# LOGICAL DESIGN FOR FAST SERIAL COMPUTER

TECHNICAL REPORT NO. ESD-TR-65-81



JULY 1965

H. B. Enderton

Prepared for

DIRECTORATE OF COMPUTERS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 508

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts

Contract AF19(628)-2390

# LOGICAL DESIGN FOR FAST SERIAL COMPUTER

TECHNICAL REPORT NO. ESD-TR-65-81

JULY 1965

H. B. Enderton

Prepared for

DIRECTORATE OF COMPUTERS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts


Project 508

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts

Contract AF19(628)-2390

LOGICAL DESIGN FOR FAST SERIAL COMPUTER


ABSTRACT

The detailed logical design is given for a serial digital computer using a 0.5-microsecond magnetic memory, 100-mc logical circuits, and one-word delay lines. The computer performs most instructions in 1 microsecond. A list is given of the amount of hardware used.


REVIEW AND APPROVAL

This technical documentary report has been reviewed and is approved.

James A Mellin Major USAF

SEYMOUR JEFFERY
Major, USAF
Chief, Computer Division

CONTENTS



LIST OF FIGURES

LIST OF FIGURES (Cont.)

LIST OF FIGURES (Cont.)

# LOGICAL DESIGN FOR FAST SERIAL COMPUTER

## SECTION I

## INTRODUCTION

This report presents a design for a fast serial digital computer. This design was made to answer the questions:

What is currently feasible for this type of computer?

What characteristics might such a machine have?

The logic circuits used are the 100 mc circuits developed at MITRE. The memory is a parallel magnetic storage unit with 0.5-$\mu$s cycle time. Ten delay lines (developed at MITRE) are used for one-word registers.

In Sections II and III, the performance of the computer and the amount of hardware required are summarized. The remainder of the report is devoted to the detailed design.

## SECTION II

## PERFORMANCE

Assume a 100-mc clock rate and a 0.5-$\mu$s memory. Each word consists of 24 usable bits including sign. The time required for typical instructions is:

| | |
|---|---|
| Add | 1 $\mu$s |
| Shift | 1 $\mu$s (either direction, any amount) |
| Multiply | 6.75 $\mu$s |
| Divide | 6.75 $\mu$s |
| Normalize | 1.25 $\mu$s (transfers control if number is zero) |

All other built-in instructions require 1 $\mu$s (load, store, etc.).

Floating point operations are not built-in, but can be done by subroutine. Such subroutines have been written with the following average times:

| | |
|---|---|
| Floating add | 24 $\mu$s |
| Floating multiply | 15 $\mu$s |
| Floating divide | 25 $\mu$s |

Three index registers are provided. A program interrupt feature is included. A parity bit is generated for each word stored in memory. When a word is read from memory, its parity is checked.

There is a single channel which can be used for both input and output, but not both simultaneously. Once initiated, an I/O operation can transfer any given number of words between the I/O device and the memory without attention from the main program. When the I/O operation is complete, a program interrupt signal is generated.

## SECTION III

## COST

Table I contains the number of circuit packages used for each type of logic circuit.

Table I

Circuit Packages

| Circuit | Figure No. in Appendix | AND Gates | OR Gates | FF | I & A* |
|---|---|---|---|---|---|
| A register | 5-8 | 65 | 4 | 10 | 1.5 |
| Arithmetic | 9-11 | 34 | 3 | 6 | 0 |
| Q register | 12-15 | 46 | 2 | 5 | 1 |
| B register | 16 | 12 | 2 | 3 | 1 |
| Shift counter | 17-18 | 41 | 3 | 5 | 1.5 |
| PC | 19-20 | 20 | 2 | 3 | 0.5 |
| Index regs, G | 21-24 | 77 | 3 | 16 | 2 |
| MAS, MA | 25-26 | 56 | 0 | 30 | 3 |
| MIO, MB | 27-32 | 199 | 5 | 56 | 15 |
| PI | 33-34 | 27 | 2 | 7 | 1 |
| IR, decoder | 35-36 | 56 | 2 | 12 | 5.5 |
| Timing | 37-40 | 45 | 1 | 15 | 26.5 |
| I/O | 41-47 | 97 | 4 | 27 | 3.5 |
| GATES | | 775 | 33 | 195 | 62 |
| PKGS | | 388 | 33 | 195 | 62 |
| Logic packages | 678 | | | | |
| Clock pulse packages | 88 | | | | |
| TOTAL | 766 packages | | | | |
| Other: 10 delay lines with drive amplifiers and phase inverters. 1 memory, 0.5-$\mu$s cycle, 25 bits, up to 32 K. | | | | | |

*I represents inverting amplifiers

A represents noninverting amplifiers

## SECTION IV

## ORGANIZATION

The block diagram of the fast serial digital computer is shown in Figure 1. The memory is connected to two registers of flip-flops, MB and MIO. The MIO is also a shift register, and it is here that the parallel to serial conversion takes place.

The A register is the main arithmetic register. The Q register is used in multiply and divide. Both are delay lines, as are the three index registers, XRA, XRB, and XRC.

The MAS and MA are flip-flop registers used for the memory address. The program counter (PC) is a delay line.

The I/O circuitry includes three delay lines, the work counter (WC), address counter (AC), and word buffer (WB). The device buffer (DB) is an eight-bit flip-flop register.

Word format is as follows:

| | | |
|---|---|---|
| Data: | Bits 0-22: | magnitude |
| | Bit 23: | sign (zero for plus) |
| | Bit 24: | guard bit (not usable). |
| | Ones complement is used for negative numbers. | |
| Instructions: | Bits 0-14: | address |
| | Bits 15-17: | index tag |
| | Bits 18-23: | operation code |
| | Bit 24: | guard bit. |

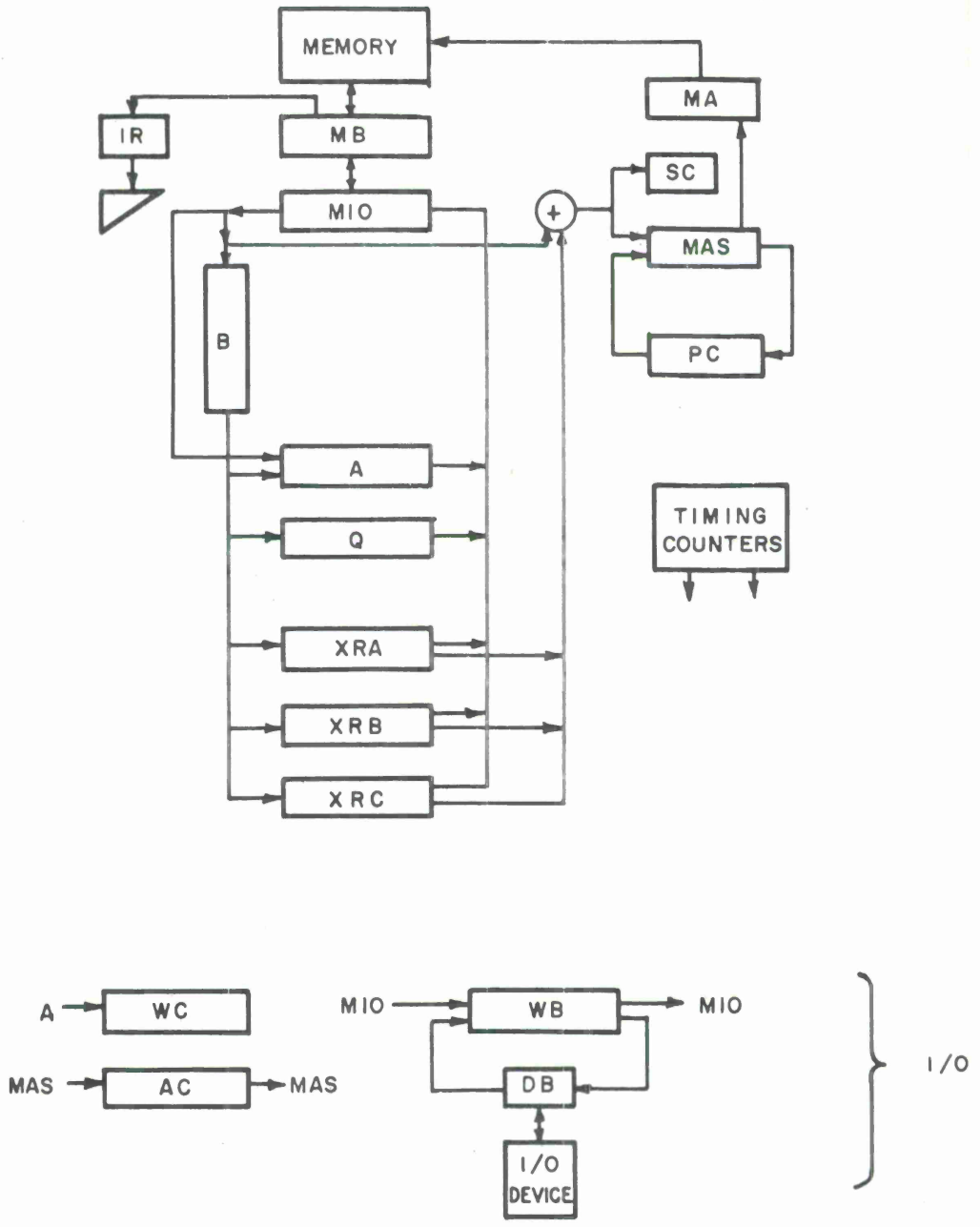



Figure 1. Fast Serial Digital Computer

In memory storage the guard bit is replaced by a parity bit.

One word time is 250 ns. Normally an instruction requires four word times (1 $\mu$s), i.e., two memory cycles. These four word times are designated T1 to T4. Within each word time there are 25 bit times, designated $F_0$ to $F_{24}$. The bit time (Ti and $F_j$) is abbreviated $Ti_j$ in the mechanization equations.

In preparation for starting a computer program, a master reset signal is provided. This signal:

(1) Clears I/0 control flip-flops;

(2) Clears AC, WC, CC, and DB;

(3) Clears $\lambda$ and injects HLT into IR;

(4) Sets the program interrupt controls to accept only the "input complete" interrupt, and clears PI, IC.

(It also destroys the contents of most registers by suspending the t pulses.) A console signal can then set the input status FF (IS) to initiate an input operation. This operation will fill memory, starting with location 0, until an end-of-file signal sets PI, IC. This causes the computer to execute the instruction in location 1.

Instruction List

| | |
|---|---|
| LDX | Load index register |
| LDQ | Load Q register |
| CLA | Clear and add into A |
| CLS | Clear and subtract |
| CAM | Clear and add magnitude |
| SHL | Shift left |
| SHR | Shift right |
| CYR | Cycle right |
| NMT | Normalize and transfer |
| NEG | Complement A |
| LGS | Logical sum |
| LGM | Logical multiply |
| STA | Store A |
| STQ | Store Q |
| STX | Store index |
| STZ | Store zero |
| ADD | Add |
| SUBT | Subtract |
| ADM | Add magnitude |
| SBM | Subtract magnitude |
| MLY | Multiply |
| DIV | Divide |
| TRU | Transfer unconditionally |
| TRN | Transfer if negative |
| TRP | Transfer if positive |
| TRZ | Transfer if zero |
| TROV | Transfer if overflow |
| TRX | Transfer and decrement index |
| TSX | Transfer and set index |
| RPI | Read program interrupt requests |
| SPI | Set program interrupt controls |
| SIO | Start input/output |
| HLT | Halt |
| NOP | No operation |

## Memory Timing

| | Memory | PC/MAS |
|---|---|---|
| **Normal** | | |
| T3 | Read data | MAS → PC if branch |
| T4 | Write data | PC → MAS |
| T1 | Read instruction | PC+1 → PC |
| T2 | Write instruction | Index into MAS |
| **Program Interrupt** | | |
| T3 | Read data | |
| T4 | Write data | 0 → MAS |
| T1 | Clear 0 | PC → MIO |
| T2 | Write PC | 1 → MAS |
| T1 | Read instruction in 1 | PC+1 → PC |
| T2 | Write instruction | Index into MAS |
| **I/O Cycle** | | |
| T3 | Read data | |
| T4 | Write data | AC → MAS |
| T1 | Read I/O word | |
| T2 | Write I/O word | PC → MAS |
| T1 | Read instruction | PC+1 → PC |
| T2 | Write instruction | Index into MAS |

SECTION V

CIRCUIT CHARACTERISTICS

The timing assumption is as follows:

For any signal subject to change, at least every fourth gate must be an AND gate with a clock pulse input. (An exception is the timing levels, for which special methods are used.) The typical configuration is shown in Figure 2.

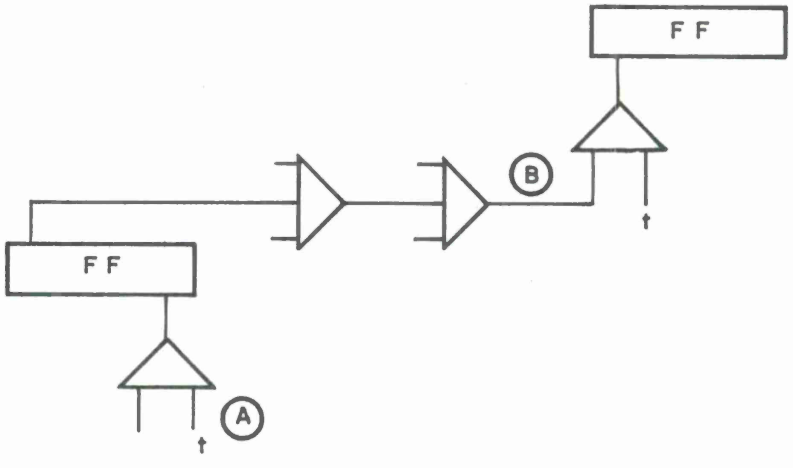



Figure 2. Typical Timing Circuit Configuration

For example if

(1) the clock pulse has a rise time of 1 ns;

(2) the logic circuits have rise and fall times of 2 ns; and

(3) the logic circuits have propagation delays of 2 ns;

then we have the situation in Figure 3.


Figure 3. Signal Propagation Time

The level at B must fall before the pulse at B rises. Thus there is only 0.5 ns available in the above situation for point-to-point propagation, assuming a 100 mc clock rate.

But the assumptions are only a crude approximation to the actual characteristics of the logic circuits. Only a detailed study of the timing situation will determine the maximum permissible clock rate.

The maximum fan-out is six. There is the possibility that this is overly conservative, and that in fact a fan-out of eight (for levels) could be used. The fan-out restriction implies that certain circuits must be duplicated. Duplication is often preferable to using amplifiers, which increase the number of stages of delay.

## SECTION VI

## MECHANIZATION EQUATIONS

The notation used is: When the conditions to the left of the colon are satisfied, then the action specified on the right side is taken. The index of abbreviations is in Section VII.

Memory

1. $(T1_{23})(SI')$ : [ 0 => MIO]

   $(T1_{23})(SI)$ : [ 0 => MB]

   $(T1_{24})(SI')$ : [MB => MIO]

   $(T1_{24})(SI)$ : [MIO => MB]

   $(T1_{24})$ : [ 1 → WR] [ 0 → SI] Rewrite instruction

2. $(T2_{23})$ : [ 0 => MA]

   $(T2_{24})$ : [MAS => MA] [ 1 → RD] [ 0 => MB] Read data

3. $(T3_{23})(SI')$ : [ 0 => MIO]

   $(T3_{24})(SI')$ : [MB => MIO]

   $(T3_{24})$ : [ 1 → WR] [ 0 → SI] Rewrite data

4. $(T4_{23})(\lambda')$ : [ 0 => MA]

   $(T4_{24})(\lambda')$ : [MAS => MA] [ 1 → RD] [ 0 => MB] Read Instruction

Instruction Access

1. $(T4_{0-14})(IIC')$ : [ PC → MAS]

2. $(T4_{24})(IIC')(\lambda')$ : [ 1 → IPC]

3. $(T2_{0-14})(IIC')$ : [Index adder → MAS]

4. $(T2_{23})(IIC')$ : $[MB_{18-23} \Rightarrow IR]$ Double-line

Note: IIC = IOF + PI + HLT.

Program Interrupt

1. $(T3_{24})(IOF')(IPI')\left\{(PI, IC) + (\Sigma PI)(PIC)\right\}$ : [ 1 → IPI ] [ 1 → PIC]

2. $(PI)(T4_{0-14})$ : [ 0 → MAS]

3. $(PI)(T4_{24})$ : [ 1 → SI ]

4. (PI)(TI) : [PC → MIO]

5. $(PI)(T2_{22})$ : [ 1 → λ]

   $(PI)(T2_{23})$ : [ 1 → $MAS_0$] [ NOP => IR] [ 1 → T] [ 0 → λ]

   $(PI)(T2_{24})$ : [ 0 → PIC ]

Note: PI = (IPI)(PIC)

Note: The instruction in location 1 should be a TRU, since the above does not alter PC. The instruction whose address is stored in location 0 has not yet been executed.

Input

1. $(PI')(IDR)(T2_{23})$ : [ 1 → ICF]

2. (ICF)(T3)(CCF) : [DB → WB] [ 0 → DB]

3. $(ICF)(T3_{23})$ : [ CC+1 → CC] [ 0 → IDR] [ 0 → ICF]

   $(ICF)\ (T3_{23})(CC=2)$ : [ 1 → IF] [ 1 → IAC, DWC]

Output

1. $(PI')(ODR)(T2_{23})$ : [ 1→OCF]

2. (OCF)(T3)(CCF) : [WB → DB]

3. $(OCF)(T3_{23})$ : [ CC+1 →CC] [ 0 → ODR] [ 0 →OCF]

   $(OCF)(T3_{23})(CC=2)(WC\neq 0)$ : [ 1 →OF]

   $(OCF)(T3_{23})(CC=2)(WC=0)$ : [ 1 → PI, OC] [ 0 → OS]

Note: CC counts modulo 3.

$CCF = (CC=O)(F_{0-7}) + (CC=1)(F_{8-15}) + (CC=2)(F_{16-23})$

Note: Device sets IDR or ODR.

Clearing IDR or ODR allows device to proceed to fill or read DB.

Note: DB shift signal = (T3)(CCF)(ICF + OCF).

Input/Output

1. $(IOF)(T4_{0-14})$ : [AC → MAS]

2. $(IF)(T4_{24})$ : [ 1 → SI]

3. (IF)(T1) : [WB → MIO]

4. $(IOF)(T2_{0-14})$ : [ PC → MAS]

5. (OF)(T2) : [ MIO → WB]

6. $(IOF)(T2_{22})$ : [ 1 → λ ]

   $(IOF)\ (T2_{23})(HLT')$ : [NOP → IR]

7. $(IOF)(T2_{23})$ : [ 1 → T] [ 0 → λ ]

   $(IOF)(T2_{24})$ : [ 0 → IOF]

$(IF)(T2_{24})(WC=O)$ : [ 0 → IS] [ 1 → PI, IC]

$(OF)(T2_{24})$ : [ 1 → IAC, DWC]

$(IOF)(T2_{24})(HLT')$ : [ 1 → IPC]

Miscellaneous

1. $(T4_{24})(\lambda')$ : [ 0 → IMS]
2. (T4) : [MIO → B] unless specifically overruled
3. $(T1_{24})$ : [ 0 → SN]
4. (F23) : [ 0 → SUB]
5. $(T4)(\lambda)$ : [ B → B]

Instructions

LDX

1. (T1) : [B → XRG]

LDQ

1. (T1) : [B → Q]

CLA

1. (T1) : [B → A]

CLS

1. (T4) : [ MIO' → B]
2. (T1) : [ B → A]

CAM

1. $(MB_{23})(T4)$ : [ MIO' → B]
2. (T1) : [ B → A]

SHL

1. $(T2_{24})$ : [1 → SI]

2. (T3) : [A → MIO] [0 → A]

3. $(T3_{23})$ : [1 → IMS]

4. (T4) : [SC-1 => SC]

5. (T4)(SC=0) : [0 → IMS]

6. (T4)(IMS' ) : [MIO → A]

SHR

1, 2, 4 as in SHL.

8. $(T4_{0-23})$(SC=0) : [1 → IMS]

9. (T1) : [MIO → A]

Note: Copies of sign enter from the left.

CYR

1, 2, 4, 8, 9 as above.

7. (T4) : [MIO → MIO]

Note: The three above instructions shift one place more than the number in the address field.

NMT Normalize and Transfer

1. $(T2_{24})$ : [-0 => SC] [1 → SI]

2. $(T3_{0-14})(A_Z)$ : [MAS → PC]

3. (T3) : [A → MIO]

4. $(T3_{0-22})(A \neq A_{SN})$ : [-0 => SC]

   $(T3_{0-22})(A = A_{SN})$ : [SC-1 => SC]

5. $(T3_{23})$ : [1 → IMS]

6. $(T4_{0-4})(\lambda')$ : [SC → A] [SC' → SC]

   $(T4_{5-24})(\lambda')$ : [1 → A]

7. $(T4_{22})(\lambda')$ : [1 → λ]

8. $(T4_{24})$(SC=0) : [0 → IMS]

   (T4)(λ)(SC=1) : [0 → IMS]

9. (T4)(λ) : [SC-1 => SC]

10. (T4)(IMS) : $[MIO_{23} \rightarrow Q]$

    (T4)(IMS') : [MIO → Q]

11. $(T4_{22})(\lambda)$ : [0 → λ]

Note: A contains the negative of the number of shifts. The normalized number (if non-zero) is in Q.

NEG

1. (T3) : [A' → A]

LGS Logical Sum

1. (T1) : [A ∨ B → A] (inclusive or)

LGM Logical Multiply

1. (TI) : [A · B → A]

STA Store A

1. ($T2_{24}$) : [1 → SI]

2. (T3) : [A → MIO]

3. ($T3_{23}$) : [0 => MB]

4. ($T3_{24}$) : [MIO => MB]

STQ Store Q

1, 3, 4 as above.

2. (T3) : [Q → MIO]

STZ Store Zero

1, 3, 4 as above

2. (T3) : [0 → MIO]

STX Store Index

1, 3, 4 as in STA.

