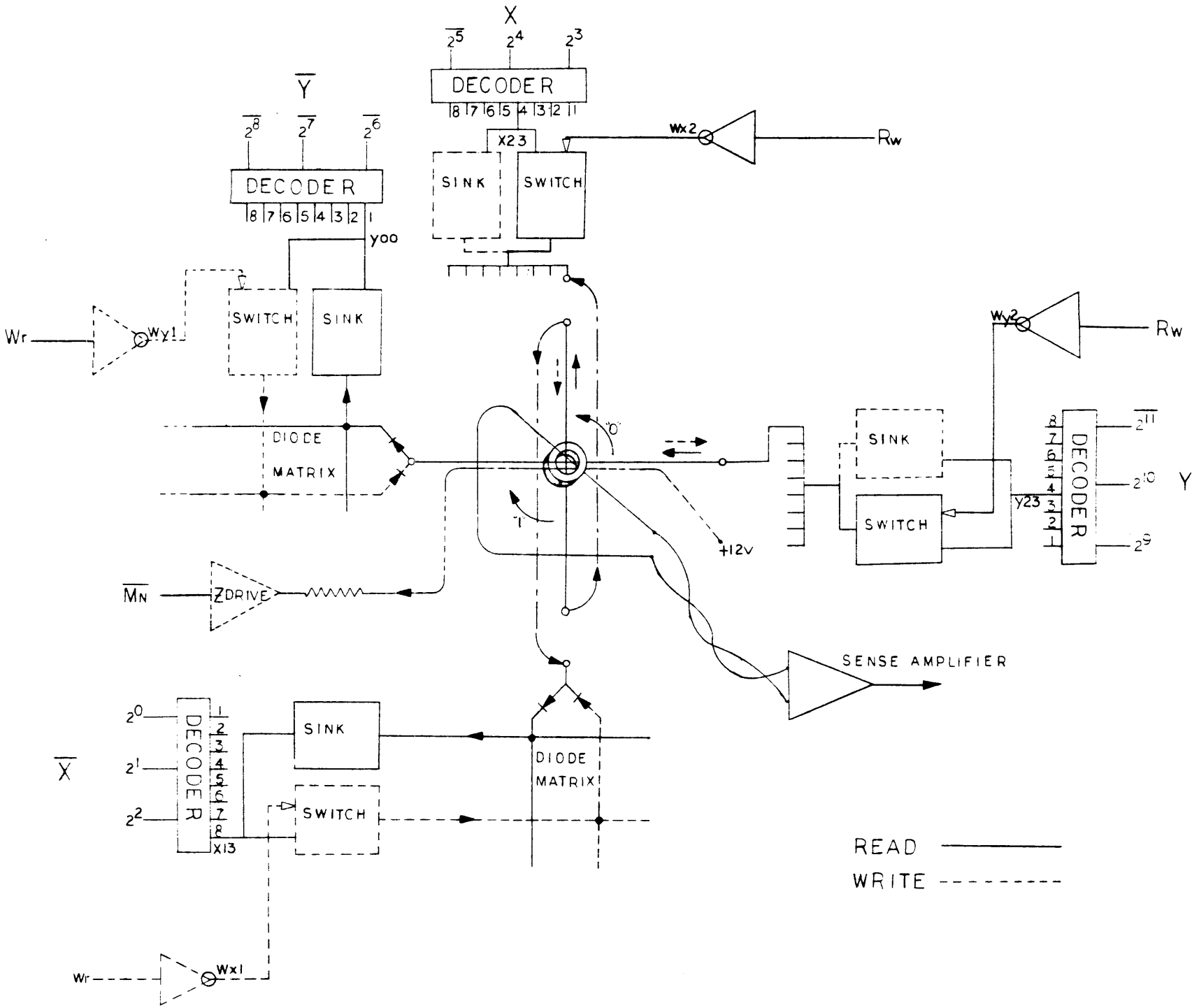


Figure 44. Operation of Address 3037

Figure 44 shows the selection of address 3037 and the control of the direction of the X and Y drive currents by the above-described Rw and Wr terms.

Figure 43 shows the overall structure of an individual 4096-word core stack with the circuit module types and numbers designated. These modules are listed and briefly described below:

Module	Quantity	Function
QK52	4	Decoding of an octal character for address selection.
QK53	4	8 sw-sink combinations for driving the selected line.
HK52	3	Z coordinate, inhibit drivers.
HK59	5	Sense Amplifier.
HK51	2	Discriminator, detects the digital information from the amplifier.
SK58	1	Current regulator, gates and supplies the current to the sinks and switches.
HK55	2	Control, generates and distributes the inhibit terms for 2^{12} and 2^{13} and also distributes the current direction selection terms Wr and Rw.
SK53	1	Voltage Regulator, generates the + 12 voltage for the inhibit lines and the current regulator.
SK57	1	Operates in parallel with the SK53.

Figure 46 shows the core stack in detail. Note that this is approximately a one-sixteenth model in that one-sixteenth the actual number of cores in a single digit plane are shown. It is suggested that the reader trace the directions of current flow, for both the read and write phases of the memory cycle, for the following addresses:

0070	3760
3037	3747

This diagram also shows the inhibit and sense windings. Note that the inhibit winding passes through all cores in the direction of the Y-axis, and that the sense winding traces diagonally through all cores on the digit plane.

