

*Univac*<sup>®</sup>

LARC

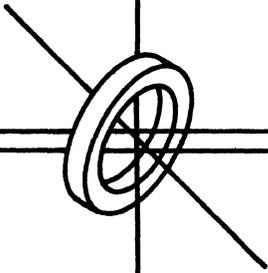
PROCESSOR

DISPATCHER

***Remington Rand Univac***<sup>®</sup>

DIVISION OF SPERRY RAND CORPORATION

UNIVAC ENGINEERING CENTER • PHILADELPHIA



*Univac*<sup>®</sup>  
LARC

---

# PROCESSOR

## DISPATCHER

COMPANY CONFIDENTIAL

Registered Circulation

The information contained in this publication is the property of the Sperry Rand Corporation. It is submitted in confidence and must not be disclosed to others except in confidence with the written permission of an officer of the Remington Rand Univac Division, Sperry Rand Corporation. This copy, issued in your name, is not to be reproduced or duplicated without permission in writing from a duly authorized representative of the Sperry Rand Corporation. This publication is subject to recall.

AUGUST 1960

***Remington Rand Univac***<sup>®</sup>  
DIVISION OF SPERRY RAND CORPORATION  
UNIVAC ENGINEERING CENTER • PHILADELPHIA

This manual has been reviewed by the engineering design staff and approved for issue.

Suggestions for additions, deletions, corrections, or improvements to this manual should be forwarded to

Publications Engineering Group (120)

Remington Rand Univac

Division of Sperry Rand Corporation

Box 5616

Philadelphia 29, Pa.

# CONTENTS

Heading	Title	Page
SECTION I. INTRODUCTION		
1-1.	Scope of This Manual . . . . .	1-1
1-2.	The Dispatcher: General . . . . .	1-1
1-3.	Main Functions . . . . .	1-3
1-4.	General Description . . . . .	1-3
SECTION II. PRIORITY CIRCUITS		
2-1.	General . . . . .	2-1
2-2.	Priority Within the Memory . . . . .	2-1
2-3.	Overflow and Underflow . . . . .	2-2
2-4.	Order of Priority . . . . .	2-2
2-5.	Special Underflow Precautions for Drum Write Synchronizers . . . . .	2-3
2-6.	Operation of Priority Circuits . . . . .	2-4
2-7.	Avoiding Time-Slot Waste . . . . .	2-5
2-8.	Conditions of Priority . . . . .	2-6
2-9.	Synchronizers Other Than Drum Synchronizers . . . . .	2-6
2-10.	Drum Read Synchronizers 1, 2, and 3 (S1, S2, S0) . . . . .	2-7
2-11.	Drum Write Synchronizer 1 (S3) . . . . .	2-9
2-12.	Drum Write Synchronizer 2 (S4) . . . . .	2-10
2-13.	Order of Time-Slot Assignment: Examples . . . . .	2-11
SECTION III. ADDRESS AND BUFFER REGISTERS		
3-1.	General . . . . .	3-1
3-2.	Basic Characteristics . . . . .	3-1
3-3.	Address-Register Matrix . . . . .	3-2
3-4.	Input-Buffer-Register Matrix . . . . .	3-4
3-5.	Output-Buffer-Register Matrix . . . . .	3-9
SECTION IV. ADDRESS AND INPUT-OUTPUT BUFFER CONTROLS		
4-1.	General . . . . .	4-1
4-2.	Address-Transfer Control . . . . .	4-1
4-3.	First Section . . . . .	4-3

Heading	Title	Page
4-4.	Second Section . . . . .	4-4
4-5.	Address-Register and Buffer-Register Controls . . . . .	4-5
4-6.	Address- and Buffer-Register Control for an Input Synchronizer j (Other Than Tape) . . . . .	4-5
4-7.	Address- and Buffer-Register Control for an Output Synchronizer j (Other Than Tape) . . . . .	4-9
4-8.	Address- and Buffer-Register Control for Tape Input-Output Synchronizer j	4-10
4-9.	Timing for Address and Input-Output Buffer Controls . . . . .	4-11

#### SECTION V. SYNCHRONIZER ADDRESS MODIFIER

5-1.	General . . . . .	5-1
5-2.	Capabilities of Address Modifier . . . . .	5-1
5-3.	Pattern of Modification . . . . .	5-3
5-4.	Address-Modifier Controls . . . . .	5-5
5-5.	Functions of a Digit Adder-Subtractor . . . . .	5-6
5-6.	Digit 1 Adder-Subtractor . . . . .	5-8
5-7.	Translating the Input Digit . . . . .	5-8
5-8.	Modifying the Translated Digit . . . . .	5-11
5-9.	Determining How Modification Affects Digit 2 . . . . .	5-12
5-10.	Translating Modified Digit Back to Larc Code . . . . .	5-13
5-11.	Differences Between Adder-Subtractors . . . . .	5-13
5-12.	Differences in Translation . . . . .	5-13
5-13.	Differences in Modification . . . . .	5-14
5-14.	Binary-Modification Detection . . . . .	5-15
5-15.	Parity-Modification Detection . . . . .	5-16
5-16.	Difference in Retranslation . . . . .	5-16
5-17.	Address-Modifier Timing . . . . .	5-16

#### SECTION VI. TYPICAL WORD-TRANSFER SEQUENCES

6-1.	General . . . . .	6-1
6-2.	Word Transfer 1 . . . . .	6-1
6-3.	Word Transfer 2 . . . . .	6-5

## ILLUSTRATIONS

Figure	Title	Page
1-1.	The Dispatcher: Simplified Block Diagram . .	1-2
2-1.	Priority Circuit For Synchronizer Other Than Drum Synchronizer: Simplified Logic .	2-8
2-2.	Priority Circuit for Drum Read Synchronizer: Simplified Logic . . . . .	2-8
2-3.	Priority Circuit for Drum Write Synchronizer 1 (S3): Simplified Logic . . . . .	2-9
2-4.	Priority Circuit for Drum Write Synchronizer 2 (S4): Simplified Logic . . . . .	2-11
2-5.	Example of Time-Slot Assignment for Drum Read Synchronizers 1, 2, and 3 (S1, S2, S0) . . . . .	2-12
2-6.	Examples of Time-Slot Assignment for Drum Read Synchronizer 1 (S1), Drum Write Synchronizer 1 (S3), and Line-Printer Synchronizer 2 (S6) . . . . .	2-13
3-1.	Core Windings: Schematic . . . . .	3-2
3-2.	Address-Register Matrix: Simplified Diagram	3-4
3-3.	Timing of Read-Write Cycle of an Address Register . . . . .	3-5
3-4.	Half Registers of Drum Read Synchronizers 1 and 2 (S1, S2) . . . . .	3-7
3-5.	Half Registers of Drum Write Synchronizer 1 (S3) and Line-Printer Synchronizer 1 (S5)	3-11
4-1.	Address-Transfer Control. Part 1 . . . . .	4-2
4-2.	Address-Transfer Control. Part 2 . . . . .	4-2
4-3.	Address- and Buffer-Register Controls for an Input Synchronizer j: Detailed Logic . . .	4-6
4-4.	Address- and Buffer-Register Controls for an Output Synchronizer j (Two Buffer Registers): Detailed Logic . . . . .	4-6
4-5.	Address- and Buffer-Register Controls for Tape Input-Output Synchronizer j: Detailed Logic . . . . .	4-7
4-6.	Memory-Transfer Control Cycle for Input Synchronizer j . . . . .	4-12
4-7.	Memory-Transfer Control Cycle for Output Synchronizer j . . . . .	4-13
5-1.	Synchronizer Address Modifier: General Block Diagram . . . . .	5-2
5-2.	Pattern of Modification by Add 1, Subtract 1, or No Change . . . . .	5-4
5-3.	Derivation of Control Signals . . . . .	5-7
5-4.	Digit 1 Adder-Subtractor: Simplified Logic .	5-9
5-5.	Derivation of Quinary Result of 2 . . . . .	5-11
5-6.	Address Modification: Timing Diagram . . .	5-17

Figure	Title	Page
6-1.	Typical Word-Transfer Sequences . . . . .	6-2
6-2.	The Dispatcher: Simplified Logic Diagram . .	6-3

Note on

**ENGINEERING DRAWINGS**

Larc Dispatcher

The following engineering drawings are referred to by number in figure 6-2 and alluded to throughout the text. They are listed here appropriately divided by section for this manual and by subject.

Section I

IOP Dispatcher: General Block Diagram D 809 411

Section II

IOP Priority Circuits. Part IA D 807 520  
 IOP Priority Circuits. Part II D 807 521  
 IOP Priority Circuits. Part III D 807 522  
 IOP Priority Circuits. Part IV D 807 523  
 W.E.C. Priority, Address, and Buffer Register Controls.  
 Part IA D 807 544

Section III

IOP Synchronizer Address Register D 807 534  
 IOP Input Buffer Register. Part I D 807 560  
 IOP Input Buffer Register. Part II D 807 561  
 IOP Input Buffer Register Expanded System. Part III D 807 562  
 IOP Input Buffer Register Expanded System. Part IV D 807 563  
 IOP Output Buffer Register. Part I D 807 564  
 IOP Output Buffer Register. Part II D 807 565  
 IOP Output Buffer Register Expanded System. Part III D 807 566  
 IOP Output Buffer Register Expanded System. Part IV D 807 567

Section IV

IOP Address Register Output Inverters D 807 529  
 IOP Synchronizer Address Pulseformers. Part I D 807 530  
 IOP Synchronizer Address Pulseformers. Part II D 807 531  
 Synchronizer Address Recirculation Gates D 807 532  
 IOP Synchronizer Address Transfer Control. Part IV D 807 533

IOP Address and Buffer Register Control Drum Read Synchronizers 1 and 2	D 807 550
IOP Address and Buffer Register Controls Drum Write Synchronizers 3 and 4	D 807 551
IOP Address and Buffer Register Control Line Printers	D 807 552
IOP Address and Buffer Register Control Card Reader and Charactron	D 807 553
W.E.C. Priority, Address, and Buffer Register Controls. Part IA	D 807 554
IOP Address and Buffer Register Control Tape Read-Write Synchronizer 1	D 807 555
IOP Address and Buffer Register Control Tape Read-Write Synchronizer 2	D 807 556
IOP Address and Buffer Register Control Tape Read-Write Synchronizer 3	D 807 557
IOP Address and Buffer Register Control Tape Read-Write Synchronizer 4	D 807 558
IOP Synchronizer Memory Address Control. Part V	D 807 559

Section V

Synchronizer Address Modifier Controls. Part II	D 807 535
Synchronizer Address Modifier Control. Part I	D 807 536
Synchronizer Address Modifier. Part IA	D 807 540
Synchronizer Address Modifier. Part IB	D 807 541
IOP Synchronizer Address Modifier. Part IIA	D 807 542
Synchronizer Address Modifier. Part IIB	D 807 543
IOP Synchronizer Address Modifier. Part IIIA	D 807 544
IOP Synchronizer Address Modifier. Part IIIB	D 807 545
IOP Synchronizer Address Modifier. Part IVA	D 807 546
Synchronizer Address Modifier. Part IVB	D 807 547
Synchronizer Address Modifier. Part VA	D 807 548
Synchronizer Address Modifier. Part VB	D 807 549

## TABLES

Table	Title	Page
2-1.	Order of Time-Slot Priority of Synchronizers	2-3
3-1.	Number and Type of Buffer Registers for Each Synchronizer . . . . .	3-13
5-1.	Address-Modifier Control Signals . . . . .	5-6
5-2.	Comparison of Larc and Seven-Bit Codes . . .	5-8



# SECTION I

## INTRODUCTION

### 1-1. SCOPE OF THIS MANUAL

The primary purpose of this manual is to provide training and reference information about the theory of operation of the logical circuits in the dispatcher section of the Larc processor. It applies equally to any Larc system, from the basic to the fully expanded. It does not, however, describe the physical characteristics of the system or the detailed circuit operation of basic logic elements such as gates, delay flops, and pulse-formers. Information on circuitry and the properties of these elements is available in Larc Circuitry, Section I, Standard Logic-Circuitry Types.

For practical reasons not every detail of every logical operation is described in this manual, but the content should provide the reader with the necessary background information for tracking down details on the engineering drawings, which define completely the logic of the dispatcher.

### 1-2. THE DISPATCHER: GENERAL

The dispatcher is a closely integrated group of circuits which coordinates and controls all data transfers to and from the memory for all of the synchronizers except the console-printer synchronizer.\*

A summary of the major functions and a general description of the dispatcher follow in this section. The description is based on and illustrated by the block diagram of figure 1-1. Sections II through V contain more detailed descriptions and illustrations of each of the subdivisions of the dispatcher, which are indicated by the blocks in figure 1-1. The descriptions of the dispatcher subdivisions provide the reader with the background for understanding Section VI, which describes the coordinated operation and timing of all sections of the dispatcher in performing typical sequences of memory transfers for the synchronizers. Both the detailed descriptions of the dispatcher subdivisions and the description of the sequences of transfer operations are illustrated by the simplified logic diagram of figure 6-2 and the timing diagram of figure 6-1.

---

\* Data transfers for the console-printer synchronizer are handled by the processor program.

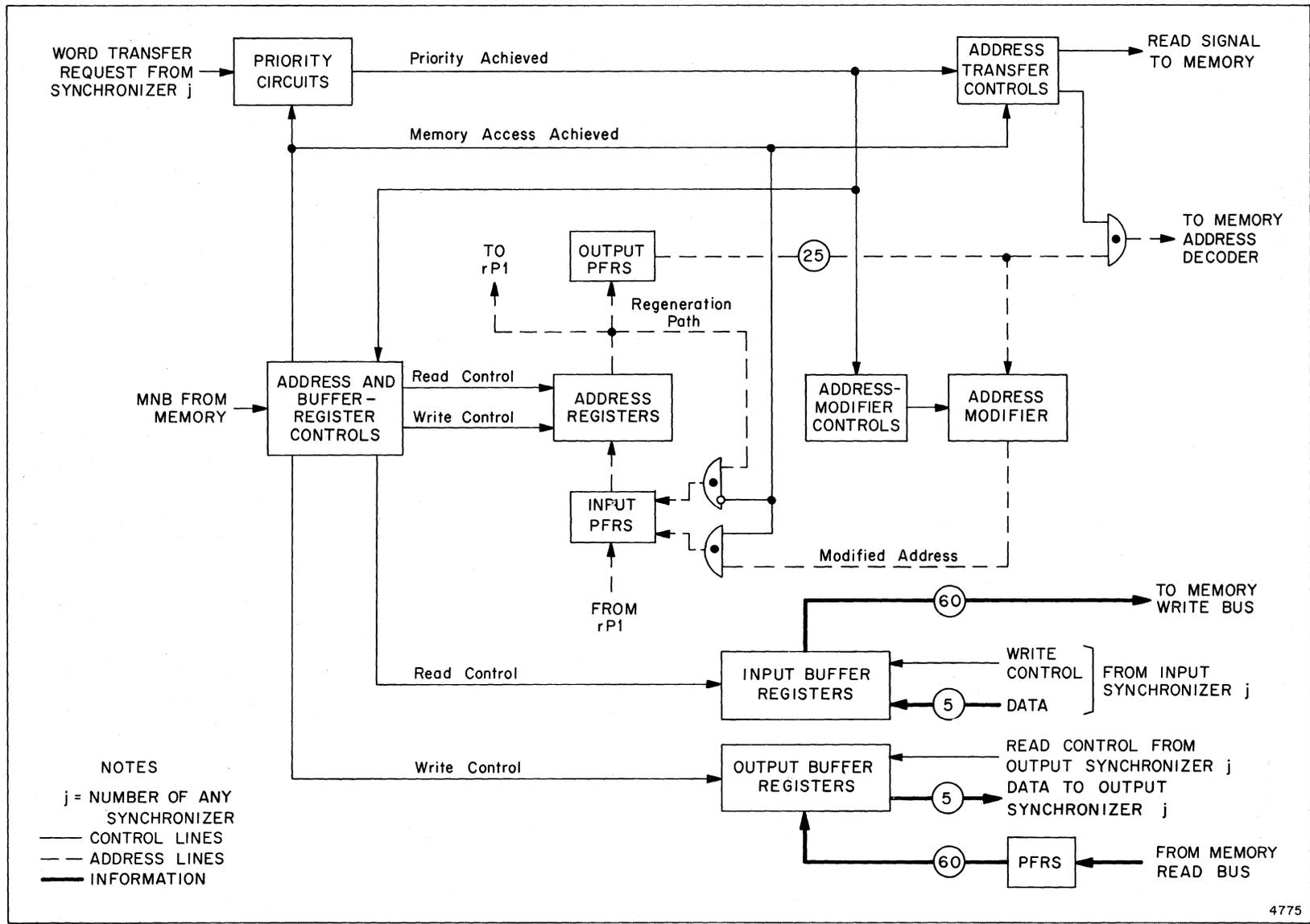


Figure 1-1. The Dispatcher: Simplified Block Diagram

### 1-3. MAIN FUNCTIONS

These are the main functions of the dispatcher.

(1) To define, on the basis of preassigned priority ratings, the time and order in which memory transfers requested by the synchronizers will be performed.

(2) To store a memory address for each operating synchronizer and use the address to initiate a memory-transfer operation requested by the synchronizer.

(3) To modify the memory address after each memory reference and, after access to the memory is assured, store the modified address in preparation for the next memory reference for the synchronizer.

(4) To regenerate the address for another attempt if access to the memory is denied.

(5) To control the transfer of data between the memory bus and a buffer register for the synchronizer.

(6) To inform the synchronizer of the completion of the transfer and of any errors detected while the transfer was being made.

### 1-4. GENERAL DESCRIPTION

The dispatcher is essentially an auxiliary of the synchronizers for dispatching input-output data between the memory and a number of input or output buffer registers associated with the synchronizers (figure 1-1). There are from one to four 12-digit buffer registers within the dispatcher for each synchronizer in the system; they are tape-wound magnetic-core registers. The cores of the input and output registers are arranged in separate matrices. They provide temporary storage for data being transferred between the memory and an input-output device and provide the means by which the data are converted from serial-parallel to parallel form or from parallel to serial-parallel form.

The flow of data between the buffer registers and the input-output devices is controlled by the synchronizers. The rate and order of flow is governed primarily by the characteristics and operating mode of the particular synchronizer and input-output device concerned. In most cases data are transferred to or from a buffer register by a synchronizer one digit at a time. The time interval between digits is dependent on the reading or recording rate of the synchronizer.

Whenever a synchronizer completes the processing and transfer of a word of data to or from a buffer register, it signals the dispatcher to

(1) Transfer the word to the main memory if it is an input synchronizer, or

(2) Obtain a new word from memory if it is an output synchronizer.

Since the synchronizers process data asynchronously with respect to one another, several may signal the dispatcher at the same time for a transfer of data. Such conflicts are resolved by priority circuits within the dispatcher which determine, on the basis of preassigned priority ratings, the order in which the data will be transferred for the synchronizers. The priority rating assigned to a synchronizer reflects, in general, the rate at which the synchronizer must have information transferred in order to maintain without interruption the flow of data required by the input-output device.

The synchronizers and the dispatcher together are designed to process a number of words of input-output data automatically without direct intervention by the processor program. For example, once it is started, a drum read synchronizer with the aid of the dispatcher reads from a drum and transfers a multiple of 100 words of data to successive locations in the memory. To accomplish this, means must be available to the dispatcher for keeping track of the memory address of every word transferred. The dispatcher therefore contains for each synchronizer a register in which is stored for the synchronizer the memory address of the next word to be transferred to or from the memory. Whenever the synchronizer signals for and receives priority, this address is read out of the address register and sent to the memory-address decoder to initiate a word-transfer operation. The address is also sent to an address modifier where it is modified before being written back into the same register after access to the memory is assured. The modified address then is the address of the next word to be transferred for the synchronizer.

The beginning memory address for a particular input-output operation is supplied by the processor program. Instructions are provided for transferring a beginning address from register P1 to any one of the address registers or for transferring the contents of any address register to register P1. The latter transfer is normally made to perform a program check on the address.

The address modifier is a special-purpose adder-subtractor which is time-shared to modify addresses for the input-output operations of all the synchronizers. Because data transfers are usually made to or from successive memory locations in ascending order, an address will, in most cases, be modified by adding 1. However, depending upon the type of synchronizer and the mode in which it is operating, an address is sometimes modified by subtracting a number or by adding a number other than 1. For example, during a backward tape-read operation the number 1 is subtracted from the address. Mode signals indicating what number is to be added or subtracted from the address are sent to the address-modifier controls by the synchronizers.

After initiating a word-transfer operation for a synchronizer by sending an address to the memory-address decoder, the dispatcher subsequently completes the transfer, provided that the memory is not busy. The transfer is completed during an input operation through reading the word out of an input-buffer register of the synchronizer and sending it to the memory over the write bus. During an output operation the word is received from the memory over the read bus and is written into an output-buffer register of the synchronizer. When a word-transfer operation is completed, the dispatcher signals the appropriate synchronizer. If a memory-select or HSB odd-even error is detected during the transfer operation, the dispatcher sends an error signal to the appropriate synchronizer.