2. (T3) : [XRG → MIO]

ADD

1. ($T4_{24}$) : [$A_{SN}$ → SN]

2. (T1 + T2) : [A ⊕ B → A] Arithmetic sum

3. ($T2_{23}$)(SN = $B_{SN}$ ≠ A ⊕ B) : [1 → OV]

SUBT

0. (T4) : [ MIO' → B]

1, 2, 3 as above.

ADM

0. $(MB_{23})$(T4) : [ MIO' → B]

1, 2, 3 as above.

SBM

0. $(MB_{23}')$(T4) : [ MIO' → B]

1, 2, 3 as above.

MLY

1. $(T3_{23})(A_{SN} \neq MB_{23})$ : [ 1 → SN]

   $(T3_{23})$ : [ -0 => SC]

2. (T4)(λ') : [ 0 → A]

   (TR)(λ')$(A_{SN}')$ : [ A → Q]

   (T4)(λ')$(A_{SN})$ : [ A' → Q]

   (T4)(λ')$(MB_{23})$ : [ MIO' → B]

3. $(T4_{22})(A_Z')$(λ') : [ 1 → λ ]

   $(T4_{23})$ : [ SC-1 => SC]

   $(T4_{22})$(SC = 23) : [ 0 → λ ]

4. (λ)$(F_{0-21})$ : [ $Q_T$ → Q]

   (λ)$(F_0)(Q_0)$ : [ A ⊕ B → $Q_+$]

$(\lambda)(F_0)(Q_0')$ : $[A \rightarrow Q_+]$

$(\lambda)(F_0')(Q_0)$ : $[A \oplus B \rightarrow A_T]$

$(\lambda)(F_0')(Q_0')$ : $[A \rightarrow A_T]$

$(\lambda)(F_{22})$ : $[Q_+ \rightarrow Q]$

$(\lambda)(F_{23-24})$ : $[0 \rightarrow Q]$

$(\lambda)(F_{24})$ : $[Q_T \rightarrow Q_0]$ (normal)

5. $(T1)(SN)$ : $[A' \rightarrow A]$ $[Q' \rightarrow Q]$

Note: During (MLY)(λ), the normal $Q_T \rightarrow Q_0$ is suppressed.

DIV

1. $(T3)(A_{SN}')$ : $[A' \rightarrow A][Q' \rightarrow Q]$

$(T3_{23})(A_{SN} \neq MB_{23})$ : $[1 \rightarrow SN]$

$(T3_{23})$ : $[-0 \Rightarrow SC][0 \rightarrow Q_+]$

2. $(\lambda')(MB_{23})(T4)$ : $[MIO' \rightarrow B]$

3. $(\lambda')(T4_{22})$ : $[1 \rightarrow \lambda]$

$(T4_{23})$ : $[SC-1 \Rightarrow SC]$

$(T4_{22})(SC = -23)$ : $[0 \rightarrow \lambda]$

$(\lambda)(F_{24})(A_+')$ : $[1 \rightarrow OV]$ Abort if

$(\lambda)(F_{22})(OV)$ : $[0 \rightarrow \lambda]$ overflow

4. $(\lambda)(SUB)$ : $[A \oplus B \rightarrow A_+]$

$(T4)(SUB')$ : $[A \rightarrow A_+]$

5. $(T4_0)(SC \neq -23)$ : $[Q_H \rightarrow A]$

$(T4_{1-24})(SC \neq -23)$ : $[A_+ \rightarrow A]$

6. $(T4) : [Q \rightarrow Q_+]$

$(T4_0) : [SUB \rightarrow Q]$

$(T4_{1-24}) : [Q_+ \rightarrow Q]$

7. $(T4_{24})(B \leq |A|) : [1 \rightarrow SUB]$

8. $(SC = -23)(SUB) : [A \oplus B \rightarrow A]$

$(SC = -23)(SUB')(\lambda) : [A \rightarrow A]$

9. $(T1)(SN') : [Q' \rightarrow Q]$

Note: Remainder in A is negative.

TRU

1. $(T3_{0-14}) : [MAS \rightarrow PC]$

TRN

1. $(T3_{0-14})(A_{SN}) : [MAS \rightarrow PC]$

TRP

1. $(T3_{0-14})(A_{SN}') : [MAS \rightarrow PC]$

TRZ

1. $(T3_{0-14})(A_Z) : [MAS \rightarrow PC]$

TROV

1. $(T3_{0-14})(0V) : [MAS \rightarrow PC]$

2. $(T3_{24}) : [0 \rightarrow 0V]$

TRX

1. $(T3_{0-14})(XRGZ') : [MAS \rightarrow PC]$

2. $(T3_{24})(XRGZ') : [1 \rightarrow DXRG]$

TSX

1. $(T3_{0-14})$ : [MAS → PC] [ PC → XRG]

RPI Read Program Interrupt Requests

1. $(T3_{0-4})$ : [PIR → A] [ 0 → PIR]

2. $(T3_{5-24})$ : [0 → A]

Note: No FF in PIR should be set during $F_0$ - $F_4$.

Last FF in PIR should not be set during $F_0$ - $F_{22}$.

SPI Set Program Interrupt Controls

1. $(T3_{24})$ : $[MA_0 \rightarrow PIC]$ $[MA_1 \rightarrow IPI]$

Note: The effect of these controls is:

(IPI')(PIC) : Any interrupt will be recognized.

(IPI')(PIC') : Only PI, IC will be recognized.

(IPI)(PIC') : No interrupts will be recognized.

Note: After clearing IPI, at least one instruction is executed before the next interrupt is recognized.

SIO Start input/output

1. (T3) : [MAS → AC] [ A → WC]

2. $(A_Z')(A_{SN})(T3_{23})$ : [ 1 → OS] [ 1 → OF]

$(A_{SN}')(T3_{23})$ : [ 1 → IS]

HLT Halt

1. $(T2_{23})$(Start Button) : [NOP = > IR]

Note: A program interrupt will also start machine.

## SECTION VII

## ABBREVIATIONS

The number in parentheses represents the most relevant illustration.

| | |
|---|---|
| A, $A_{SN}$, $A_T$, etc. | (5). |
| AC. | Address counter. |
| ADD. | An instruction. |
| Add class. | ADD + SUBT + ADM + SBM (36). |
| ADM. | Add magnitude (instruction). |
| B, $B_{SN}$, $B_{24}$. | (16). |
| CAM. | Clear and add magnitude (instruction). |
| CC. | Character count (46). |
| CCF. | (46). |
| CLA. | Clear and add (instruction). |
| CLS. | Clear and subtract (instruction). |
| COA. | Complementary output amplifier (4). |
| CYR. | Cycle right (instruction). |
| DA. | Drive amplifier (4). |
| DB. | Device buffer (47). |
| DIV. | Divide (instruction). |
| DL. | Delay line (4). |
| DWC. | Decrease word count (45). |
| DXR(A). | Decrease index register (A) (21). |
| EOF. | End of file (34). |
| $F_i$. | Timing function number i (38). |
| FF. | Flip-flop (4). |
| G. | (22). |
| HLT. | Halt (instruction). |

| | |
|---|---|
| I. | Inverter (4). |
| IAC. | Increase address counter (44). |
| ICF. | Input control FF (41). |
| IDR. | Input data ready (41). |
| IF. | Input flag (41). |
| IIC. | Inhibit instruction control (35). |
| IMS. | Inhibit MIO shift (30). |
| IOF. | Input/output flag (41). |
| IP. | Input parity (31). |
| IPC. | Increase program counter (19). |
| IPI. | Inhibit program interrupt (33). |
| IR. | Instruction register (35). |
| IS. | Input status (42). |
| LDX. | Load index (instruction). |
| LDQ. | Load Q register (instruction). |
| LGM. | Logical multiply (instruction). |
| LGS. | Logical sum (instruction). |
| MA. | Memory address register (26). |
| MAS. | Memory address storage (26). |
| MB. | Memory buffer register (28). |
| MIO. | Memory input-output register (28). |
| MLY. | Multiply (instruction). |
| MRE. | Master reset signal. |
| NEG. | Negate (instruction). |
| NMT. | Normalize and transfer (instruction). |
| NOP. | No operation (instruction). |
| OCF. | Output control FF (41). |
| ODR. | Output data ready (41). |

| | |
|---|---|
| OF. | Output flag (42). |
| OP. | Output parity (31). |
| OS. | Output status (42). |
| OV. | Overflow (10). |
| PC. | Program counter (19). |
| PI. | Program interrupt level (33). |
| PIC. | Program interrupt control (33). |
| PIR. | Program interrupt register (34). |
| Q, $Q_H$, etc. | (12). |
| RD. | Read signal (29). |
| RPI. | Read program interrupt requests (instruction). |
| SBM. | Subtract magnitude (instruction). |
| SC. | Shift counter (17). |
| Shift. | SHL + SHR + CYR (36). |
| SHL. | Shift left (instruction). |
| SHR. | Shift right (instruction). |
| SI. | Strobe inhibit (32). |
| SIO. | Start input/output operation (instruction). |
| SN. | Sign (10). |
| SPI. | Set program interrupt controls (instruction). |
| STA. | Store A (instruction). |
| Store. | STA + STQ + STX + STZ (36). |
| STQ. | Store Q (instruction). |
| STX. | Store index (instruction). |
| STZ. | Store zero (instruction). |
| SUB. | Subtraction control (33). |
| SUBT. | Subtract (instruction). |
| t. | Clock pulse. |

| | |
|---|---|
| Ti. | Timing level (39). |
| TP. | Test parity (31). |
| TRN. | Transfer if A is negative (instruction). |
| TROV. | Transfer on overflow (instruction). |
| TRP. | Transfer if A is positive (instruction). |
| TRU. | Transfer unconditionally (instruction). |
| TRX. | Transfer and decrement index (instruction). |
| TRZ. | Transfer if A is zero (instruction). |
| TSX. | Transfer and set index (instruction). |
| WB. | Word buffer (43). |
| WC. | Word counter (45). |
| WCZ. | Word count is zero (45). |
| WR. | Write signal (32). |
| X. | Index signal (23). |
| XAD. | Index added to address (24). |
| XR(A) | Index register (A) (21). |
| XRG. | The index register specified by G. |
| $\alpha$ | Amplifier (2). |
| $\lambda$ | (40). |
| $\emptyset$I | Phase inverter (4). |
| $\Sigma$ PI | (34). |

## SECTION VIII

## FLOATING POINT SUBROUTINES

We give programs for multiply, divide, and add.

### FLOATING MULTIPLY SUBROUTINE

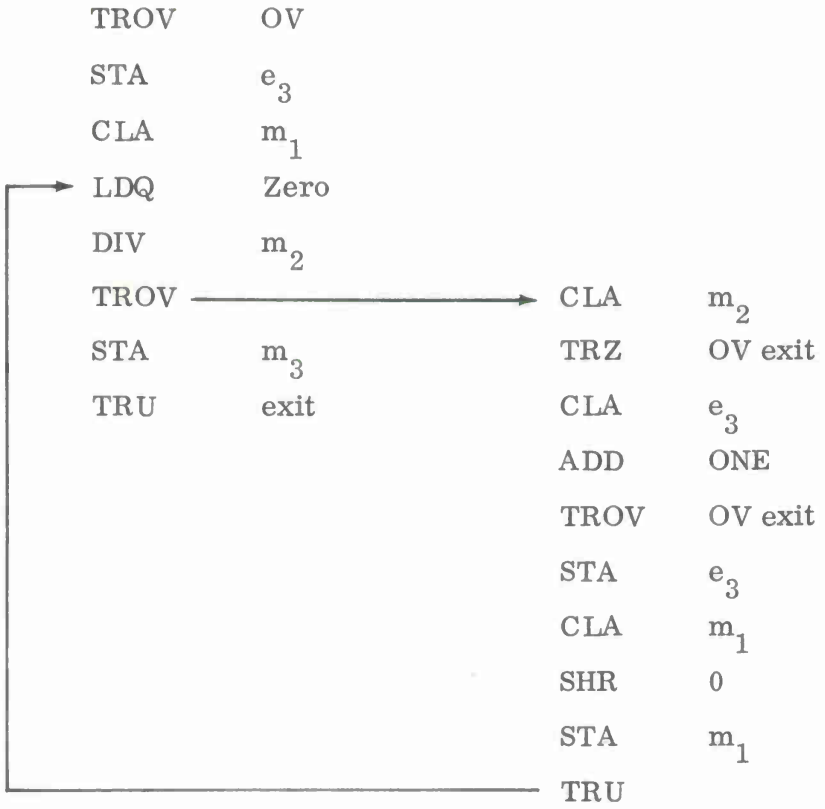
$$2^{e_3} m_3 = \left(2^{e_1} m_1\right)\left(2^{e_2} m_2\right)$$

```
        CLA     m1              Average time 15 μs
        MLY     m2
        NMT     Z
        STQ     m3
        ADD     e1
        ADD     e2
        STA     e3
        TROV    *+2
        TRU     exit
        TRN     OV exit
Z       STZ     m3              (Zero or
        CLA     min. exp.        underflow)
        STA     e3
        TRU     exit
```

### FLOATING DIVIDE SUBROUTINE

$$2^{e_3} m_3 = \left(2^{e_1} m_1\right) / \left(2^{e_2} m_2\right)$$

```
        CLA     e1              Average time 25 μs
        SUBT    e2
```




```
OV      TRN   OV exit
        CLA   min. exp.
        STA   e3
        STZ   m3
        TRU   exit
```

FLOATING ADD SUBROUTINE

$2^{e_3} m_3 = 2^{e_1} m_1 + 2^{e_2} m_2$.

Average time 24 $\mu$s.

```
        CLA   e1
        STA   e3
```

```
XCH     CLA   e2
        STA   e3
```

```
        SUBT    e2
        TROV    α
        TRZ     No shft
        TRN     XCH
        SUBT    "22"
        TRP     No add
        ADD     SHR 21
        STA     * +2
        CLA     m2
        ---
AD      ADD     m1
β       TROV    OV
        NMT     Zero
        STQ     m3
        ADD     e3
        TROV    Zero
        STA     e3
        TRU     exit

α       TRN     No ADD
        CLA     e2
        STA     e3
NA      CLA     m2
        STA     m3
        TRU     exit
No Shft CLA     m2
        TRU     AD
```

```
        SUBT    e1
        SUBT    "22"
        TRP     NA
        ADD     SHR 21
        STA     * +2
        CLA     m1
        ---
        ADD     m2
        TRU     β

No Add  CLA     m1
        STA     m3
        TRU     exit

OV      SHR     0
        TRN     * +3
        LGS     Bit 23
        TRU     * +2
        LGM     Bit 23'
        STA     m3
        CLA     e3
        ADD     ONE
        TROV    OV exit
        STA     e3
        TRU     exit
Zero    CLA     min. exp.
        STA     e3
        STZ     m3
        TRU     exit
```

# APPENDIX

## ILLUSTRATIONS

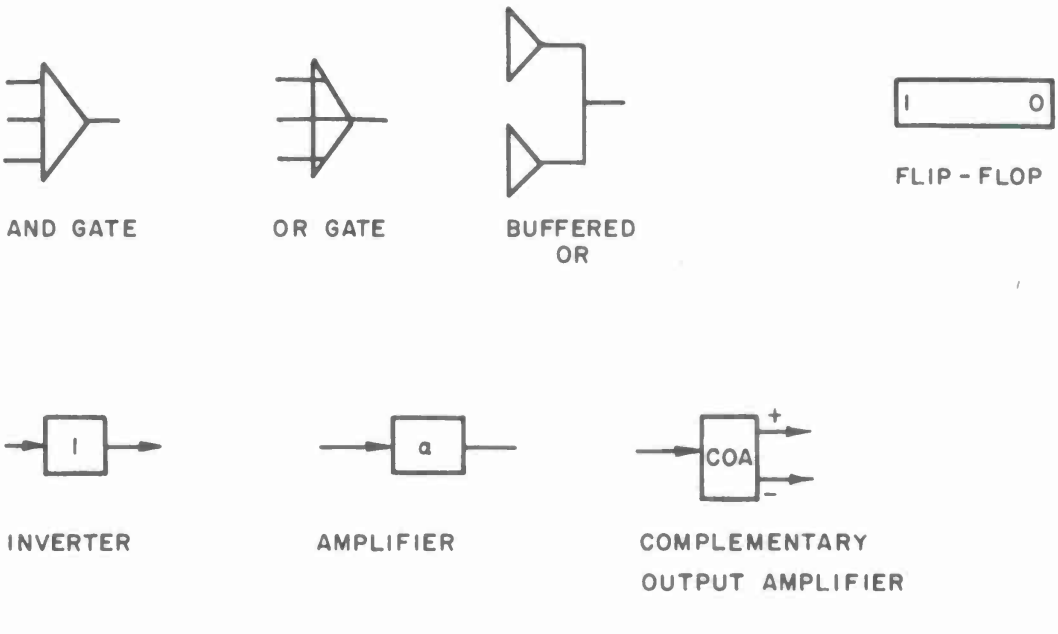
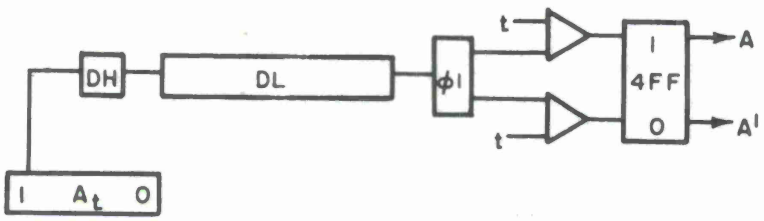
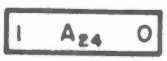