READ CYCLE

At pulse time T10 of the computer cycle, the S (Address) register is cleared by Sc in preparation for receiving a new address. At T9 time, a new address is transferred to the S register, the decoders will immediately decode the new information, and the outputs of the decoders will assume a new configuration. At pulse time T8, the M (Memory) register will be cleared by Mc in preparation for

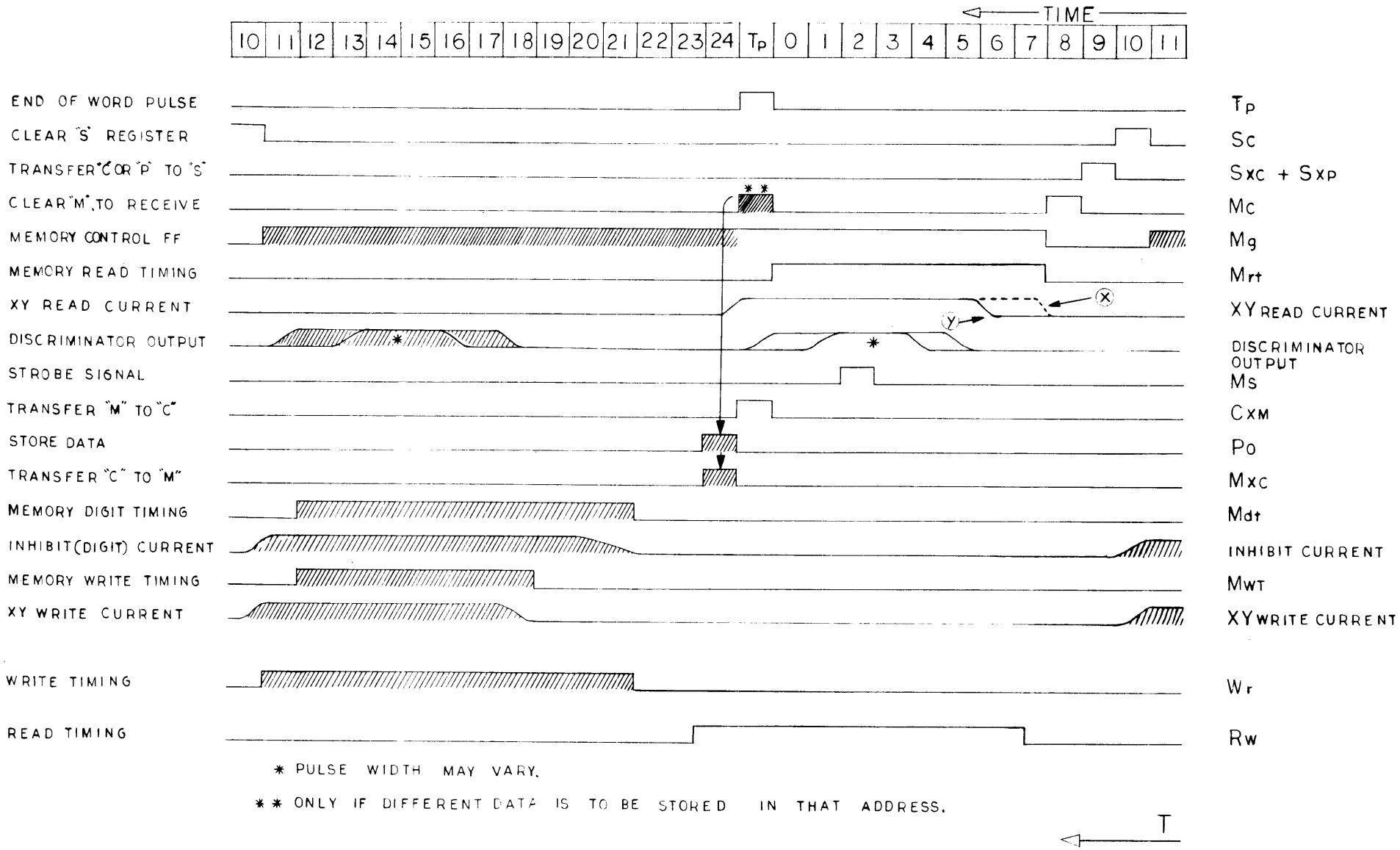


Figure 45. Computer Memory Cycle

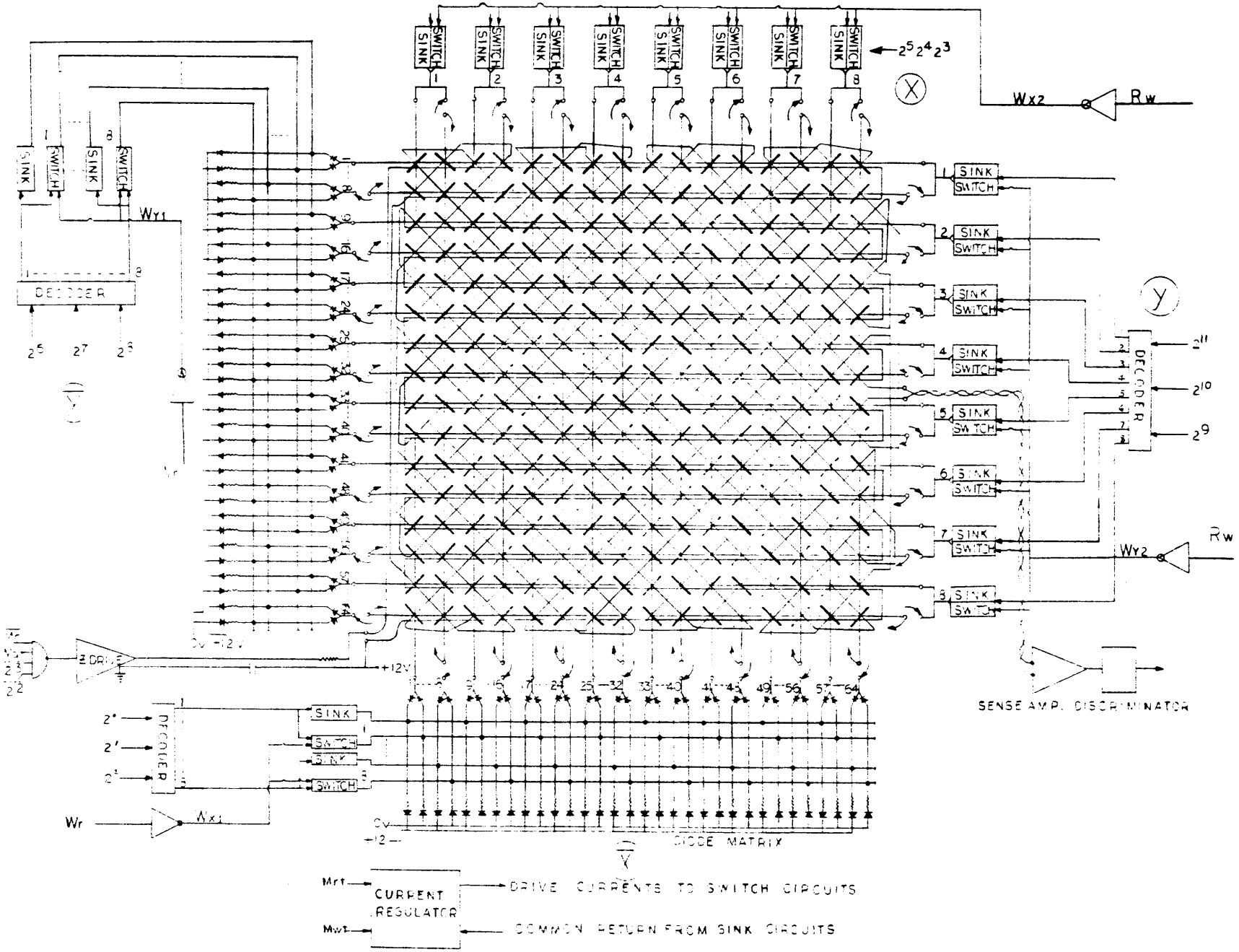
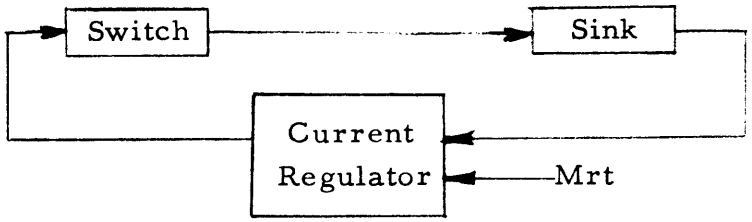


Figure 46. Memory Digit Plane Diagram

receiving data from memory, and the Mg flip-flop will be set, initiating a new memory cycle. When Mg goes true, this will also qualify the current direction signals Rw and Wy1.

At pulse time T7, the memory read timing signal (Mrt) will be enabled for a duration of 8 pulse times. Mrt controls the current regulator circuits which supply the current to all of the switch and sink circuits:



The decoders selected the proper switches and sinks at T7 time, and current is supplied to the drive line by Mrt which, in turn, activates the current regulator.

During the read cycle, all of the X and Y drive line half-currents are applied in a direction which will drive all of the cores in the zero direction. The voltage outputs of the digit planes are sensed by 25 sense amplifiers which feed their respective outputs to 25 discriminator circuits which detect the large signal change; and at T2 time, this information is clocked into the M register by the Ms (strobe) signal.

At Tp time, the Mrt signal is disabled which, in turn, disables the current regulator, and no more current is supplied to the X and Y drive line paths.