## SECTION II

### PRIORITY CIRCUITS

#### 2-1. GENERAL

Every 2  $\mu\text{s}$  the synchronizers as a group are allotted a 0.5- $\mu\text{s}$  time slot, during which any one synchronizer is permitted to attempt a word transfer over the memory high-speed-bus system. Since the synchronizers process information asynchronously with respect to one another, more than one synchronizer may require access to the memory at the same time. The priority circuits are therefore designed to remedy this problem by determining which synchronizer shall use the time slot on a basis of preassigned priority. This system of priority ratings reflects primarily the characteristic data-transfer rates of the input-output devices controlled by the synchronizers, so that the faster an input-output device, the higher the memory-access priority allotted to its synchronizer. Priority ratings also depend, though to a lesser extent, upon the buffer-storage capacity of the various synchronizers—that is, whether a synchronizer has one or two buffer registers.

#### 2-2. PRIORITY WITHIN THE MEMORY

In transferring information a synchronizer may be required not only to compete with other synchronizers for the available time slots but also to compete with other synchronizers, the computing unit, and the central processor for access to a particular memory unit.

Consequently, even though a time slot is assigned to a synchronizer on a priority basis, it may still be wasted, which results in a delay in the transfer of a word if the memory unit addressed happens to be busy. Although the programmer will avoid, in general, simultaneous reference to the same memory unit by two or more of the sections of the system (computing units, central processor, or dispatcher), such situations will occasionally be unavoidable. If a system of priorities on the memory did not exist, continuous reference to the memory unit (once every 4  $\mu\text{s}$ ) by the computing unit or the central processor would continuously prevent the dispatcher from gaining access. This condition could not be tolerated since a synchronizer must transfer a word within a definite period of time to keep pace with the continuous data-transfer rate of its input-output device.

Each memory unit is therefore provided with a priority system which insures that a synchronizer will gain access to it within a maximum time of 4  $\mu$ s after addressing it. If a synchronizer addresses a memory unit within 4  $\mu$ s after a memory cycle is initiated, a synchronizer-interlock FF is set in that memory unit to prevent either of the computing units or the central processor from gaining access to the unit when the current memory cycle ends, so that the synchronizer will at that time be able to make a successful memory reference to the unit. Since the central processor must monitor within a definite period of time each input-output operation being performed, it also receives the same type of priority over the computing units by means of an IOP-interlock FF. This flip-flop is set if the central processor addresses a busy memory unit, to prevent either computing unit from gaining access to the memory unit until the processor successfully completes its memory reference. When the synchronizer-interlock FF and the IOP-interlock FF are both set, the synchronizer-interlock FF takes precedence.

### 2-3. OVERFLOW AND UNDERFLOW

The whole scheme of assigning memory-reference priorities among the synchronizers and of giving the dispatcher precedence over the other sections of the Larc system in memory-unit reference has essentially one purpose: to insure that each operating synchronizer will gain memory access frequently enough to keep pace with the continuous data-transfer rate of its input or output device. Failure to maintain this frequency of access for a synchronizer would disrupt the input-output operation because it would result in either of two conditions—overflow or underflow, depending upon the direction of information transfer.

The condition of overflow would occur if the synchronizer for an input device failed to have the contents of its buffer register transferred to the memory before the input device was ready to load the same buffer register again with another word of information. Similarly, underflow would occur if the synchronizer for an output device did not have its buffer register loaded with a word obtained from the memory by the time the output device was ready to accept the new word. Either condition would interrupt the continuity of information flow for the input-output device concerned. The mnemonic designation of a signal indicating the nearing-underflow condition is OF.

### 2-4. ORDER OF PRIORITY

The priority circuits assign the time slots to the synchronizers in the order of priority listed in table 2-1. Note, however, that the numerical designations of the synchronizers, of their signal names, and of the priority circuits associated with them, are not an indication of priority rating. The drum synchronizers have the highest priority rating because they operate at the highest data-transfer rate and are most susceptible to overflow and underflow. Although the line printers have an output rate which is less than that of the tape units, the printer synchronizers have higher priority because they contain only one output-buffer register each, as opposed to two each in the tape synchronizers. Further, since a line printer requires numerous references to the memory for every word printed, its synchronizer is therefore more susceptible to underflow than a tape synchronizer. The

Table 2-1. Order of Time-Slot  
Priority of Synchronizers

Priority	Name of Synchronizer	Number of Synchronizer
1	Drum read 1	1
2	Drum read 2	2
3	Drum read 3	0
4	Drum write 1	3
5	Drum write 2	4
6	Line printer 1	5
7	Line printer 2	6
8	Card reader	11
9	Tape 1	7
10	Tape 2	8
11	Tape 3	9
12	Tape 4	10
13	Electronic page recorder	12

card reader has a comparatively high data-input rate per card, which places its synchronizer in a priority rating below the synchronizers of the line printers but above those of the tapes, even though the card-reader synchronizer has two input-buffer registers. The electronic-page-recorder synchronizer is last in order of priority because the recorder has the characteristic of not requiring information at a continuous rate.

The priority assignments among synchronizers of the same type are arbitrary.

## 2-5. SPECIAL UNDERFLOW PRECAUTIONS FOR DRUM WRITE SYNCHRONIZERS

The arbitrary assignment of priority among drum synchronizers might, under certain conditions, lead to underflow, since the drum write synchronizers are assigned a lower priority, although they operate at the same data-transfer rate as the drum read synchronizers. The worst possible condition is one in which more than one drum synchronizer refers to the same memory unit. Under such a condition a low-priority drum write synchronizer nearing underflow might not be able to achieve memory access in time because (1) its requests for a time slot might be overridden by requests from a drum synchronizer of higher priority, or (2) if it was allotted a time slot, the time slot might be wasted because the memory was busy with a read or write operation for a drum synchronizer of higher priority. To obviate such a situation, the priority circuits for the drum synchronizers are designed

in such a way that a drum write synchronizer of low priority nearing underflow will override any drum synchronizer of higher priority for memory reference. Thus, if either drum write synchronizer 1 (S3) or 2 (S4) approaches underflow, it receives first priority; if both approach underflow, drum write synchronizer 2 (S4) receives first priority and 1 (S3) second.

## 2-6. OPERATION OF PRIORITY CIRCUITS

An operating synchronizer will require access to the memory (1) if it is an input synchronizer whose input-buffer register is filled with a word ready to be sent to the memory, or (2) if it is an output synchronizer whose output-buffer register is empty and ready for a new word from the memory.

Whenever memory access is required by synchronizer j,\* a word-transfer-request FF in the synchronizer is set. The set output signal of this FF (SjRF) alerts the synchronizer-j priority circuit to begin competing with the priority circuits of all other operating synchronizers for the earliest time slot to which synchronizer j is entitled. The word-transfer-request FF will remain set until synchronizer j makes a successful memory reference; it is reset by the signal  $\overline{SjRFR}$  generated in the buffer-register control circuits when the memory-not-busy signal is received.

In the case of a fast-operating drum write synchronizer (S3 or S4), in which two output-buffer registers are used, the word-transfer-request FF will be set as soon as one of the registers is empty. If the synchronizer still has not gained memory access for this register—that is, the word-transfer-request FF is still set—by the time a counter indicates that only eight digits remain in the other register, a nearing-underflow signal  $\overline{S30F}$  or  $\overline{S40F}$  is sent by the synchronizer to the priority circuits to insure that it receives first priority.

Whenever the correct conditions prevail for synchronizer j to request memory access, the output signal of the set synchronizer-j word-transfer-request FF (SjRF) sets the synchronizer-j priority-bid FF at T26 for a minimum of 2  $\mu$ s (figures 2-1 through 2-4). The output of this FF is gated with the outputs of the priority-bid FFs for the other synchronizers in such a way that signal PjA,B, indicating that priority is granted to synchronizer j, will be produced only if the priority-bid FFs of no other synchronizers of higher priority are set. While an address is being transferred between an address register and register P1, function signal  $\overline{820}$  is present at the output gating of all priority-bid FFs to block all priority assignments.

When produced, priority signal PjA,B alerts circuits within the address- and buffer-register controls, and the address-modifier controls. These controls, at the correct time, set the address-modifier controls for synchronizer j, read the address out of the synchronizer-j address register, gate the address to the memory-address decoder, and write the same address back into the address register. The priority signal PjA,B is also delayed by the delayed-priority FF to produce delayed-priority signal PjD for 2  $\mu$ s. Signal PjD alerts the circuits in the address- and buffer-register controls; if the memory is not busy, these circuits will

---

\* j is any synchronizer.

(1) Clear the old address out of the synchronizer-j address register;

(2) Write in the modified address;

and

(3) Write the new word coming from memory into the correct output-buffer register if synchronizer j is an output synchronizer;

or

(4) Read the word out of the correct input-buffer register for transmission to the memory if synchronizer j is an input synchronizer.

A signal  $\overline{S_jMD}$  derived from gating  $P_jD$  with the MNB signal is used to indicate that priority for synchronizer j is no longer required and resets the synchronizer-j priority-bid FF. The  $\overline{S_jRFR}$  signal, which is similarly derived, resets the word-transfer-request FF in the synchronizer.

## 2-7. AVOIDING TIME-SLOT WASTE

The  $\overline{S_jMD}$  signal, indicative of whether synchronizer j did or did not make a successful memory reference when a time slot was assigned to it, is returned to the synchronizer-j priority circuit after priority to the following time slot has already been assigned. Whether or not synchronizer j makes a successful reference on the first time slot, the following time slot is given to the next calling synchronizer; it is not reserved for synchronizer j to use if it failed to gain access on its first attempt. If the following time slot were reserved in this way and synchronizer j gained access on its first attempt, the reserved time slot would be wasted. If under normal conditions synchronizer j fails to gain memory access on its allocated time slot, it has to relinquish the following time slot to another calling synchronizer and then compete again for a time slot. Because of the delay in returning the  $\overline{S_jMD}$  signal, if synchronizer j happens to be the only synchronizer requesting a time slot, its priority circuits will assign it one more time slot after the one on which it made a successful memory reference. This extra time-slot assignment will be ineffective, however, since the  $\overline{S_jMD}$  signal is also used to inhibit the address control circuits from gating the address to the memory, thereby preventing the initiation of the second memory reference. Since synchronizer j is the only synchronizer addressing the memory, this case cannot be considered as wasting a time slot.

There are, however, instances in which time slots will pass unused in order to insure that (1) all drum synchronizers requesting priority will achieve memory access before any synchronizer requesting priority other than a drum synchronizer and (2) any drum write synchronizer nearing underflow will gain memory access before any other synchronizer requesting priority. The basic system of priority ratings will always give a drum write synchronizer near underflow the highest time-slot priority of all, and will always give the drum synchronizers higher time-slot priority than the synchronizers other than drum synchronizers, but this system alone does not assure the drum synchronizers of first memory access. To provide such assurance, the

priority circuits of the drum synchronizers are designed to operate in the following manner:

(1) If there are no other drum synchronizers requesting priority, the last requesting drum synchronizer receives as many successive time slots as may be necessary to gain memory access, plus an extra unused one, regardless of any outstanding priority demand by a synchronizer other than a drum synchronizer.

(2) If a drum write synchronizer is in the nearing-underflow state, it will receive highest priority and as many successive time slots as may be needed to gain memory access, plus an extra unused one, regardless of which other synchronizers have outstanding demands (drum write, drum read, or other). The one exception to this situation is when drum write synchronizer 1 (S3) is nearing underflow and drum write synchronizer 2 (S4) is requesting priority in the normal way. In this event drum write synchronizer 1 (S3) will receive the higher priority but must relinquish the next time slot to drum write synchronizer 2 (S4) even though it (S3) may not have gained memory access. Then, of course, whether drum write synchronizer 2 (S4) gained memory access or not, priority will revert to drum write synchronizer 1 (S3). If the demand of drum write synchronizer 2 (S4) is still outstanding when drum write synchronizer 1 (S3) gains access, there will be no extra unused time slot.

(3) If both drum write synchronizers 1 and 2 (S3 and S4) approach underflow simultaneously, drum write synchronizer 2 (S4) will immediately have the kind of priority described first in the preceding item (2), and drum write synchronizer 1 (S3) will then assume precisely the same kind of priority after drum write synchronizer 2 (S4) had gained memory access and the unused time slot has elapsed.

## 2-8. CONDITIONS OF PRIORITY

The synchronizer is granted or refused priority under the conditions listed under headings 2-9 through 2-12. Refer to logical drawings D807520 ...524. The signals representing the conditions are in parentheses. The letter j represents the designatory number of any synchronizer, and the 'ands' and 'ors' used between conditions are logical ANDs and ORs.

### 2-9. SYNCHRONIZERS OTHER THAN DRUM SYNCHRONIZERS

Figure 2-1 shows the priority circuit for synchronizers other than drum synchronizers.

Priority is granted to synchronizer j when the following conditions prevail:

(1) The synchronizer-j word-transfer-request FF is set (SjRF low);

and

(2) Synchronizer j was not granted priority on the preceding time slot ( $\overline{PjD}$  low);

and

(3) No higher priority demands are outstanding ( $\overline{\text{PnF}}$  low\*).

Priority for synchronizer j is refused when one of the following conditions occurs:

(1) A demand of higher priority arises ( $\overline{\text{PnF}}$  high);

or

(2) A demand of lower priority arises ( $\overline{\text{PmF}}$  high\*\*), and synchronizer j was assigned priority on the preceding time slot ( $\overline{\text{PjD}}$  high);

or

(3) Synchronizer j makes a successful memory reference ( $\overline{\text{SjMD}}$  high).

#### 2-10. DRUM READ SYNCHRONIZERS 1, 2, AND 3 (S1, S2, S0)

Figure 2-2 shows the priority circuit for drum read synchronizers 1, 2, and 3 (S1, S2, and S0).

Priority is granted to drum read synchronizer 1, 2, or 3 (S1, S2, or S0) when the following conditions prevail:

(1) The synchronizer word-transfer-request FF is set (S1RF, S2RF, or SORF low);

and

(2) Underflow is not imminent on drum write synchronizers 1 or 2 (S3 or S4) ( $\overline{\text{S30F}}$ ,  $\overline{\text{S40F}}$  low);

and

(3) The synchronizer was not granted priority to the preceding time slot ( $\overline{\text{P1D}}$ ,  $\overline{\text{P2D}}$ , or  $\overline{\text{P0D}}$  low);

and

(4) A demand of higher priority is not outstanding ( $\overline{\text{P1F}}$  low for drum read synchronizer 2 (S2),  $\overline{\text{P1F}}$  or  $\overline{\text{P2F}}$  low for drum read synchronizer 3 (S0).

Priority for drum read synchronizer 1, 2, or 3 (S1, S2, or S0) is refused when one of the following conditions arises:

(1) Underflow becomes imminent on drum write synchronizers 1 or 2 (S3 or S4) ( $\overline{\text{S30F}}$  or  $\overline{\text{S40F}}$  high);

\* n is any synchronizer of higher priority rating than j.

\*\* m is any synchronizer of lower priority rating than j.

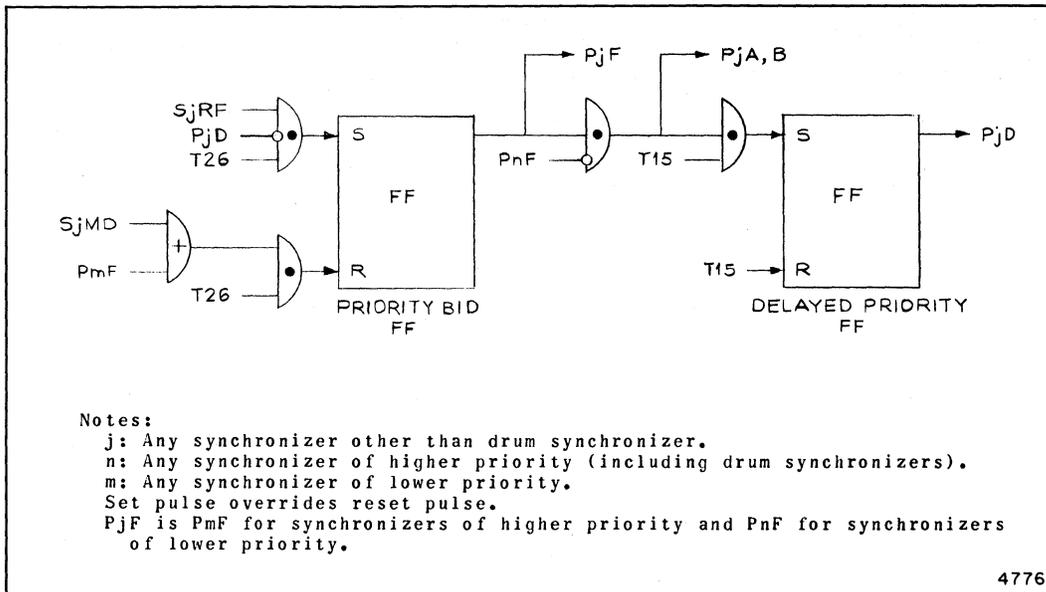


Figure 2-1. Priority Circuit for Synchronizer Other than Drum Synchronizer: Simplified Logic

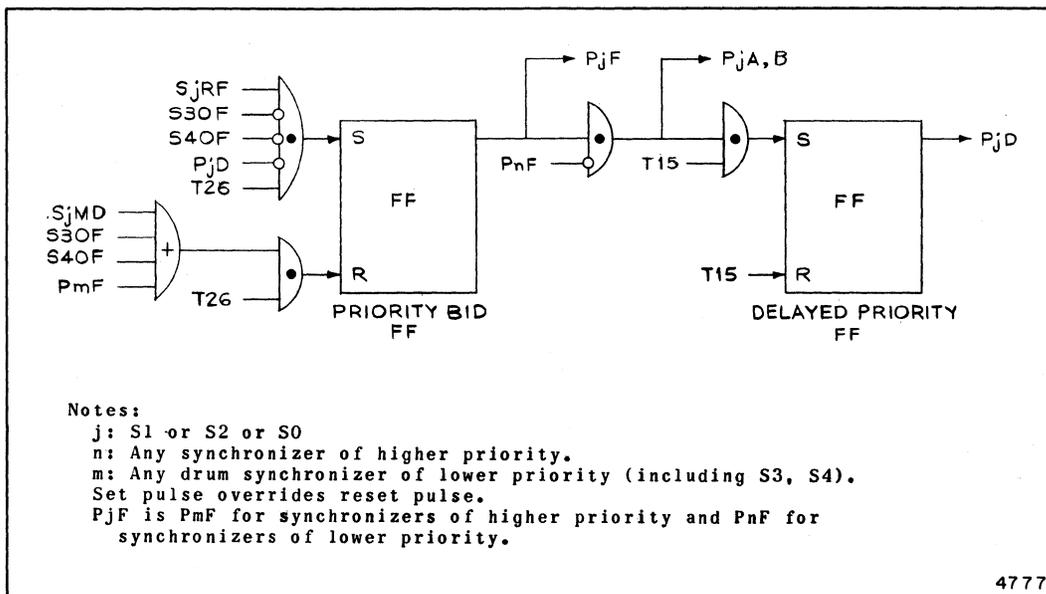


Figure 2-2. Priority Circuit for Drum Read Synchronizer: Simplified Logic

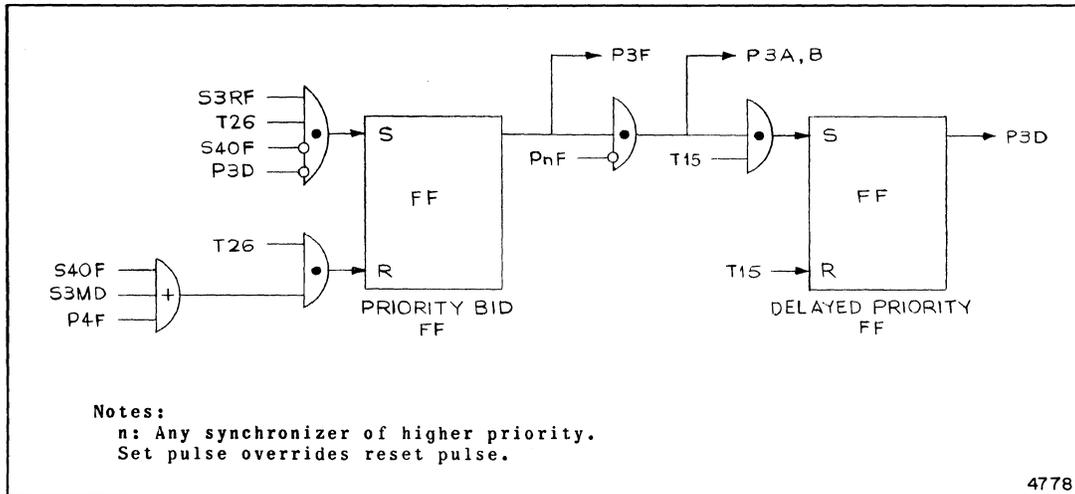


Figure 2-3. Priority Circuit for Drum Write Synchronizer 1 (S3): Simplified Logic

or

(2) The synchronizer makes a successful memory reference ( $\overline{S1MD}$ ,  $\overline{S2MD}$ , or  $\overline{S0MD}$  high);

or

(3) A demand of higher priority arises ( $\overline{P1F}$  high for drum read synchronizer 2 (S2);  $\overline{P1F}$ ,  $\overline{P2F}$  high for drum read synchronizer 3 (S0));

or

(4) The synchronizer was granted priority on the preceding time slot ( $\overline{P1D}$ ,  $\overline{P2D}$ , or  $\overline{P0D}$  high) and a demand of lower priority arises from a drum synchronizer ( $\overline{P2F}$ ,  $\overline{P0F}$ ,  $\overline{P3F}$ , or  $\overline{P4F}$  high for drum read synchronizer 1 (S1);  $\overline{P0F}$ ,  $\overline{P3F}$  or  $\overline{P4F}$  high for drum read synchronizer 2 (S2);  $\overline{P3F}$ , or  $\overline{P4F}$  high for drum read synchronizer 3 (S0)).