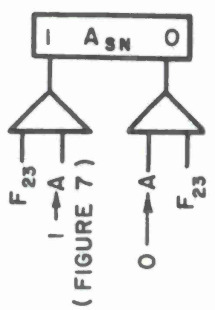

Figure 4. Notation for Logical Diagrams




SEE $A_t$ GATING




SEE A GATING



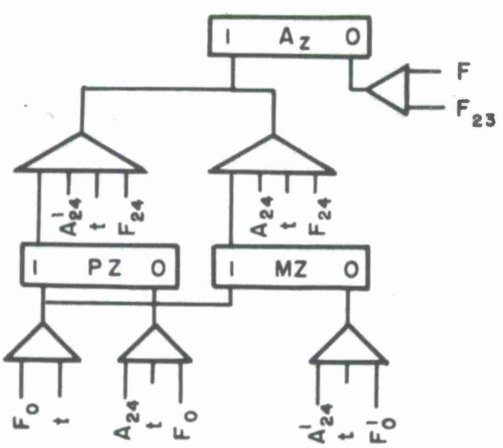




Figure 5. A Register

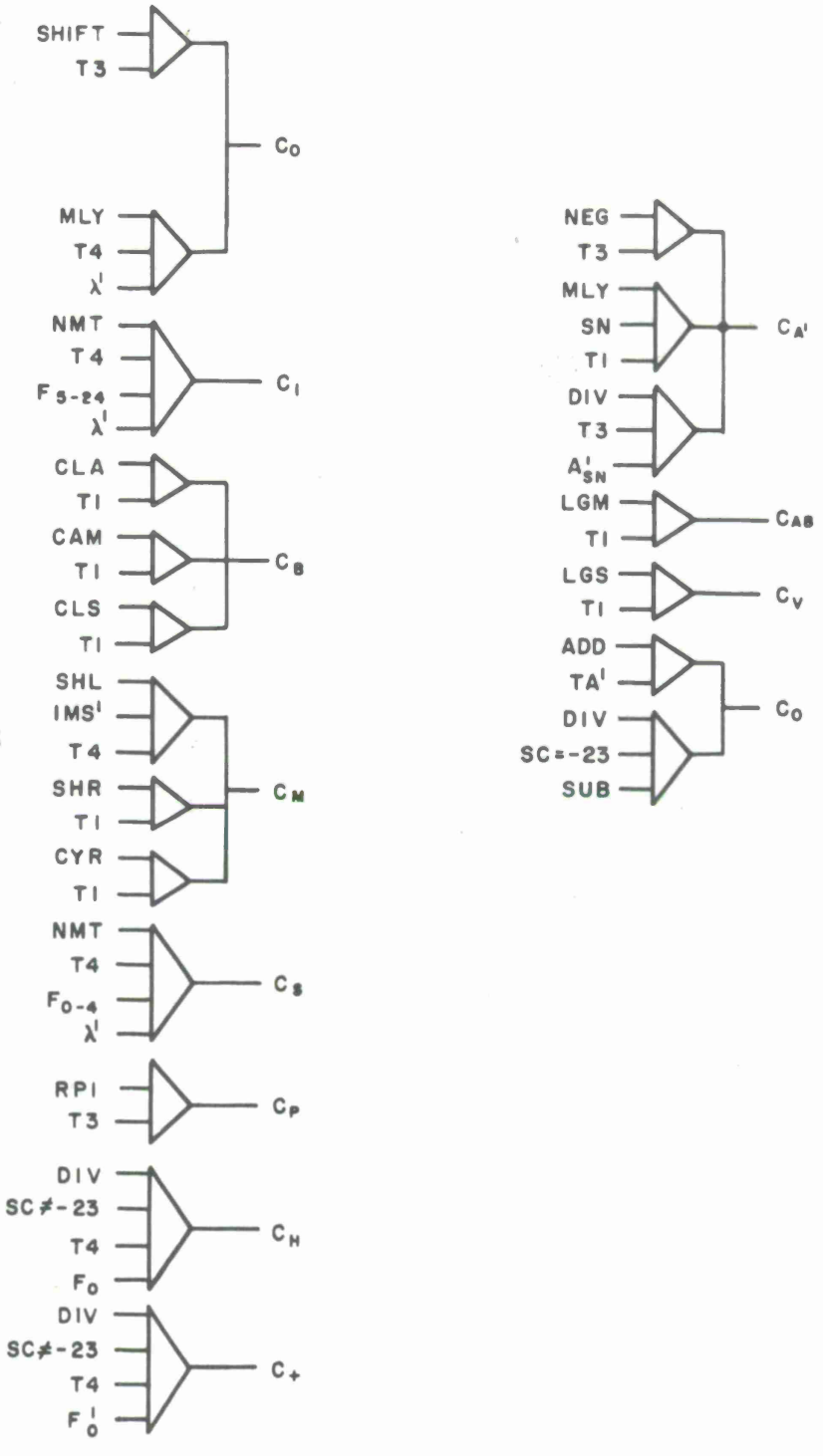



Figure 6. A Gates I

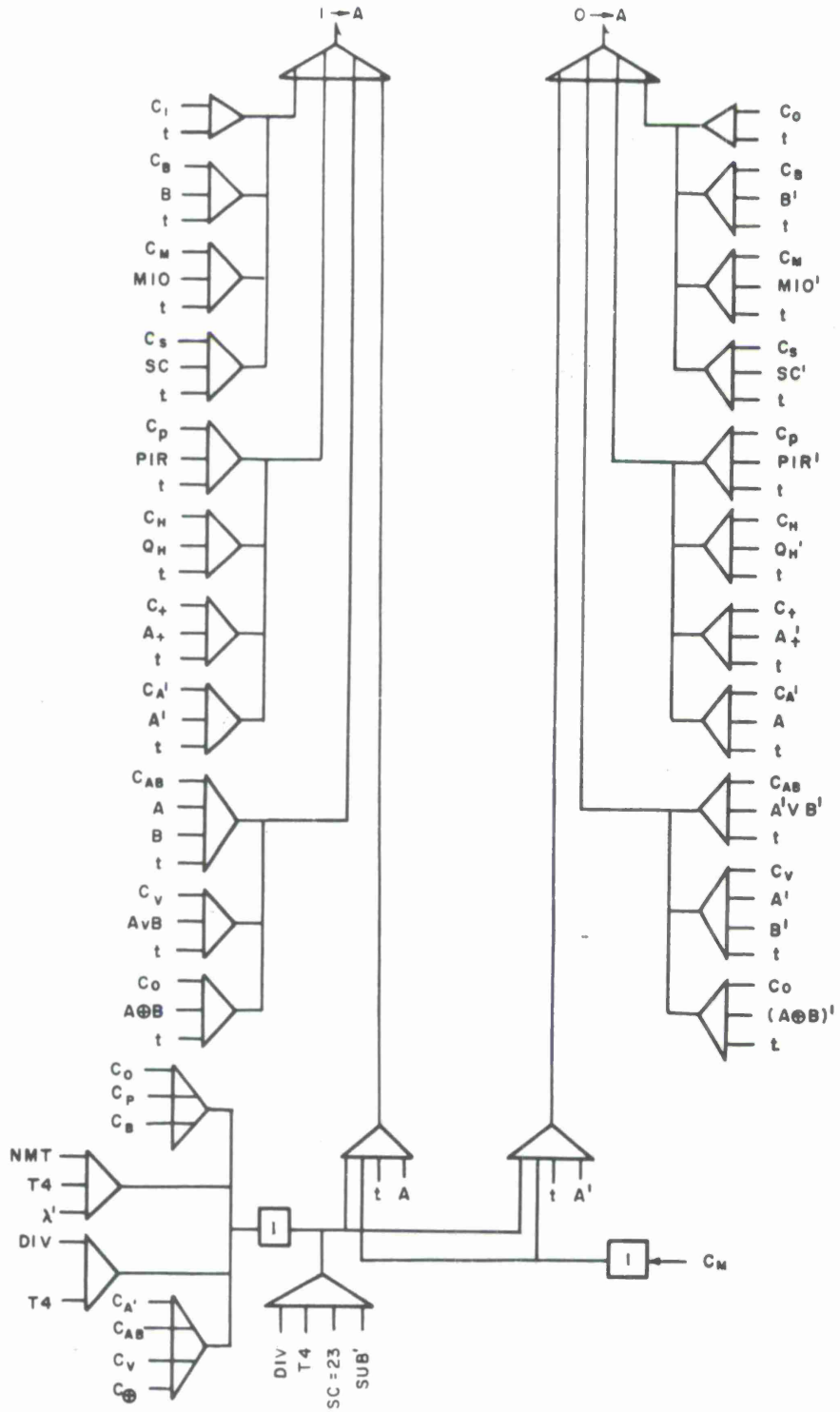



Figure 7. A Gates II

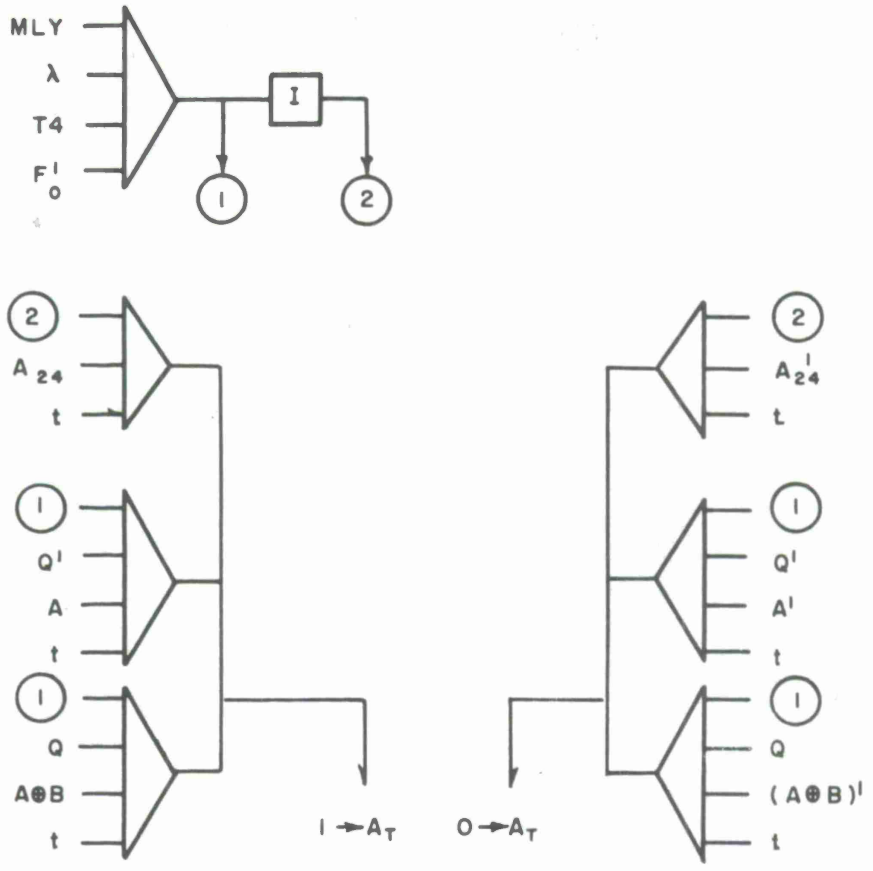
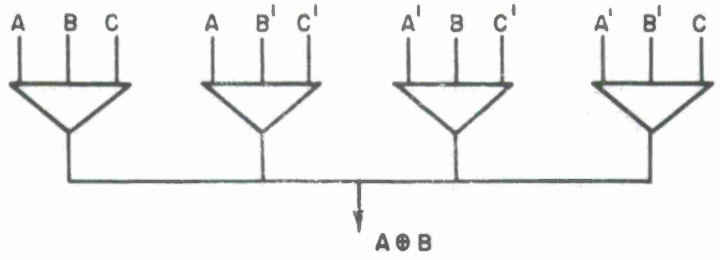
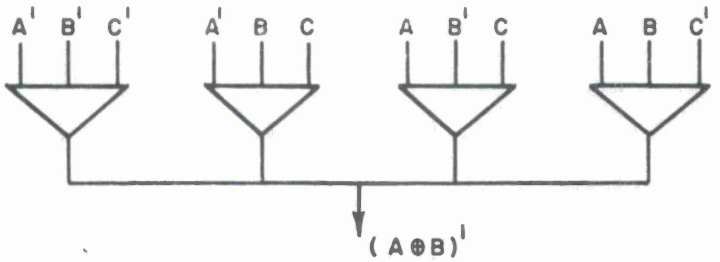
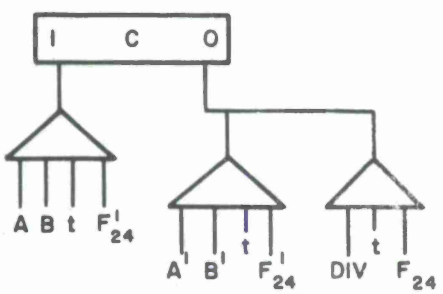
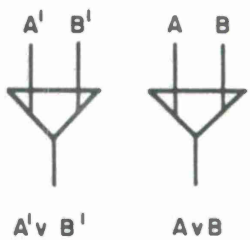