WRITE CYCLE

The write cycle consists of regenerating the present contents of M into the address from which it was just read, or of inserting new information into the M register at T24 time for storage into the address that was just read.

In either case, the contents of the S (address) register will not change, as the address must be the same; therefore, the output configuration of the decoders will remain as they were during the read cycle.

At pulse time T21, the Mdt (memory digit timing) signal is enabled. The Mdt signal enables the Z (inhibit) drivers whose inputs are qualified by the $\overline{Mn}$ signal. The current for the Z drivers is not supplied by the current regulator modules, but by the voltage regulator so that inhibit current will flow from T21 until T8.

At T18 time, the Mwt (memory write timing) signal will be enabled which, in turn, will enable the current regulator to supply current to the sinks and switches in a fashion similar to Mrt. Note that during the write phase the address selection is the same, but the direction of all drive line half-currents is reversed due to Wr and Wy2 now being enabled.

Voltages will be generated on the sense lines during the write cycle due to the write

drive currents setting cores to the one position and will be generated from the inhibit currents, but these voltages are not strobed by Ms and are always ignored.

At T12 time, the Mwt signal is disabled and the X and Y drive current sources are inhibited.

MEMORY EXPANSION

The memory of the computer can be expanded to contain 4 core stacks of 4096 words each. Figure 47 shows the addressing system used when expanding the memory to other configurations.