#### 2-11. DRUM WRITE SYNCHRONIZER 1 (S3)

Figure 2-3 shows the priority circuit for drum write synchronizer 1 (S3).

Priority is granted to drum write synchronizer 1 (S3) under the following conditions:

(1) The word-transfer-request FF of drum write synchronizer 1 (S3) is set (S3RF low);

and

(2) Drum write synchronizer 1 (S3) was not granted priority on the preceding time slot ( $\overline{P3D}$  low);

and

(3) Underflow on drum write synchronizer 2 (S4) is not imminent ( $\overline{S40F}$  low);

and

(4) No demands of higher priority are outstanding ( $\overline{P1F}$ ,  $\overline{P2F}$  and  $\overline{POF}$  low).

Priority for drum write synchronizer 1 (S3) is refused when

(1) Underflow on drum write synchronizer 2 (S4) is imminent ( $\overline{S40F}$  high).

or

(2) Drum write synchronizer 1 (S3) was granted priority on the preceding time slot ( $\overline{P3D}$  high) and a demand of lower priority arises from a drum synchronizer ( $\overline{P4F}$  high).

or

(3) A demand of higher priority arises ( $\overline{P1F}$ ,  $\overline{P2F}$ , or  $\overline{POF}$  high);

or

(4) Drum write synchronizer 1 (S3) makes a successful memory reference ( $\overline{S3MD}$  high).

## 2-12. DRUM WRITE SYNCHRONIZER 2 (S4)

Figure 2-4 shows the priority circuit for drum write synchronizer 2 (S4).

Priority is granted to drum write synchronizer 2 (S4) when the following conditions prevail.

(1) The word-transfer-request FF of drum write synchronizer 2 (S4) is set ( $\overline{S4RF}$  low);

and

(2) Drum write synchronizer 2 (S4) was not assigned priority to the preceding time slot ( $\overline{P4D}$  low);

and

(3) No demand of higher priority is outstanding ( $\overline{P1F}$ ,  $\overline{P2F}$ ,  $\overline{POF}$ , and  $\overline{P3F}$  low).

Priority for drum write synchronizer 2 (S4) is refused when

(1) A demand of higher priority arises ( $\overline{P1F}$ ,  $\overline{P2F}$ ,  $\overline{POF}$ , or  $\overline{P3F}$  high);

or

(2) Drum write synchronizer 2 (S4) makes a successful memory reference (S4MD high).

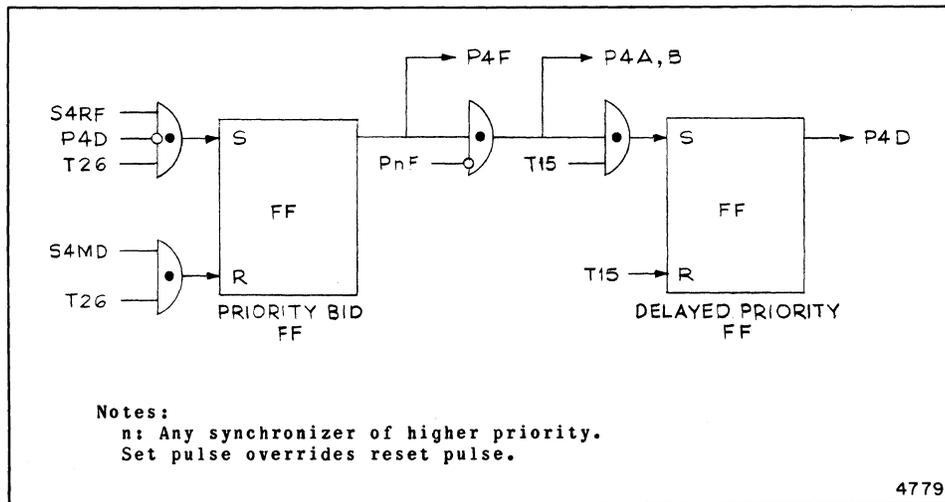


Figure 2-4. Priority Circuit for Drum Write Synchronizer 2 (S4): Simplified Logic

## 2-13. ORDER OF TIME-SLOT ASSIGNMENT: EXAMPLES

Figures 2-5 and 2-6 are timing diagrams chosen as examples of the sequence in which calling synchronizers receive priority to the synchronizer time slots, depending upon various conditions such as whether the memory is busy when addressed or whether the priority circuits of the drum synchronizers are able to insure that the drum synchronizers will achieve first memory access.

Figure 2-5 illustrates a competition for synchronizer time slots among drum read synchronizers 1, 2, and 3 (S1, S2, and S0). Since drum read synchronizer 1 (S1) has the highest priority rating, it receives priority to the first time slot. The second time slot is allotted to drum read synchronizer 2 (S2). In the meantime, however, drum read synchronizer 1 (S1) has failed to gain memory access, and its priority request is therefore outstanding to override that of drum read synchronizer 3 (S0) for the third time slot. The fourth time slot then is assigned to drum read synchronizer 3 (S0), and, since by this time drum read synchronizer 3 (S0) is the only calling synchronizer, it also receives the fifth time slot, although it does not need it.

The timing chart of figure 2-6 shows the competition for synchronizer time slots and for memory access between drum read synchronizer 1 (S1), drum write synchronizer 1 (S3) nearing underflow, and line-printer synchron-





that no drum synchronizer is requesting priority, line-printer synchronizer 2 (S6) receives the sixth time slot, on which it gains memory access in this example. Since line-printer synchronizer 2 (S6) is now the only calling synchronizer of any kind, it also receives but does not use the seventh time slot.

## SECTION III

### ADDRESS AND BUFFER REGISTERS

#### 3-1. GENERAL

Within the dispatcher there is an address register for each synchronizer in the system, two input-buffer registers for each input synchronizer, and one or two output-buffer registers for each output synchronizer. Each address register is a five-digit register which stores the memory address of the next word to be transferred to or from the memory for a particular operating synchronizer. Each buffer register is a 12-digit register used as temporary storage for data in the process of being transferred between the memory and an input or output device controlled by a particular synchronizer. Table 3-1 lists the number and type of one-word buffer registers required for each type of synchronizer.

#### 3-2. BASIC CHARACTERISTICS

The address and buffer registers are composed of magnetic cores arranged in separate address-register, input-buffer-register, and output-buffer-register matrices. The same circuit elements are used in all three matrices, although there is considerable difference in the overall configurations of the matrices which reflects the differences in their function.

The basic circuit element of the register matrices is a fast-switching tape-wound magnetic core (figure 3-1). Each core has three windings—a write winding, a read winding, and an output winding. The read and write windings are driven, respectively, by magnetic-amplifier read and write drivers.

Reading or clearing a core requires a negative pulse from the read driver at point 1. This negative pulse will tend to switch the core to the cleared or 0 state. If a binary 1 is initially stored in the core, the read driver will switch the core to the 0 state and produce a positive output pulse representing a 1 at point 2. If, however, the core is initially in the 0 state, the core will not be switched and point 2 will remain low, indicating a 0 output.

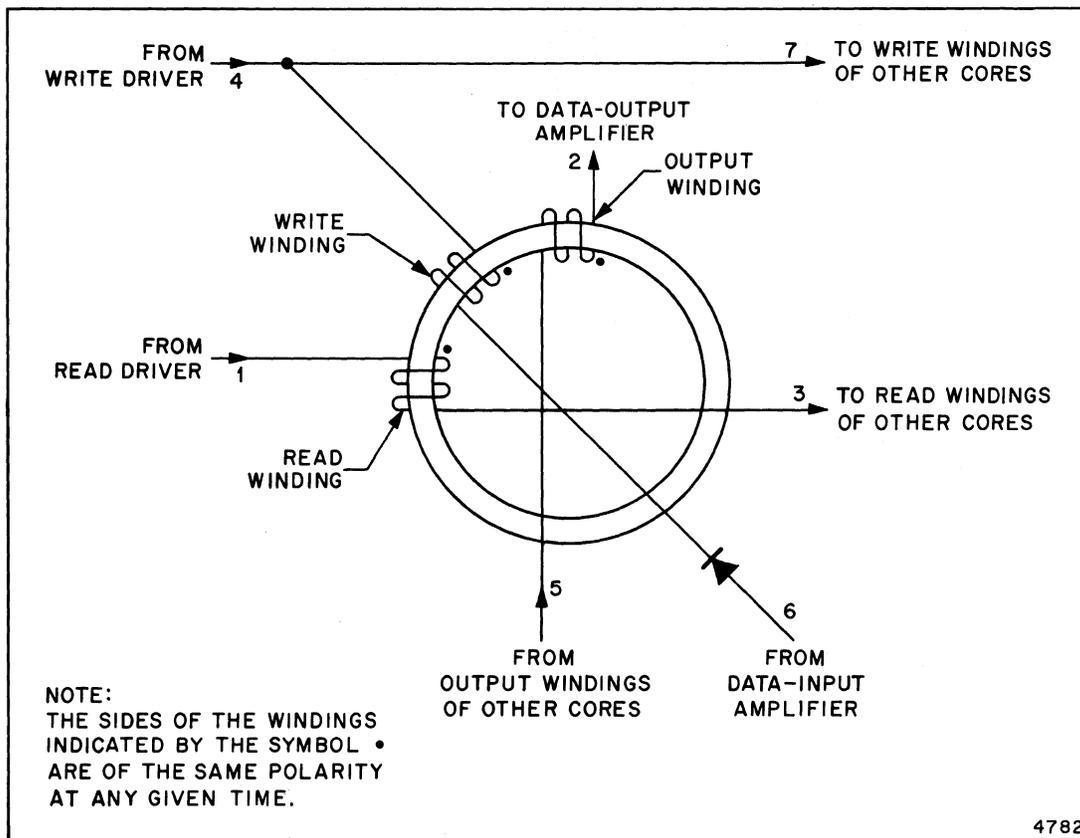


Figure 3-1. Core Windings: Schematic

To write information into a core requires a negative pulse from the write driver at point 4. Whether or not the core switches depends on the data input at point 6. If the core is assumed to be initially in the cleared or 0 state, it will switch to the 1 state only if a positive pulse (representing a 1) appears at point 6 coincident with the negative pulse from the write driver at point 4. If either point 4 is positive or point 6 is negative, the core will remain in the initial 0 state.

A magnetic-amplifier read or write driver is set by a 1- $\mu$ s negative pulse. When set, it generates a 0.5- $\mu$ s negative clock pulse. The read- and write-driver clock pulses are 180 degrees out of phase with respect to each other. All the read magnetic amplifiers are driven by an A-phase clock, and all of the write magnetic amplifiers by a B-phase clock, except in the output-buffer registers, where the write magnetic amplifiers are driven by an A-phase clock, and the read magnetic amplifiers by a B-phase clock. Both clocks are synchronized with the pulseformer clock that re-times signals throughout the logical circuits.

### 3-3. ADDRESS-REGISTER MATRIX

An address register is a 25-bit magnetic-core register which is used to store the address of the next word to be transferred between the memory and the buffer register of a particular synchronizer. A different address register is associated with each synchronizer. The registers for all of

the synchronizers in the system are contained within a 25 x n matrix of magnetic cores.† The address-register matrix is shown in simplified form in figure 3-2.

The matrix has 25 parallel input and output circuits corresponding to the 25 bits of the address. Output windings of corresponding cores of each register are connected in series with a transistor output amplifier. The write windings of corresponding cores of each register are connected by means of isolating diodes to a transistor input amplifier.

Table 3-1. Number and Type of Buffer Registers for Each Synchronizer

Synchronizer	Each Synchronizer	
	Input-Buffer Registers	Output-Buffer Registers
Drum read	2	0
Drum write	0	2
Tape read-write	2	2
Line printer	0	1
Electronic page recorder	0	1
Card reader	2	0

Each register has an A-phase magnetic-amplifier read driver which is set by a 1- $\mu$ s read signal from the address- and buffer-register control circuit to drive the 25 cores of the register when an address is being read out or when the cores of the register are being cleared. Each register also has a B-phase magnetic-amplifier write driver which drives the 25 cores of the register as an address is being written into the register. Unlike the buffer registers in which the read and write drivers are set independently of each other, an address-register write driver is always set by the output of the read driver.

Whenever the next address is ready to be written into address register j,\* the address-register-j read driver is set by signal S<sub>j</sub>AR from the address- and buffer-registers control. The output of the read driver reads out or clears the address already in address register j (output signals SAR1...25), and sets the register write driver. The output of the write driver coincides with the arrival at the address-register-matrix input amplifiers of the next address (input signals SMR1...25), writing it into address register j.

The timing for this read-write cycle is illustrated in figure 3-3.

† n represents the number of synchronizers in the system.

\* j is any synchronizer in the system.

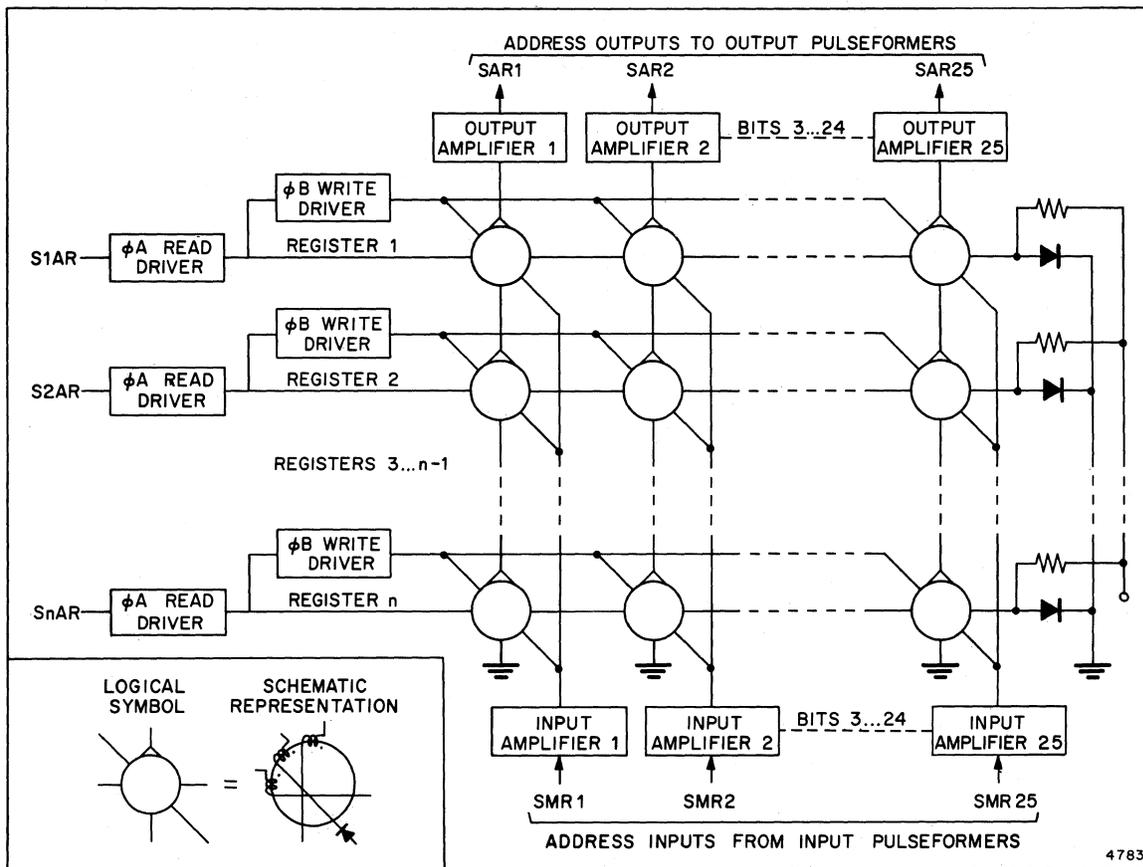
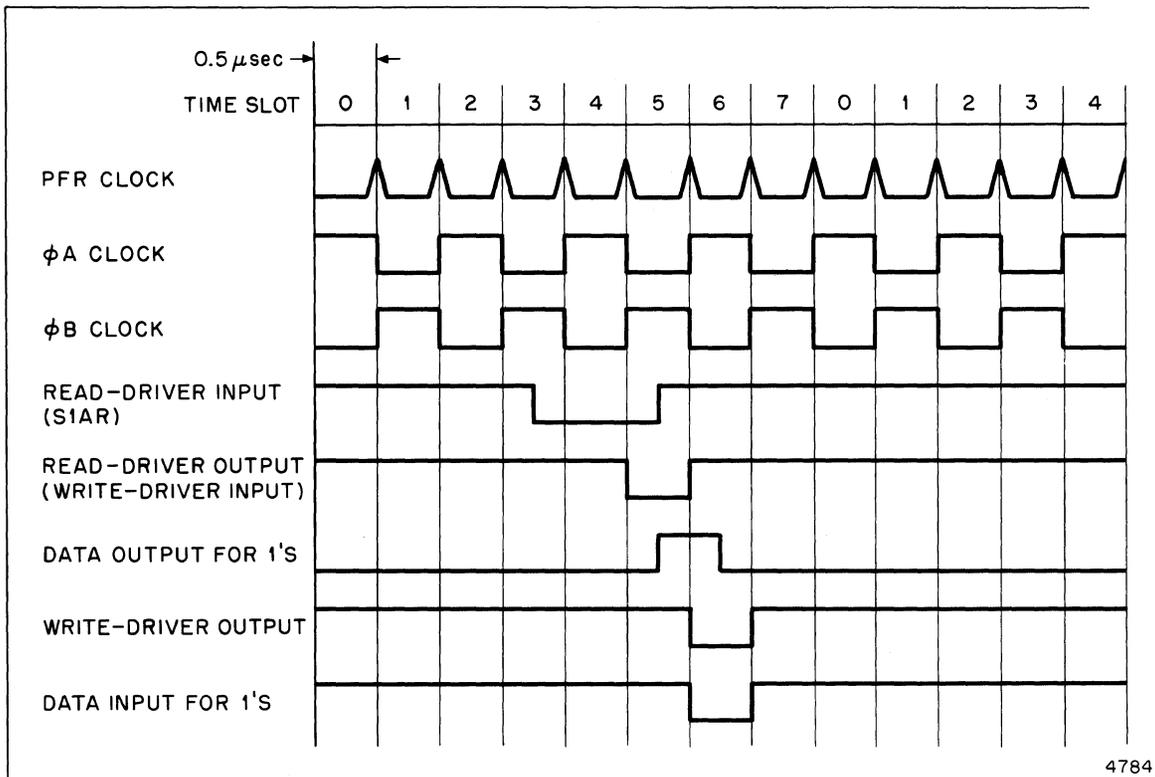


Figure 3-2. Address-Register Matrix: Simplified Diagram

### 3-4. INPUT-BUFFER-REGISTER MATRIX

An input-buffer register is a 60-bit magnetic-core register used by an input synchronizer as buffer storage for a word being transferred from an input device to the memory. (See figure 3-4 and logic drawings D807560 ...563). Since the input device fills its input-buffer register one digit at a time, and a word sent over the memory HSB system is 60-bit parallel, the register is also designed to act as a series-to-parallel converter. A word is written into the register under control of the synchronizer and is read out by signals from the address- and buffer-register controls of the dispatcher.

Since all the input synchronizers operate at a comparatively high rate of data transfer, the time interval between words is not sufficient for successive words to gain access to the memory; consequently two one-word input-buffer registers are used alternately for each input synchronizer. While one word is being written into one register ('odd' register), the preceding word is waiting in the other register ('even' register) for the dispatcher to gain memory access for it. By the time the odd register is full, the word in the even register will have been sent to the memory. Then the roles of the registers are reversed. The terms 'odd' and 'even' are used in this case solely as a means of distinguishing between the two one-word registers of each synchronizer; they have no further meaning.



**Figure 3-3. Timing of Read-Write Cycle of an Address Register**

Since a word is written into an input-buffer register with each parallel-bit digit in series, the 60 cores of the register are connected in a 12 x 5 input matrix. The input-buffer registers for all the input synchronizers in the system are within a  $t \times 60$  matrix.\*

Each register has 12 B-phase magnetic-amplifier write drivers, each of which energizes the write windings of the five cores in one digit position (figure 3-4). The write drivers are selected sequentially by a digit counter in the synchronizer, so that each digit of five parallel bits, fed to the register through five transistor amplifiers, will be stored in the five cores of the digit position specified by the sequential digit counter.

Because input to an input-buffer register is by one digit at a time, each successive digit should in theory be applied through the same five input amplifiers (one for each bit). The actual wiring of the registers, however, is such that 30 cores of the register are on one plane and the remaining 30 on another; for this reason there are two sets of five input transistor amplifiers, one set for each half of the register. One set of input amplifiers is used for the first 30 cores of both the odd and even registers of one synchronizer; the other is used for the last 30 cores of both the odd and even registers. (See figure 3-4.)

Since the two registers of each synchronizer therefore have their own sets of input amplifiers and write drivers, the input circuits of the reg-

\*  $t$  is the total number of input-buffer registers in the system.

isters of each synchronizer are independent of those of the registers of other synchronizers.