Figure 8. $A_T$ Gating




A ⊕ B IS ARITHMETIC SUM

AvB IS LOGICAL SUM

Figure 9. Adder

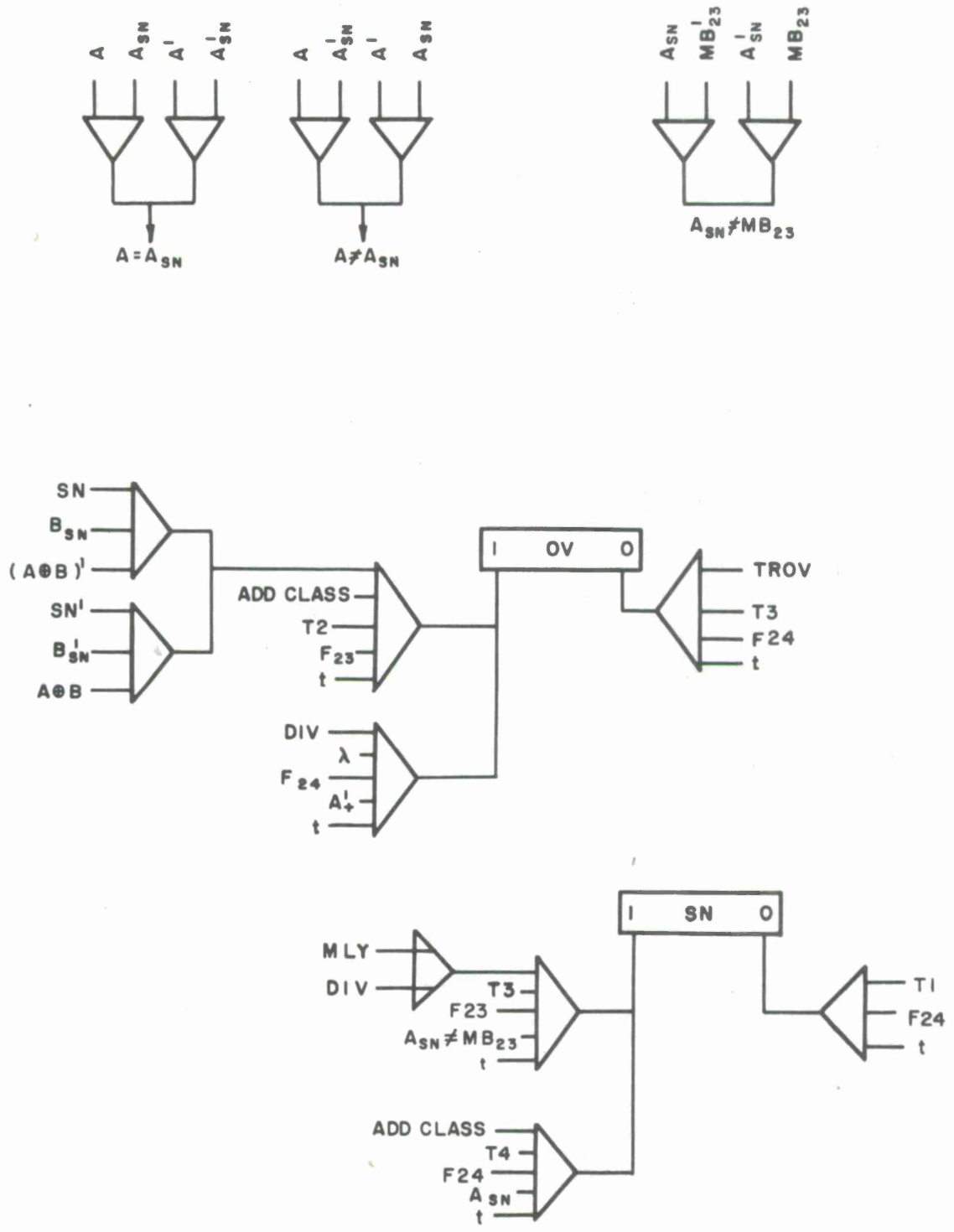



Figure 10. Miscellaneous Controls

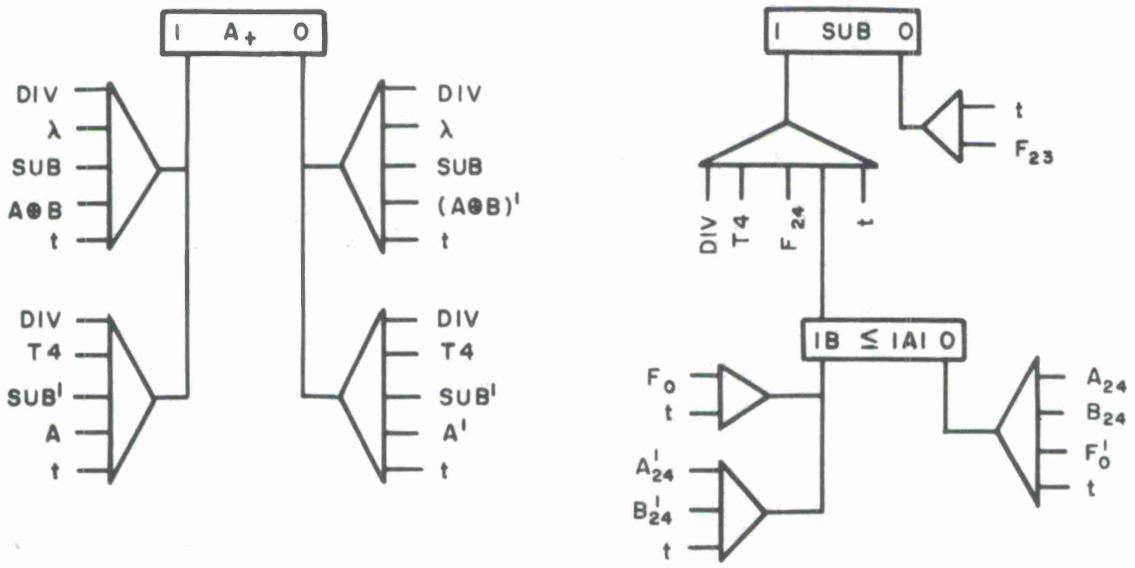



Figure 11. Divide Control

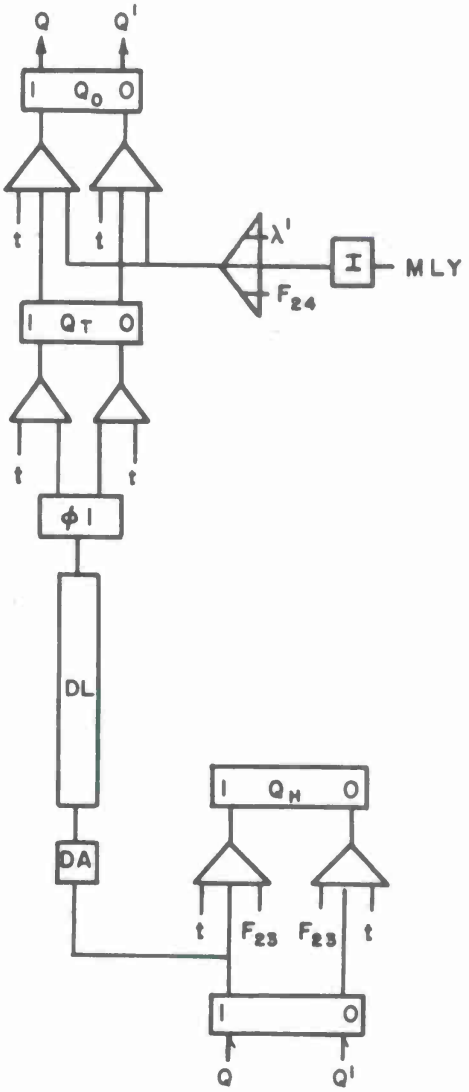



Figure 12. Q Register

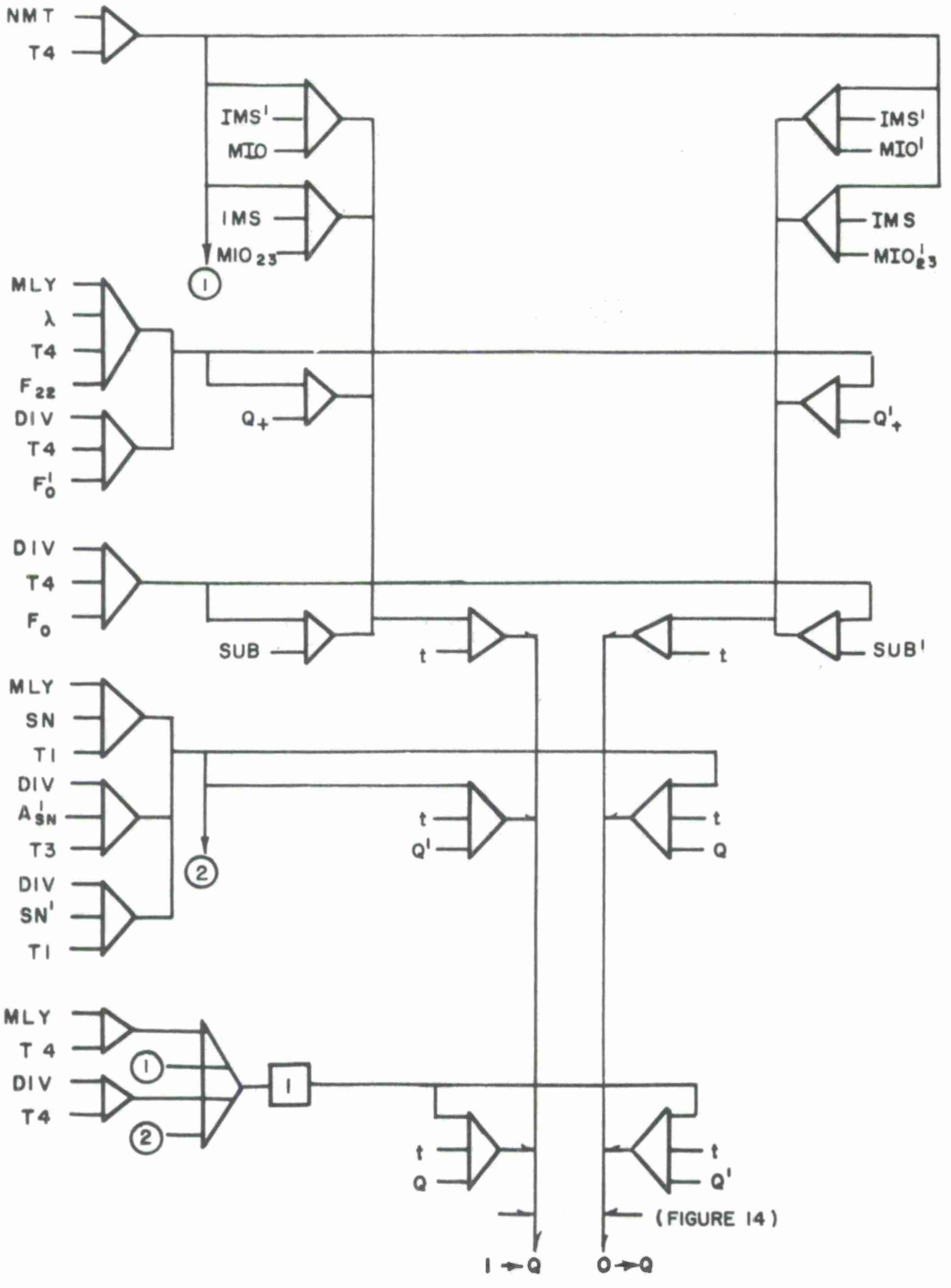



Figure 13. Q Gates I

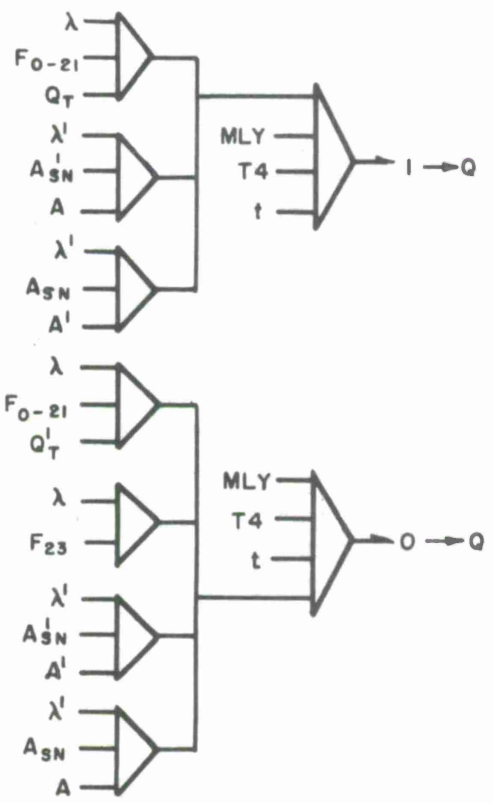



Figure 14. Q Gates II

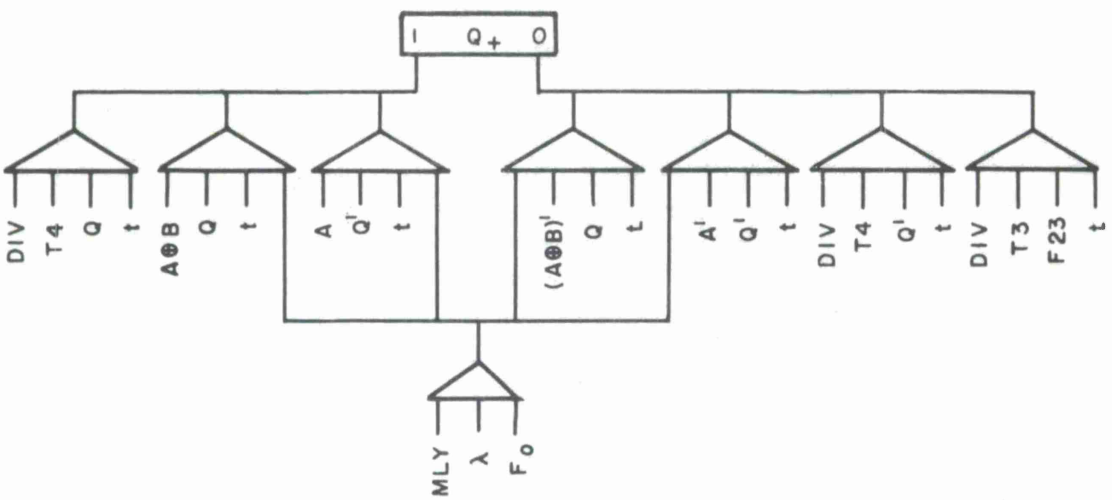