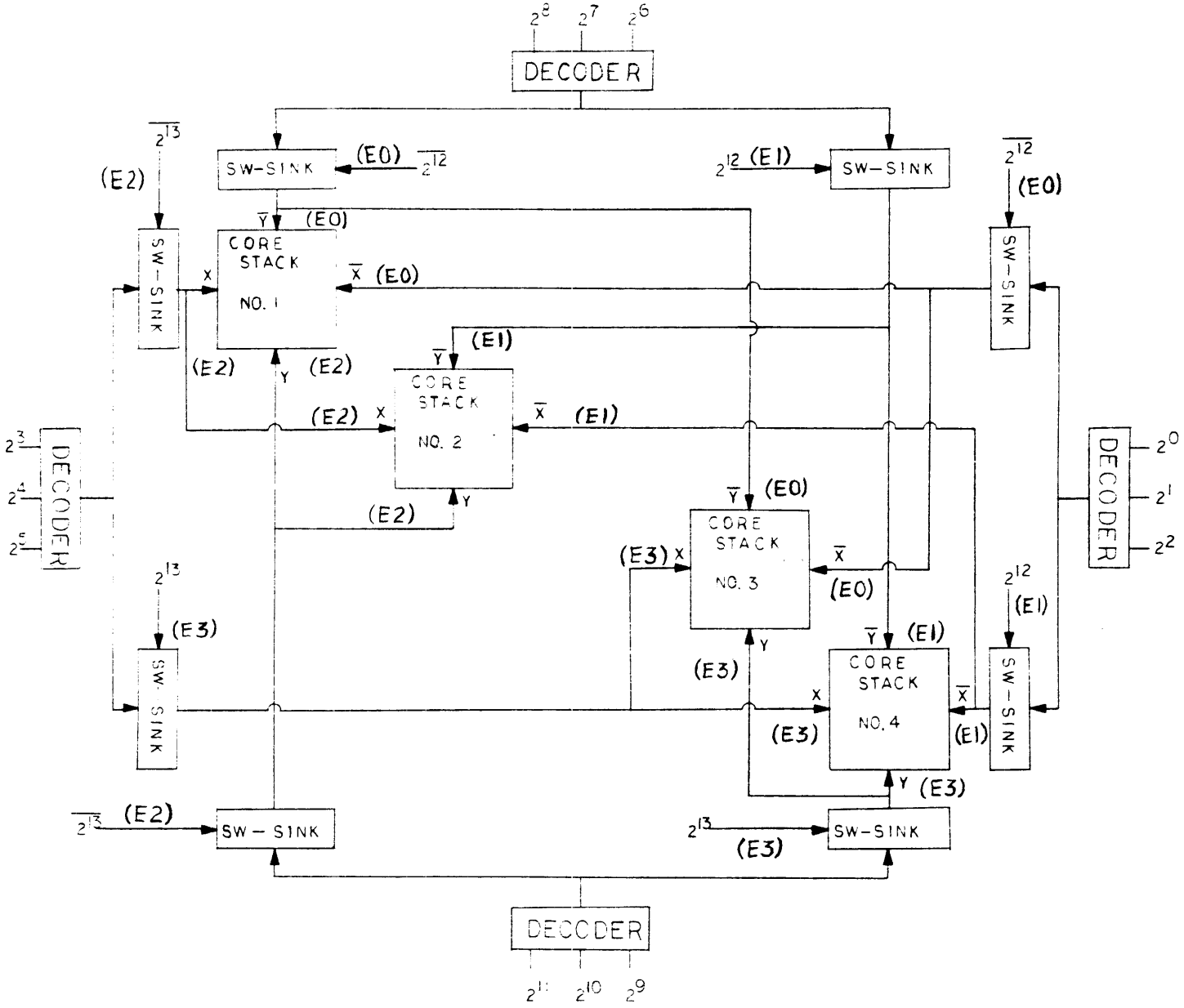


Figure 47. Block Diagram – Memory Expansion

MODULE OPERATION

DECODER

The decoder module, designated QK52, produces an output signal (+) on one of eight output lines to represent one of eight possible input configurations. Both the true and false sides of the three bits to be decoded ($2^n 2^{n+1} 2^{n+2}$) are supplied to the module.

The decoding is performed by a series of negative "AND" gates such that the inverted output will be the correct result. The eight output lines feed to one or more sets of 8 sw-sink modules (depending upon the size of memory) and will act as the enabling address terms.

SELECTOR CONTROL

The selector control module, designated HK55, generates the current direction control terms, the stack selection control terms, and the stack selection terms, for the inhibit drivers.

Two HK55 modules are connected together to generate the required complement of control terms. On the first HK55, two current direction terms are generated from Wr. These are Wx1 and Wy1. On the second HK55, two similar terms, Wx2 and Wy2, are generated from Rw. These four current direction control terms are fed to the four sw-sink modules.

The two most significant computer address terms L1 and L2 are used to generate four stack selection terms E0, E1, E2, and E3. These stack selection terms are coded with Mdt (inhibit or digit timing) to generate four inhibit control signals, designated D1, D2, D3, and D4. The stack selection signals E0, E1, E2, and E3 are also used to enable the sw-sink modules in each of the four stacks.

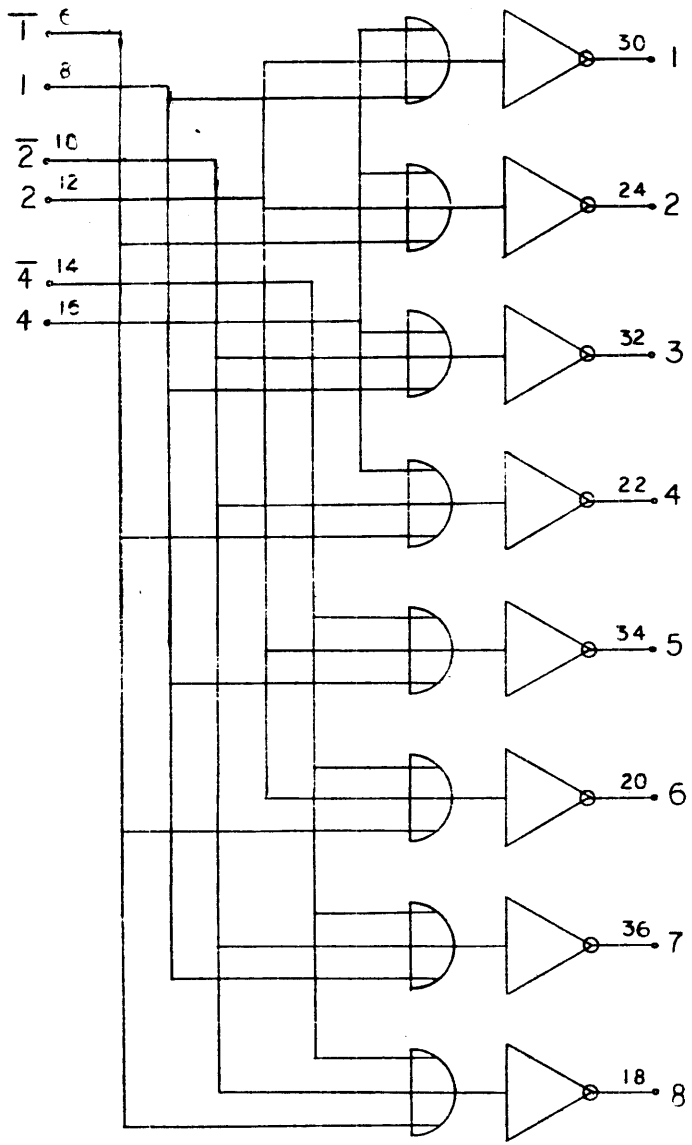
XY SELECTOR

The XY Selector modules designated QK53 contain eight sets of switch-sink circuits for selecting the proper X or Y drive line.