One complete word is read out of an input-buffer register at one time, all 60 bits in parallel. Each register has two A-phase magnetic-amplifier read drivers activated by a 1- $\mu$ s signal derived from the set state of the memory-ready FF in the address- and buffer-register control circuits. These amplifiers drive one half of the register each—that is, 30 cores—as illustrated in figure 3-4, but, since they are activated simultaneously, the contents of all 60 cores are read out in parallel through 60 transistor output amplifiers to the memory-write buffers. Only one set of 60 output amplifiers is provided for the 60 parallel-bit output of all  $t$  registers, since only one register can transfer information to the memory at any one time. Thus, although each register has its own two read drivers, the corresponding cores of all  $t$  registers have a common output path—that is, the output windings of all corresponding cores are connected in series.

Figure 3-4 illustrates the connections between the cores of two half registers, both odd and even, of drum read synchronizers 1 and 2 (S1 and S2). The connections between the cores of the second halves of registers 1 and 2 are identical to those shown in figure 3-4. The cores representing digit positions 1, 2, and 6 are shown fully, but those representing digit positions 3, 4, and 5 are omitted for brevity.

The first six digits of a word of information transferred from its input device by drum read synchronizer 1 (S1), for example, will be written into the first 30 cores of one of two input registers of the synchronizer in the following way.

Assume that the odd register for the synchronizer is already filled, and the synchronizer therefore specifies that the word to be transferred will go into the even register. When the first digit of the new word is fed through the five input amplifiers (IA) at the input to both the odd and even registers of the synchronizer, signal S1WV1 from the synchronizer activates the B-phase write driver controlling the first five cores of the even register. The first digit will therefore be written only into the first digit position of the even register. When the second digit is applied to the same five input amplifiers, signal S1WV2 will cause it to be written only into the second digit position of the even register, and so on until signal S1WV12 writes the 12th digit into the 12th digit position of the even register (at the end of the second half of the register not shown in figure 3-4). Since the signals that activate the write drivers of an input register are derived from the digit counter within the synchronizer, the order in which the digit positions in the register are filled is dictated by the order in which the digit counter of the synchronizer is stepped. Because of mode considerations, for example, a tape synchronizer has several sequences for writing digits into its input-buffer register.

When the even register is filled with the word and the synchronizer has gained access to the memory, signal S1RV from the address- and buffer-register control activates two read drivers, one for each half register (only one being shown, therefore), to read out the register content through the output amplifiers to the memory write buffers and onto the memory write HSB.

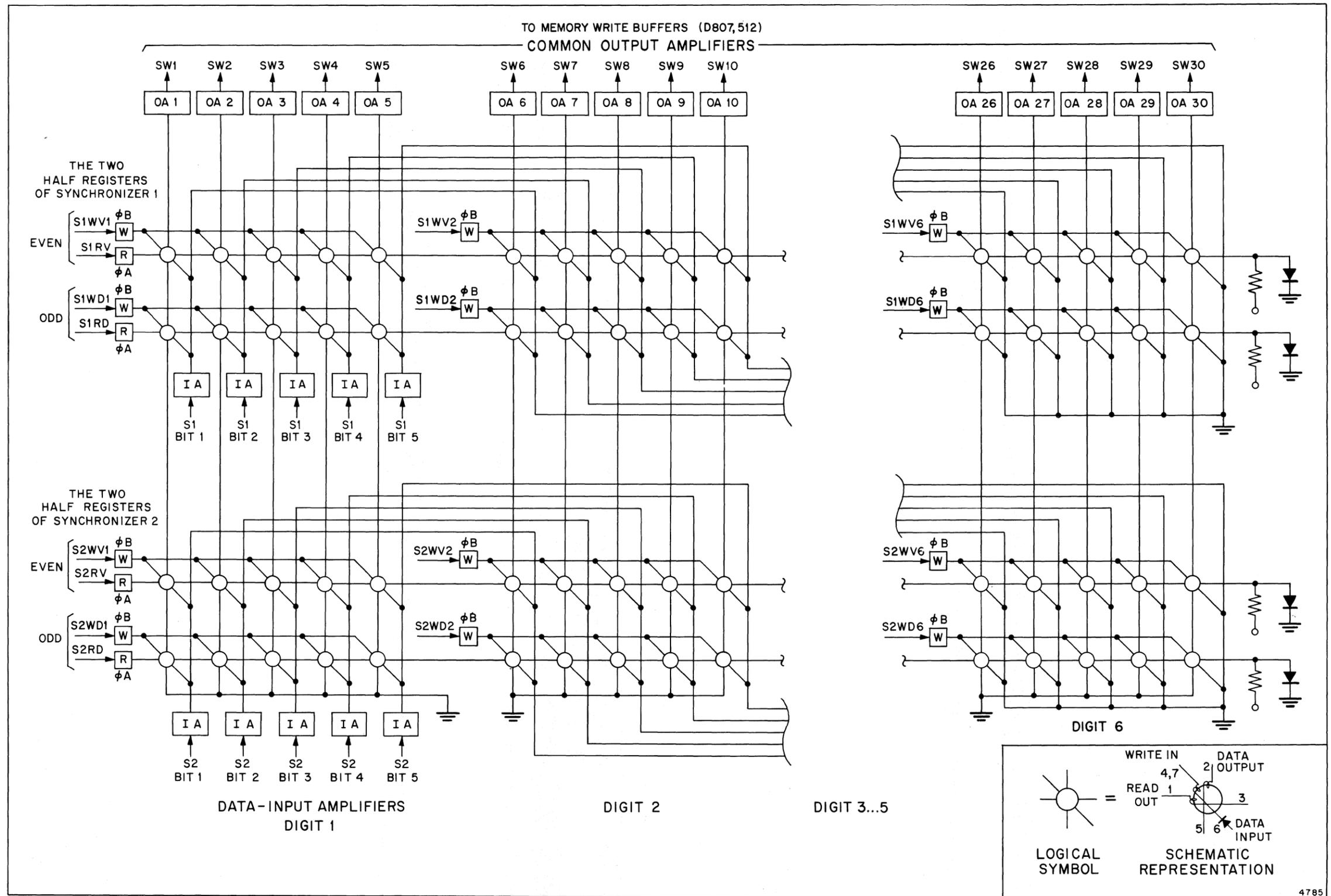


Figure 3-4. Half Registers of Drum Read Synchronizers 1 and 2 (S1, S2)

### 3-5. OUTPUT-BUFFER-REGISTER MATRIX

An output-buffer register is a 60-bit magnetic-core register used by a synchronizer as buffer storage for a word being transferred from the memory to an output device. Such a word is sent from the memory on the HSB, all 60 bits in parallel, and is transferred to the output device one digit at a time; thus the output-buffer register is designed to act as a parallel-to-series converter. (See figure 3-5 and logic drawings D807564...567.)

The writing of a word into the output register is controlled by the address- and buffer-register controls of the dispatcher, and the reading out of a word by the synchronizer.

Where two output-buffer registers are used for one synchronizer, they are distinguished by the terms 'odd' and 'even,' as shown in figure 3-5.

Since a word is read out of an output-buffer register with each parallel-bit digit in series, the 60 cores of each register are arranged in a 12 x 5 output matrix. The output-buffer registers for all the output synchronizers in the system are within a  $t \times 60$  array.\*

Each register has two A-phase magnetic-amplifier write drivers, each of which energizes the write windings of one half of the register, and 12 B-phase magnetic-amplifier read drivers, one to energize the read windings of the five cores of each digit position in the register (figure 3-5).

For any output device except the line printer in an alphanumeric mode, a complete word of data from the memory is written into an output-buffer register, all 60 bits in parallel. Coincident with the arrival of the word, a signal from the address- and buffer-register controls energizes both of the write-amplifier drivers of the appropriate register to write the 60 bits in parallel into the 60 cores. Only one set of 60 input amplifiers is provided as input to all the  $t$  registers, since only one register can receive information from the memory at any one time. Thus, although each register has its own two write drivers, the corresponding cores of all the  $t$  registers have a common input path—that is, the data input windings of all corresponding cores are connected in series.

When an output device is ready to receive a word from its output-buffer register, sequential signals derived from a digit counter within the synchronizer set each of the 12 read amplifier drivers in turn to read out the 12 digits one after the other through a set of five output transistor amplifiers. As with input-buffer registers, the actual wiring of the output buffer registers is such that one set of five output amplifiers is provided for each half of each register (figure 3-5). Where a synchronizer has two registers, one set of output amplifiers serves the same half of both the odd and even registers. The output path of the registers of each synchronizer is therefore independent of the output paths of the registers of other synchronizers.

The common 60-bit parallel input path to all registers is arranged on the logic drawings in this digit-position order, from left to right:

---

\*  $t$  is the total number of output-buffer registers in the system.

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

This digit-position order is dictated by the printout characteristics of the line printer in the alphanumeric mode. When operating in this mode, the line printer prints the three characters in digit positions 1, 2, 5, 6, 9, and 10 of a word during one print cycle (figure 3-5), and then in a later print cycle prints the three characters in digit positions 3, 4, 7, 8, 11, and 12. Because of these two separate print cycles required for the printing of one alphanumeric word, the line-printer output-buffer register is used as two separate 30-core registers, which are filled and emptied at different times. The first time that a word to be printed is sent from the memory, only the write driver for the first 30 cores is activated, so that the first 30 cores are filled with digits 1, 2, 5, 6, 9, and 10 of the alphanumeric word. These digits are read out serially to the line printer. When the word is sent again from the memory, only the write driver for the second 30 cores is activated, so that digits 3, 4, 7, 8, 11, and 12 are written into the remaining cores and read out serially to the line printer.

The 60 cores of the line-printer output-buffer register are therefore arranged in this digit-position order:

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

Since the data-input path to all output-buffer registers is common, the cores of all output registers are arranged in the same order. This irregular digit-position order does not, however, affect the order in which digits are read out of the other registers, since their data input from memory (including that of the line-printer register in a numeric mode) is of 12 digits in 60-bit parallel form and is a simultaneous transfer so that each digit can be read out in the sequence dictated by the digit counter in each synchronizer, regardless of the arrangement of digit positions in the registers.

Figure 3-5 shows the connections between the cores of the half output-buffer registers of drum write synchronizer 1 and line-printer synchronizer 1 (S3 and S5). The line-printer synchronizer requires only one register and the drum write synchronizer has both an odd and an even register. Only the cores representing digit positions 1, 2, and 10 are shown; those representing digit positions 5, 6, and 9 are omitted for brevity. The connections between the cores of the second halves of registers 3 and 5 are identical to those illustrated in figure 3-5. As an example of output-buffer-register operation, assume that a word is obtained from the memory by drum write synchronizer 1 (S3) to be written into the even output-buffer register of drum write synchronizer 1 (S3). When the word is applied to the 60 input amplifiers to the registers, signal S3WV from the address- and buffer-register control activates the A-phase write driver of the even register of drum write synchronizer 1 (S3) to write digits 1, 2, 5, 6, 9, and 10 into the 30 cores shown. S3WV also simultaneously activates the A-phase write driver for the second half of the same register to write in digits 3, 4, 7, 8, 11, and 12.

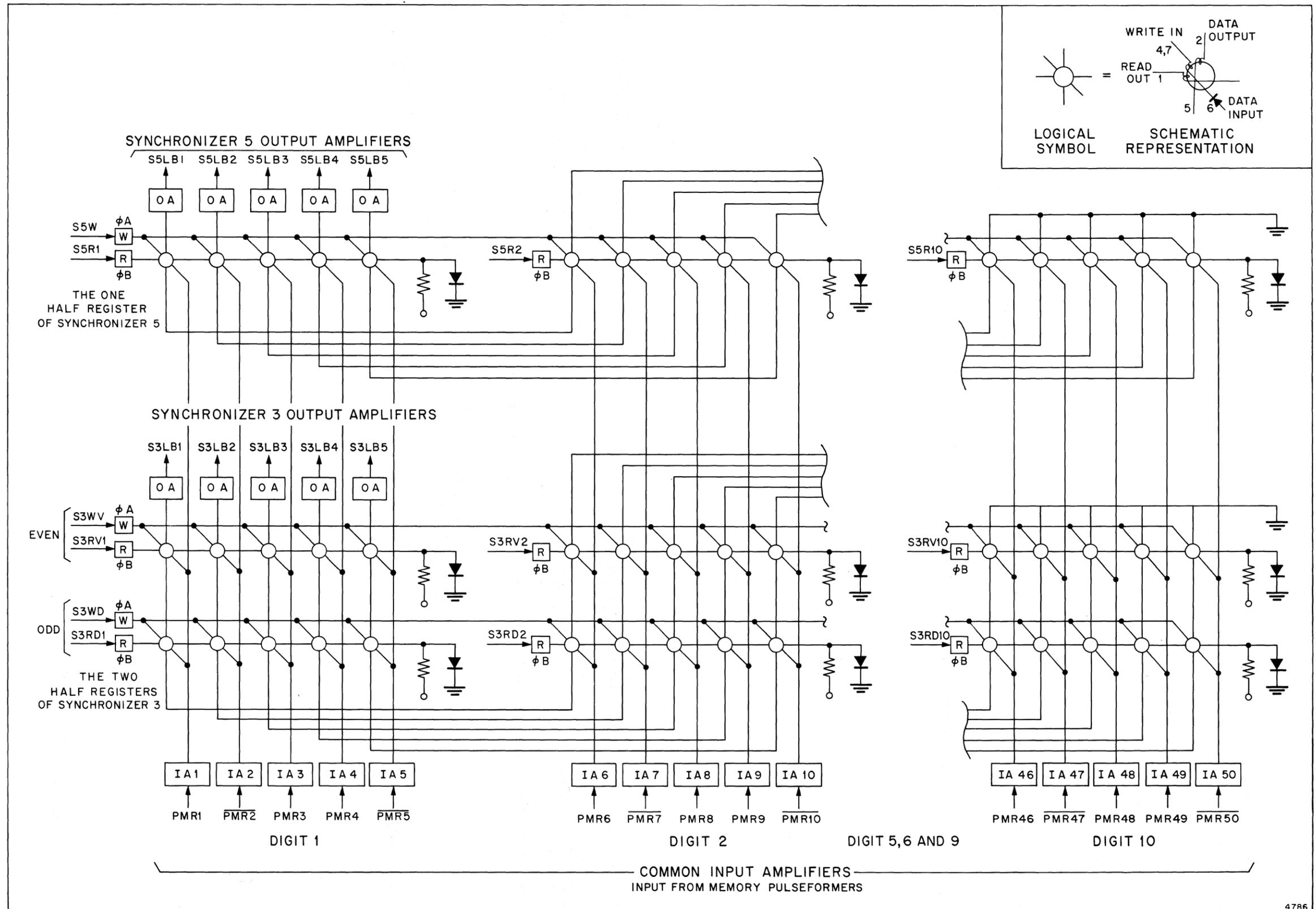


Figure 3-5. Half Registers of Drum Write Synchronizer 1 (S3) and Line-Printer Synchronizer 1 (S5)

If the contents of the register are then to be read out in ascending numerical sequence, the digit counter in drum write synchronizer 1 (S3) generates the sequential read signals S3RV1 through S3RV12 to activate each of the B-phase read drivers in turn and thus read out each of the stored digits one after the other to the output device which drum write synchronizer 1 (S3) controls.

On the assumption that line-printer synchronizer 1 (S5) is operating in an alphanumeric mode, the half register shown in figure 3-5 for line-printer synchronizer 1 (S5) will be used independently of the half which is not shown—that is, S5W will activate the write drivers for the two halves at different times. When the line printer is operating in a numeric code, however, the two halves of the line-printer synchronizer 1 (S5) register will be used together as described for drum write synchronizer 1 (S3).



## SECTION IV

### ADDRESS AND INPUT-OUTPUT BUFFER CONTROLS

#### 4-1. GENERAL

The address and input-output buffer controls are a group of circuits within the dispatcher designed to control and time the transfer of

- (1) Addresses both to and from the address registers, and
- (2) Words of information between the buffer registers and the memory high-speed bus system.

The address and input-output buffer control circuits may be divided into two separate sections:

- (1) The common address-transfer control circuit, which is shared by all the synchronizers in the system, and
- (2) The individual address-register and input-output buffer-register control circuits, of which there is one for each synchronizer in the system.

In actual operation, these two sections are interdependent.\*

#### 4-2. ADDRESS-TRANSFER CONTROL

The address-transfer control circuit functions as the switching center between the address-register matrix and the address modifier, the memory-address decoder, and register P1. It also serves to produce a memory-read signal for transmission to the memory along with the address whenever a memory readout operation is to be performed.

The address-transfer control circuit divides readily into two sections, as illustrated by figures 4-1 and 4-2. The first section (figure 4-1) consists of 25 address-register input pulseformers, 25 address-register output pulseformers (one pulseformer for each bit of the five-digit address), and associated gating circuits which control the routing of the address between the address-register matrix and its possible destinations.

---

\* See the timing diagrams of figures 4-6 and 4-7.

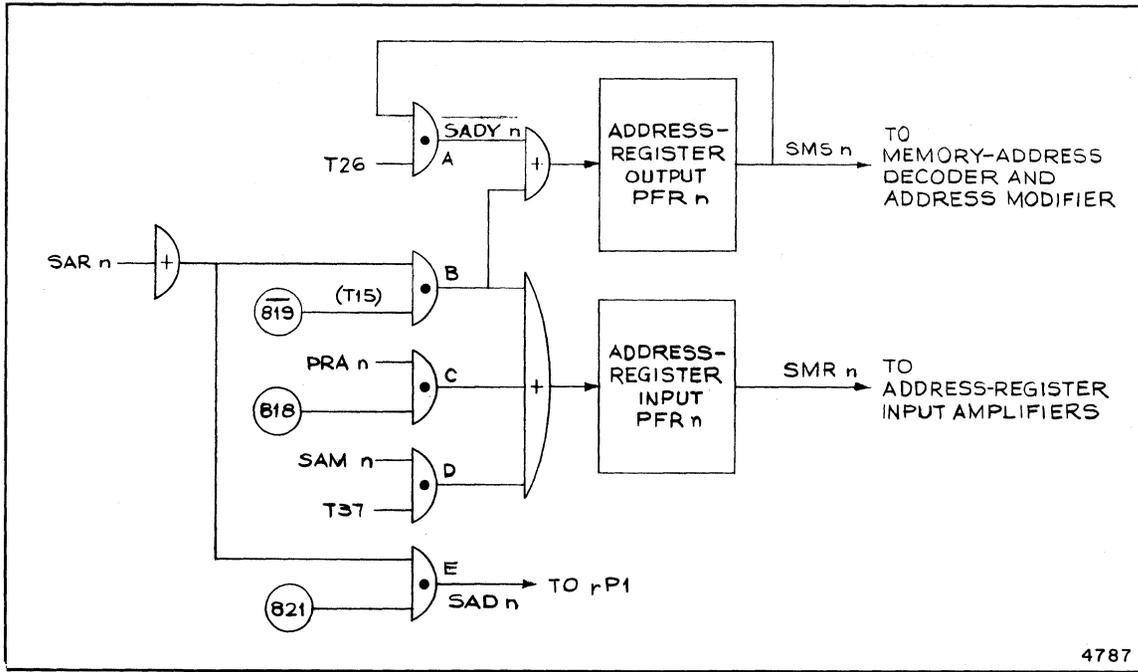


Figure 4-1. Address-Transfer Control. Part 1

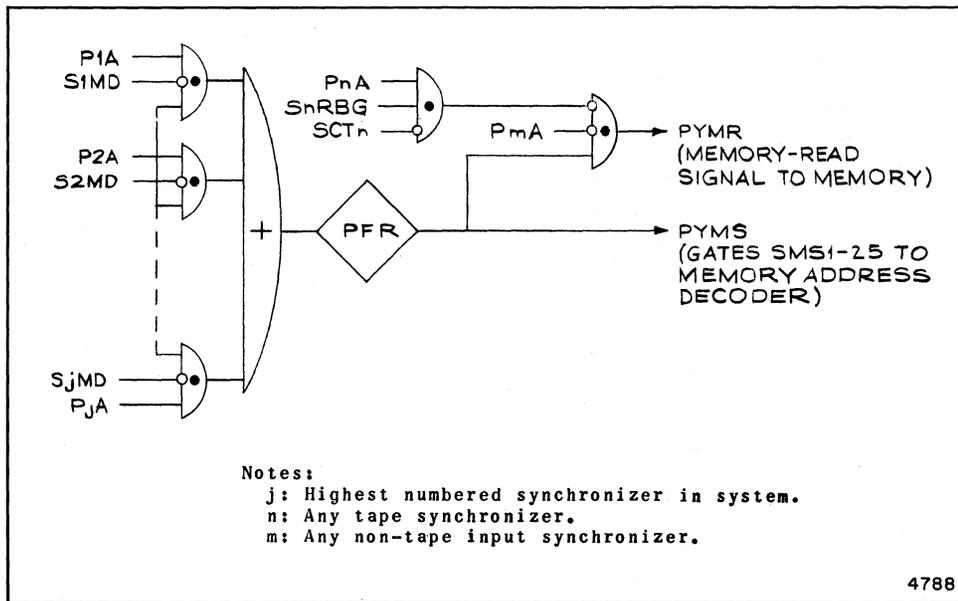


Figure 4-2. Address-Transfer Control. Part 2

The second section of the address-transfer control circuit (figure 4-2) is composed of a series of gates whose outputs are buffed to produce the signal which gates the address to the memory-address decoder. Also included in this second section is a series of gates whose outputs are gated to generate the read signal to the memory whenever an output synchronizer has gained priority to address the memory.

#### 4-3. FIRST SECTION

Figure 4-1 shows one of the 25 address-register input pulseformers and one of the 25 address-register output pulseformers, both used for bit position  $n$  of an address.\* The input gates to the input pulseformer are labeled A and B, and those to the output pulseformer B, C, and D. (B is a common input gate.) Gate E bypasses the pulseformers and is an input path to register P1.