Figure 15. $Q_+$

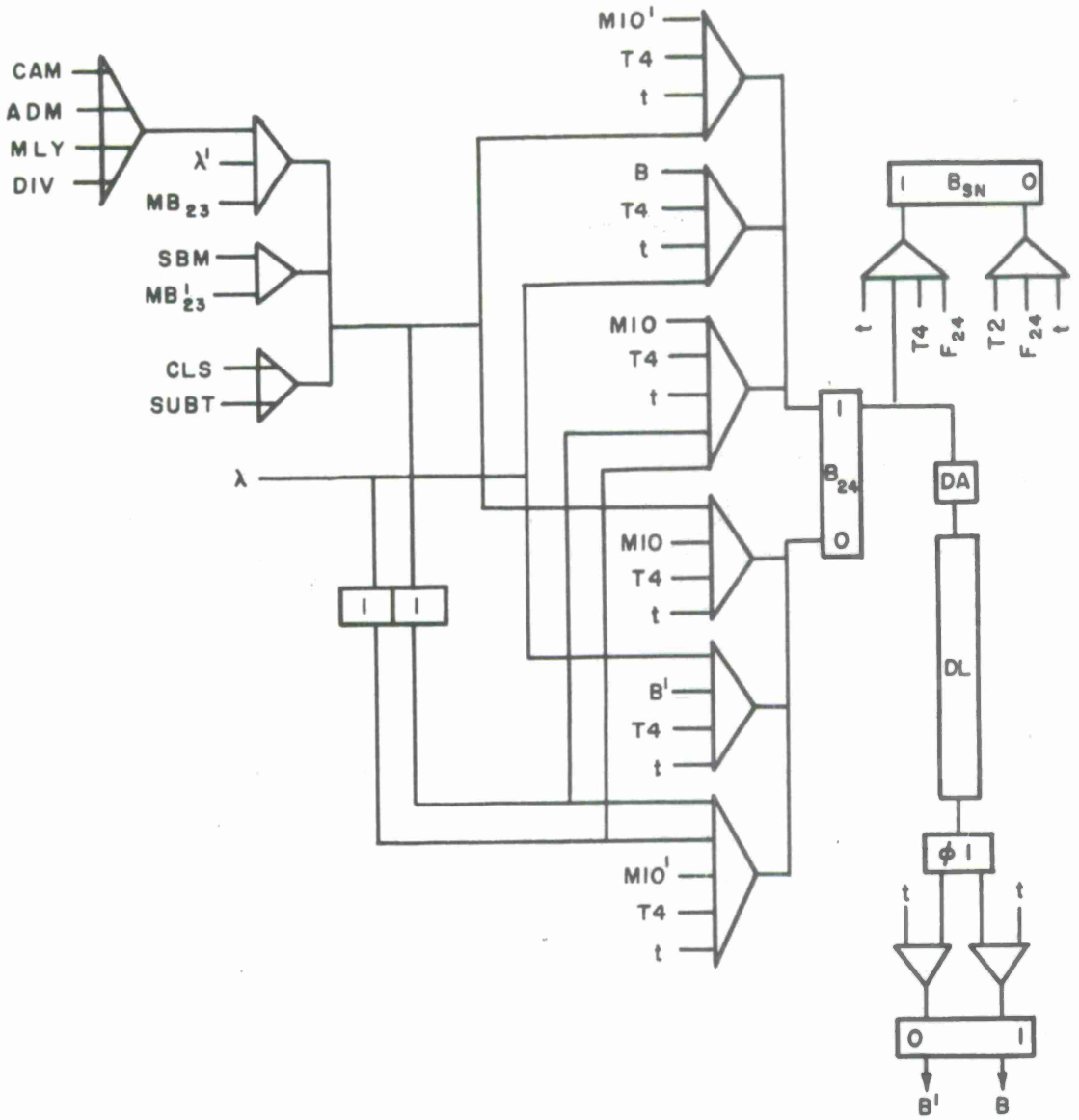



Figure 16. B Register

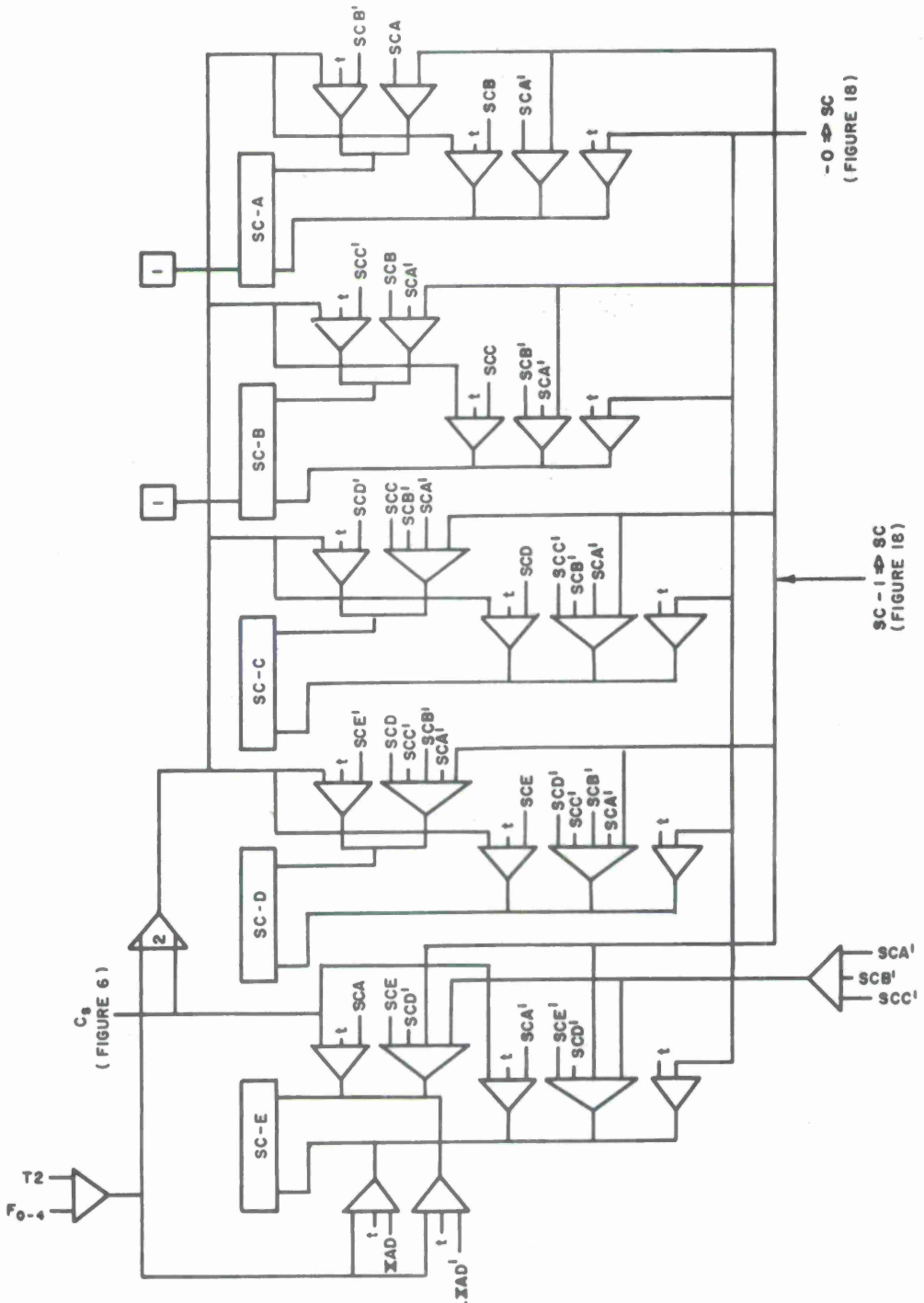



Figure 17. Shift Counter

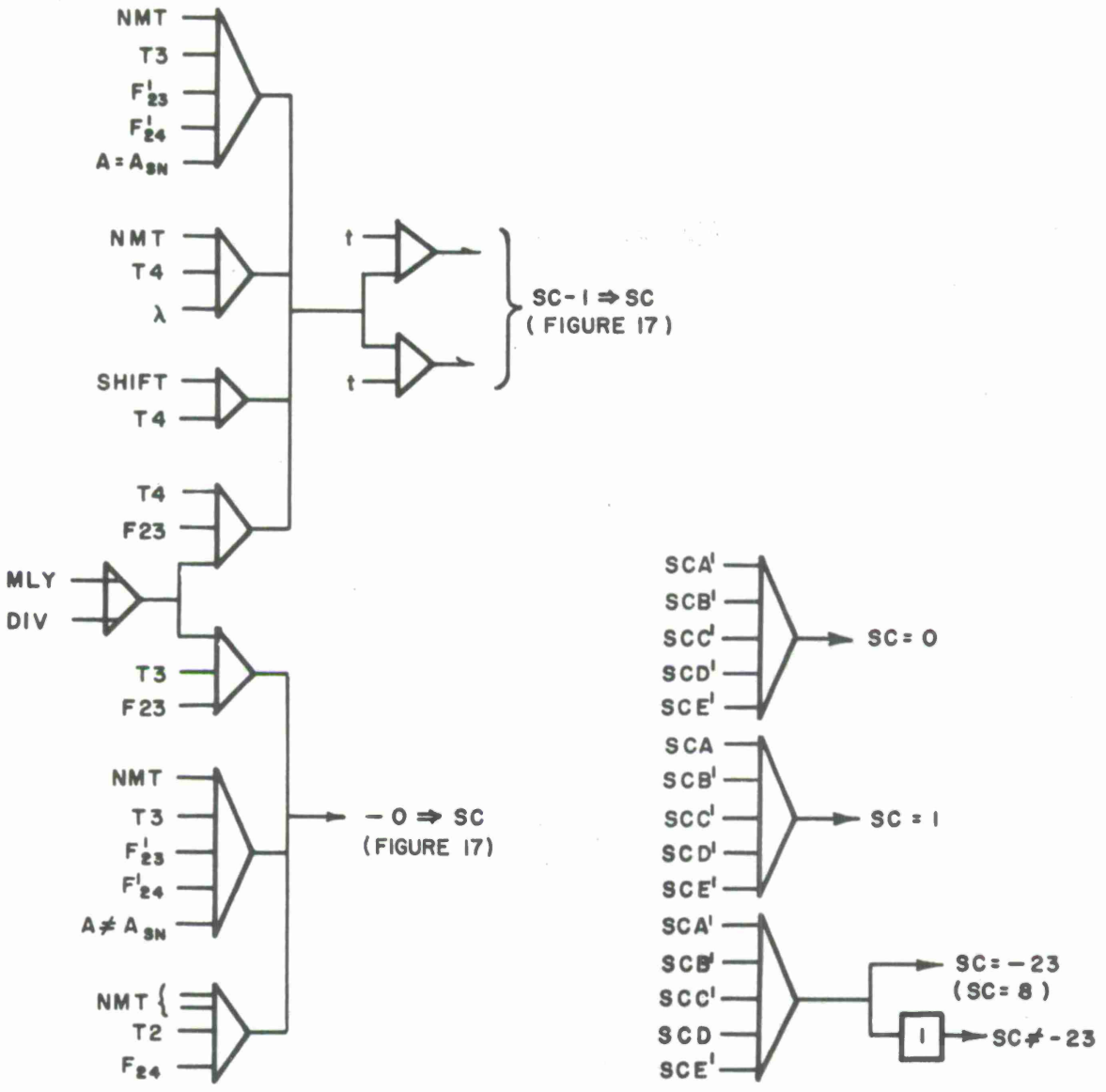



Figure 18. Shift Counter Controls

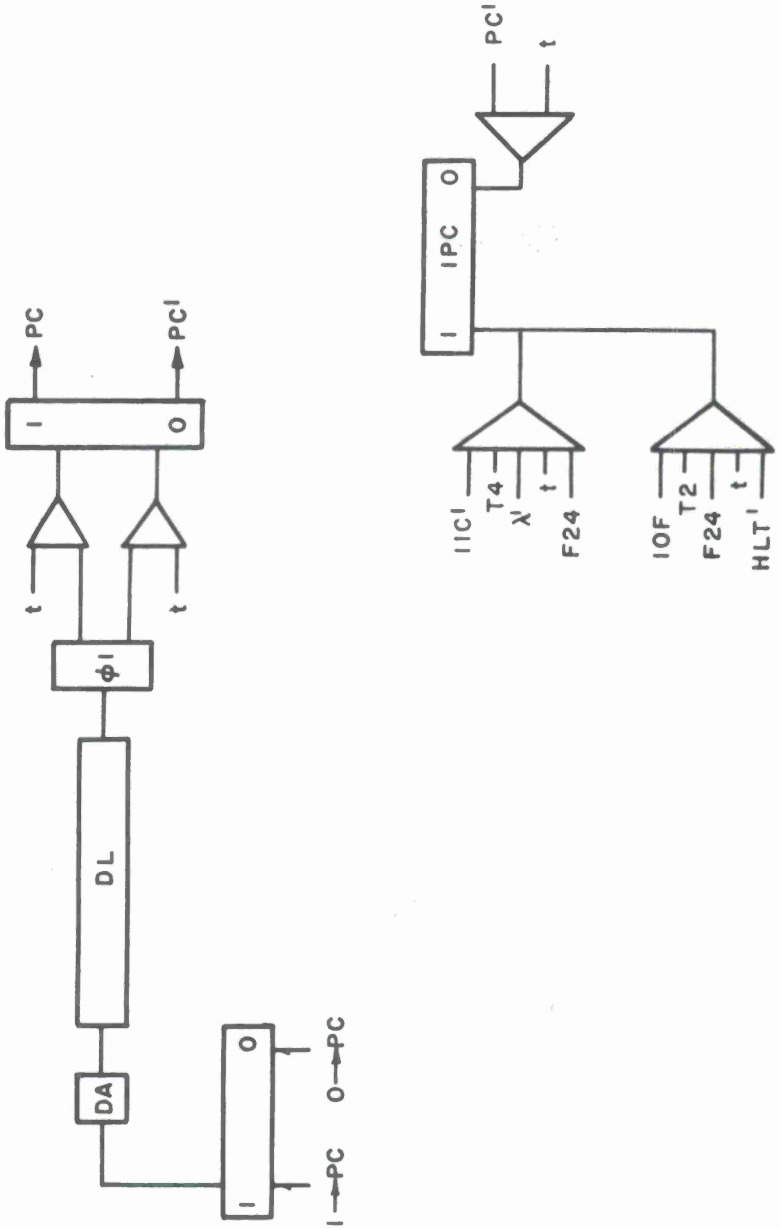



Figure 19. Program Counter

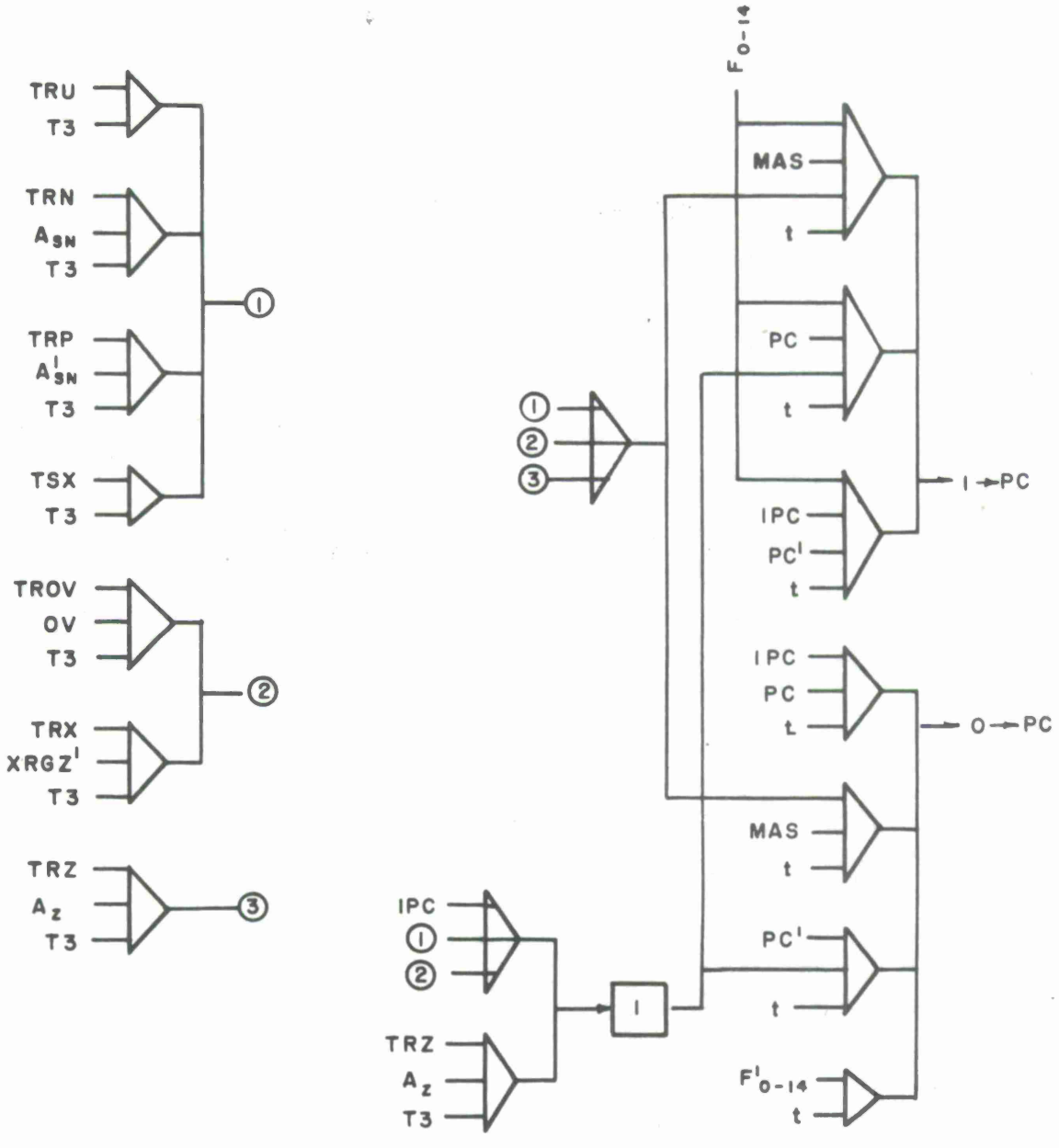



Figure 20. Program Counter Gates

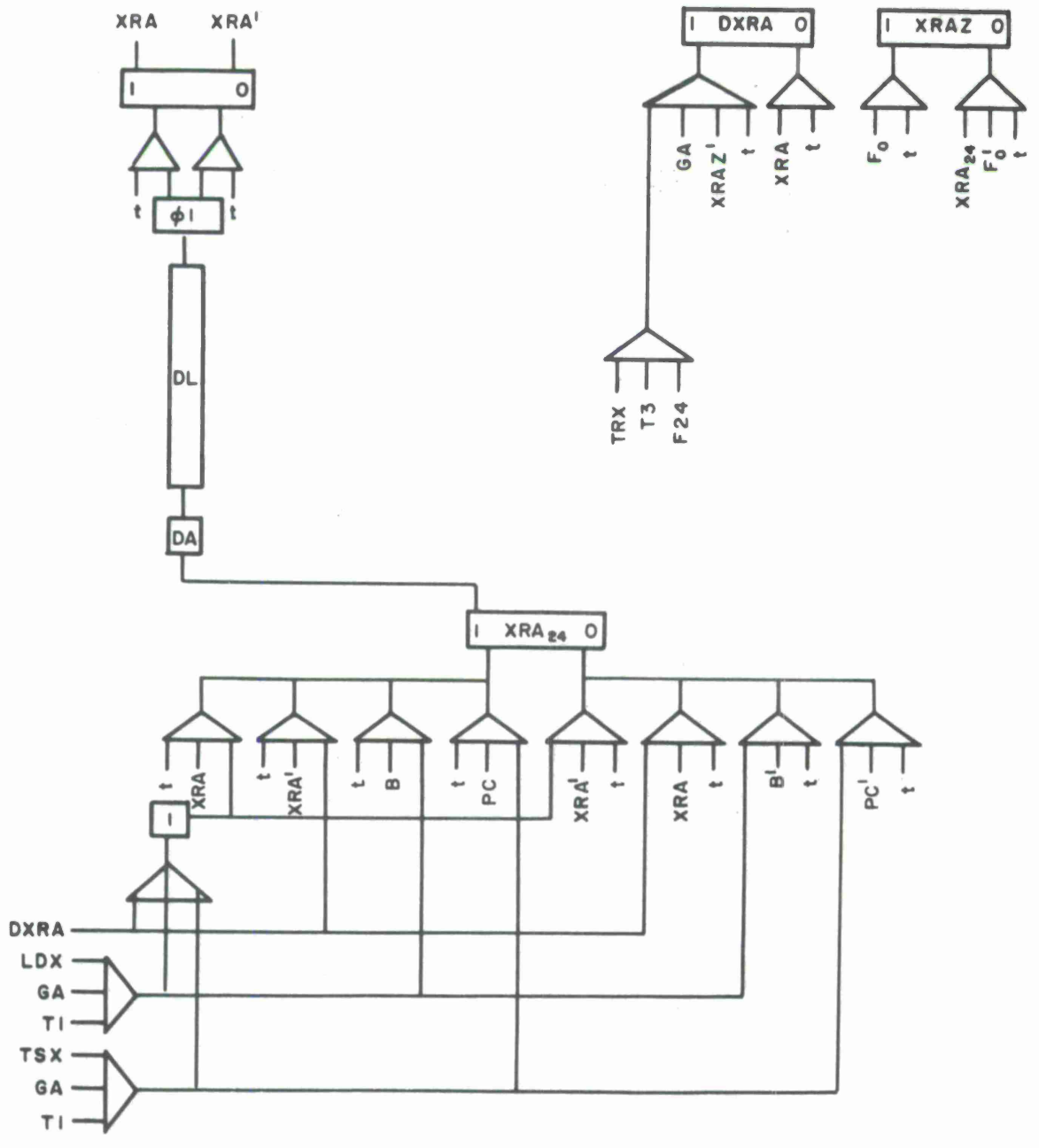



Figure 21. Index Register A