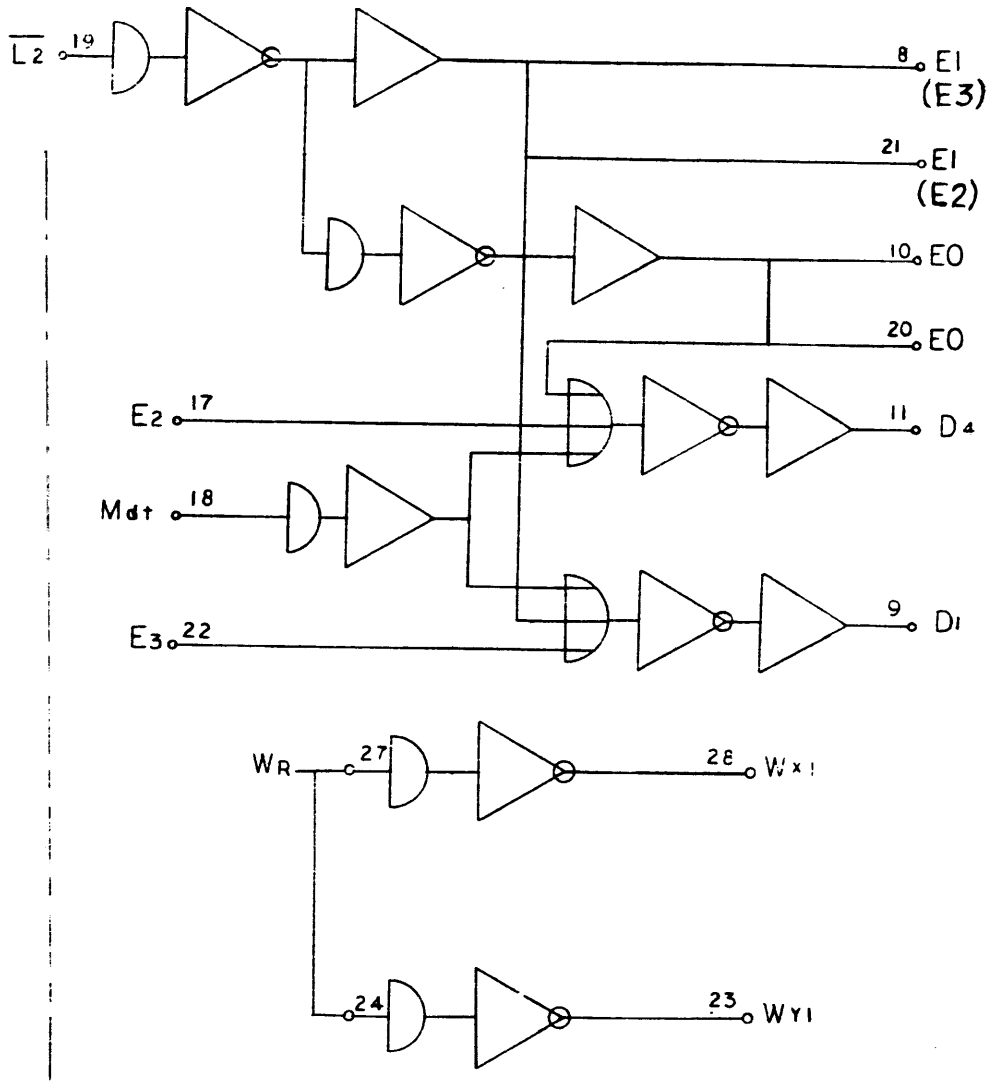
The eight address signals from the respective decoder module are connected to pins 39, 37, 33, 30, 24, 20, 14, and 11, respectively. If the specific address input signal is true (ANDed with the stack zero signal E0), it will enable the switch-sink pair to conduct current when directed by the Rw and Wr signals.

The switch circuit gates the drive current onto the appropriate XY drive line where the current passes through a sink circuit to return to the current regulator.

When the current direction control signals invert (between read and write), the mating sink circuit will be enabled and will receive current from the same XY drive line.