When an address is read out of an address register, the output of the output amplifiers is polarity-inverted and designated SAR 1 ... 25. Address bit SAR  $n$  is fed to output pulseformer  $n$  and input pulseformer  $n$  through gate B at T15. Through the input pulseformer SAR  $n$  becomes SMR  $n$  at T26, and will be written back into the same address register 0.5  $\mu$ s after being read out. Through the output pulseformer SAR  $n$  becomes SMS  $n$  at T26. SMS  $n$  then is sent to the memory-address decoder, and at T26 is also gated back through gate A as SADY  $n$  to the output pulseformer, so that SMS  $n$  is also present at T37. At T37 SMS  $n$  is gated into the address modifier. The address modifier takes 2  $\mu$ s to modify an address so that at the following T37 the modified address bit SAM  $n$  is applied to the input pulseformer through gate D and is written into the address register as SMR  $n$  at T04. If during the modification the SMS  $n$  sent to the memory-address decoder finds the memory not busy, the regenerated SMS  $n$  is read out of the address register at T37 and is lost—since gate B is not permissive at T37—and is replaced by the modified SMS  $n$  (now SMR  $n$ ) at T04. If the memory is busy when addressed by SMS  $n$ , the regenerated SMS  $n$  is not read out of the address register and it is the modified SMS  $n$  that is lost.

When SAR  $n$  represents a bit of an address that is to be replaced by a new address from rP1, function signal 819 (generated by instructions 53, 55, and 74) is high at T15 and blocks gate B, thus preventing regeneration and destroying the address read out of the address register. At the same T1, function signal 818 will gate the new address from the five LSD positions of rP1, PRA  $n$ , through gate C to the input pulseformer, so that the address is written into the register at T2.

If SAR  $n$  represents a bit of an address which is to be sent to the five LSD positions of rP1, and not to be regenerated (instruction 41), it is gated to rP1 by function signal 821 through gate E as SAD  $n$  at T1, and gate B is inhibited by function signal 819.

If SAR  $n$  is a bit of an address which is to be sent to rP1 and also regenerated (instructions 42, 43), function signal 821 gates it to rP1 as SAD  $n$  through gate E at T1, and function signal 819 is not present at T1, thus allowing regeneration.

---

\*  $n$  is 1...25.

#### 4-4. SECOND SECTION

Figure 4-2 shows the second section of the address-transfer control. This section produces the control signal required to gate the address to the memory-address decoder and the signal which indicates to the memory that a memory read operation is to be performed. A gate for each synchronizer in the system feeds into a buffer. The inputs to each gate are (1) timing signal T15, (2) the priority signal for the particular synchronizer, and (3) the output of the memory-ready FF of the synchronizer. (The memory-ready FF is set when the synchronizer gains memory access.) Each gate is permissive only when the synchronizer it represents has priority but has not yet achieved memory access; under these conditions at T15, the gate causes the buffer to activate a high-powered driver, producing signal PYMS low at T26. PYMS then is used to gate the address read out of the address register for the particular synchronizer into the memory-address decoder.

When a synchronizer receives priority to one time slot more than is necessary to achieve memory access,\* the memory-ready FF will be set by the time the additional time slot is granted to prevent the generation of PYMS and thereby to prevent the initiation of a second memory reference.

Whenever an address is sent to the memory for a synchronizer to initiate a memory cycle, the memory logic requires an indication of whether it is to be a read cycle—that is, an output synchronizer is addressing—or a write cycle—that is, an input synchronizer is addressing. This indication is derived from the address-transfer control in the form of a memory-read signal ( $\overline{\text{PYMR}}$  high), which is generated whenever the address-transfer control gates an address to the memory-address decoder for an output synchronizer. When an address is transferred to the memory-address decoder for an input synchronizer, signal  $\overline{\text{PYMR}}$  will not be generated; that is, it will remain low and indicate to the memory logic that the cycle to be initiated is a write cycle.

Memory-read signal  $\overline{\text{PYMR}}$  high is produced by gating the address-to-memory-transfer control signal (PYMS low) with signals indicating that the synchronizer, which has produced the PYMS low, is not performing an input (memory write) operation. Thus the gating will be inhibited and  $\overline{\text{PYMR}}$  generated low if the synchronizer that has priority to produce PYMS low is either solely an input synchronizer or a tape input-output synchronizer performing an input operation. For a solely input synchronizer the priority signal ( $\overline{\text{Pm}}$  high) is sufficient to inhibit the gating. For a tape input-output synchronizer that has priority ( $\text{Pn}$  low) the gating is inhibited if the synchronizer is set up for a tape read operation ( $\text{SnRBG}$  low) and is not performing a check-read operation with the positioning checker connected ( $\text{SCT } n$  low). Although a check-read operation with the positioning checker connected to the tape synchronizer would not generate a priority signal, a priority signal could be generated by a write operation being performed simultaneously on the same tape synchronizer.

---

\* Section II, Priority Circuits.

#### 4-5. ADDRESS-REGISTER AND BUFFER-REGISTER CONTROLS

The address-register and buffer-register controls make up the section of the address and input-output buffer control circuits which is designed to control the reading-out and writing-in of addresses for the address registers and to control the reading-out of information from the input-buffer registers and the writing-in of information to the output-buffer registers.

Each synchronizer has its own address register and its own input- or output-buffer registers. It therefore has its own address- and buffer-register control circuit. The address- and buffer-register control circuit of each synchronizer generates the signals which set the address-register read and write driver of the synchronizer, its input-buffer-register read drivers if it controls an input device, or its output-buffer-register write drivers if it controls an output device. Since a tape synchronizer controls an input and output device and therefore uses both input- and output-buffer registers, its address- and buffer-register control circuit has to generate both input-buffer-register read signals and output-buffer-register write signals.

The address- and buffer-register control circuit for a synchronizer comprises a memory-ready FF, a memory-transfer FF, and a number of gates which together control the clearing, reading, writing and regenerating operations for the address and buffer registers associated with the synchronizer (figures 4-3, 4-4, and 4-5). The memory-ready FF is always set for 2  $\mu$ s when the synchronizer makes a successful memory reference, and the memory-transfer FF is always set afterward for the next 2  $\mu$ s. For an input synchronizer the address- and buffer-register control signals are derived from the gating at the output of the memory-ready FF. For an output synchronizer the address-register control signal is also derived from the gating at the output of the memory-ready FF, but the buffer-register control signals are produced from the gating at the output of the memory-transfer FF.

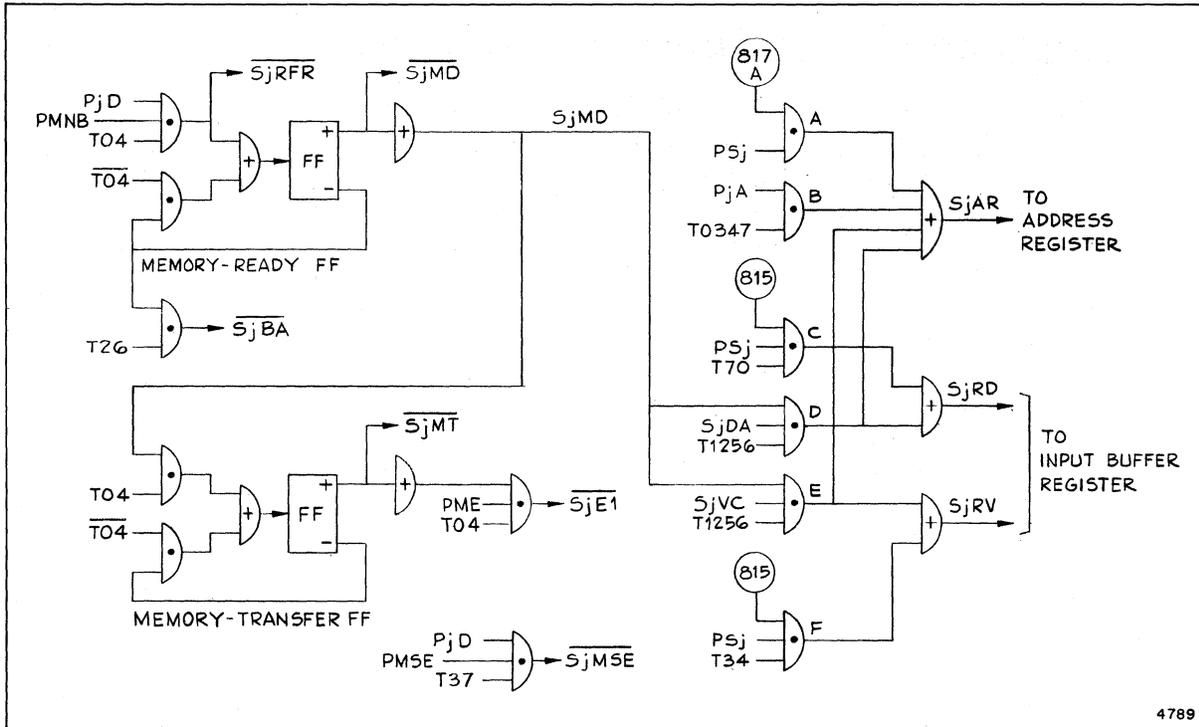
Included in the memory-transfer FF output circuit is a gate which samples the output of the memory HSB odd-even error checker whenever information is transferred across the memory HSB lines for the particular synchronizer. Signals indicating to the priority circuit that the synchronizer no longer requires priority and to the synchronizer that the information transfer between the buffer register and the memory has taken place are also derived from the address- and buffer-register control circuit. Also shown in each of figures 4-3, 4-4, and 4-5—though not strictly a part of the address- and buffer-register control circuits—is a gate alerted by the delayed-priority signal for synchronizer  $j$  to sample the memory-address-checker output signal  $PMSE$  at T37. Should  $PMSE$  indicate an address error, error signal  $S_jMSE$  high is generated to set the synchronizer- $j$  error FF.

#### 4-6. ADDRESS- AND BUFFER-REGISTER CONTROL FOR AN INPUT SYNCHRONIZER $j$ (OTHER THAN TAPE)

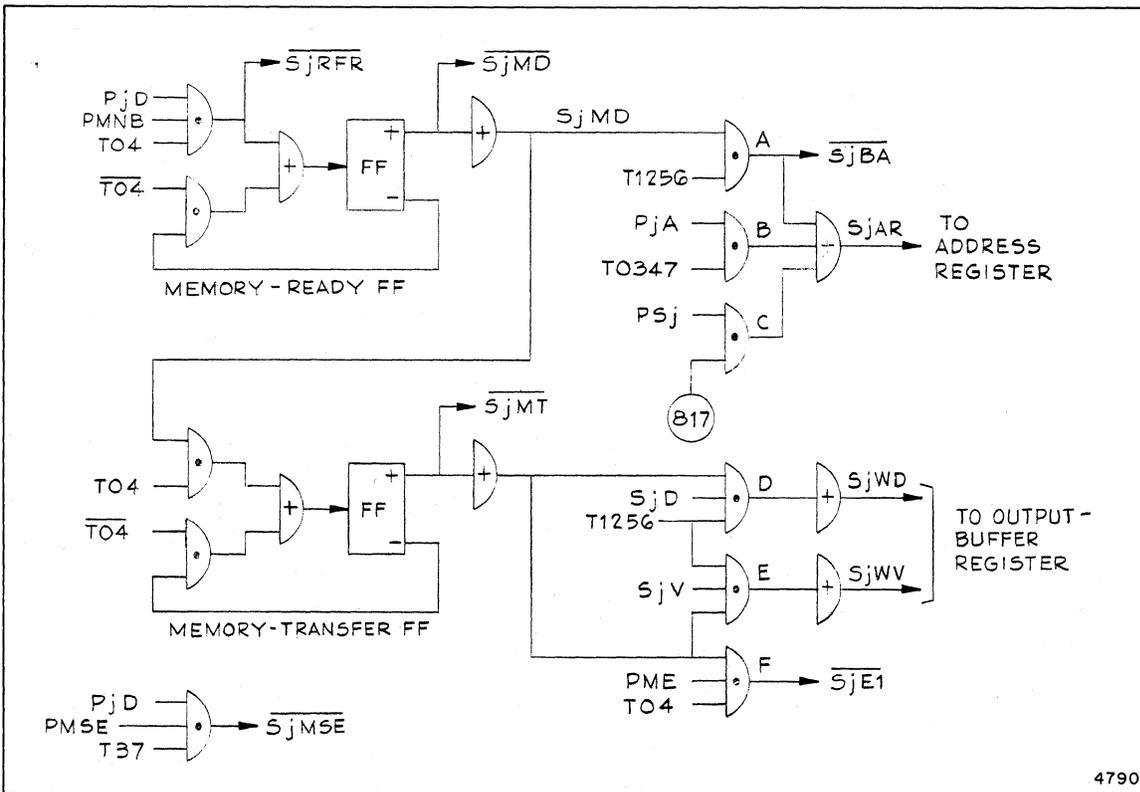
Figure 4-3 illustrates the address- and buffer-register control circuit for input synchronizer  $j$ .<sup>\*</sup> The gates through which the control signals are produced are labeled A through F.

---

<sup>\*</sup>  $j$  is other than a tape synchronizer.



**Figure 4-3. Address- and Buffer-Register Controls for an Input Synchronizer j: Detailed Logic**



**Figure 4-4. Address- and Buffer-Register Controls for an Output Synchronizer j (Two Buffer Registers): Detailed Logic**

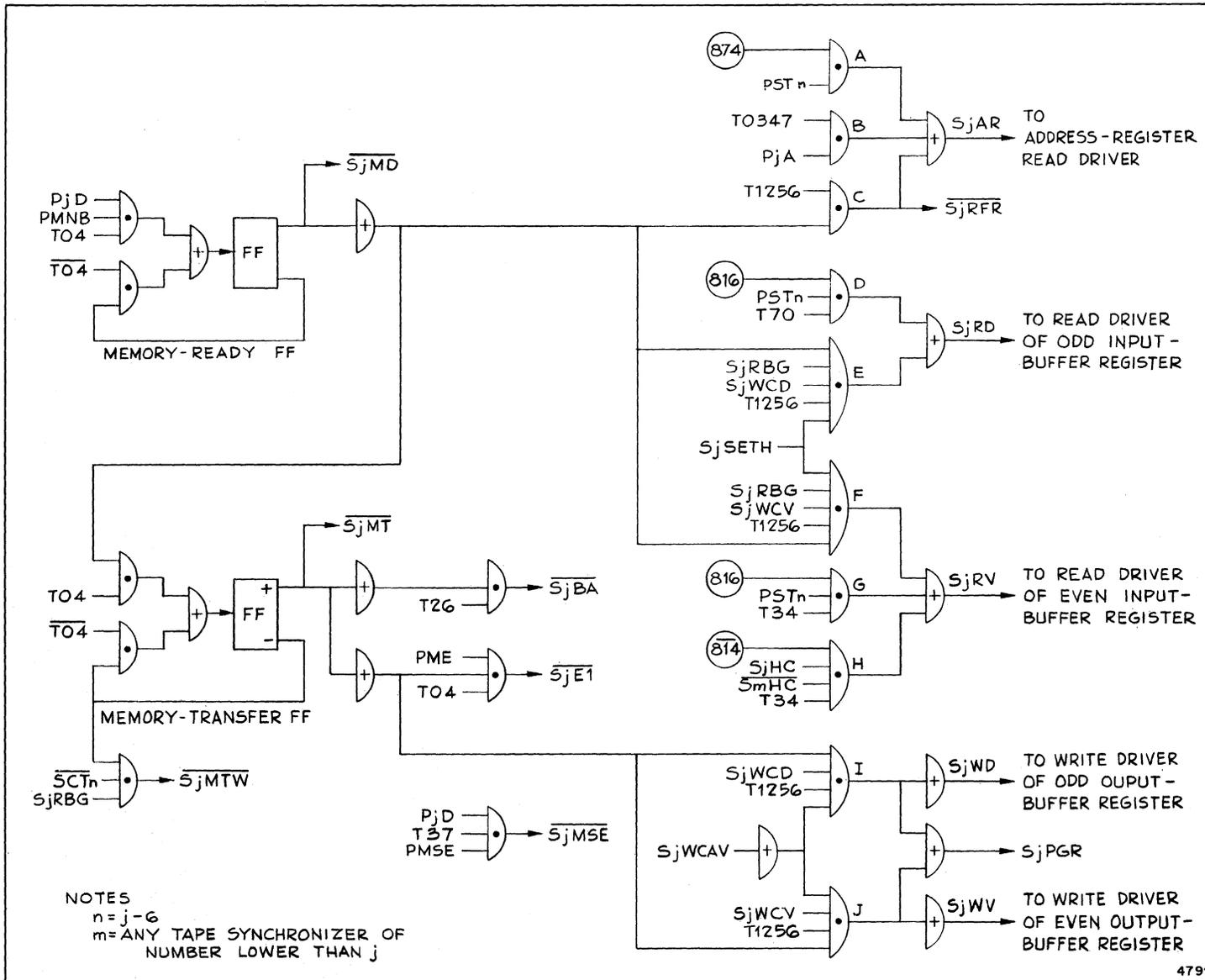


Figure 4-5. Address- and Buffer-Register Controls for Tape Input-Output Synchronizer j: Detailed Logic

When synchronizer  $j$  receives priority to address the memory, priority signal  $PjA$  is applied to gate B, which will then be permissive at either T34 or T70 and produce address-register- $j$  read signal  $SjAR$ .  $SjAR$  sets the synchronizer- $j$  address-register read driver, the output of which reads out the stored address at T15 to the memory-address decoder and the address modifier and also sets the write driver of the register to write the same address back into the register at T26. If the memory is not busy when it is addressed in this way, the delayed-priority signal  $PjD$  sets the memory-ready FF at T04 with the arrival of the memory-not-busy signal  $PMNB$ . The FF remains set for the next  $2 \mu s$ , alerting gates D and E. One of gates D and E will be permissive at T1256, depending on whether the synchronizer specifies the use of the odd ( $SjD$ ) or the even ( $SjV$ ) input-buffer register, and will produce buffer-register read signal  $SjRD$  (odd) or  $SjRV$  (even) and read out the contents of the buffer register to the memory write buffers.

Either gate D or gate E permissive will also produce  $SjAR$  at T1256 to clear out the regenerated address from the synchronizer- $j$  address register and write in the address now modified from the address modifier at T04.

At the same T04 the set output of the memory-ready FF sets the memory-transfer FF for the next  $2 \mu s$  so that at the following T04 the set output of the memory-transfer FF is used to sample the memory HSB odd-even error-checker output  $PME$ , producing error signal  $SjEI$  high if an error has been detected.

Gates C and F are used by instructions 53 and 55, which produce function signal 815 to clear out the contents of an input-buffer register. It does so by generating read signal  $SjRD$  at T70 for the odd register or  $SjRV$  at T34 for the even register.

Function signal 817, produced by instructions 41, 42, 53, and 55, is applied to gate A to produce  $SjAR$  at T70 in order to clear out an end address from the synchronizer- $j$  address register and write in a starting address from register P1, or to read out an address to register P1 and regenerate it in the address register. (Instruction 41 does not regenerate the address read to rP1.)

The same conditions which set the memory-ready FF at T04 also produce signal  $\overline{SjRFR}$  high, which is used to indicate to the word-transfer-request FF of the synchronizer that the word transfer requested by the FF is now about to take place, and the word-transfer request is no longer needed. The set output of the memory-ready FF ( $\overline{SjMD}$  high) is tapped off and used to reset the priority-bid FF of the synchronizer in the priority circuits.

If synchronizer  $j$  is a drum read synchronizer, the set output of the memory-ready FF ( $SjMD$  low) is gated at T26 to produce signal  $\overline{SjBA}$ , which signifies to the synchronizer that the input-buffer register is now ready to be filled again. If  $j$  is the card-reader synchronizer,  $\overline{SjBA}$  is derived from the set output of the memory-transfer FF at T26.

The set output of the memory-transfer FF ( $\overline{SjMT}$  high) indicates to the IOP-memory-write checker that a memory write operation is taking place and also alerts the input gate of the dispatcher HSB-error FF so that the flip-flop may be set if an error occurs.

#### 4-7. ADDRESS- AND BUFFER-REGISTER CONTROL FOR AN OUTPUT SYNCHRONIZER j (OTHER THAN TAPE)

Figure 4-4 shows the address- and buffer-register control for an output synchronizer j. j has two output-buffer registers and is not a tape synchronizer. The control gates are labeled A through F.

When synchronizer j achieves memory-reference priority, gate B is alerted by priority signal PjA. Gate B is then permissive at T34 or T70, corresponding to the synchronizer time slot used, and generates the 1- $\mu$ s synchronizer-j address-register read signal SjAR at T34 or T70. SjAR reads out the address at T15 to the memory-address decoder and the address modifier and also writes the same address back into the register at T26. When the synchronizer does gain memory access, the memory-not-busy signal PMNB sets the memory-ready FF at T04 for 2  $\mu$ s through the gate alerted by PjD, the delayed-priority signal. With the memory-ready FF set, gate A will be permissive at T1256 and will generate address-register read signal SjAR, which clears the regenerated address and replaces it with the now modified address at T04.

At the same T04 at which the memory-ready FF is reset, the memory-transfer FF is set for the next 2  $\mu$ s. With the memory-transfer FF set, gates D, E, and F are all alerted. If the synchronizer specifies that it is the odd output-buffer register into which the word from memory is to be written (SjD), gate D will be permissive at T1256 to produce the 1- $\mu$ s odd output-buffer-register write signal SjWD. If the even register is specified (SjV), gate E will be permissive at T1256 to generate the even output-buffer-register write signal SjWV. Gate F samples the output PME of the memory HSB odd-even error checker to generate  $\overline{\text{SjEI}}$  high for an error.

If j is the electronic-page-recorder synchronizer, which uses only one output-buffer register, only one buffer-register write signal needs to be generated. This signal is produced as described in the preceding paragraph by the set state of the memory-transfer FF at T1256. If j is a line-printer synchronizer, however, two buffer-register write signals are generated, even though the synchronizer uses only one register. Thus gates D and E are both included in the line-printer address- and buffer-register control circuit, but their input signals are Sj1256 and Sj3478 instead of SjD and SjV, and the write signals they produce are SjWA and SjWB instead of SjWD and SjWV. The reason for producing the two write signals for only one register is that in an alphanumeric mode the line-printer synchronizer uses its output-buffer register as two separate 30-core registers\* which are filled and emptied independently of one another and at different times. When the line printer is in a numeric mode, SjWA and SjWB are produced at the same T1256 to act as one write signal for the whole register.

Signal  $\overline{\text{SjRFR}}$  high, which indicates to synchronizer j that the requested word transfer is being carried out and thus eliminates the need for the request, is derived from the same conditions that set the memory-ready FF if j is a drum write synchronizer. If j is a line-printer synchronizer or the electronic-page-recorder synchronizer,  $\overline{\text{SjRFR}}$  is generated from the set output of the memory-ready FF at T1256.

---

\* Section III, Output-Buffer Registers.

For any output synchronizer it is the set state of the memory-ready FF ( $\overline{SjMD}$  high) that resets the priority-bid FF.

The signal indicating to the synchronizer that the output buffer register is now about to be filled with the word from memory ( $SjBA$  high) is generated as shown in figure 4-4 only for the drum write synchronizers; for the line-printer and electronic-page-recorder synchronizers it is derived from the set state of the memory-transfer FF at T26.

The set state of the memory-transfer FF ( $\overline{SjMT}$  high) is used to alert the dispatcher HSB-error FF.

Whenever an end address is to be cleared from the address register for output synchronizer  $j$  and is to be replaced by a starting address from register P1, or whenever an address is to be read out of the synchronizer- $j$  address register to register P1, address-register read signal  $SjAR$  is produced by the presence of function signal 817 at gate C at T70.

#### 4-8. ADDRESS- AND BUFFER-REGISTER CONTROL FOR TAPE INPUT-OUTPUT SYNCHRONIZER $j$

Figure 4-5 shows the address- and buffer-register controls for a tape input-output synchronizer  $j$ .\*

When synchronizer  $j$  achieves memory-reference priority, address-register- $j$  read signal  $SjAR$  is produced by priority signal  $PjA$  through gate B at T34 or T70, depending on which synchronizer time slot was gained by synchronizer  $j$ . The address in the register is read out to the memory-address decoder and the address modifier, and is regenerated in the register in case the memory is busy. As soon as synchronizer  $j$  gains memory access, the memory-not-busy signal  $PMNB$  sets the memory-ready FF at T04 through the gate alerted by delayed-priority signal  $PjD$ . The FF remains set for the next 2  $\mu$ s, producing  $SjAR$  at T1256 through gate C to clear out the regenerated address from the address register and write in the address, by now modified. Signal  $\overline{SjRFR}$  high is also produced simultaneously with  $SjAR$  to reset the synchronizer- $j$  word-transfer-request FF. Signal  $\overline{SjMD}$  high is taken from the set output of the memory-ready FF to reset the synchronizer- $j$  priority-bid FF. The set output of the memory-ready FF ( $\overline{SjMD}$  low) is applied to both gates E and F. If tape synchronizer  $j$  is set up to perform an input or tape read operation, signals  $SjRBG$  and  $SjSETH$  from the synchronizer will also alert both gates E and F. At T1256, depending on whether the synchronizer- $j$  word-driver FF indicates an odd word ( $SjWCD$ ) or an even word ( $SjWCV$ ), either gate E will open to produce the odd input-buffer register read signal  $SjRD$  or gate F will open to produce the even input-buffer register read signal  $SjRV$ .

When the memory-ready FF is reset, the memory-transfer FF is set for the next 2  $\mu$ s. Signal  $\overline{SjMT}$  low alerts both gates I and J. At T1256, depending on whether the synchronizer- $j$  word-driver FF indicates an odd word ( $SjWCD$ ) or an even word ( $SjWCV$ ), either gate I will open to generate the odd output-buffer-register write signal  $SjWD$ , or gate J will open to generate the even output-buffer-register write signal  $SjWV$ . Both gates I and

---

\*  $j$  is any tape synchronizer.

J, however, will be blocked by signal  $S_jWCAV$  from the synchronizer if the tape synchronizer  $j$  is not set up for an output—that is, a tape write—operation. The production of either of these write signals generates signal  $S_jPGR$ , which indicates to the synchronizer that an output-buffer register has been filled.

The set output of the memory-transfer FF is further used for several purposes:

(1) At  $T_{04}$  it samples the output PME of the memory HSB odd-even error checker, producing  $\overline{S_jE1}$  high for an error.

(2) At  $T_{26}$  it generates signal  $\overline{S_jBA}$  high to inform the synchronizer that the requested word transfer has now taken place.

(3) As  $\overline{S_jMT}$  high it is used to alert the input gate to the dispatcher HSB-error FF.

(4) As  $S_jMT$  low, gated with

(a) The tape read signal  $S_jRBG$  low, and

(b) The signal indicating that the tape read operation is not a read check with the position checker ( $\overline{SCTn}$  low),

it generates signal  $\overline{S_jMTW}$  high, which signifies to the memory write checker that a tape read—that is, a memory write—operation is being performed.

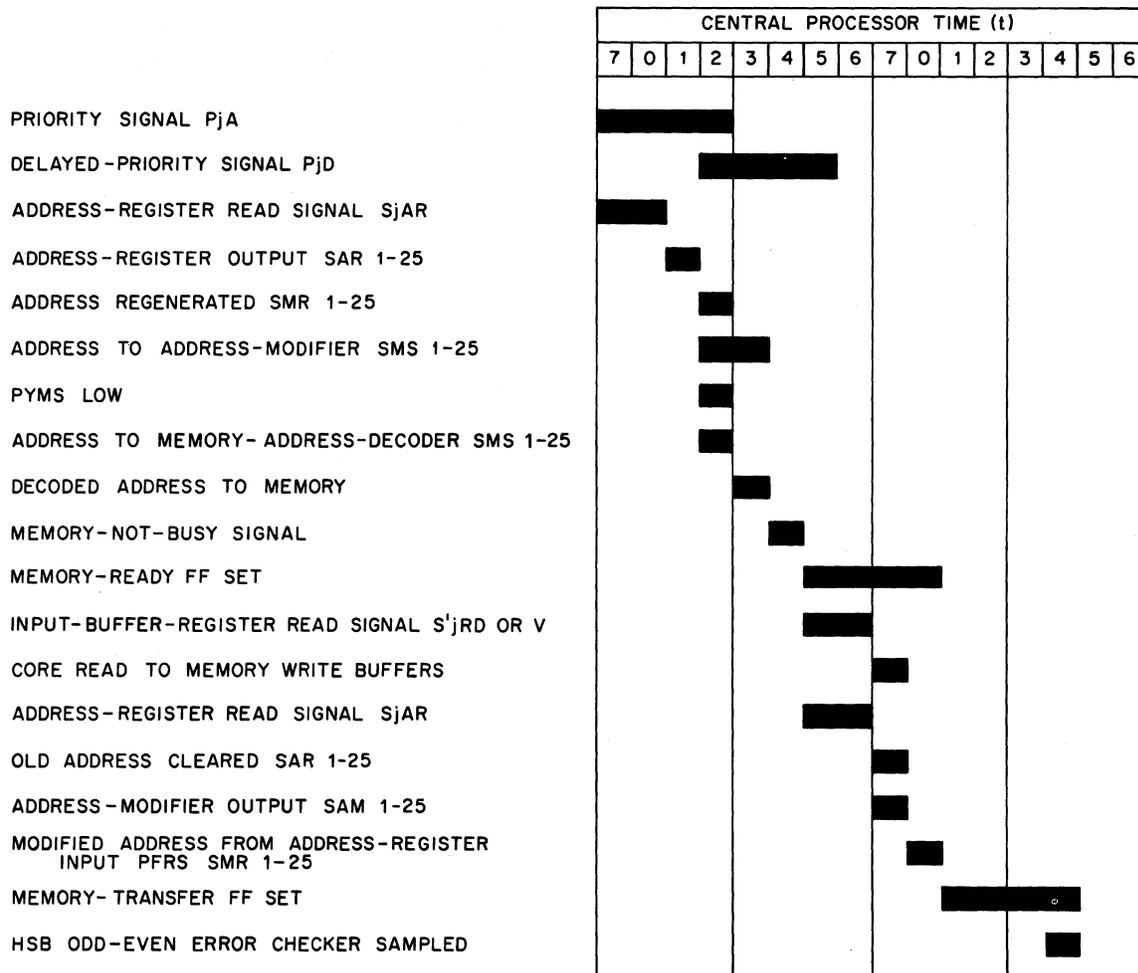
When a starting address is to be sent from register P1 to replace an end address in the tape-synchronizer- $j$  address register, or when an address in the register is to be sent to register P1 and also regenerated, function signal 874 produces read signal  $S_jAR$  through gate A at  $T_1$ . For a start-tape instruction (68), function signal 816 clears the synchronizer- $j$  input-buffer registers by generating  $S_jRD$  and  $S_jRV$  through gates D and G at  $T_{70}$  and  $T_{34}$ , respectively.

If hash is detected by synchronizer  $j$  on a tape-read-to-memory operation, hash-pulse signal  $S_jHC$  clears the even input-buffer register at  $T_{34}$  through gate H, except while a start instruction is being executed, at which time  $\overline{814}$  is high, or when a hash-pulse signal is simultaneously generated by a tape synchronizer of higher priority.

#### 4-9. TIMING FOR ADDRESS AND INPUT-OUTPUT BUFFER CONTROLS

The operation of the address and input-output buffer control circuits is summarized in the timing diagrams of figures 4-6 and 4-7.

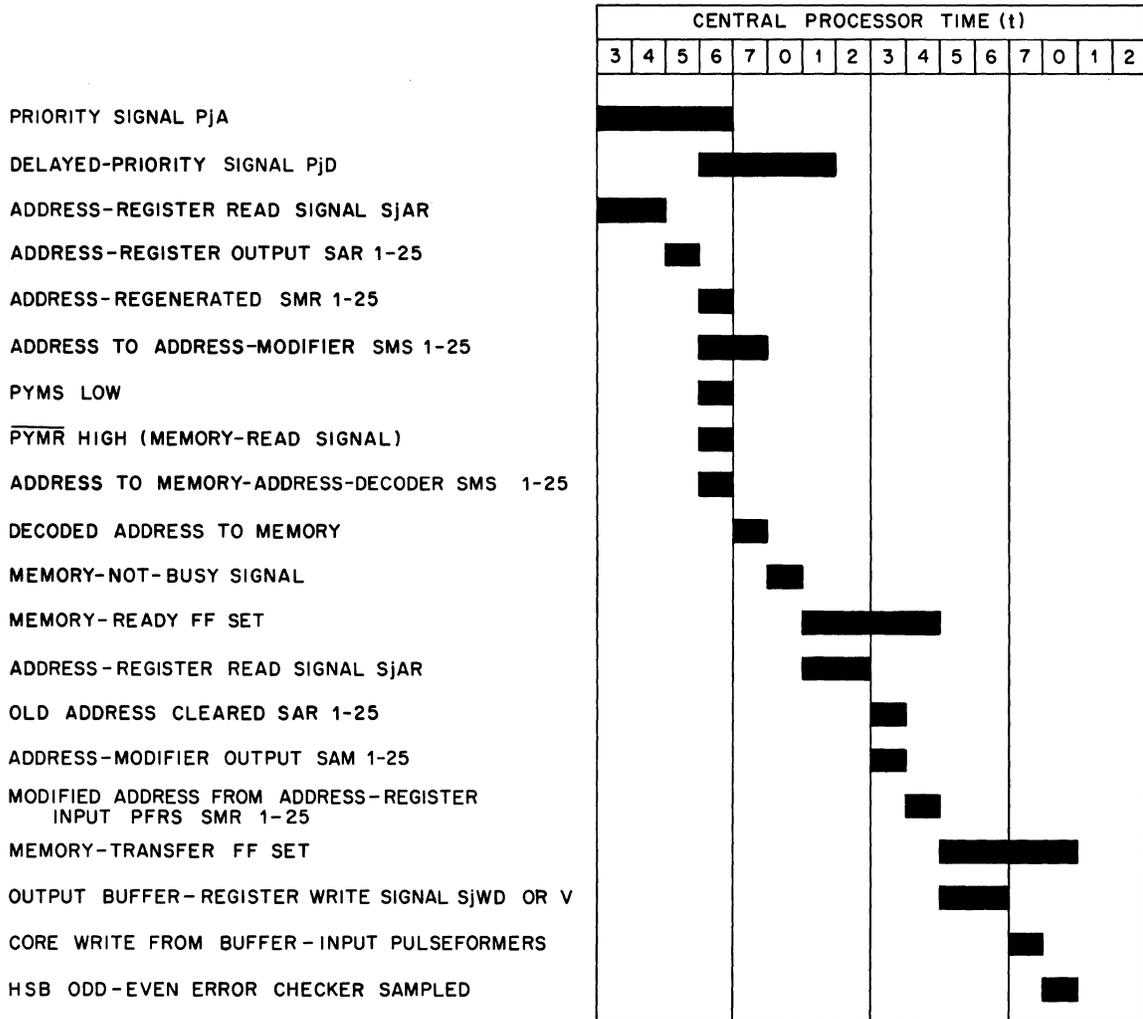
Figure 4-6 shows the timing involved in the control cycle necessary for transferring a word of information from an input-buffer register into the memory—on the assumption that the memory is not busy when the calling input synchronizer is granted time-slot priority.



4793

Figure 4-6. Memory-Transfer Control Cycle for Input Synchronizer j

Figure 4-7 illustrates the timing of the control cycle needed to transfer a word of information from the memory into an output buffer register—again on the assumption that the memory is not busy when the calling output synchronizer receives time-slot priority.



4792

Figure 4-7. Memory-Transfer Control Cycle for Output Synchronizer j



## SECTION V

### SYNCHRONIZER ADDRESS MODIFIER

#### 5-1. GENERAL

The address modifier is a special-purpose five-digit adder-subtractor which is time-shared by the synchronizers to modify the memory address each time a word is transferred to or from the memory for a synchronizer.

Whenever an address is read out of an address register to start a memory transfer operation, it is sent to each of three destinations: (1) it goes to the memory-address decoder to start the memory reference; (2) it is written back into the address register to be available for another attempt if the memory reference started should prove unsuccessful; and (3) it is sent to the address modifier.

On entering the address modifier (figure 5-1), the address is translated to a seven-bit biquinary code. Then it is modified in that form. The now modified address is translated back into Larc code and stored by the address-modifier output FFs so that 2.5  $\mu$ s after the address first enters the modifier the modified address is gated to the address-register input pulse-formers. If the memory reference proves successful—that is, the memory unit is not busy—the modified address is written into the address register to replace the old address that was regenerated.

#### 5-2. CAPABILITIES OF ADDRESS MODIFIER

The address modifier is capable of adding to the five-digit address the numbers 00001 or 97501 or subtracting the numbers 00001, 00009, 00010, or 00021. Since word transfers for the synchronizers are usually made to or from successive memory locations in ascending order, an address is modified in most cases by adding 00001. However, because of the peculiarities of a particular input or output device and the different modes in which its synchronizer may operate, the address modifier, under the following circumstances, will be called upon to add 97501 or subtract 00001, 00009, 00010, or 00021.

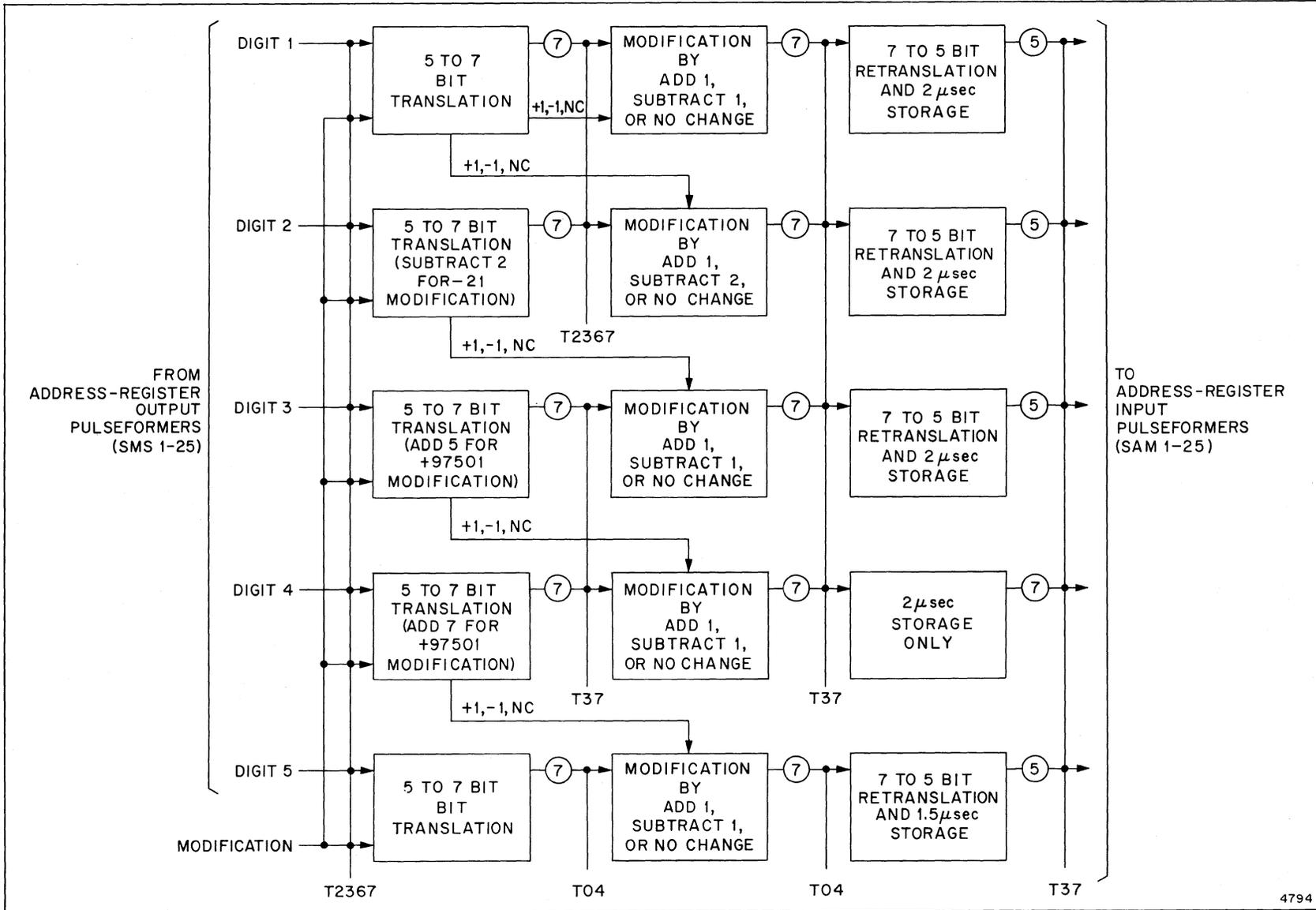


Figure 5-1. Synchronizer Address Modifier: General Block Diagram

(1) If a full band transfer is made between a drum and the memory,\* 97501 is added to the memory address between sectors 24 and 00 of the drum so that succeeding word transfers will begin with the first word location of the memory area to or from which the information is to be transferred.

(2) During a backward read operation on the magnetic-tape units the words are transferred to memory locations in descending order. Consequently 00001 is subtracted from the memory address.

(3) In the numeric and alphanumeric unedited modes (modes 2 and 4) of the printer synchronizer the printer prints ten words on a line in a standard format. The requirements of the printer are such that the ten words of a line must be transferred from the memory and sent, in succession, several times to the printer synchronizer for comparison with character combinations corresponding to successive printing positions of the printer typewheels. After every transfer of the tenth word 00009 is subtracted from the address so that the succeeding ten transfers will begin with the first of the ten words.

(4) In the numeric edited mode (mode 1) of the printer synchronizer 11 numeric words are printed on a line with the spacing determined by space or ignore digits within the words themselves. The 11 words of a line must be transferred from the memory and sent, in succession, to the printer synchronizer a total of 26 times. After every transfer of the 11th word 00010 is subtracted from the address, so that the succeeding 11 transfers will begin with the first of the 11 words.

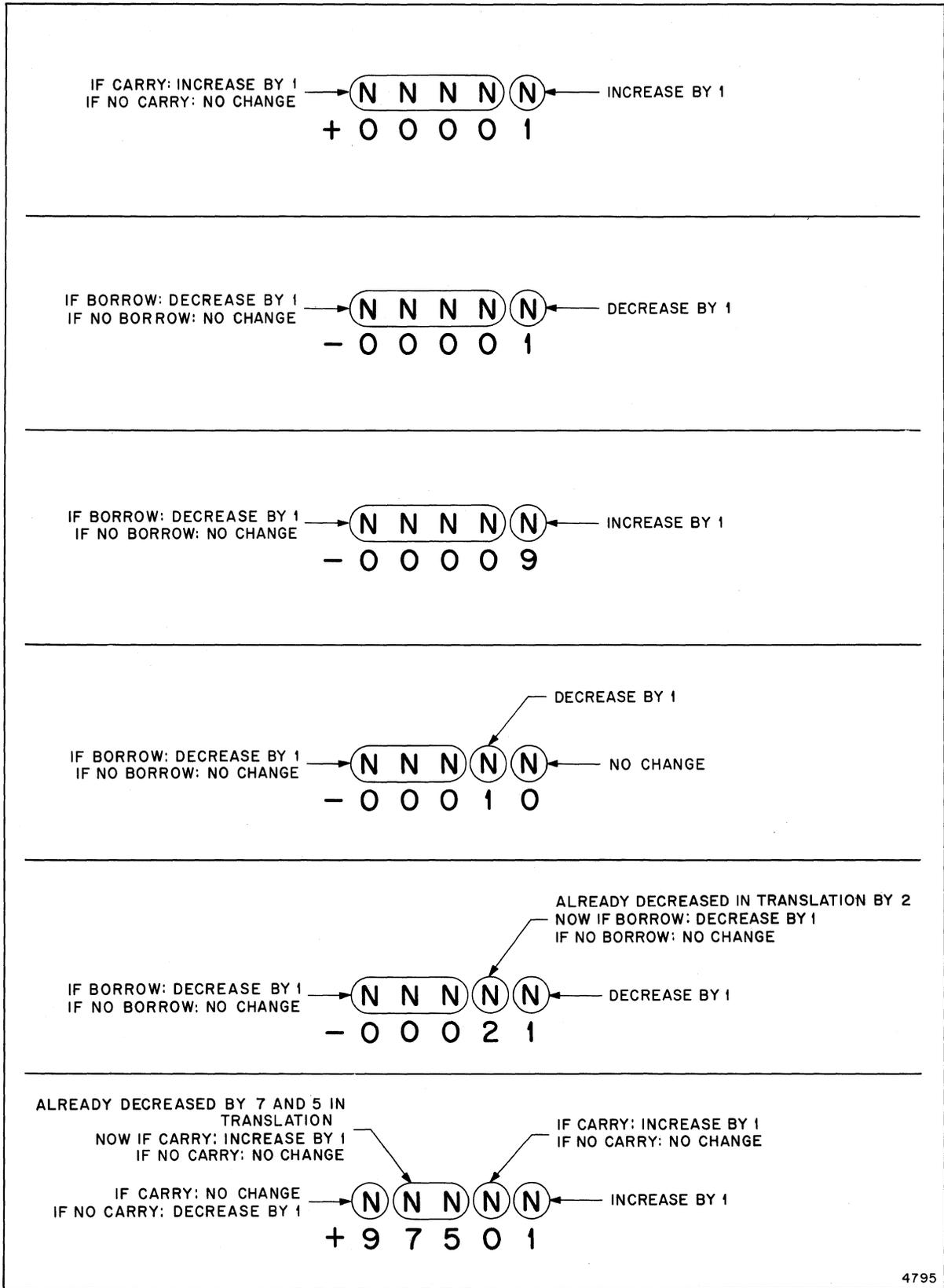
(5) In the alphanumeric edited mode (mode 3) 11 alphanumeric words are printed on a line with the spacing determined by space digits within the words. These 11 alphanumeric words are represented in the Larc two-digit code by 22 words. The 22 words must be transferred from the memory and sent to the printer synchronizer in succession a total of 102 times. After every transfer of the 22nd word 00021 is subtracted from the address.

### 5-3. PATTERN OF MODIFICATION

The pattern of modification (figure 5-2) is designed to take advantage of the fact that, since this is a special-purpose adder-subtractor, the modifying numbers are already known. In practically every instance the modifying digit causes the address digit to be increased by 1, to be decreased by 1, or to remain unchanged. For address digit 1 these changes occur because the modifying digits +1, -1, -9, and -0 can have only one of these three effects; since for address digits 2, 3, 4, and 5 the modifying digit is nearly always a 0, the only change to each address digit will be by the addition of a carry, or the subtraction of a borrow, as demanded by the modification of the preceding digit. Thus a pattern of modification by adding 1, subtracting 1, or no change, holds good for all digits. The 9 of +97501 is no misfit because it, too, has the effect of either decreasing the fifth digit by 1 if there is no carry from the fourth, or of leaving it unchanged if there is a carry.

---

\* Processor instruction 55: Start Drum Synchronizer, Mode 2.



4795

Figure 5-2. Pattern of Modification by Add 1, Subtract 1, or No Change

The -2 of -21 and the +7 and +5 of +97501 do, however, present a difficulty, since -2 will obviously decrease the second digit by 2 if there is no borrow demanded by the first, or by 3 if there is a borrow. Similarly, +7 will increase the fourth digit by 7 with no carry from the third, or by 8 with a carry, and +5 will increase the third digit by 5 with no carry from the second or by 6 with a carry. So that these digit positions may follow the pattern of add 1, subtract 1, or no change, the 2 is subtracted, and the 5 and 7 are added beforehand, during the translation stage, by a process of deliberate mistranslation. Thus the only control signals needed for a given digit are those indicating an addition of 1, a subtraction of 1, or no change, and, where necessary, those required to distinguish the modifications of -21 and +97501 from all others, and +00001 from +97501. The signal indicating the addition or subtraction of 1 is sometimes propagated directly by the preceding digit, and sometimes by combining one signal indicating change—generated by the preceding digit—with external signals indicating either an addition or subtraction modification. For an add modification, the signal change will mean an addition of 1, while for a subtract modification the change will indicate a subtraction of 1.

#### 5-4. ADDRESS-MODIFIER CONTROLS

The signals which specify the amount by which an address is to be modified, and which control modification accordingly, originate from the synchronizers which use the address modifier.

The modification requests are derived as follows:

(1) SnF, the request for +00001 modification, is generated at T1256 by the synchronizer n which has priority, if n is

(a) A drum synchronizer operating in mode 1 with the Subtract 2499 FF reset;

or

(b) A tape synchronizer reading forward or writing;

or

(c) A card-reader synchronizer;

or

(d) The electronic-page-recorder synchronizer;

or

(e) A line-printer synchronizer with the last-word FF reset.

(2) SnB, the -00001 modification, is requested at T1256 if the synchronizer n which has priority is a tape synchronizer reading backward.

(3) SnX, the request for modification by -00009, is generated at T1256 by the synchronizer n, which has priority, if n is a line-printer synchronizer in mode 2 or 4 with the last-word FF set.

(4) SnY, the call for -00010 modification, is produced at T1256 when the synchronizer n which has priority is a line-printer synchronizer in mode 1 with the last-word FF set.

(5) SnW, the request for -00021 modification, is generated at T1256 if the synchronizer n which has priority is a line-printer synchronizer in mode 3 with the last-word FF set.

(6) SnA, the call for modification by +97501, is produced at T1256 if the synchronizer n which has priority is a drum synchronizer in mode 2 with the subtract 2499 FF set.

These various calls for different modifications are buffed and pulse-formed in such a way as to produce at T2367 all the necessary control signals for the address modifier, as shown in figure 5-3.

Table 5-1 summarizes the indications of all the control signals in the low state.

Table 5-1. Address-Modifier Control Signals

Signal	Modifications					
	+00001	+97501	-00001	-00009	-00010	-00021
SF low	★					
$\overline{SF}$ low		★	★	★	★	★
SA low		★				
$\overline{SA}$ low	★		★	★	★	★
SX low				★		
$\overline{SX}$ low	★	★	★		★	★
SY low					★	
$\overline{SY}$ low	★	★	★	★		★
SW low						★
$\overline{SW}$ low	★	★	★	★	★	
SB low			★	★	★	★
$\overline{SB}$ low	★	★				

## 5-5. FUNCTIONS OF A DIGIT ADDER-SUBTRACTOR

Each of the five digit adder-subtractors is designed to perform the same basic functions and to operate according to the same pattern. These are the functions of the digit adder-subtractor.

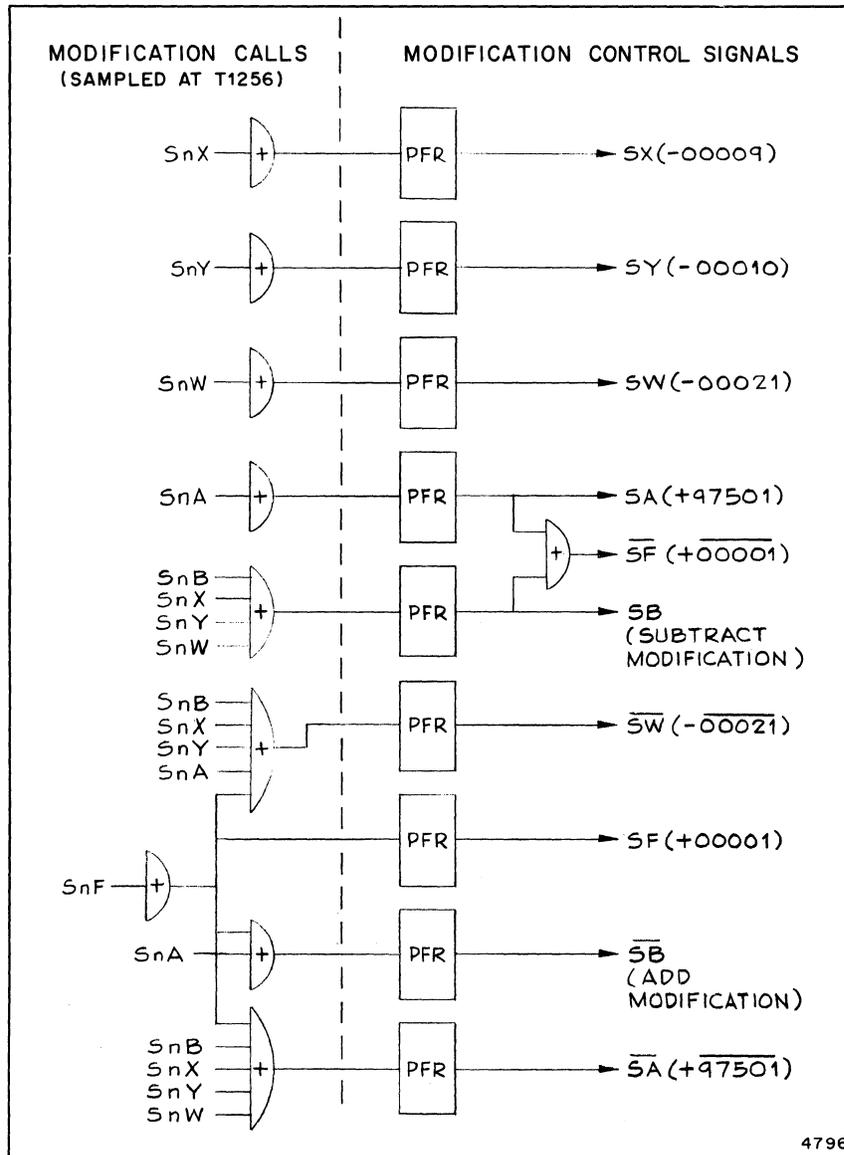


Figure 5-3. Derivation of Control Signals

- (1) Translating its input digit from Larc code to a straight biquinary code.
- (2) Modifying the translated digit in accordance with modification control signals.
- (3) Determining how modification of its input digit will affect modification of the next more significant digit.
- (4) Translating the modified digit back to Larc code and storing it for 2  $\mu$ s.

The design and operation of the digit 1 adder-subtractor will first be described under the following four functions (headings 5-6 through 5-10). The differences between the remaining four digit adder-subtractors and the first will follow (headings 5-11 through 5-16).

## 5-6. DIGIT 1 ADDER-SUBTRACTOR

The input digit to the digit 1 adder-subtractor of the address modifier is the least significant digit of an address, SMS1...5 from the address-register output pulseformers. Refer to figure 5-4.

## 5-7. TRANSLATING THE INPUT DIGIT

Entering the adder-subtractor, digit 1 (SMS1-5) is applied to a gating matrix which decodes the Larc-code digit and produces an equivalent output digit in straight biquinary code. The straight biquinary code provides a bit position for each quinary and either binary state of the digit—a total of 7 bit positions (table 5-2). Thus the quinary state indicated in bit positions 1 to 3 of the Larc-code digit produces a 1 bit in one of the first five bit positions of the biquinary code. Similarly the binary state denoted by the content of the fourth bit position in Larc code provides a 1 bit in either the sixth or seventh bit position of the biquinary code.

The parity information contained in the fifth bit position of the Larc-code digit is also decoded in the matrix to generate a 1 bit in either of two further bit positions added onto the biquinary digit—the eighth and ninth.

Table 5-2. Comparison of Larc and Seven-Bit Codes

Decimal Value	Bit Positions of											
	Larc Code				Seven-Bit Code							
	4	3	2	1	7	6	5	4	3	2	1	
0	0	0	0	0	0	1	0	0	0	0	0	1
1	0	0	0	1	0	1	0	0	0	0	1	0
2	0	0	1	0	0	1	0	0	1	0	0	0
3	0	1	1	1	0	1	0	1	0	0	0	0
4	0	1	1	0	0	1	1	0	0	0	0	0
5	1	0	0	0	1	0	0	0	0	0	0	1
6	1	0	0	1	1	0	0	0	0	0	1	0
7	1	0	1	1	1	0	0	0	1	0	0	0
8	1	1	1	1	1	0	0	1	0	0	0	0
9	1	1	1	0	1	0	1	0	0	0	0	0

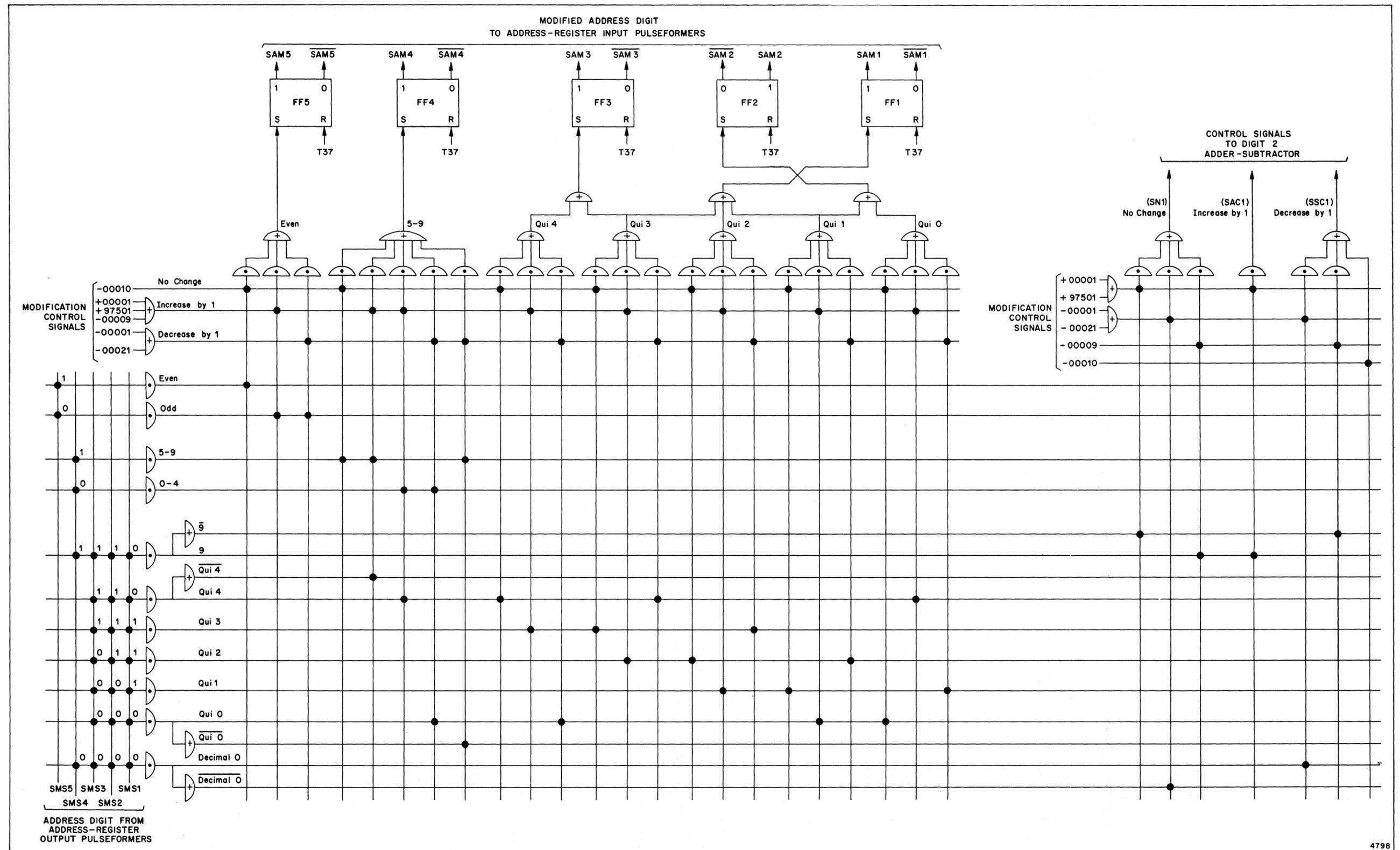


Figure 5-4. Digit 1 Adder-Subtractor: Simplified Logic

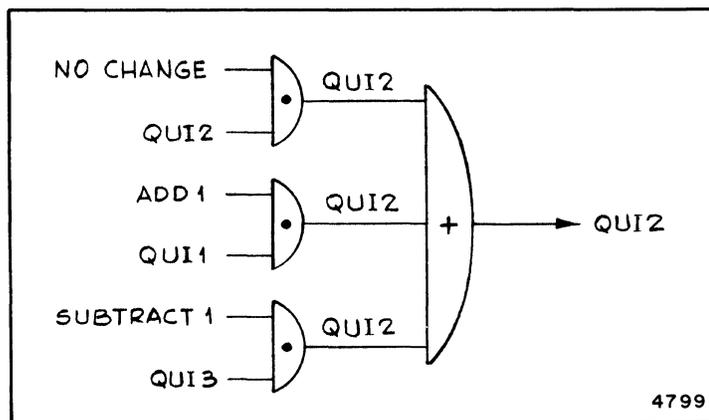


Figure 5-5. Derivation of Quinary Result of 2

#### 5-8. MODIFYING THE TRANSLATED DIGIT

Any of the six different modifications that may be requested by the synchronizers can have only one of three effects on the first address digit:

- (1) It will cause the digit to be increased by 1 (+00001, +97501, -00009).
- (2) It will cause the digit to be decreased by 1 (-00001, -00021).
- (3) It will cause the digit to remain unchanged (-00010).

Buffing the signals which indicate the +00001, +97501, and -00009 modifications therefore produces a control signal for increasing the first digit by 1; similarly, buffing the -00001 and -00021 modification signals provides a decrease-by-1 control signal, while the -00010 modification signal is synonymous with a no-change control signal.

Modification is effected by gating the translated digit with the appropriate control signal to produce a digit modified in accordance with the control signal.

The modification of the quinary part of the digit is performed according to the following pattern: each of the five quinary bit positions is gated with each of the three control signals. When gated with the no-change control signal, a quinary number produces the same quinary number; when gated with the increase-by-1 control signal, it produces the same quinary number plus 1; when gated with the decrease-by-1 control signal, a quinary number produces the same quinary number minus 1.

Each quinary result is produced by each of three gates. For example, as shown in figure 5-5, quinary 2 is the result of gating quinary 2 and no-change, quinary 1 and increase-by-1, or quinary 3 and decrease-by-1. The three gates producing the same quinary result are buffed.

As the quinary content of the digit is modified, the binary state must also be modified accordingly, whenever necessary. Since there are essentially only two binary states, modification consists simply of inverting the binary state.

The no-change control signal will not change the quinary or binary state of the digit. When the quinary state is to be increased by 1, the binary state of the digit will be inverted only if the quinary state is 4. When the quinary state is to be decreased by 1, the binary state will be inverted only if the quinary state is 0.

The modified binary state of the digit can therefore be ascertained by gating the original binary state with the relevant control signal and an indication of whether or not the quinary state is 0 or 4. Furthermore, only those combinations of this information that result in one of the binary states need to be detected by gating since all other combinations must necessarily result in the inverse state. In the digit 1 adder-subtractor those combinations resulting in binary 1 are detected:

(1) The no-change control signal is present with an address digit in the range 5...9.

(2) The increase-by-one control signal is present with an address digit in the range 4...8.

(3) The decrease-by-one control signal is present with an address digit of 0, 6, 7, 8, or 9.

Finally, when an address digit is increased or decreased by 1, its check bit must be inverted—that is, an odd digit will become even and an even digit odd. Since the direction of modification (increase or decrease) is irrelevant in determining the modified check bit, the only control information needed is whether or not the address digit is changed by modification.

The eighth and ninth bits of the translated biquinary digit are therefore gated with the no-change control signal and its inverse in such a way as to detect both cases in which digit-1 modification results in an even digit:

(1) When the digit is already even and is not changed by modification.

(2) When the digit is odd and is changed by modification.

## 5-9. DETERMINING HOW MODIFICATION AFFECTS DIGIT 2

In certain cases, modification of digit 1 will cause a carry to be added to—or a borrow to be subtracted from—the next most significant digit. Such cases must be detected in the digit 1 adder-subtractor in order to provide the necessary control signals to the digit 2 adder-subtractor.

By gating the modification control signals with signals indicating whether digit 1 is or is not a 9 or 0, and by buffering the results of gating, the following control signals to the next digit adder-subtractor are derived:

(1) SN1: no change to the second digit—if the first digit

(a) Is not 0 with the -00001 or -00021 modification;

or

(b) Is 9 with the -00009 modification;

or

(c) Is not 9 with the +00001 or +97501 modification.

(2) SAC1: increase the second digit by 1—if the first digit is 9 with the +00001 or +97501 modification.

(3) SSC1: decrease the second digit by 1—if

(a) The first digit is not 9 with the -00009 modification;

or

(b) The first digit is 0 with the -00001 or -00021 modification;

or

(c) The modification is -00010.

#### 5-10. TRANSLATING MODIFIED DIGIT BACK TO LARC CODE

After modification the digit is in seven-bit code rather than nine-bit code, with five quinary bit positions but only one binary bit position and only one check-bit position. As shown in figure 5-4, the five quinary bit positions are buffed and connected to three flip-flops; the sixth and seventh bit positions are connected to one flip-flop each. The contents of the seven bit positions will set those of the five flip-flops required to represent the modified digit in Larc code. Once set, the flip-flops remain set for the next 2  $\mu$ s, thus storing the modified digit for 2  $\mu$ s until it is gated to the address-register input pulseformers as SAM1...5.

#### 5-11. DIFFERENCES BETWEEN ADDER-SUBTRACTORS

The slight differences in design and operation between the digit 1 adder-subtractor and the remaining four digit adder-subtractors of the address modifier are best described under three headings:

(1) Differences in translation;

(2) Differences in modification;

(3) Differences in retranslation.

#### 5-12. DIFFERENCES IN TRANSLATION

Two chief differences in translation among the five digit adder-subtractors of the address modifier follow.

(1) The parity information in the fifth bit position of the second address digit is not decoded in the digit 2 adder-subtractor. Instead, sig-

nals SMS10 and  $\overline{\text{SMS10}}$  bypass translation and are gated directly with the no-change control signal (SN1) or the change control signals (SAC1 or SSC1) to give the parity information of the modified digit.

(2) Each adder-subtractor is designed to operate on a pattern of adding 1 to—or subtracting 1 from—the address digit, if any change in the digit is required at all (heading 5-3). To prevent the -00021 and the +97501 modifications from disrupting this pattern, the 2 of -00021 must be subtracted, and the 5 and 7 of +97501 must be added before modification—that is, as the input digit is translated from Larc to straight biquinary code. Thus, when the modification is other than -00021 ( $\overline{\text{SW2A,B}}$ ), the digit 2 adder-subtractor will translate its input digit in the manner already described. However, for the -00021 modification ( $\overline{\text{SW2A,B}}$ ) it will translate its Larc input digit into a biquinary digit, which is the same number minus 2.

Similarly, the digit 3 and 4 adder-subtractors will translate normally for any modification other than +97501 ( $\overline{\text{SA}}$ ). When the 97501 modification control signal is present (SA), the third digit adder-subtractor will translate its Larc input digit into a biquinary digit greater by 5, and the fourth digit adder-subtractor will translate its input digit into a number greater by 7.

#### 5-13. DIFFERENCES IN MODIFICATION

With the 2, 5, and 7 corrections made for the -00021 and +97501 modifications as just described, modification in each of the five digit adder-subtractors consists of the same pattern—increasing a digit by 1, decreasing it by 1, or not changing it.

In the same digit 1 adder-subtractor the increase or decrease by 1 or the lack of change is the direct effect of the first digit of the modifying numbers. In each of the remaining four digit adder-subtractors, however, the increase by 1 is the effect of a carry propagated by the next less significant digit; the decrease by 1 is the effect of a borrow demanded by the next less significant digit; and the lack of change is from the lack of either carry or borrow.

An exception to this scheme is the digit 2 adder-subtractor during a -00010 modification, in which there can be no borrow demanded by digit 1 since the first digit of the modifying number is 0. Therefore, decreasing the digit 2 by 1 is a direct result of the second digit of the modifying number.

The derivation of the three control signals to the digit 2 adder-subtractor—SN1 (no change), SAC1 (add a carry), and SSC1 (subtract a borrow)—has already been discussed for the digit 1 adder-subtractor under heading 5-9.

Each digit adder-subtractor except the fifth provides the next more significant digit adder-subtractor with control signals indicating (1) add a carry, (2) subtract a borrow, or (3) no change. These control signals are derived in each digit adder-subtractor by gating the translated digit with any carry or borrow information from the next less significant digit, and with external control signals indicating which of the six modifications

is being performed. This is done in such a way as to detect all cases in which modification to the digit will propagate a carry to, demand a borrow from, or not change, the next more significant digit.

In many cases the carry or borrow information from the next less significant digit adder-subtractor is used as a means of determining what the preceding address digits are. SAC1, for example, can only be produced by the +00001 or +97501 modifications, and only then if the first address digit is a 9.

The digit 2 and 4 adder-subtractors provide 3 and 5, respectively, with only two control signals—change or no-change. When applied to the digit 3 and 5 adder-subtractors however, the change signal implies add-a-carry if the modification is one of addition, and subtract-a-borrow if the modification is one of subtraction.

Because the fifth modifying digit of the +97501 modification is a 9, a carry propagated from the digit 4 adder-subtractor to 5 will cause the fifth address digit to remain the same; therefore it is detected as a case for generating the no-change control signal to the digit 5 adder-subtractor. For the same reason, lack of a carry to the digit 5 adder-subtractor will cause the fifth address digit to be decreased by 1 and is therefore detected as a case for producing the subtract-a-borrow control signal to the digit 5 adder-subtractor.

Furthermore, the 1 of the -00010 modification is subtracted from the second address digit by causing the digit 1 adder-subtractor to generate the subtract-a-borrow signal to the second digit adder-subtractor whenever the -00010 modification is required, even though in reality no borrow is involved.

5-14. BINARY-MODIFICATION DETECTION. Since any modification can result in only one or the other of the two binary states of a digit, only one of the resulting states needs to be detected by gating (heading 5-8, paragraph 9).

In the digit 1 adder-subtractor, gating was for all cases in which modification results in a number in the range 5 through 9—that is, binary 1. The gates detecting such cases are buffed to set the binary output flip-flop whose output is then a 1 in the Larc-code fourth-bit position.

A resultant binary-1 state is detected in all digit adder-subtractors except 2, where all cases are detected in which modification results in a number in the range 0 through 4—that is, binary 0. The gates detecting the latter cases are buffed to set the binary output flip-flop. The output connections of this flip-flop are such that, when it is set, it produces a 0 in the Larc-code fourth-bit position.

Gating in the adder-subtractors for the binary state of the modified address digit has to take into account (1) the original decimal state of the digit, (2) whether the digit has been increased by a carry, decreased by a borrow, or not changed, and (3) whether or not the digit is first decreased by 2 or increased by 5 or 7 during translation.

5-15. **PARITY-MODIFICATION DETECTION.** In the digit 5 adder-subtractor, as in 1, all instances in which modification produces an even digit are detected by gating, and any of these instances will set the parity output flip-flop to produce a 1 in the Larc-code fifth-bit position.

In the digit 2, 3, and 4 adder-subtractors, however, detection is for the cases in which an odd digit results from modification. Each of these cases will set the parity output flip-flop to produce a 0 in the Larc-code fifth-bit position.

Detection in the adder-subtractors of the oddness or evenness of a digit after modification takes into consideration (1) the original oddness or evenness of the digit, (2) whether the digit is changed, and (3) if changed, whether the digit was changed by an odd or an even number, since a number can be changed by any of the following numbers:

1	Carry or borrow or modification digit 9 or 1
2	Second digit of -00021 modification
3	Borrow and -00021 modification
5	Third digit of +97501 modification
6	Carry and +97501 modification
7	Fourth digit of +97501 modification
8	Carry and +97501 modification

#### 5-16. DIFFERENCE IN RETRANSLATION

In the digit 4 adder-subtractor the modified biquinary digit is not reconverted to Larc code as it is in all other adder-subtractors. Instead, it is stored in straight biquinary form in its seven output flip-flops for 2  $\mu$ s and then is gated in biquinary form to the address-register input pulseformers. Reconversion to Larc code is achieved at the input to the pulseformers.

#### 5-17. ADDRESS-MODIFIER TIMING

Figure 5-6 illustrates the timing of modification for three addresses. The modification of each address is a parallel operation with each digit adder-subtractor (except 5) modifying its corresponding digit at the same T37.

The address to be modified, having already been delayed by the address-register output pulseformers to T26, is recirculated through the same pulseformers to be present at the input to the address modifier at T2367. The address is recirculated for an extra 0.5- $\mu$ s delay. This delay, together with the other delays in the address modifier, insures that the modified address will be present at the time (T04) required for writing a modified address into an address register.



The control signals from the synchronizer are also present at T2367. Thus translation and modification of the first two digits take place at T2367, but the modified biquinary address is sampled only at T37, and therefore only T37 is relevant. Translation for digit 3 is at T2367, but the control signals generated from the digit 2 modification (SN2) are pulseformed to be present only at T37, and thus modification takes place and is sampled at T37. The translation of digit 4 is at T2367, but final modification is gated with a timing signal for T37, so that the modified biquinary digit is sampled at T37. The timing of the digit 5 adder-subtractor differs from that of the others: although translation is effected at T2367, modification (because of pulseforming for reasons of levels of drive) does not take place until T04, and the sampling of the modified biquinary fifth digit is therefore at T04, 0.5  $\mu$ s later than in all other digit positions.

Thus the first four modified biquinary digits are sampled at T37, retranslated (except 4), and stored in Larc code by their output flip-flops during T4...T7, or T0...T3; the modified digit 5 is sampled at T04, retranslated, and stored during T5...T0 or T1...T4. The five completely modified Larc-code digits are gated to the address-register input pulseformers simultaneously at T37, which for the first four digits is the last 0.5  $\mu$ s of storage, and for the fifth digit the next to last 0.5  $\mu$ s of storage.

## SECTION VI

### TYPICAL WORD-TRANSFER SEQUENCES

#### 6-1. GENERAL

The timing diagram in figure 6-1 shows the timing sequence for two typical word-transfer operations. These operations may also be traced on the diagram of figure 6-2, which is a pictorial summary of the complete dispatcher. Because of the large number of possibilities involved, it is impossible to describe every combination of operating conditions.

In figure 6-1 the A- and B-phase clock waveforms are shown at the top of the diagram. The first operation (in black) shows the time sequence for a synchronizer that obtains priority and successfully completes the word-transfer operation. Then the synchronizer relinquishes priority to a synchronizer of lower priority (in white) which receives priority to two successive time slots because it is the only calling synchronizer, even though it successfully completes its word transfer with the first reference. More detailed descriptions of these two word-transfer operations follow.

#### 6-2. WORD TRANSFER 1

The synchronizer indicated in black is assumed to have the highest priority of synchronizers requesting memory access. Its priority request (priority-bid FF set) therefore overrides other priority requests, and the synchronizer receives priority (PjA) for 2  $\mu$ s and delayed priority for a further 1.5  $\mu$ s (delayed-priority FF set).

The priority signal (PjA) alerts the gates which control the readout of the address from the address register, the gating of the address to the memory-address decoder, the regeneration of the address, and the setting of the address-modifier controls. The address is read out of the address register and is gated into the address-register output pulseformers, where it is reshaped and retimed. From there it is sent to the address-register input pulseformers to be written back into the same register so that it is available again if the memory is busy. The address is also sent to the address modifier for modification (that is, to become the next address to be used) and is gated as well to the memory-address decoder. The decoded address then is sent over the memory-address bus to the correct memory location.

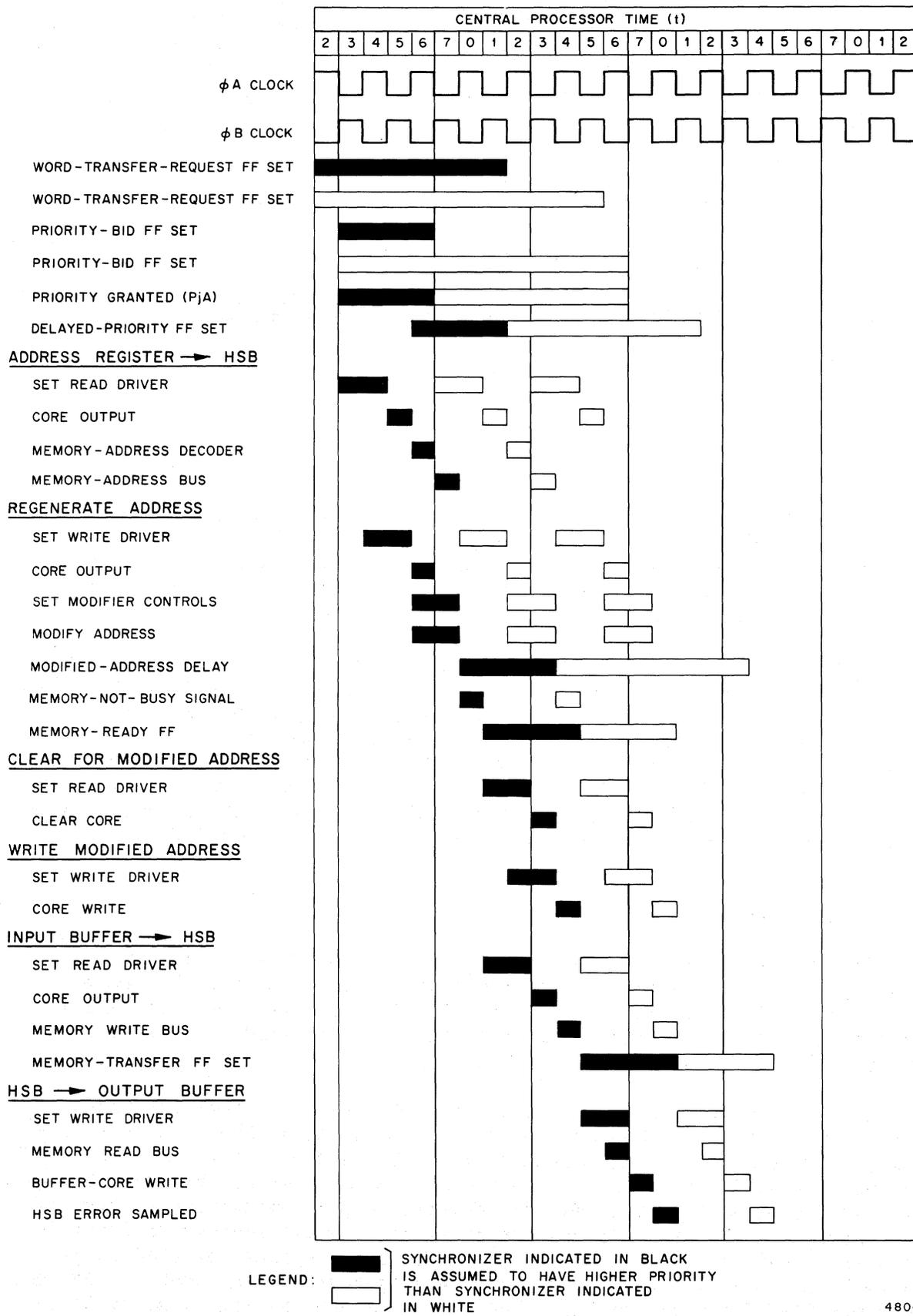
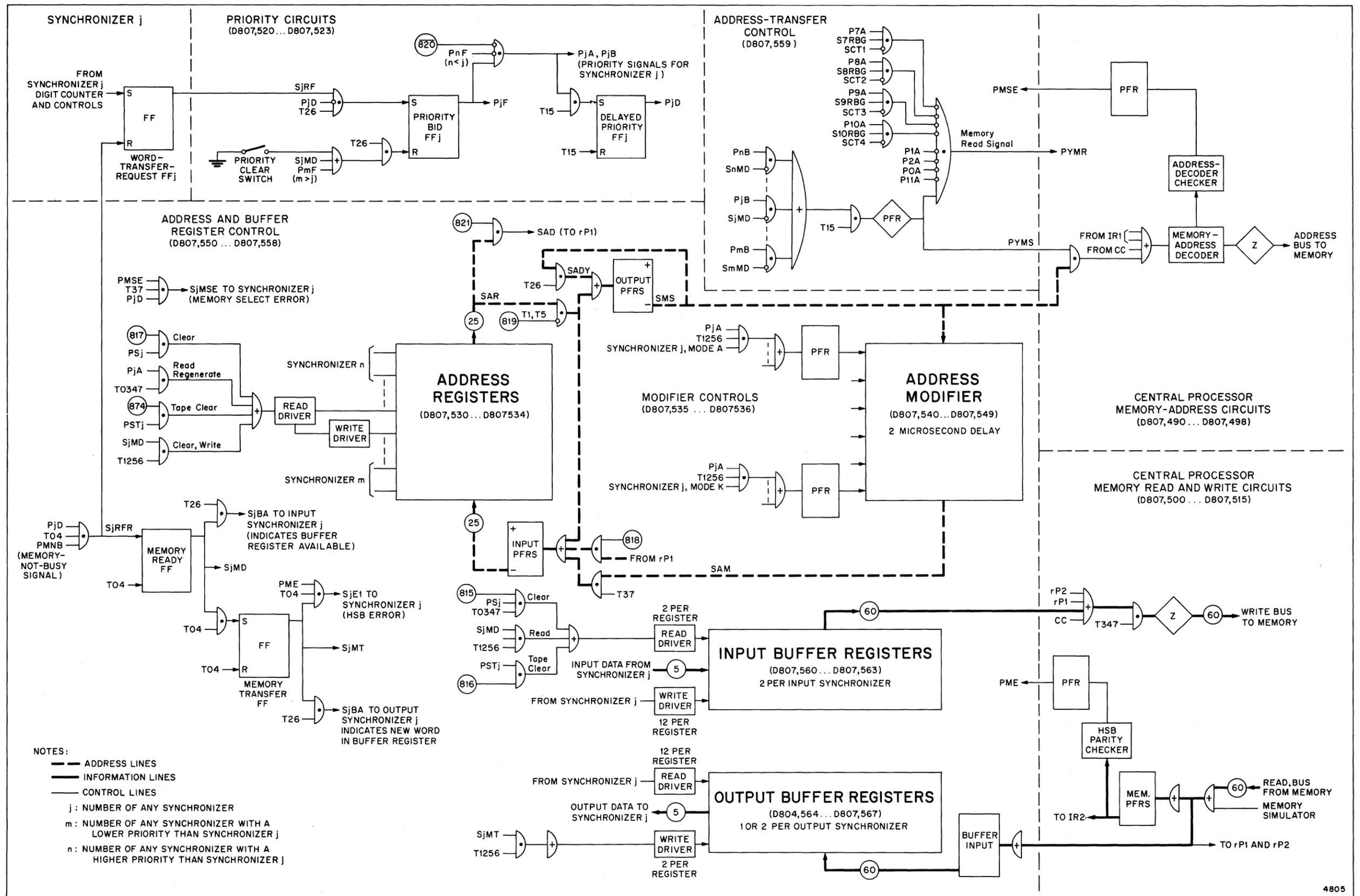


Figure 6-1. Typical Word-Transfer Sequences



4805

Figure 6-2. The Dispatcher: Simplified Logic Diagram

Delayed-priority signal PjD alerts the memory-ready FF, which is set by the memory-not-busy signal (MNB) returned from the memory unit. The memory-ready FF alerts the gates which control (1) the clearing of the now used address from the address register prior to writing in the modified address, (2) the writing into the address register of the modified address, and (3) the readout of the word in the input buffer register—if the synchronizer is an input synchronizer—for transmission to the memory over the write bus. The memory-ready FF also alerts the memory-transfer FF, which is set by a timing signal, and in turn alerts the gates controlling (1) the write-in to the output-buffer register of a word received across the memory-read bus—if the synchronizer is an output synchronizer, and (2) the sampling of the HSB-error signal.

If the memory happens to be busy when it is addressed, the MNB signal will not be returned. The memory-ready FF therefore will not be set, and the operations described as a consequence of its being set will not take place.

### 6-3. WORD TRANSFER 2

The synchronizer represented in white in figure 6-1 is assumed to have lower priority than the one in black. Its priority request (priority-bid FF set) therefore will be overridden until the first synchronizer relinquishes priority. Then the synchronizer in question receives priority and continues to hold it for 4  $\mu$ s, encompassing two time slots because it is now the only calling synchronizer. The word transfer is initiated on the first time slot and is successfully completed as described under heading 6-2. During the second 2- $\mu$ s priority period priority signal PjA continues to alert the address-register controls, and the address is read out of the address register and gated to the address-register output pulseformers. From there the address is regenerated in the same address register through the address-register input pulseformer. The address is also sent to the address modifier and modified, but it is not gated to the memory-address decoder; thus a second memory cycle is avoided. At the same time that the address would have been gated to the memory-address decoder, the memory-ready FF is still set as a result of the successful memory reference on the first time slot, and the set output of the FF inhibits the gating.

Although delayed-priority signal PjD continues to alert the set gate of the memory-ready FF during the second priority period, the FF is not set at that time. This is because the MNB signal is not returned from the memory since the memory was not addressed. Consequently the regenerated address is not cleared, the modified address is not written into the address register, and the word is not read from the input-buffer register or written into the output-buffer register.