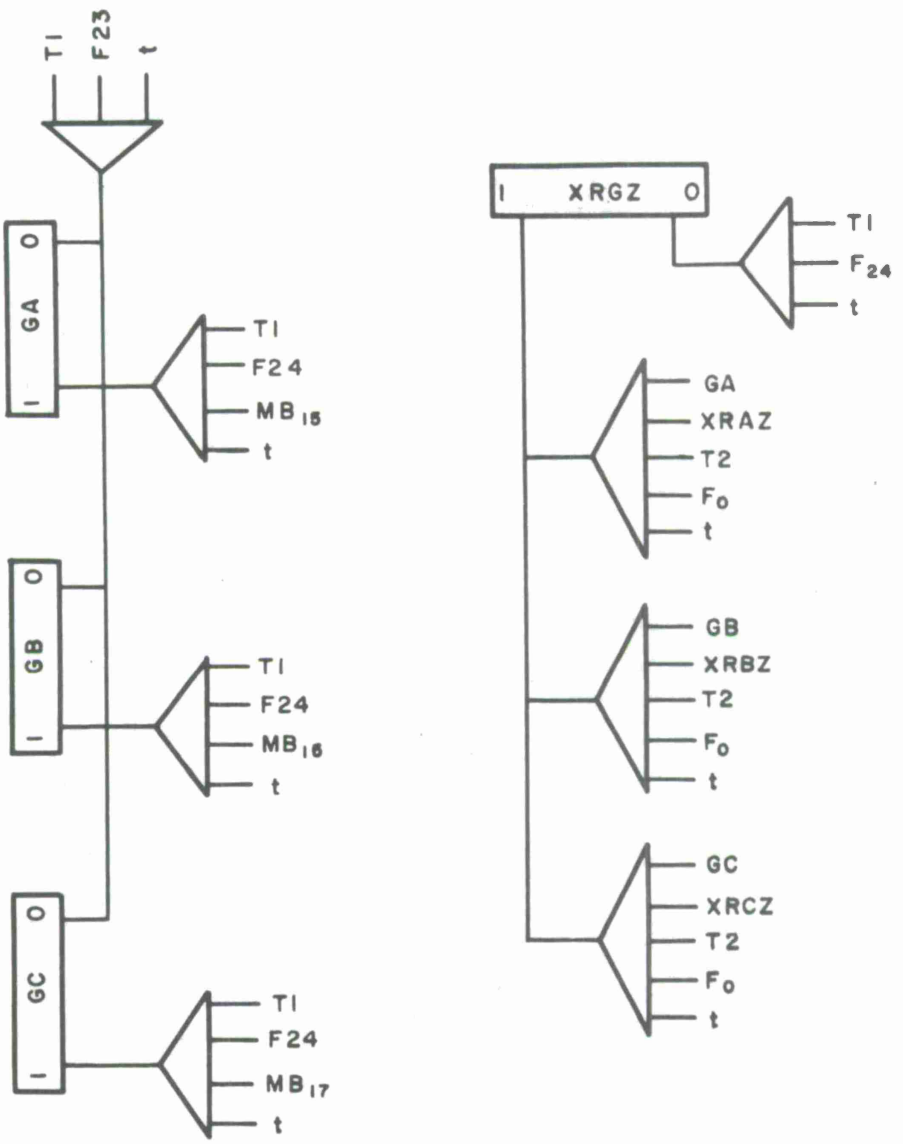



Figure 22. G Register

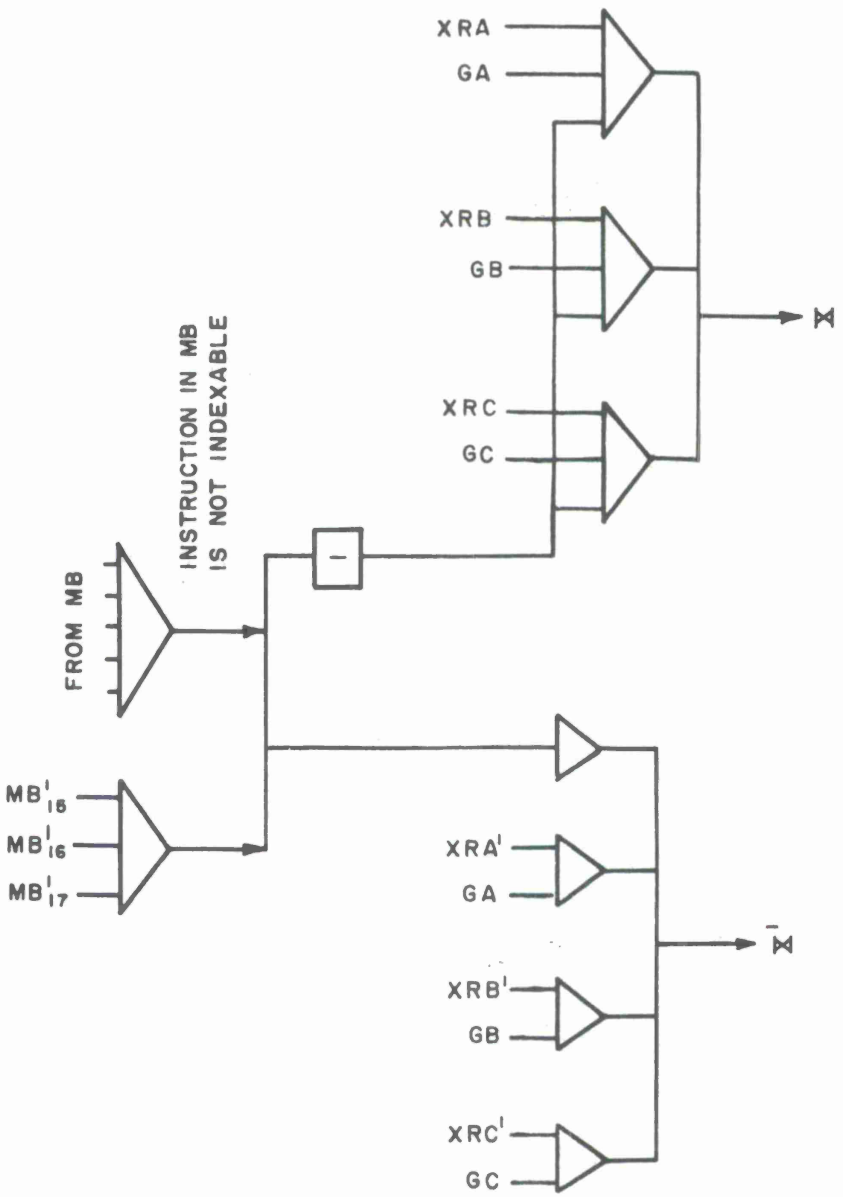



Figure 23. Index Selection

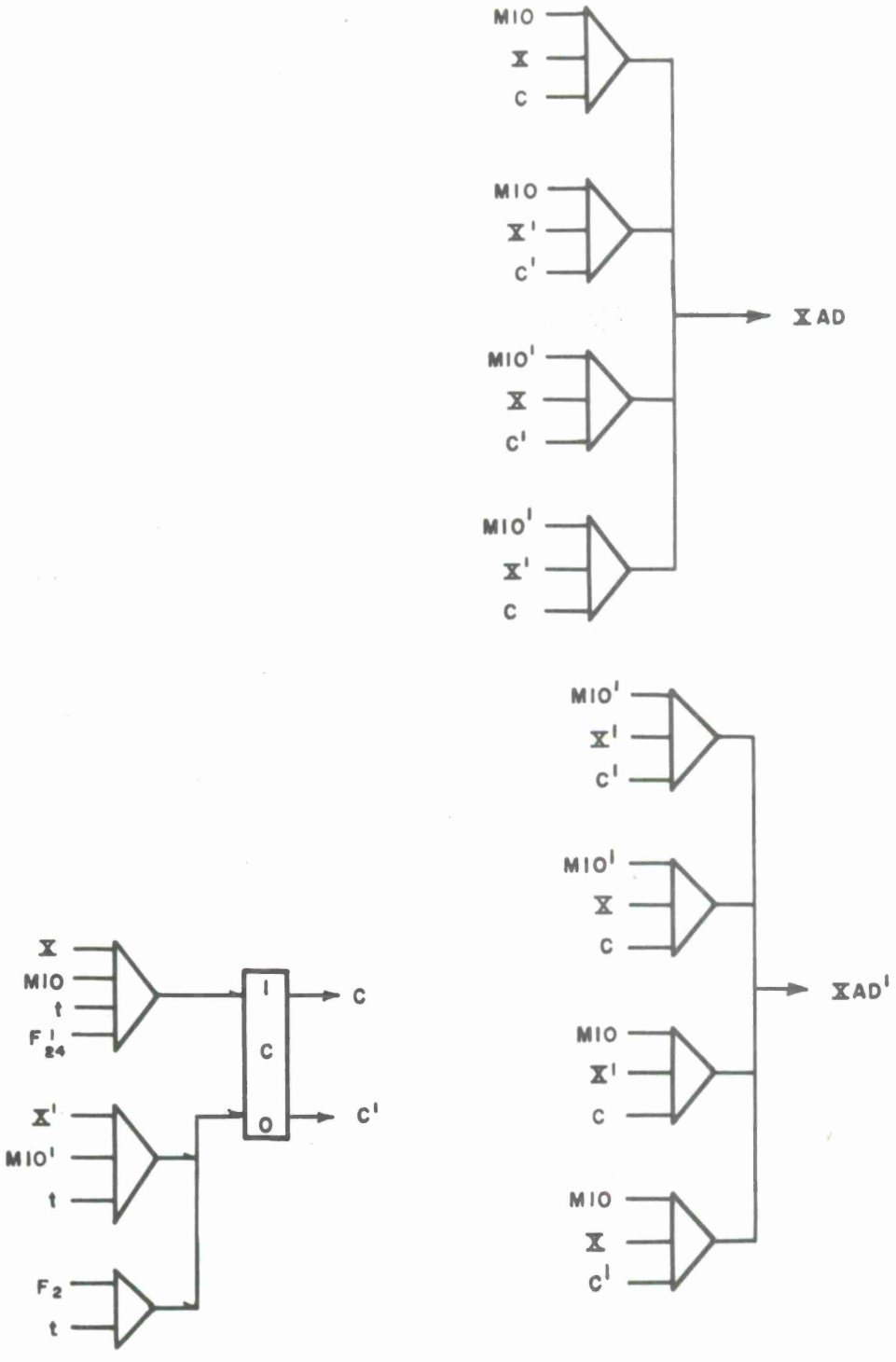



Figure 24. Index Adder (XAD)

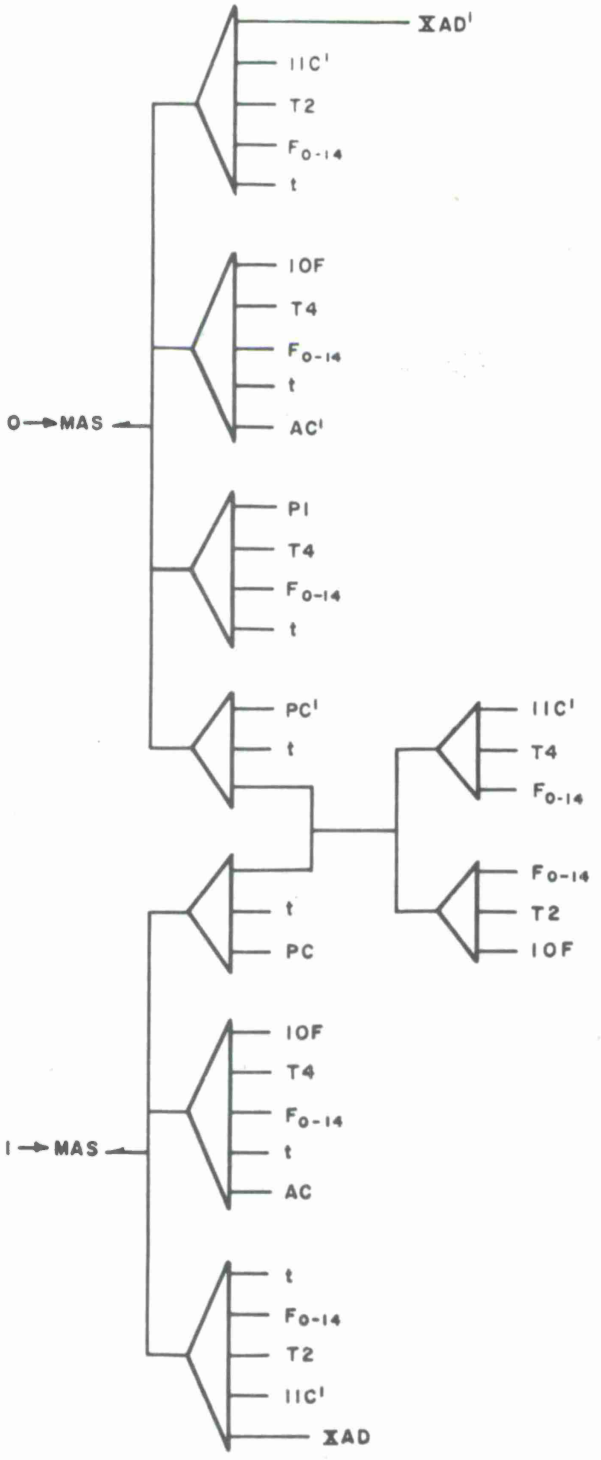



Figure 25. Memory Address Storage Gates

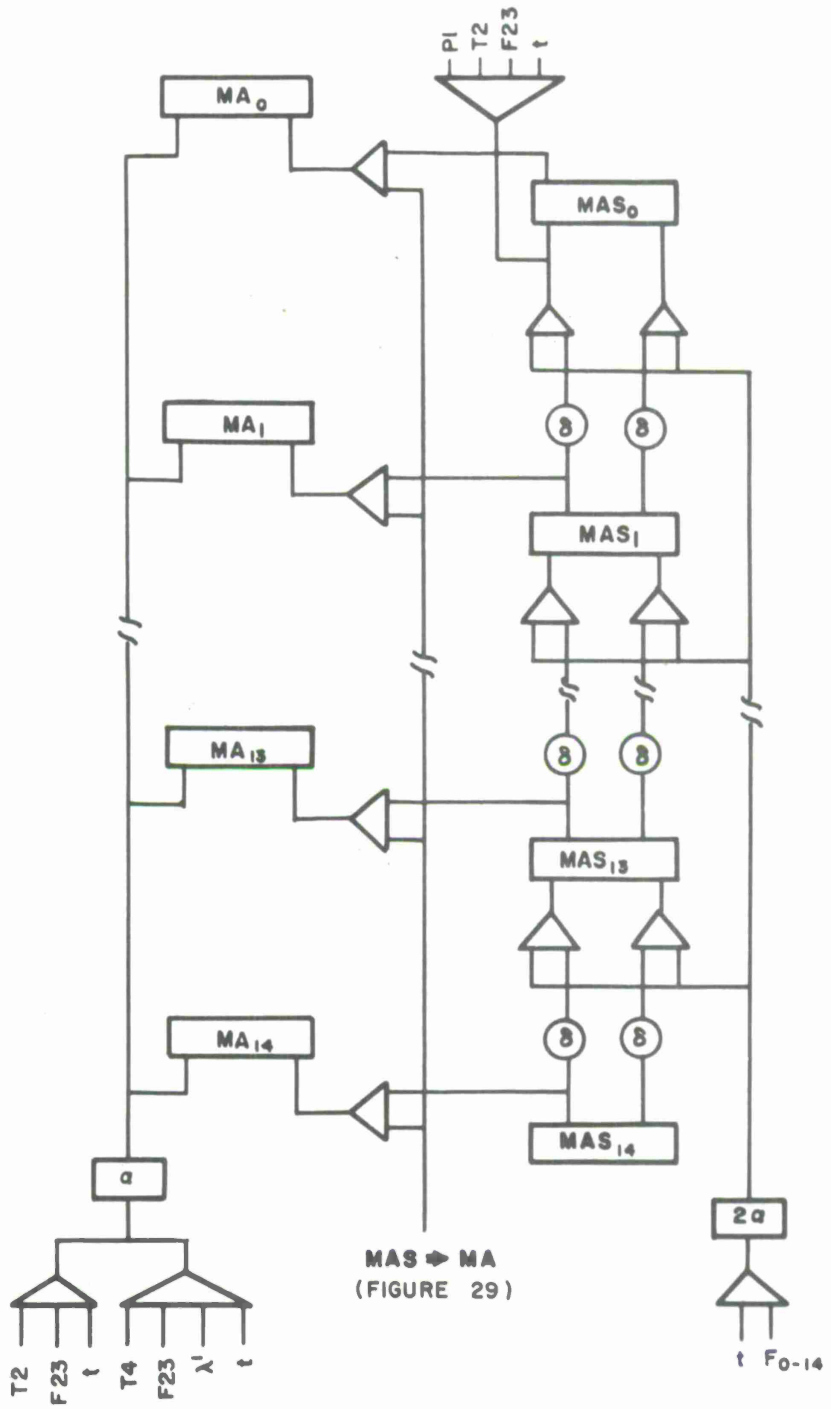



Figure 26. Memory Address Storage Register - Memory Address Register

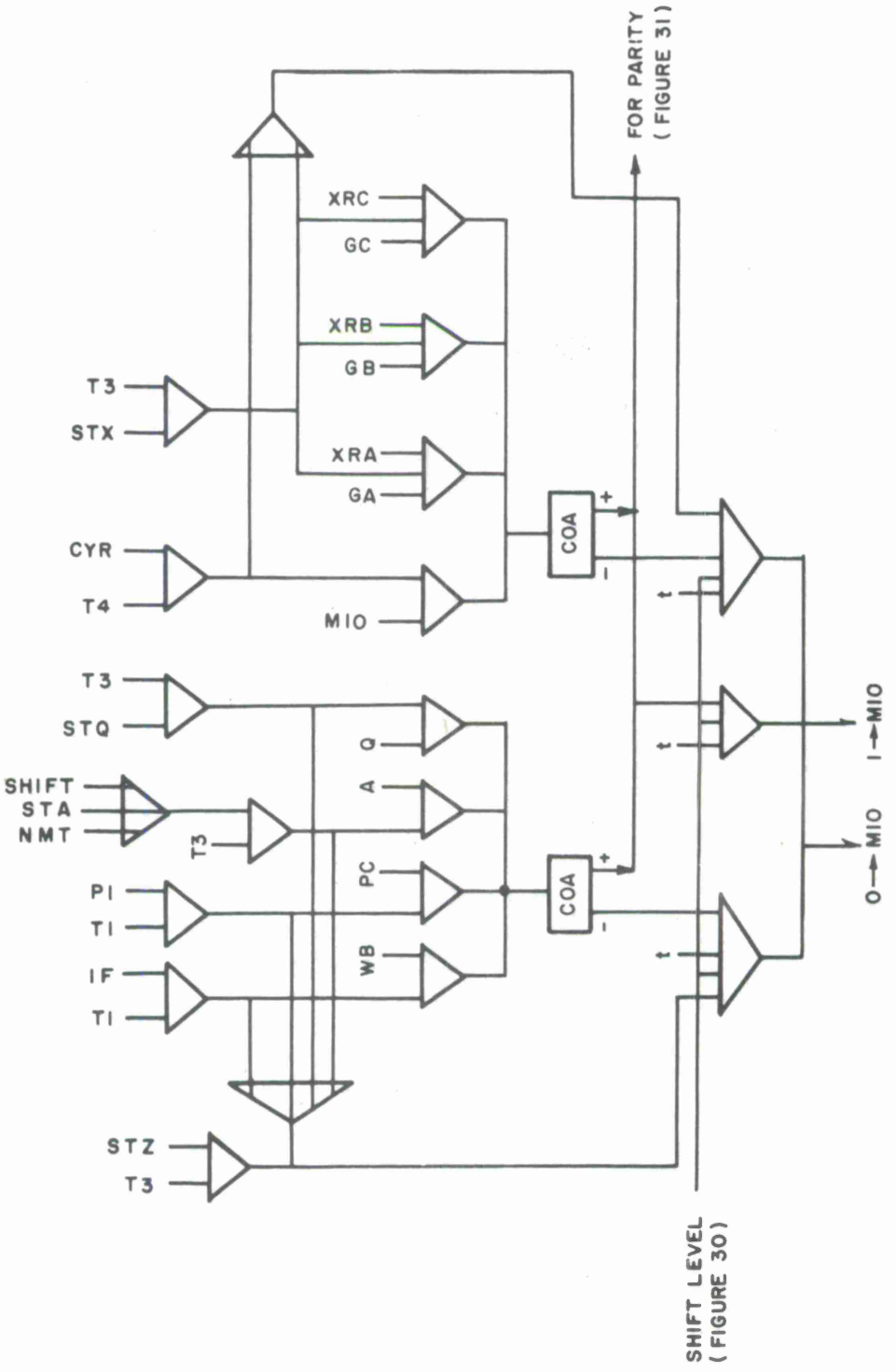
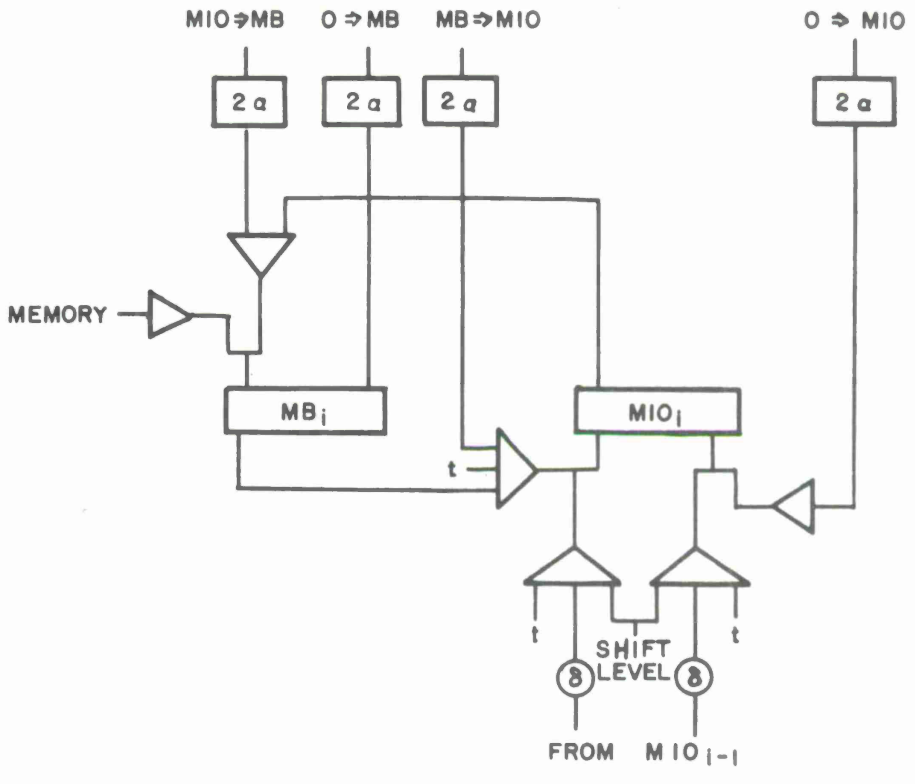