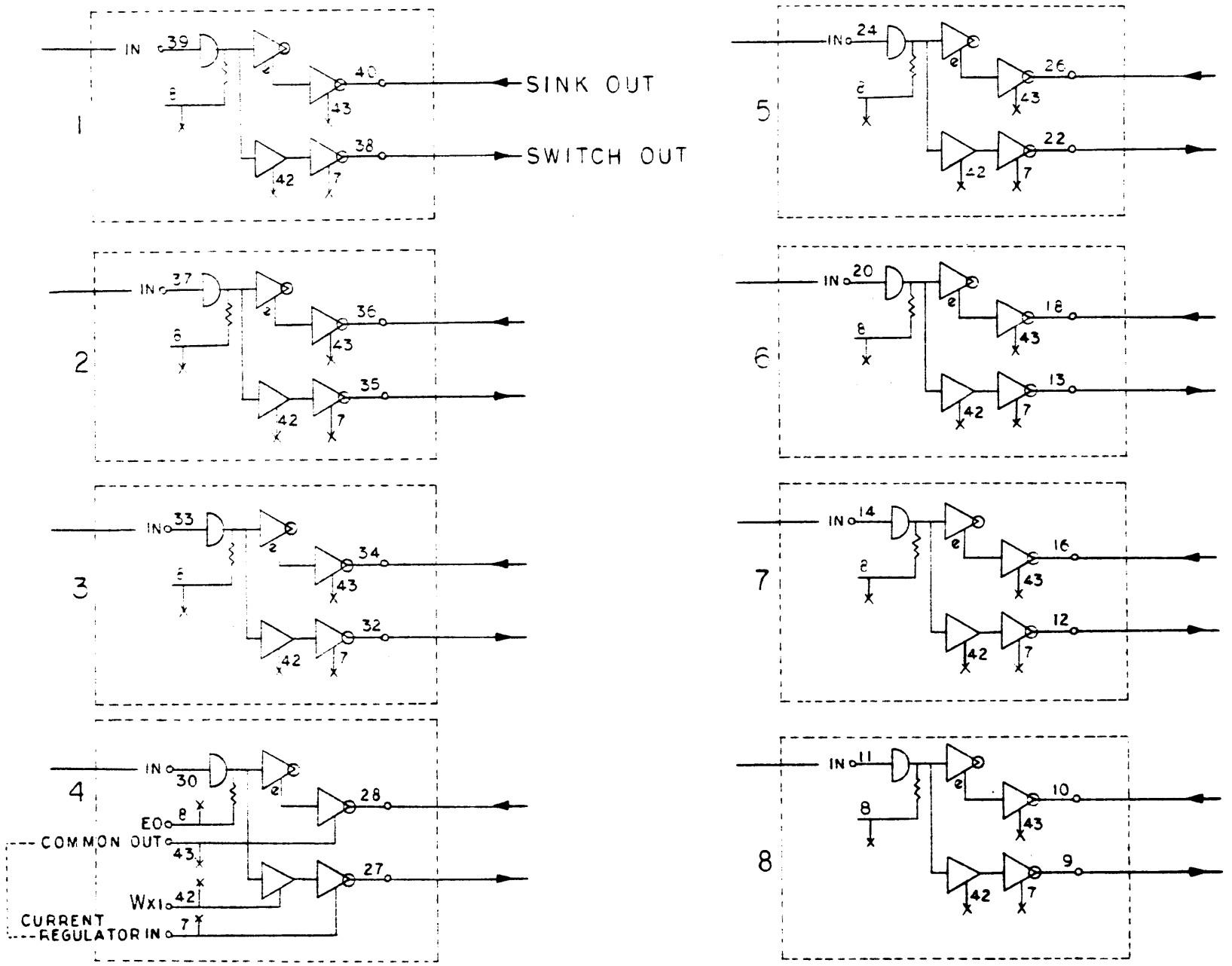


QK 52



HK 55

Figure 48. Decoder and Selector Control Logic



QK53

Figure 49. X-Y Selector

CURRENT REGULATOR

The current regulator module, designated SK58, contains three basic circuits, a voltage regulator, and two gated current stabilizers.

Pins 1, 2, 14, 16, and 18 are inputs to the voltage regulator which, in turn, controls the X and Y current stabilizers.

Pin 1 receives the +12 volt reference signal generated by the outputs of the SK53 and SK57 voltage regulators. This voltage will vary depending upon the amount of inhibit current desired. Pins 2 and 18 receive the +15 volt floating supply signal, which has a +12v. reference point. Pin 14 (control in) is used by the automatic "Start up/shut down" mechanism to inhibit all drive currents when the logical state of the computer is in an ambiguous area. Pin 16 (marginal test) is connected to the Marginal Test switch on the front panel of the computer and is used to increase or decrease the XY drive currents by approximately 6%. Pins 34 and 36 connect the Mrt (memory read timing) and Mwt (memory write timing), respectively. The read timing signal (Mrt) may be delayed and it, in turn, will delay the read current on the Y lines and, consequently, the sense signal.

The Mwt and Mrt signals gate the X and Y current stabilizers. Pins 20 and 22 are the current outputs for the X and Y switch circuits. Pins 9 and 10 are the current returns for the sink circuits. There is a potentiometer adjustment on the SK58 for varying the amount of XY drive current. This current is nominally about 500 ma, but should be adjusted according to the individual characteristics of each unit.

VOLTAGE REGULATOR

The voltage regulator, designated SK53, generates a constant voltage of approximately + 12 v from the + 18 volt line from the power supply. The + 12 v output supplies the inhibit lines and serves as a reference signal to the current regulator module.

The SK57 voltage regulator module operates in parallel with the SK53 and receives its V_R control signal from the SK53. The threshold potentiometer on the SK57 is nominally set between 6 and 6.5 v, but is optimally adjusted according to the amount of noise on the sense lines. There is a potentiometer adjustment on the SK53 that determines the magnitude of the + 12 output which, in turn, determines the amount of current that will flow through the inhibit (Z) windings. This is nominally about 200 ma, but should be adjusted according to individual stack characteristics.

Z DRIVER

The Z driver, designated HK52, drives the inhibit winding during the write cycle. The false output of the respective stage of the M (Memory) register is connected to the input. Depending upon the stack number, the input gate will be enabled by one of the stack selection digit signals D1, D2, D3, or D4.

During Mdt time, current will now flow through the inhibit winding.