Figure 27. Memory Input-Output Register Input Gates




TRIPLE FF ON MIO$_0$

Figure 28. Memory Input–Output and Memory Buffer Register Stage (1 of 24)

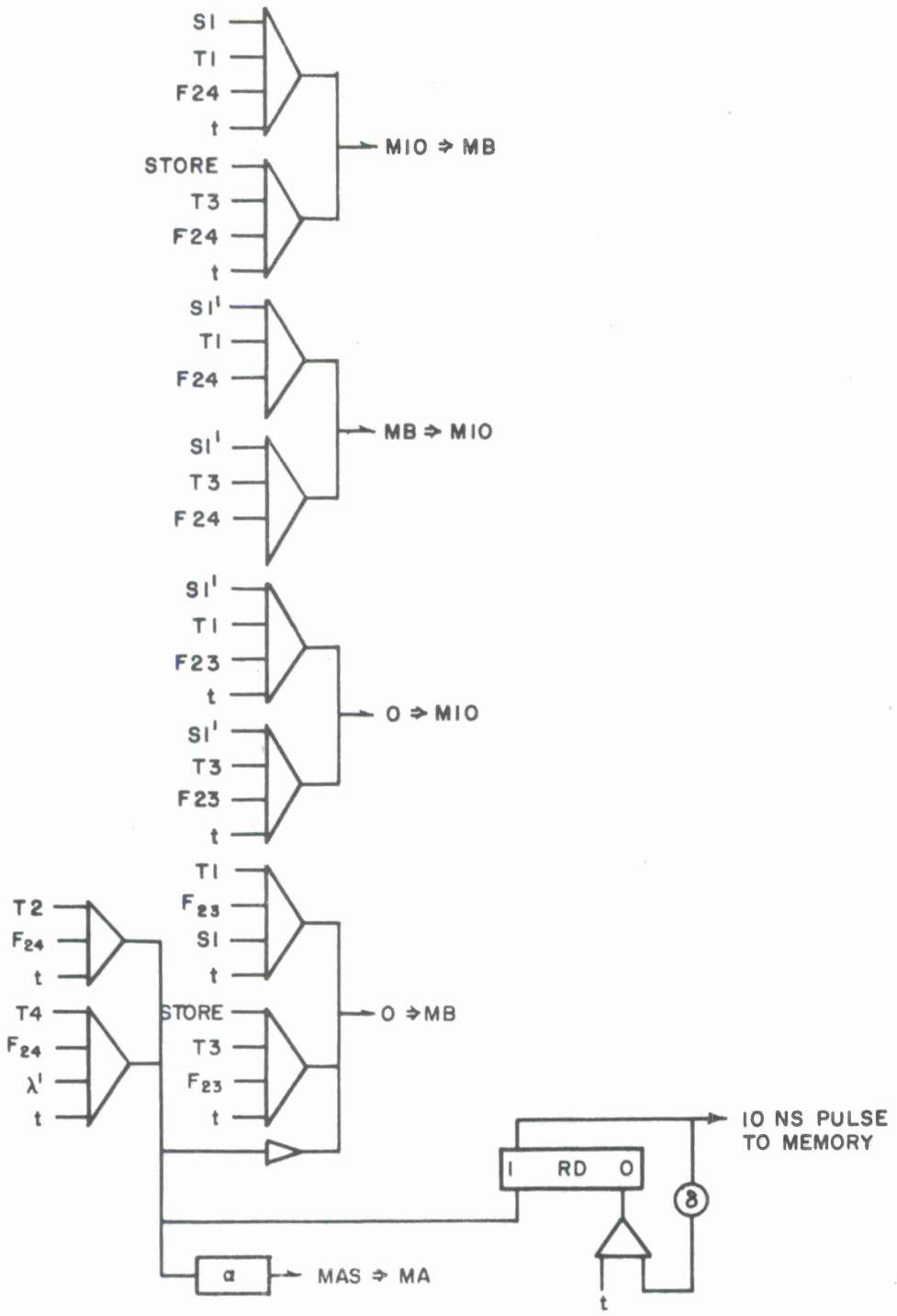



Figure 29. MIO-MB Controls

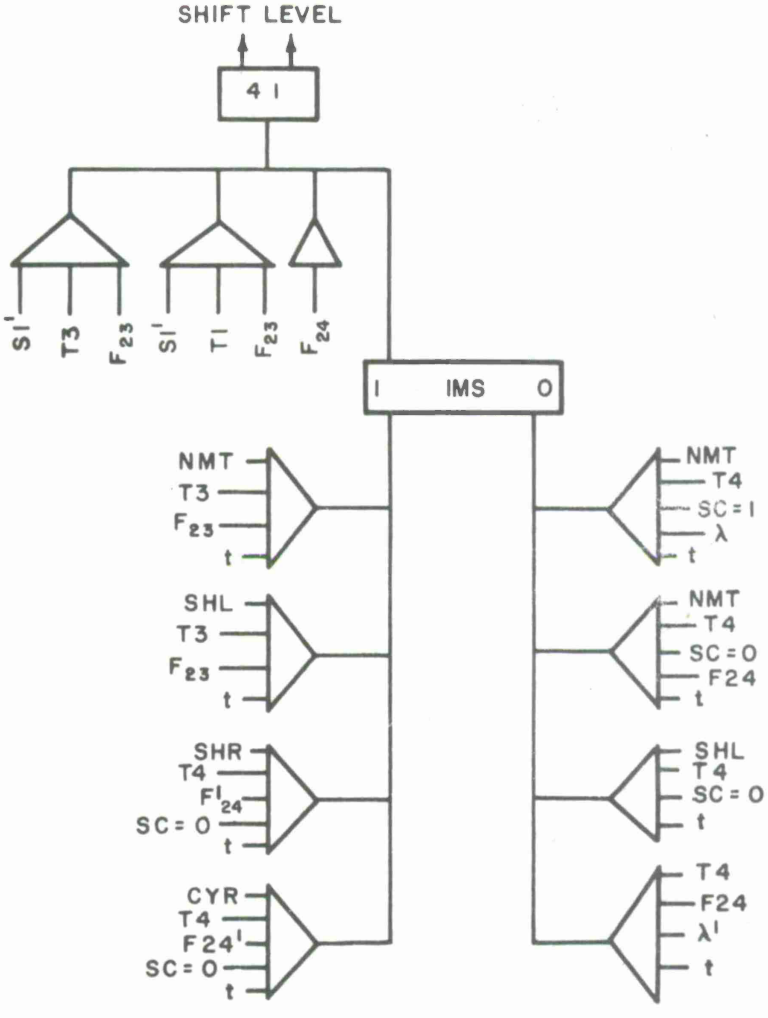



Figure 30. Memory Input-Output Register Shift Control

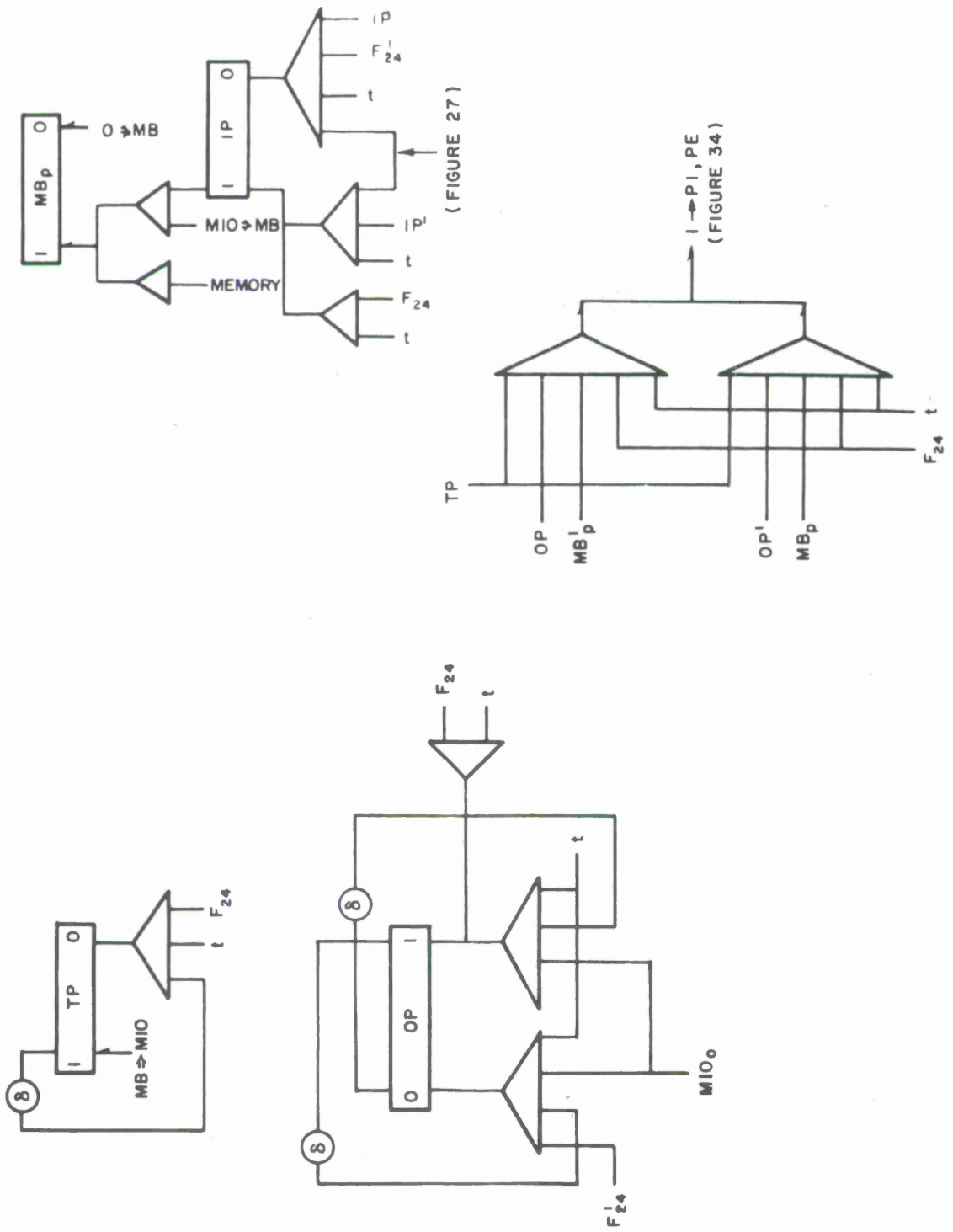



Figure 31. Parity

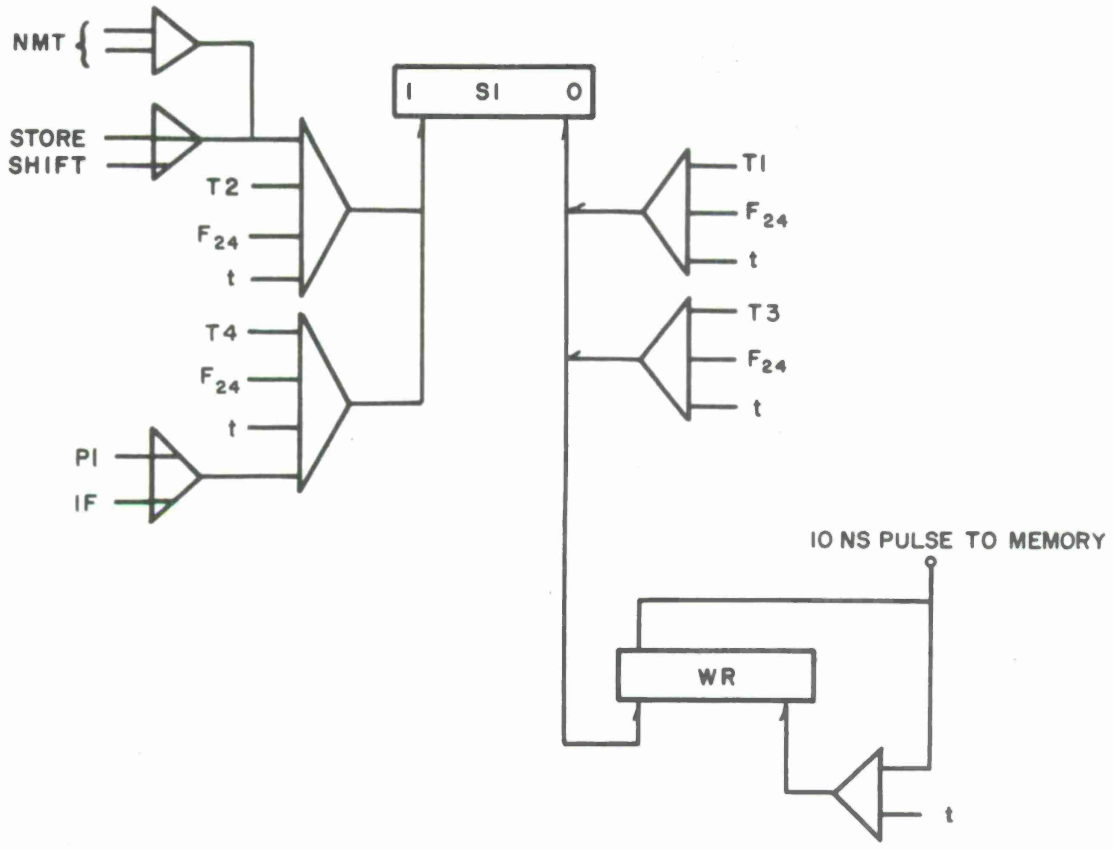



Figure 32. Memory Controls

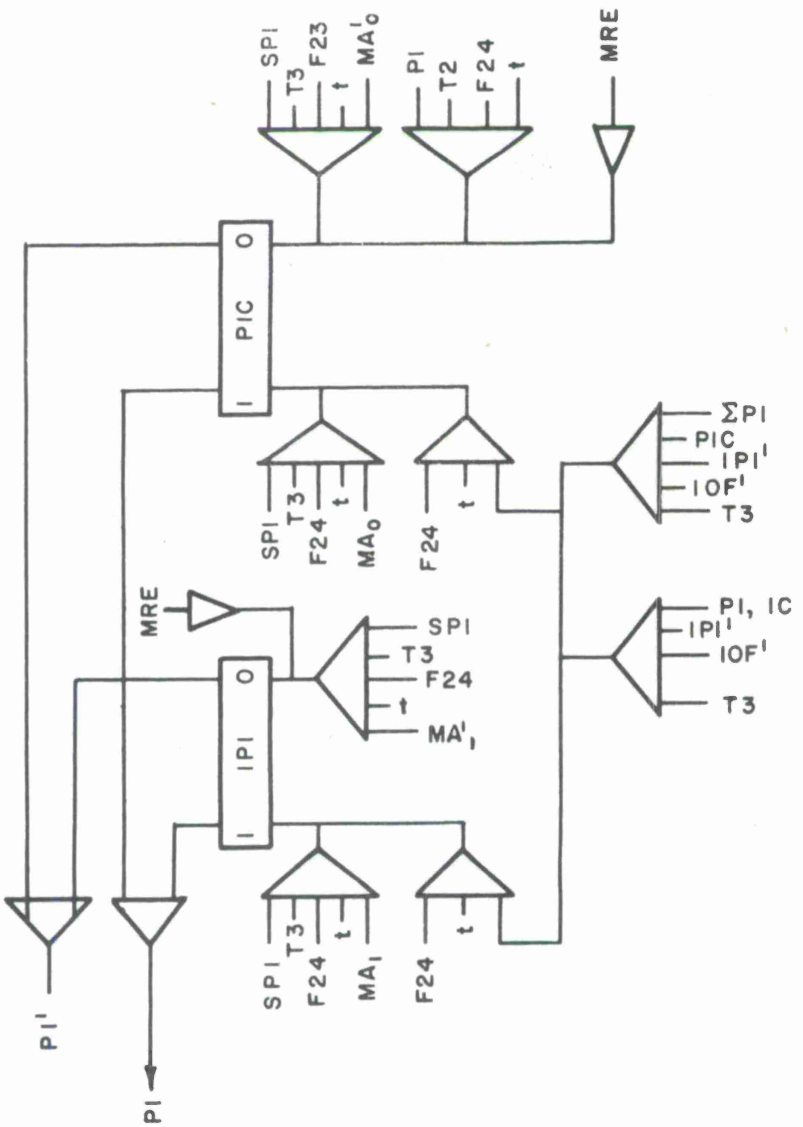



Figure 33. Program Interrupt Controls

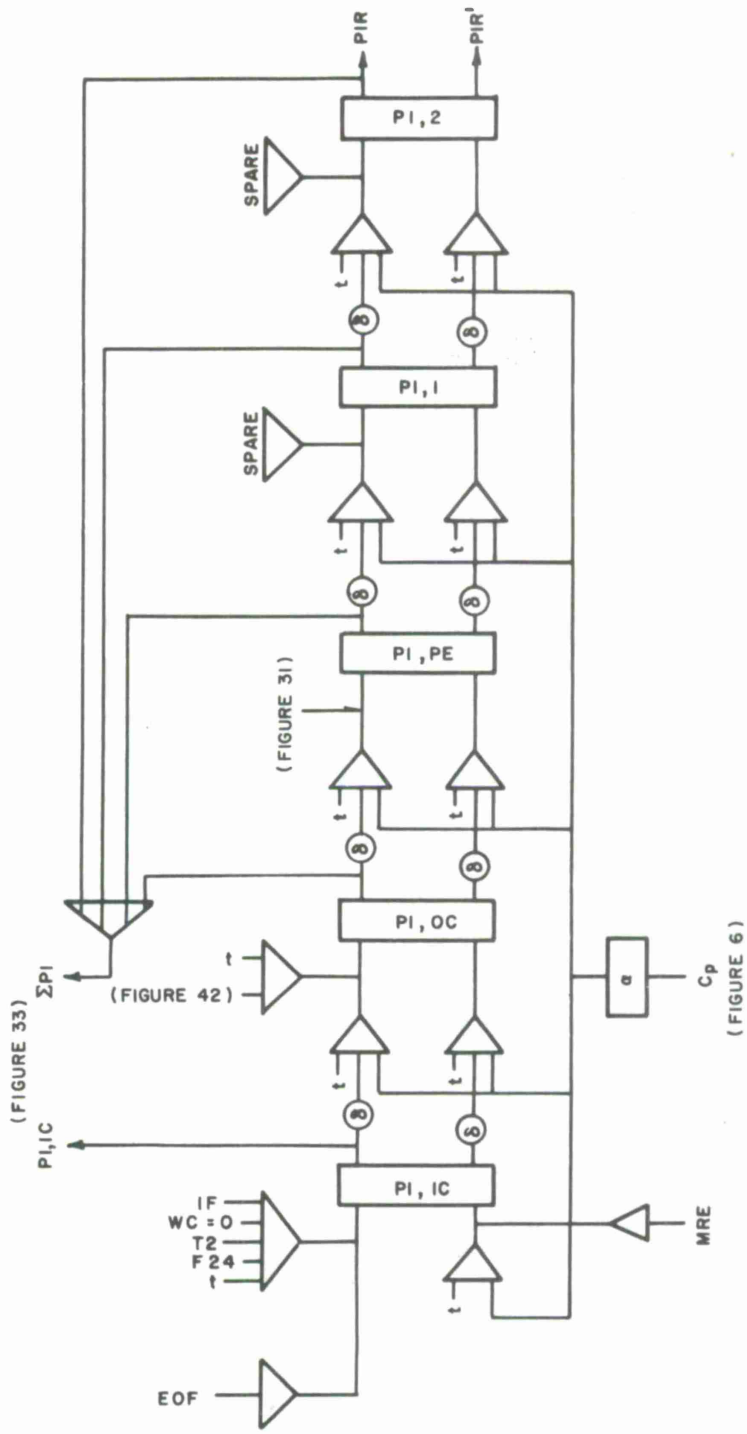



Figure 34. Program Interrupt Register

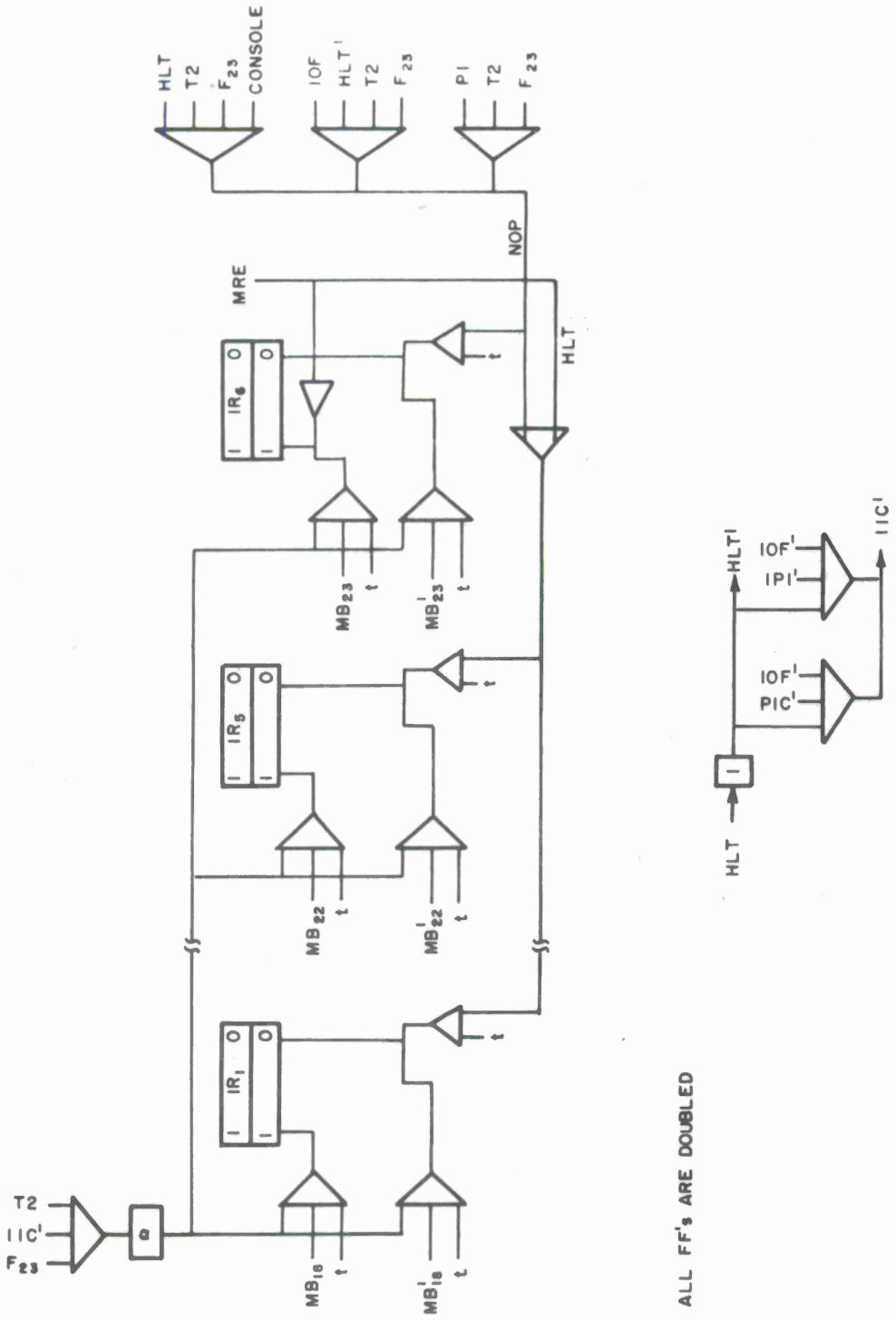



Figure 35. Instruction Register

33 INSTRUCTIONS ARE DECODED

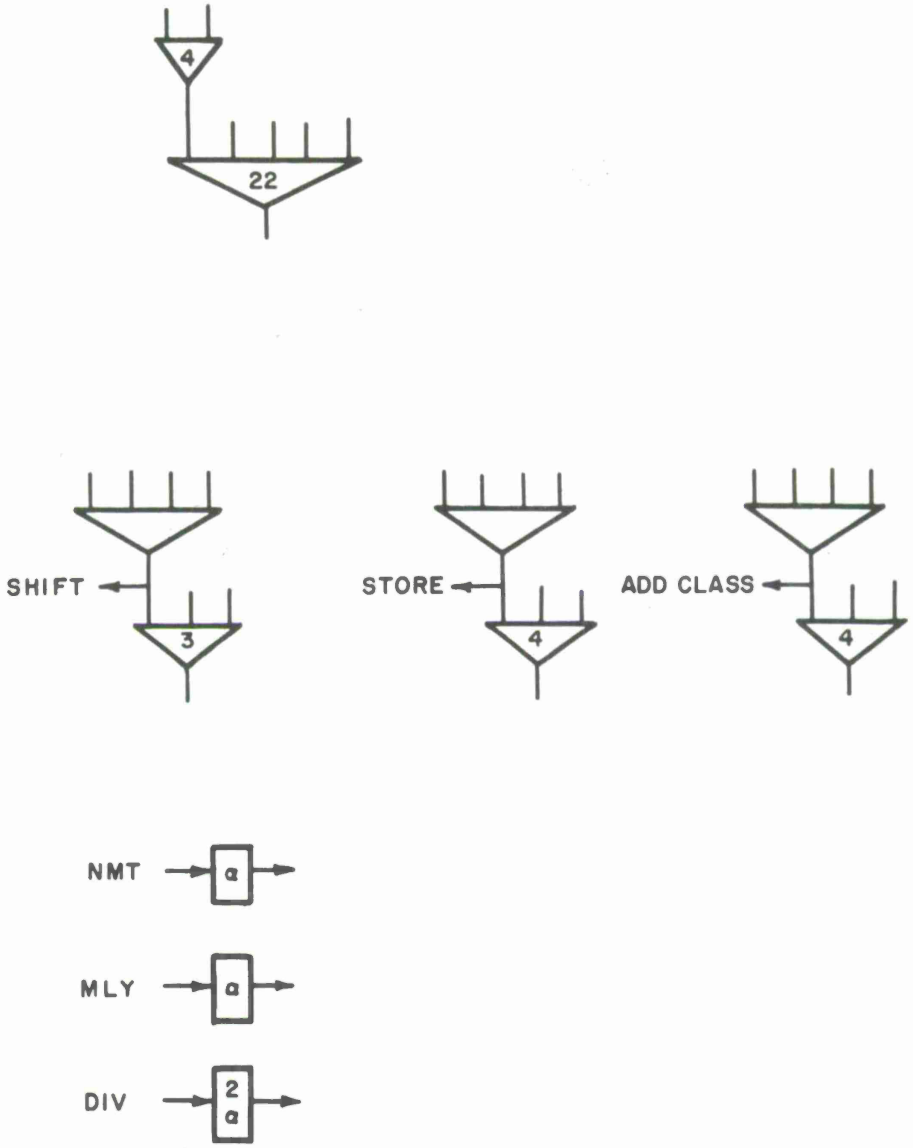



Figure 36. Decoder

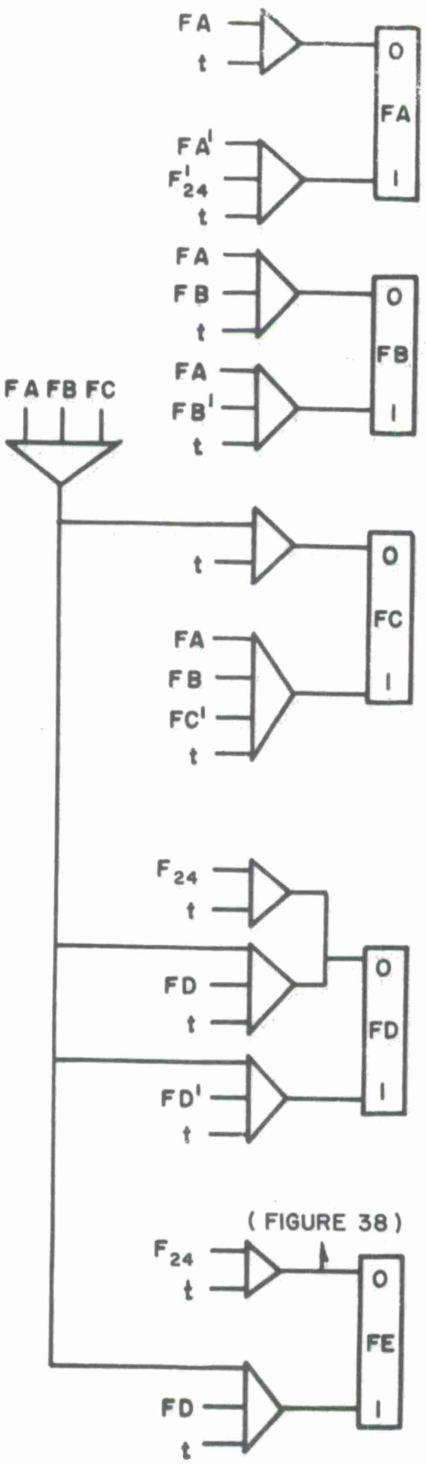



Figure 37. F Counter

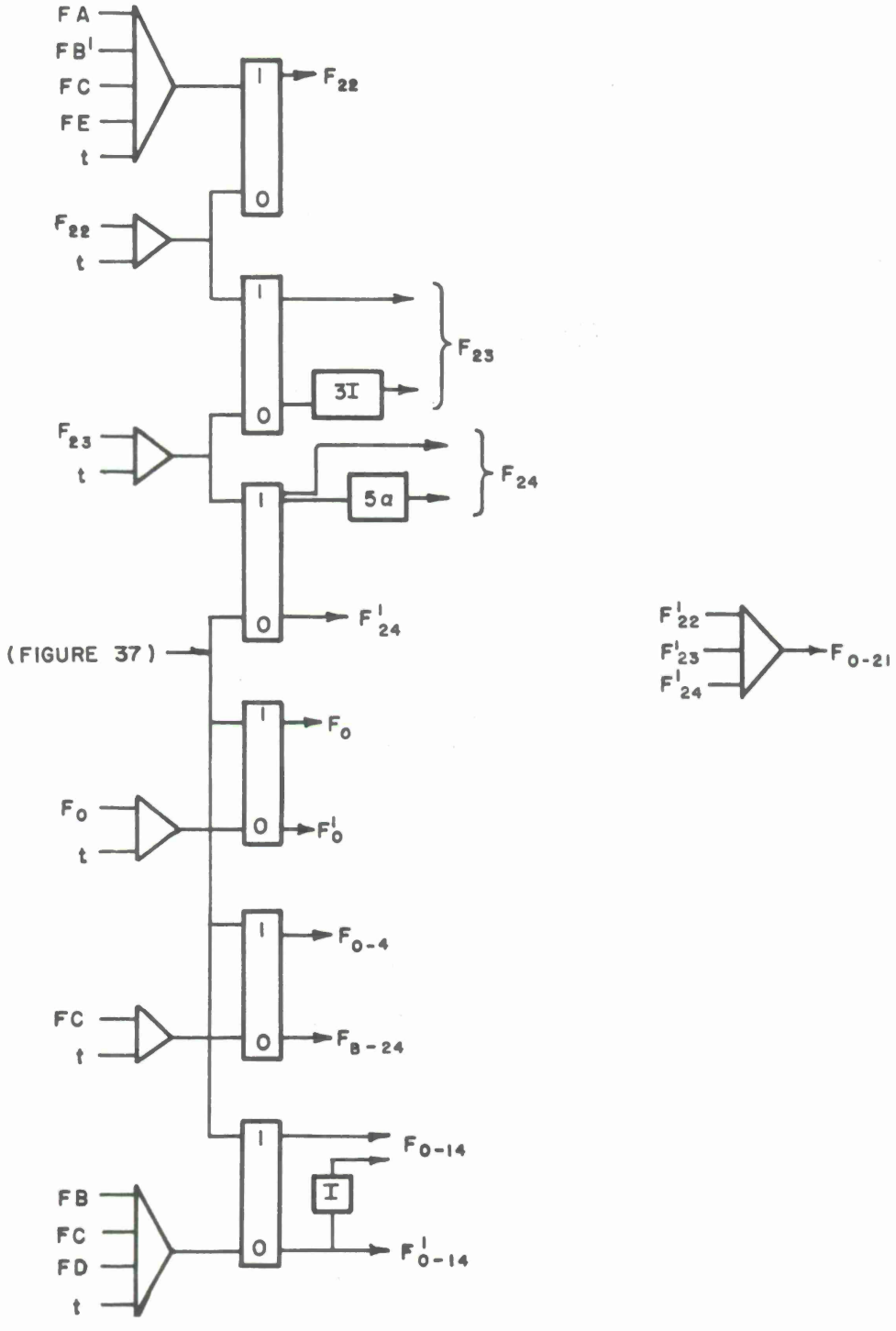
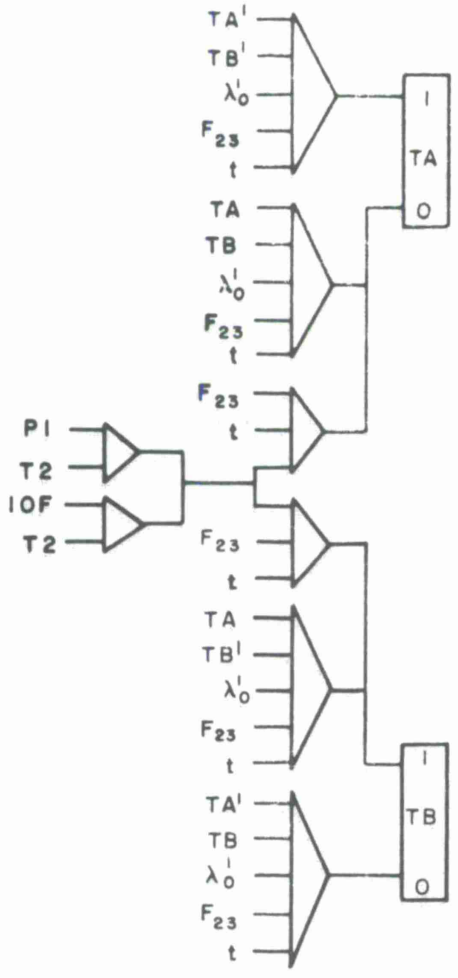
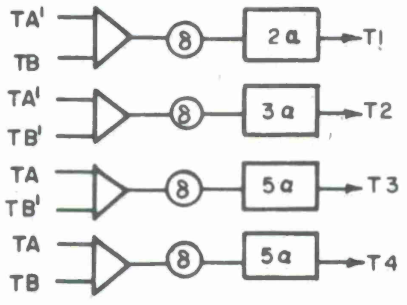