SENSE AMPLIFIER

The sense amplifier module, designated HK59, is composed of five basic sense amplifiers. The sense amplifier is basically a two-stage, balanced amplifier with

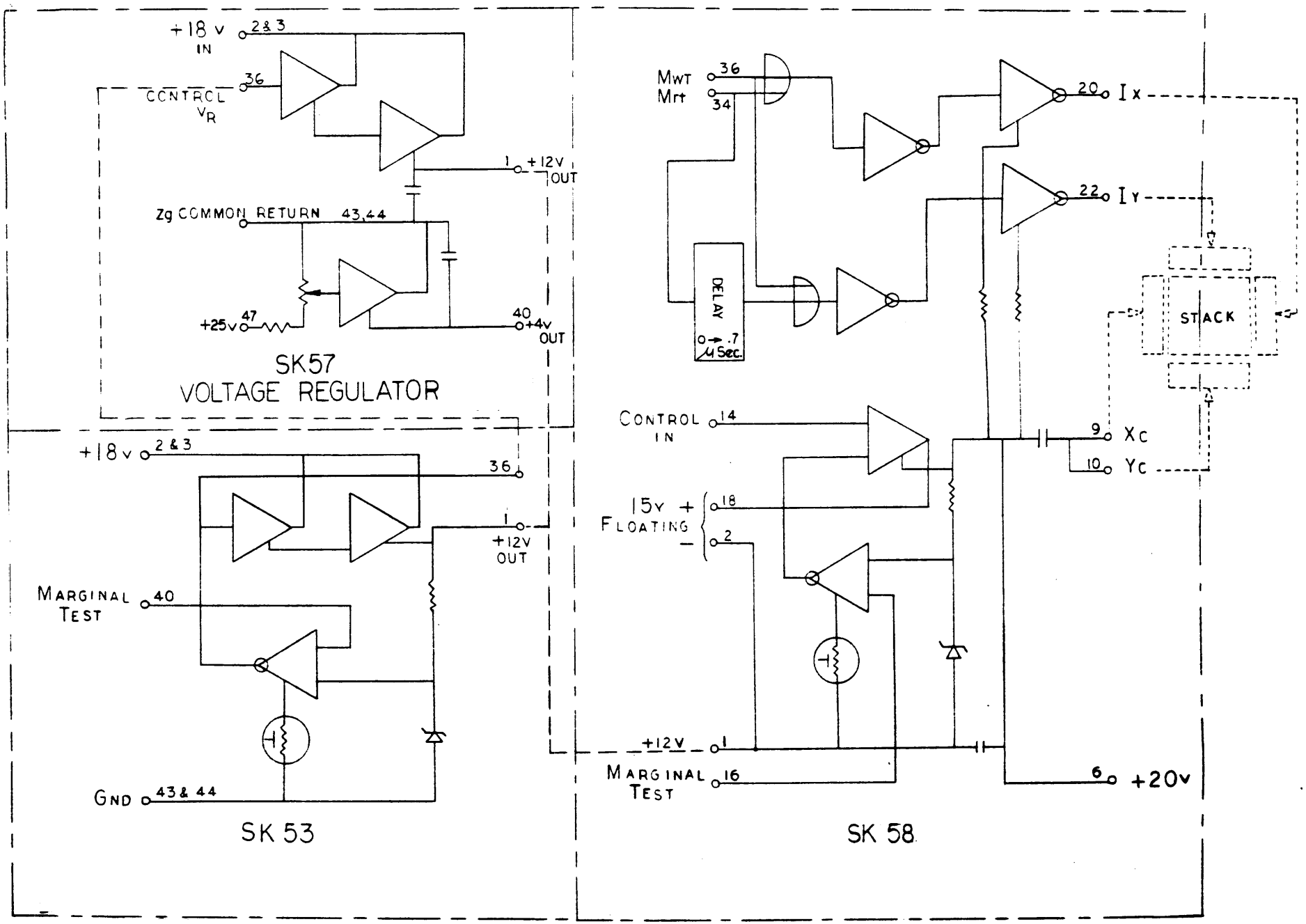


Figure 50. Voltage Regulator and Current Regulator

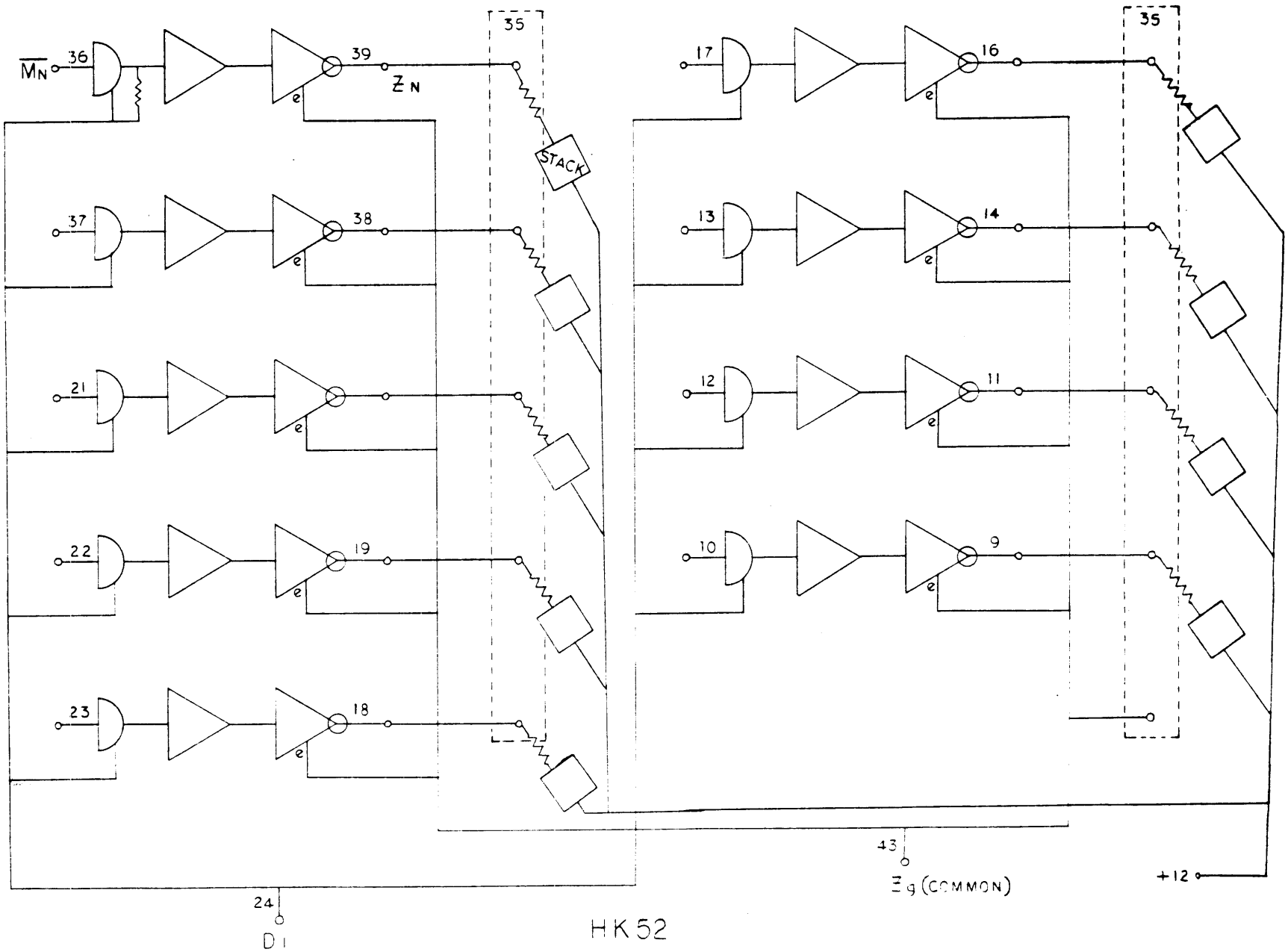


Figure 51. Z-Driver

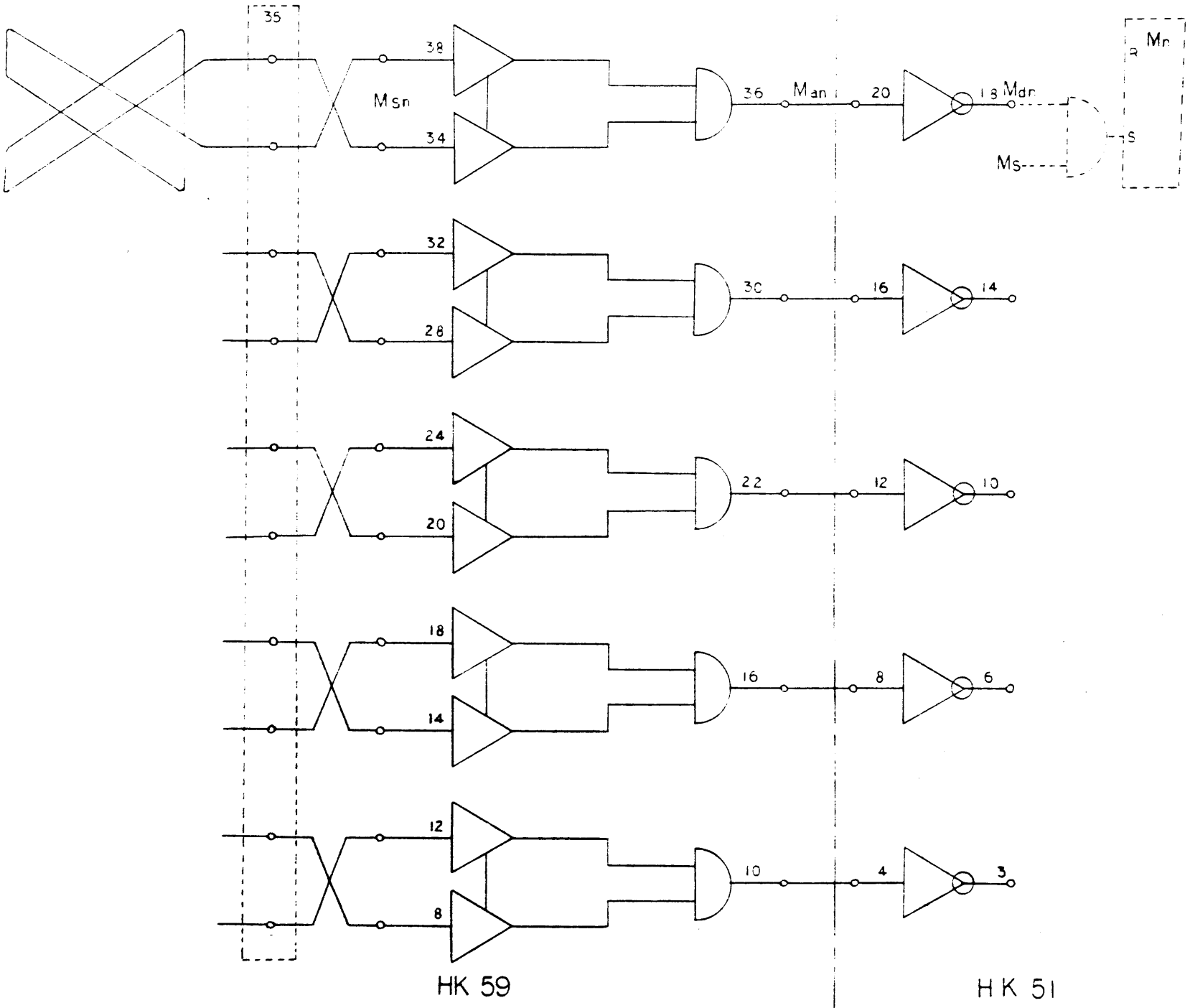


Figure 52. Sense Amplifier and Discriminator Logic

rectified output. The output of the sense amplifier is fed to a discriminator.

DISCRIMINATOR

The discriminator module, designated HK51, contains 13 separate, single-stage, inverter circuits whose inputs are biased to discriminate against the noise content of the signal from the sense amplifier.