Figure 38. F Functions




| T | A | B |
|---|---|---|
| 1 | 0 | 1 |
| 2 | 0 | 0 |
| 3 | 1 | 0 |
| 4 | 1 | 1 |




DELAY TRIMMED SO THAT T LEVEL RISES 12.5 NS AFTER $(F_{23})(t)$ PULSE

Figure 39. T Counter

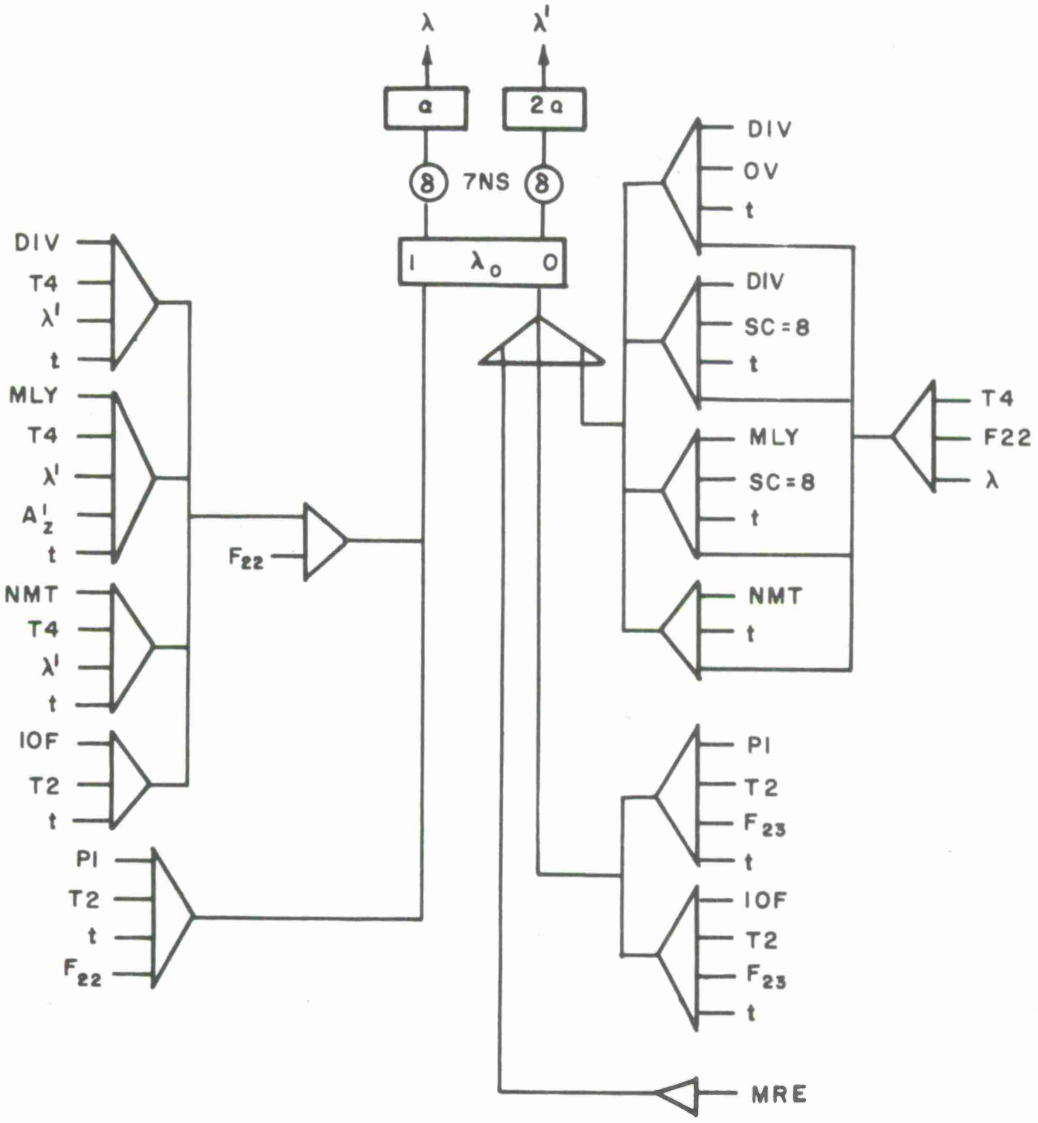



Figure 40. Lambda Flip Flop

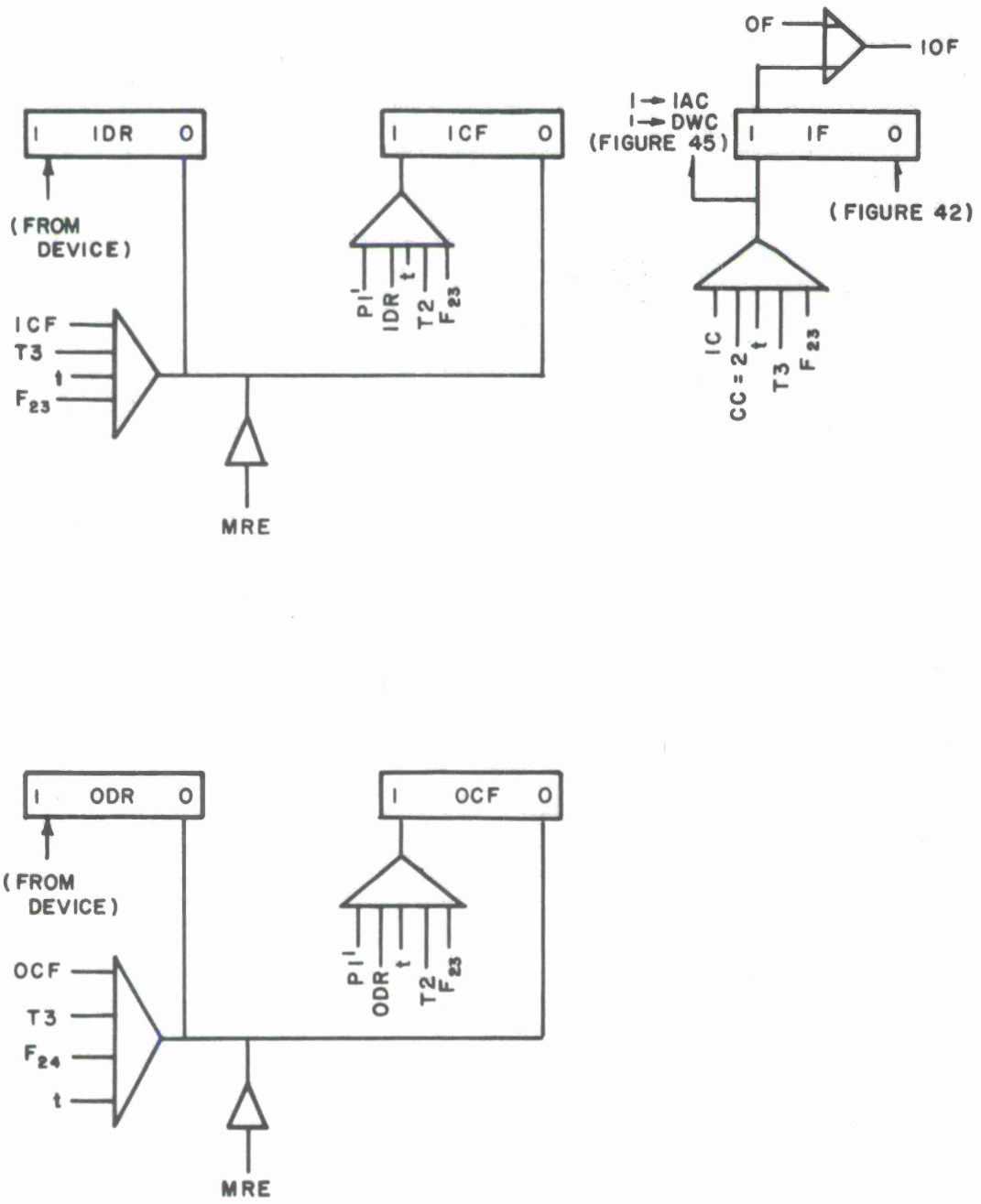



Figure 41. I/O Controls

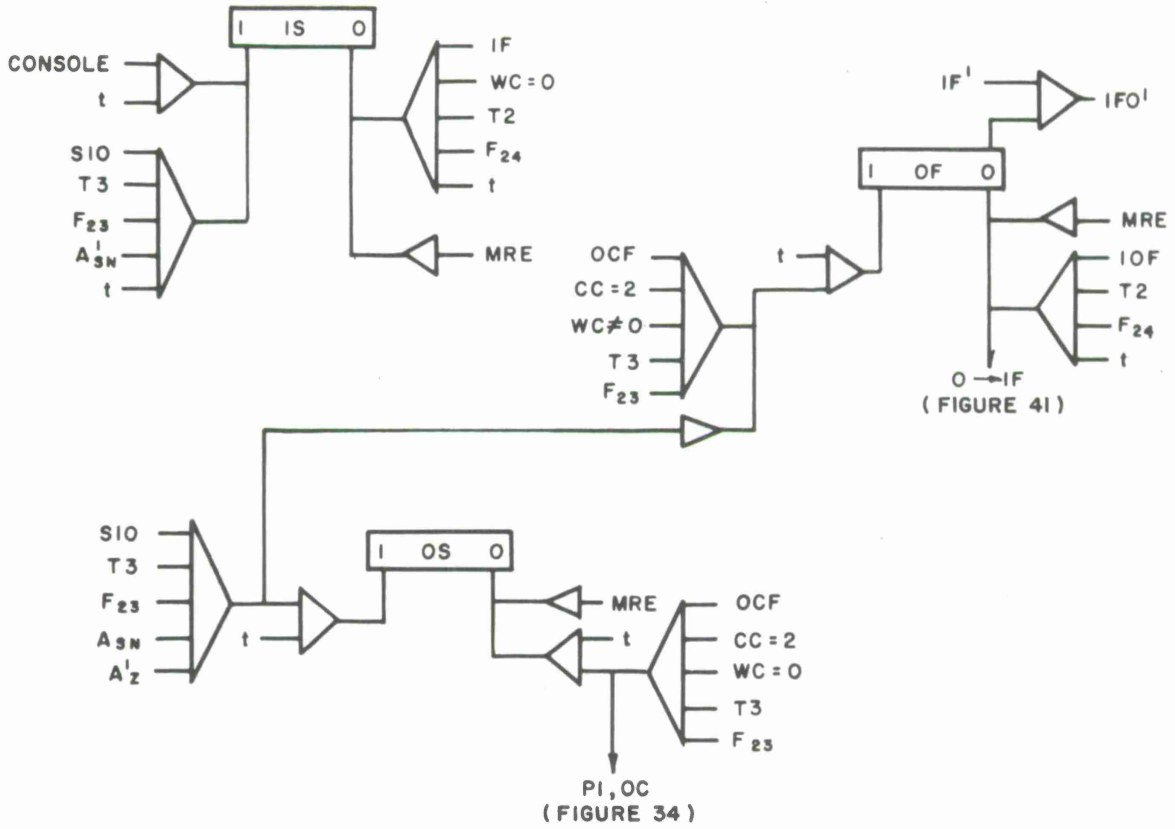



Figure 42. I/O Controls

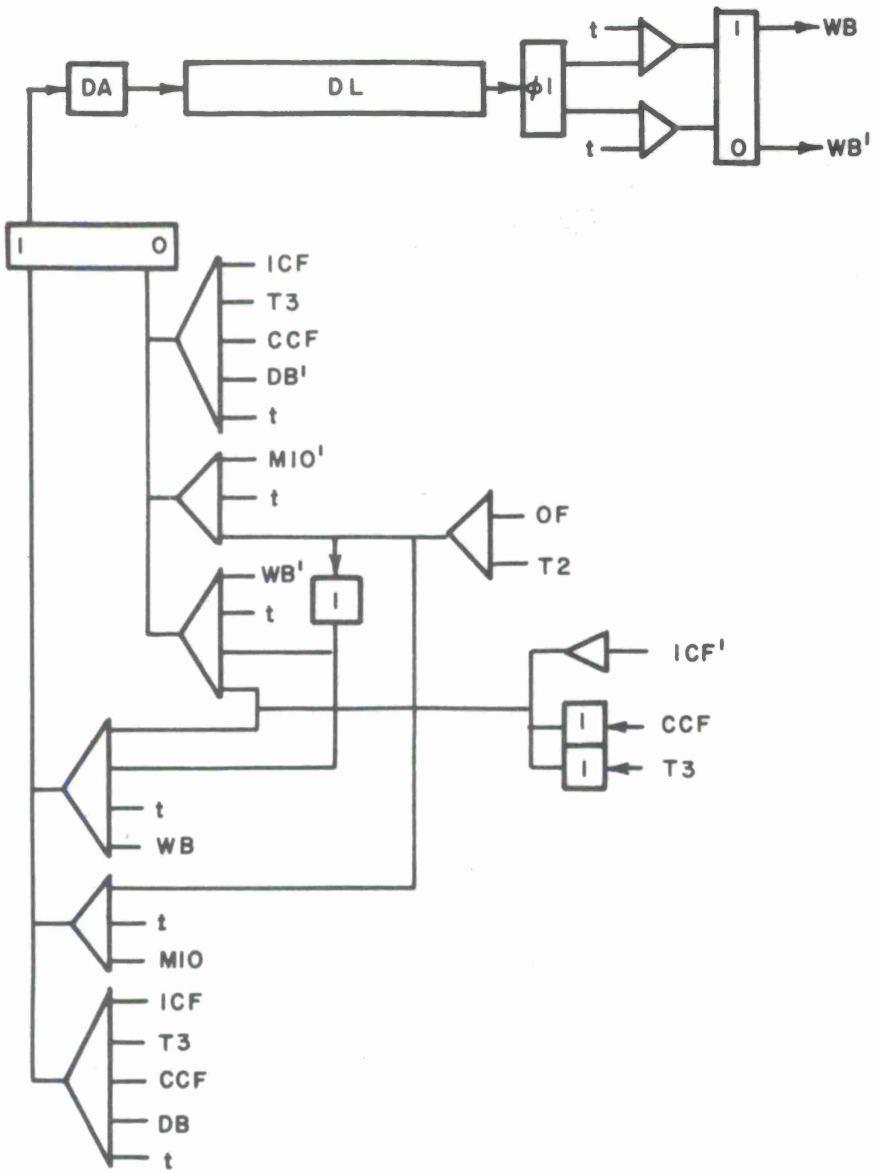



Figure 43. Word Buffer Register

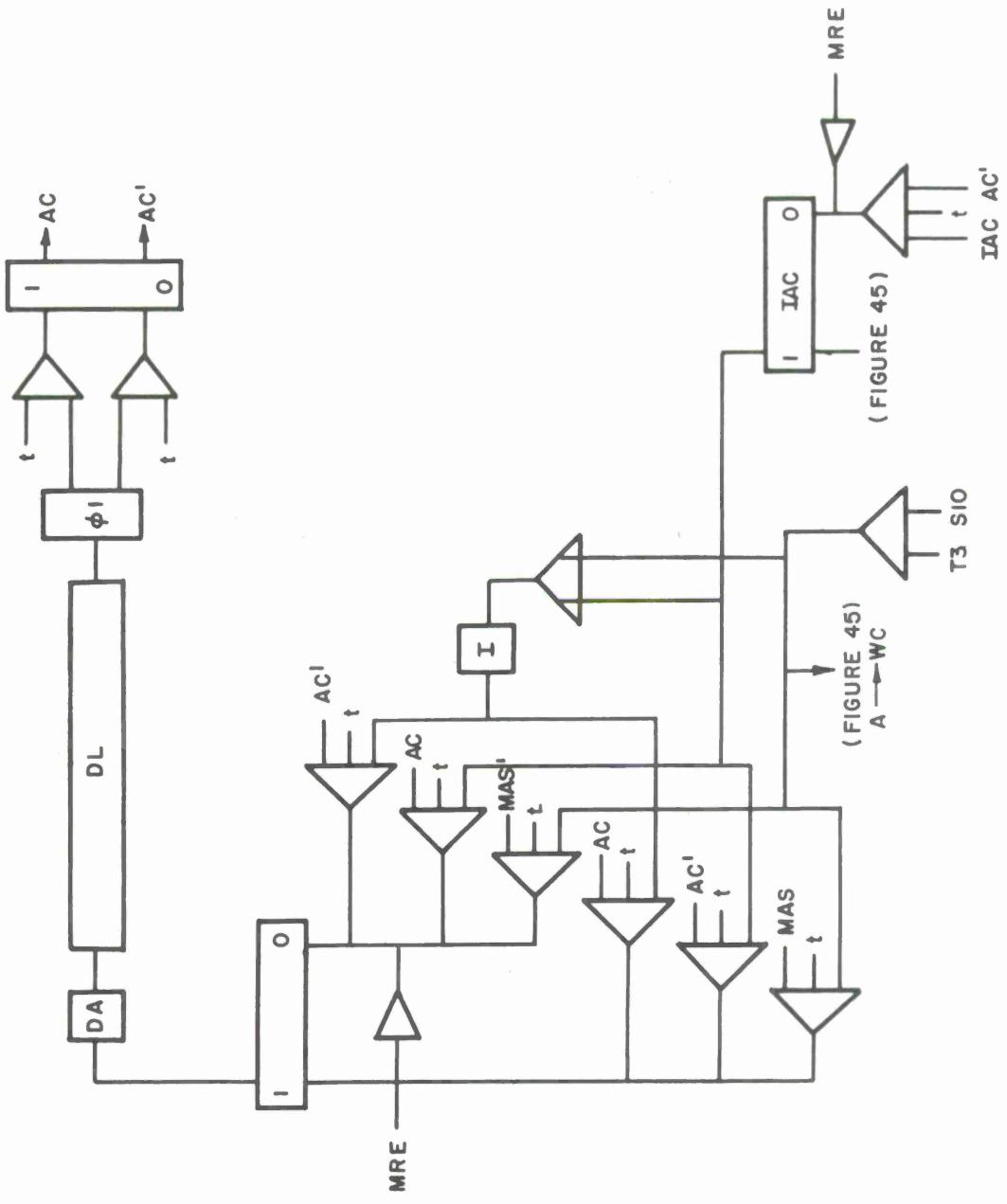



Figure 44. Address Counter

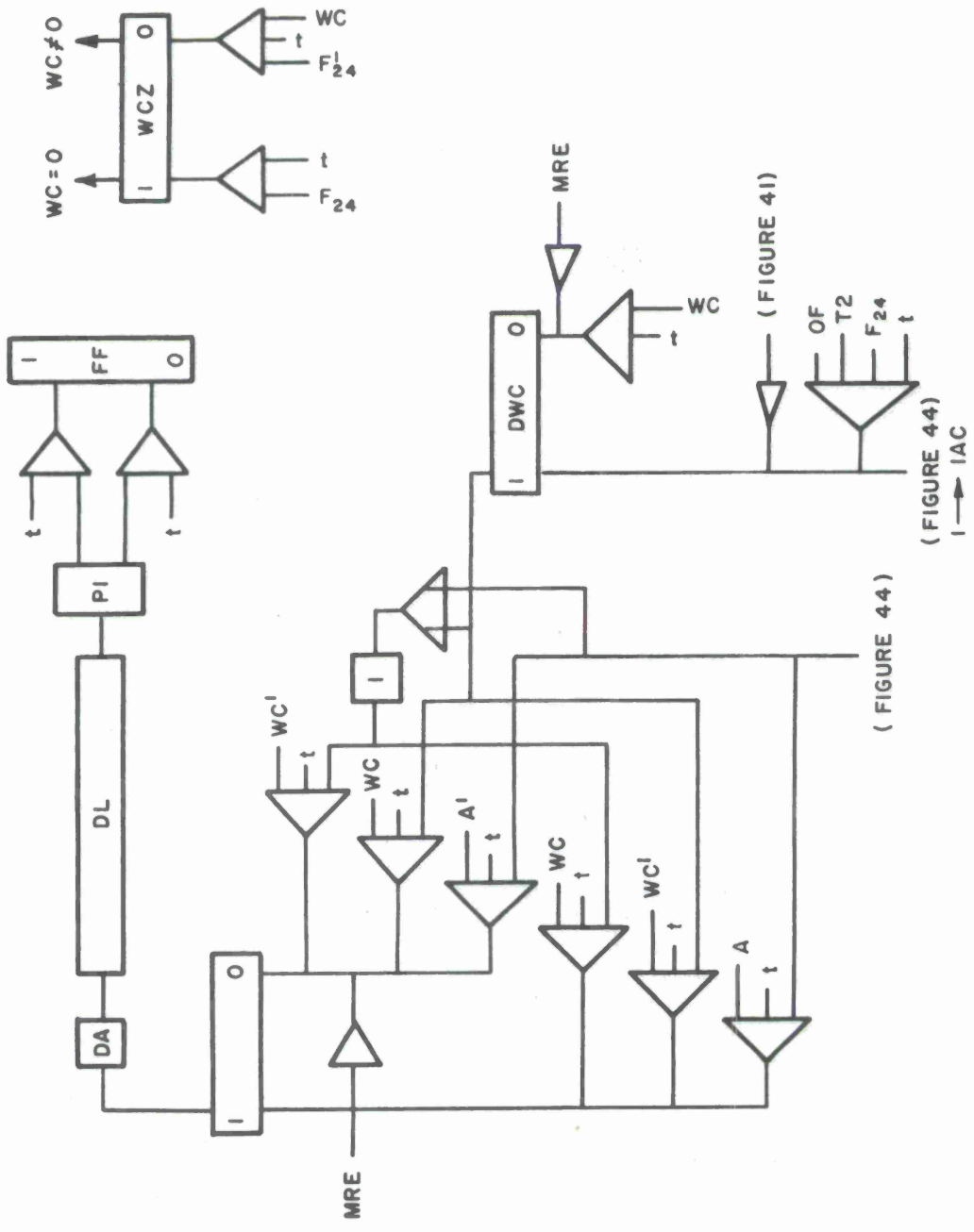



Figure 45. Word Counter

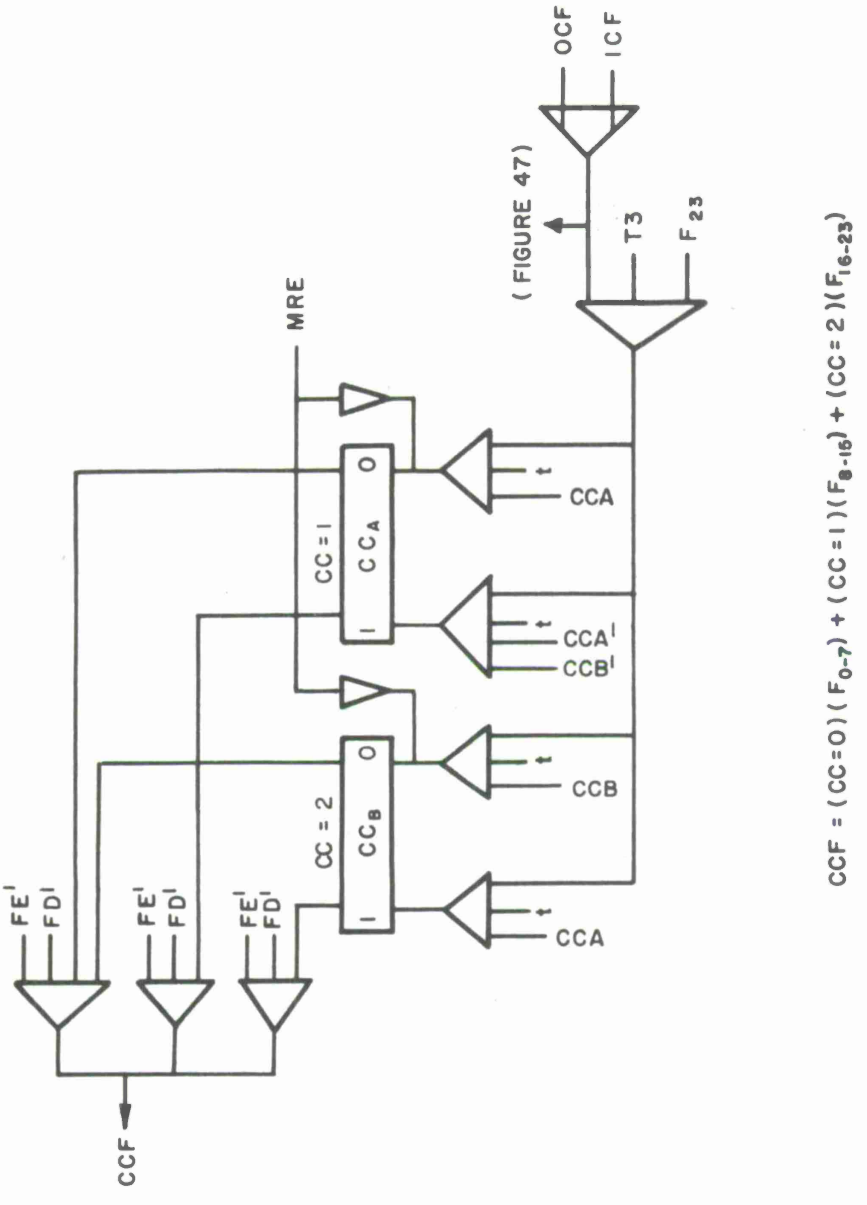



Figure 46. Character Counter

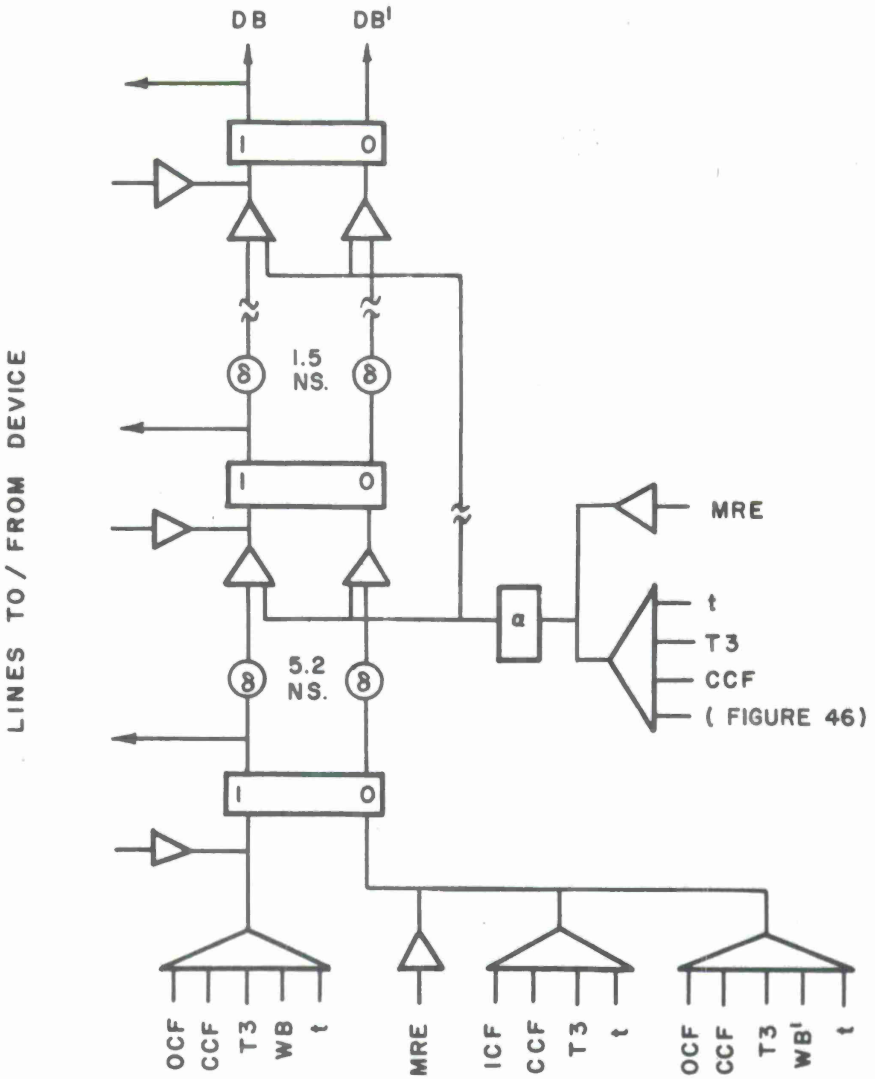



Figure 47. Device Buffer

Security Classification

## DOCUMENT CONTROL DATA - R&D

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*

| | |
|---|---|
| 1. ORIGINATING ACTIVITY *(Corporate author)*<br>The MITRE Corporation<br>Bedford, Mass. | 2a. REPORT SECURITY CLASSIFICATION<br>Unclassified<br>2b. GROUP |

3. REPORT TITLE

Logical Design for Fast Serial Computer

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

N/A

5. AUTHOR(S) *(Last name, first name, initial)*

Enderton, H. B.

| 6. REPORT DATE | 7a. TOTAL NO. OF PAGES | 7b. NO. OF REFS |
|---|---|---|
| July 1965 | 78 | 0 |

| | |
|---|---|
| 8a. CONTRACT OR GRANT NO.<br>AF19(628)-2390<br>b. PROJECT NO.<br>508<br>c.<br>d. | 9a. ORIGINATOR'S REPORT NUMBER(S)<br>ESD-TR-65-81<br>9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*<br>W-07189 |

10. AVAILABILITY/LIMITATION NOTICES

Qualified requestors may obtain from DDC.
DDC release to OTS authorized.

| 11. SUPPLEMENTARY NOTES | 12. SPONSORING MILITARY ACTIVITY |
|---|---|
| | Directorate of Computers.<br>Electronic Systems Division<br>L. G. Hanscom Field, Bedford, Mass. |

13. ABSTRACT

The detailed logical design is given for a serial digital computer using a 0.5-microsecond magnetic memory, 100-mc logical circuits, and one-word delay lines. The computer performs most instructions in 1 microsecond. A list is given of the amount of hardware used.

DD FORM 1 JAN 64 1473

Security Classification

| 14. KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|---|---|---|---|---|---|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| Digital Computers<br>Design<br>Logic | | | | | | |

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through

_____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through

_____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through

_____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional