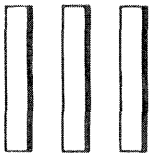
RESTRICTED DISTRIBUTION

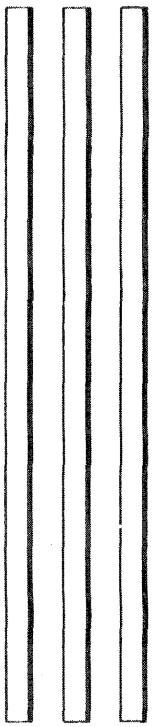
MARCH 1962



MANUAL

1

CENTRAL PROCESSOR



LOGIC DESCRIPTION

UNIVAC® III

DATA-PROCESSING SYSTEM

Remington Rand Univac®
DIVISION OF SPERRY RAND CORPORATION
315 PARK AVENUE SOUTH
NEW YORK 10, N Y

Published by
UNIVAC ENGINEERING CENTER, PHILADELPHIA
Post Office Box 500
Blue Bell, Pa.

Section 2

Introduction to the UNIVAC III Logic Components & Assemblies; and a Description of the Cycling Unit

Under Preparation

(Sections 1, 3-8 herewith)

CONTENTS

Heading	Title	Page
	SECTION 1. INTRODUCTION AND GENERAL DESCRIPTION	
1-1.	Central-Processor Specifications	1-1
1-2.	UNIVAC III Central-Processor Logic Features	1-2
1-3.	Four-bit Parallel Transfer	1-2
1-4.	Overlapping Operation	1-2
1-5.	Multiprecision Instructions	1-2
1-6.	Priority	1-2
1-7.	Multiply Instructions	1-2
1-8.	Initiate Input-Output Instruction	1-2
1-9.	Automatic-Programming Aids	1-3
1-10.	Scatter-Read/Gather-Write Control Word	1-3
1-11.	Number-System Compatibility	1-3
1-12.	Checking Features	1-3
1-13.	Functional Components of Central Processor	1-3
1-14.	Computer-Control Register	1-3
1-15.	Instruction-Control Section	1-3
1-16.	Storage-Address Section	1-3
1-17.	Field-Select Section	1-3
1-18.	Memory-Address Section	1-4
1-19.	Arithmetic Section	1-4
1-20.	Input-Output Control Section	1-4
1-21.	Priority Section	1-4
1-22.	Write Register	1-4
1-23.	Sense-Flip-Flop Section	1-4
1-24.	Interrupt Section	1-4
1-25.	Typewriter and Real-Time-Clock Section	1-6
1-26.	Cycling-Unit Section and Display Section	1-6
1-27.	UNIVAC III Word Formats	1-6
1-28.	Data Words	1-6
1-29.	Instruction Word	1-6
1-30.	Input-Output Specification Words	1-7
1-31.	Control Words	1-7
1-32.	Data Flow in the UNIVAC III Central Processor	1-8
1-33.	Data Flow During an Instruction Call	1-8
1-34.	Data Flow when Reading the Instruction from Memory	1-8
1-35.	Data Flow During the First Operand Call	1-8
1-36.	Data Flow During the Field Selection and Indirect Addressing	1-8
1-37.	Data Flow During Multiprecision Instructions	1-10
1-38.	Data Flow During Instructions which use the Main Adder	1-10
1-39.	Recomplementing	1-11
1-40.	High, Low, and Equal Flip-Flops	1-12
1-41.	Extract Instructions	1-12
1-42.	Multiply and Divide Instructions	1-12
1-43.	Store- and Load-Accumulator Instructions	1-12
1-44.	Data Flow During Instructions which Transfer Data Between the Storage-Address Registers and Memory	1-13

Heading	Title	Page
1-45.	Data Flow During Instructions which Store the Contents of a Storage-Address Register	1-14
1-46.	Data Flow During Instructions which Load a Storage-Address Register	1-14
1-47.	Data Flow During Instructions which Modify and Compare the Contents of an Index Register	1-14
1-48.	Data Flow During Shift Instructions	1-15
1-49.	Data Flow During Miscellaneous Instructions	1-15
1-50.	Sense-Flip-Flop Instructions	1-15
1-51.	Interrupt Instructions	1-15
1-52.	Data Flow During the Expand Instruction	1-16
1-53.	Data Flow During a Compress Instruction	1-16
1-54.	Data Flow During Transfer Instructions	1-17
1-55.	Data Flow During Zero-Suppress Instruction	1-17
1-56.	Data Flow During Typewriter Instructions	1-17
1-57.	Data Flow During Input-Output Operations	1-18
1-58.	Data Flow when Transferring the Specification Word to Memory	1-19
1-59.	Data Flow when Transferring the Specification Word From the Standby Location	1-21
1-60.	Data Flow for Data Word for All Devices Except Tape Units	1-21
1-61.	Data Flow for Transferring the Scatter-Read/ Gather-Write Control Word	1-21
1-62.	Data Flow for Words During Tape Operations	1-21
1-63.	UNIVAC III Instructions	1-21

SECTION 3. INSTRUCTION-CONTROL SECTION

3-1.	Introduction	3-1
3-2.	Instruction Decoder	3-1
3-3.	Program Counter	3-1
3-4.	Accumulator-Register Selector	3-2
3-5.	Index-Register Selector	3-2
3-6.	Memory-Request Network	3-3
3-7.	Instruction Decoder	3-3
3-8.	Instruction Register	3-3
3-9.	Octal Decoder	3-4
3-10.	Group Encoder	3-4
3-11.	Instruction-Line Generator	3-4
3-12.	Program Counter	3-5
3-13.	General Operation	3-5
3-14.	Program-Counter Cycles	3-5
3-15.	Normal Cycle	3-5
3-16.	Shift Cycle	3-5
3-17.	Recomplement Cycle	3-5
3-18.	Multiply Cycle	3-8
3-19.	Divide Cycle	3-8
3-20.	Accumulator-Register Selector	3-8
3-21.	AR-Select Flip-Flops	3-8
3-22.	AR-Decode Network	3-8
3-23.	AR-Selection Network	3-9
3-24.	Function Signals 235, 236, 237, and 238	3-10
3-25.	Function Signal 128	3-10
3-26.	Function Signal 129	3-10
3-27.	Selection of Accumulator Registers	3-11
3-28.	Precision Selection	3-12
3-29.	Other Selection Functions of the AR-Select Flip-Flops	3-13
3-30.	Index-Register Selector	3-13

Heading	Title	Page
3-31.	Instruction Memory Requests	3-13
3-32.	Instruction-Control Signal	3-13
3-33.	Multiprecision Instructions	3-14
3-34.	Zero-Suppress Instruction	3-15
3-35.	Shift Instructions	3-15
3-36.	Error Conditions	3-15
3-37.	Operand Memory Requests	3-15
3-38.	Instruction-Control Signals	3-15
3-39.	The CVEP Signal	3-15
3-40.	The CVRINH Signal	3-16
3-41.	The CVRECP Signal	3-19
3-42.	The CCVA Signal	3-19

SECTION 4. PRIORITY SECTION

4-1.	Introduction	4-1
4-2.	Priority Order	4-1
4-3.	Operation of Priority Channels	4-2
4-4.	Operation of Central-Processor Channels	4-2
4-5.	Automatic Operand Call	4-2
4-6.	Transfer Instructions	4-3
4-7.	Automatic Interrupt	4-4
4-8.	Multioperand Calls	4-4
4-9.	Priority Inhibits	4-4
4-10.	Priority Manual Reset	4-5

SECTION 5. COMPUTER-CONTROL REGISTER

5-1.	Introduction	5-1
5-2.	General Operation	5-1
5-3.	Logic of Computer-Control Register	5-1
5-4.	Register Functions	5-1
5-5.	Resetting	5-3
5-6.	Jamming Binary Zeros	5-3
5-7.	Decimal Shifting and Circular-Decimal Shifting	5-3
5-8.	Binary Shifting and Circular-Binary Shifting	5-5
5-9.	Register Inputs	5-5
5-10.	Memory Input	5-5
5-11.	Typewriter Input	5-5
5-12.	Real-Time-Clock Input	5-5
5-13.	Interrupt Inputs	5-8
5-14.	Accumulator-Register Inputs	5-8
5-15.	Synchronizer-Control-Register Inputs	5-8
5-16.	Register Outputs	5-11

SECTION 6. STORAGE-ADDRESS SECTION

6-1.	Introduction	6-1
6-2.	Selection of SAr's and TCWr's	6-2
6-3.	Inputs to SAr's and TCWr's	6-2
6-4.	Outputs from SAr's and TCWr's	6-2
6-5.	Logic of the SAr and TCWr Selection	6-2
6-6.	Selection of SAr's	6-4
6-7.	Selection of Index-Register	6-4

Heading	Title	Page
6-8.	Selection of Memory-Address Counters	6-4
6-9.	Selection of TCWr's	6-6
6-10.	SAr and TCWr Outputs	6-8

SECTION 7. MEMORY-ADDRESS SECTION

7-1.	Introduction	7-1
7-2.	Memory Adder	7-2
7-3.	Subtrahend Inputs	7-2
7-4.	Minuend Inputs	7-3
7-5.	CMAC Pulseformer Inputs	7-5
7-6.	Memory-Adder Addition Logic	7-6
7-7.	Memory-Address Parity Generator	7-8
7-8.	Generation of Parity Bit From the Memory-Adder Output	7-8
7-9.	Generation of Parity Bit During Store-SAr or Store-TCWr Instructions	7-10
7-10.	Generation of Parity Bit During Initiate I-O Instruction	7-10
7-11.	Modulo-3 Check-Bit Generation	7-10
7-12.	Memory-Address Selector	7-12
7-13.	Memory-Address Selector Operation	7-12
7-14.	Memory-Address Selector Inputs	7-12
7-15.	Memory-Cabinet Selector	7-15
7-16.	Zero-Detector Flip-Flop	7-16
7-17.	Tape Operation	7-16
7-18.	The 53 Instruction	7-16

SECTION 8. MEMORY-WRITE REGISTER

8-1.	Introduction	8-1
8-2.	Central-Processor Format	8-1
8-3.	Accumulator Register	8-1
8-4.	Memory Adder	8-2
8-5.	UNISERVO III Format	8-2
8-6.	Input-Output Format	8-4
8-7.	Computer-Control-Register Format, Expand Instruction Only	8-7
8-8.	Maintenance-Panel Format	8-9
8-9.	Summary of Functions	8-9

APPENDIX A. ENGINEERING-DRAWING SOURCES OF ILLUSTRATIONS

ILLUSTRATIONS

Figure	Title	Page
1-1.	Central Processor, Functional Block Diagram	1-5
1-2.	Data Flow During an Instruction Call, Functional Block Diagram	1-9

Figure	Title	Page
1-3.	Data Flow During Multiprecision Instructions, Functional Block Diagram	1-10
1-4.	Data Flow During Instructions Which Use the Main Adder, Functional Block Diagram	1-11
1-5.	Example of Double-Precision Add with Carries, Functional Block Diagram	1-12
1-6.	Loading Accumulator Registers from Adjacent Memory Locations, Diagram	1-12
1-7.	Storing Contents of Accumulator into Adjacent Memory Locations, Diagram	1-13
1-8.	Data Flow During Storage-Address-Register Load and Store Instructions, Functional Block Diagram	1-13
1-9.	Data Flow During Index-Register Modify and Compare Instructions, Functional Block Diagram	1-14
1-10.	Shifting Right with Right Shifts or with Left Shifts, Diagram	1-15
1-11.	Data Flow During Double-Precision Two-Digit Right Shift, Diagram	1-16
1-12.	Data Flow During Double-Precision Two-Digit Left Shift, Diagram	1-17
1-13.	Data Flow During Typewriter Interrupt, and Sense-Flip-Flop Instructions, Functional Block Diagram	1-18
1-14.	Data Flow During Expand and Compress Instructions, Functional Block Diagram	1-19
1-15.	Data Flow During Input-Output Operations, Functional Block Diagram	1-20
3-1.	Instruction-Control Section, Functional Block Diagram	3-2
3-2.	Instruction Register and Octal Decoder, Detailed Block Diagram	3-3
3-3.	Program Counter, Detailed Block Diagram	3-6
3-4.	Timing Diagram for Typical Program Cycles	3-7
3-5.	AR-Select FF's and Decode Network, Detailed Block Diagram	3-9
3-6.	Production of Function Signals 235, 236, 237, and 238, Detailed Block Diagram	3-11
3-7.	Production of Precision Signals, Detailed Block Diagram	3-12
3-8.	Index-Register Selector, Detailed Block Diagram	3-14
3-9.	Production of the CVEP Signal, Detailed Block Diagram and Timing Diagram	3-16
3-10.	Production of the CVRINH Signal, Detailed Block Diagram and Timing Diagram	3-17
3-11.	Production of the CCVA Signal, Detailed Block Diagram and Timing Diagram	3-18
4-1.	Typical Priority Channel, Detailed Block Diagram	4-2
4-2.	Automatic Operand Call, Detailed Block Diagram	4-3
4-3.	Transfer Instructions, Detailed Block Diagram	4-4
5-1.	Computer-Control Register, Functional Block Diagram	5-2
5-2.	Example of the Transfer of a Word from Memory Through the Computer-Control Register to Main Adder, Functional Block Diagram	5-3
5-3.	Decimal Shift, Circular-Decimal Shift, Binary-Zero Entries into the Computer-Control Register, Detailed Block Diagram	5-4
5-4.	Binary Shift, Circular-Binary Shift, and Jammed Zeros into the Computer-Control Register, Detailed Block Diagram	5-6

Figure	Title	Page
5-5.	Parallel Inputs from Memory, Real-Time Clock, and Typewriter into the Computer-Control Register, Detailed Block Diagram	5-7
5-6.	Accumulator Register Loading Computer-Control Register and XS-3 Zeros Entered During Multiply and Divide, Detailed Block Diagram.	5-9
5-7.	Synchronizer-Control Register Loading the Computer-Control Register, Detailed Block Diagram	5-10
6-1.	Storage-Address Section, Functional Block Diagram	6-1
6-2.	Storage-Address-Register and Tape-Control-Word-Register Selection, Detailed Block Diagram	6-3
6-3.	Production of Storage-Address-Register and Tape-Control-Word-Register Function Signals, Detailed Block Diagram	6-5
6-4.	Production of Function Signals 222 through 234, Detailed Block Diagram	6-7
7-1.	Memory-Address Section, Functional Block Diagram	7-1
7-2.	Production of Memory-Adder Subtrahend Inputs, Detailed Block Diagram	7-3
7-3.	Production of Memory-Adder Minuend Inputs, Detailed Block Diagram	7-4
7-4.	Production of Function Signal 104, Detailed Block Diagram	7-6
7-5.	Production of Add-1 (CMAC) and Add-2 (Function Signal 212) Signals, Detailed Block Diagram	7-7
7-6.	Memory-Adder Parity-Bit Generation, Detailed Block Diagram	7-9
7-7.	Modulo-3 Check-Bit Generation, Detailed Block Diagram	7-11
7-8.	Memory-Address Selector, Functional Block Diagram	7-13
7-9.	Memory-Address Selector, Detailed Block Diagram	7-14
7-10.	Cabinet Selector, Detailed Block Diagram	7-15
7-11.	Zero Detector, Detailed Block Diagram	7-16
8-1.	Memory-Write Register, Functional Block Diagram	8-2
8-2.	Memory-Write Register Loading from the Accumulator Registers, Detailed Block Diagram	8-3
8-3.	Production of CVRW1 and Function Signals 121, 270, and 271, Detailed Block Diagram	8-4
8-4.	Memory-Write Register Loading from the Memory Adder, Detailed Block Diagram	8-5
8-5.	Memory-Write Register Loading from the UNISERVO III Channels, Detailed Block Diagram	8-6
8-6.	Production of Function Signals 131, 148, and 149, Detailed Block Diagram	8-7
8-7.	Memory-Write Register Loading from UNISERVO II and General-Purpose Channels, Detailed Block Diagram	8-8
8-8.	Production of Function Signals 130, 132 through 139, Detailed Block Diagram	8-9
8-9.	Memory-Write Register Loading from the Computer-Control Register During Expand Instruction, Detailed Block Diagram	8-11
8-10.	Production of Function Signals 277, 278, and 280, and the CMW26EJ and CMW27EJ Signals, Detailed Block Diagram	8-12

TABLES

Table	Title	Page
1-1.	UNIVAC III Instructions	1-21
1-2.	Sense-Flip-Flop Instruction Designations	1-24
3-1.	Decoding of AR-Select Flip-Flop Outputs	3-10
3-2.	Selection of Accumulator Registers During 08-Type Instructions by Function Signals 235 through 238	3-12
3-3.	Selection of Accumulator Registers During 09-Type Instructions by Function Signals 235 through 238	3-12
3-4.	Precision Signals for All Bit Combinations in the AR-Select Flip-Flops	3-13
4-1.	Order of Priority Channel	4-1
5-1.	Interrupt Instruction Words	5-8
5-2.	Computer-Control-Register Outputs	5-11
5-3.	Summary of Function Signals, Which Control Transfer Operations to and Within Computer-Control-Register Flip-Flops, and Interrupt Inputs	5-11
5-4.	Summary of Input Sources and Word Formation	5-12
6-1.	Input and Output Timing of Memory-Address Counter, Index Register, Tape-Control-Word Register	6-2
6-2.	Production of Function Signal 222 by CSG03	6-6
6-3.	Tape-Control-Word-Register Input Information	6-8
7-1.	Production of the Final Sum and Partial Carry for the First Bit Position	7-8
7-2.	Production of Partial Sums and Partial Carries for Bit Positions 2 through 4	7-8
7-3.	Production of Final Sum Bits of 1 for Bit Positions 2 through 4	7-8
7-4.	Parity Bit Production From First Gating Network	7-9
7-5.	Production of Final Parity Bit	7-10
7-6.	Summary of Modulo-3 Check-Bit Generation	7-12
8-1.	Memory-Write-Register Inputs from UNISERVO II and General-Purpose Channels, and Function Signals	8-4
8-2.	Summary of Function Signals Which Enter and Gate Inputs in the Memory-Write Register	8-9
8-3.	Bit-Entry Sequence of Input Formats into Memory-Write Register	8-10
A-1.	Figure References and Engineering Drawings for Section 2	A-1
A-2.	Figure References and Engineering Drawings for Section 3	A-1
A-3.	Figure References and Engineering Drawings for Section 4	A-1
A-4.	Figure References and Engineering Drawings for Section 5	A-1
A-5.	Figure References and Engineering Drawings for Section 6	A-2
A-6.	Figure References and Engineering Drawings for Section 7	A-2
A-7.	Figure References and Engineering Drawings for Section 8	A-3

Section 1
INTRODUCTION AND GENERAL DESCRIPTION

This manual describes the logical operation of the UNIVAC ® III Central Processor Model 121. No attempt is made to explain in complete detail the interrelation of all logic components as shown on the logic diagrams. It is assumed that the reader will be able to determine the details of the central-processor operations from the logic diagrams. In order to aid in the transition from the manual to the logic diagrams, all signal designations mentioned in the manual are presented in the same form that they appear on the logic diagrams, and appendix A lists all logic prints and the print revisions on which the information in the manual is based.

The related manuals in the UNIVAC III manual series are as follows:

Manual 2, Synchronizers for Peripheral Equipment — Logic Description

Manual 3, Logic Diagrams

Manual 4, Signal Analysis

Manual 5, Circuitry

Manual 7, Core Storage Model 122 and Model 197 — Description and Maintenance

Section 1 of manual 1 describes, in general, the major logic areas of the central processor, and section 2 describes all UNIVAC III system logic elements and assemblies. Sections 3 through 14 describe, in detail, the operation of each logic area that was introduced in section 1. The organization of the manual is planned so that the reader may, after reading sections 1 and 2, proceed to any of the twelve sections (sections 3 through 14) without the necessity of reading all sections in sequence.

Sections 1 through 8 contain the introduction to the manual and an explanation of the logic necessary to read and index instructions. Sections 9 through 14 describe the logic necessary to execute the central-processor instructions.

1-1. CENTRAL-PROCESSOR SPECIFICATIONS

The central processor performs all the data-processing functions of the UNIVAC III computing system and coordinates the operations of all input-output devices in the system. The design methods that were so successfully used in the production and maintenance of the LARC* and other UNIVAC solid-state computing systems have been applied to the UNIVAC III central processor. Because of this experience, the UNIVAC III central processor offers features in a medium priced computing system that are normally found only in higher priced systems.

The general, programming, arithmetic, and storage specifications for the UNIVAC III central processor are as follows:

General

Synchronous operation	4 μsec per minor cycle 9 pulse times per minor cycle
Serial-parallel transfer	
27-bit words	24 data bits, 1 sign bit, 2 check bits
Typewriter input and output	
15 index registers	
Automatic sequencing of memory accesses between central processor and input-output units	

The following registered trademark of the Sperry Rand Corporation is used in this manual: UNIVAC.

*Trademark of the Sperry Rand Corporation.

Programming	
Single-address instructions	
Stored program	
61 well defined instructions in SALT code	
8 programmable sense FF's	
Automatic interrupt	Tested by the program
Automatic field selection	
Indirect addressing	
Arithmetic	
Operands	6, 12, 18, or 24 decimal digits 24, 48, 72, or 96 binary digits 4, 8, 12, or 16 alphanumeric characters
Arithmetic operations performed with either binary or decimal numbers	
Storage	
Expandable magnetic-core memory	8192, 16,384, 24,576, or 32,768 words

1-2. UNIVAC III CENTRAL-PROCESSOR LOGIC FEATURES

Some of the outstanding features of the UNIVAC III central processor are: 4-bit parallel transfer, overlapping operation, multiprecision instructions, priority, multiply instructions, initiate input-output instructions, automatic programming aids, scatter-read/gather-write control, number-system compatibility, and checking features.

1-3. Four-bit Parallel Transfer

All data format manipulation in the UNIVAC III central processor is performed four bits (a decimal digit) at a time, and all information is transferred within the central processor four bits at a time. The use of the decimal-digit-at-a-time transfer represents the most efficient way of handling data in a medium priced decimal computer because the decimal digit is the smallest unit of information that is normally required.

1-4. Overlapping Operation

The reading of an instruction from memory and the indexing of the instruction is performed while the previous instruction is being executed. This overlapping of instructions is normally found only in the most expensive data processing systems.

1-5. Multiprecision Instructions

The UNIVAC III instruction catalogue permits the programmer to designate the precision of the arithmetic operation that is to be performed. This presents two distinct advantages: one, the programmer can work directly with numbers that contain as many as 24 decimal digits or 96 binary digits without any programming manipulation; two, an arithmetic operation takes only as long to perform as is necessary. No time is wasted adding meaningless digits when they are not required by the operation. Thus, the greater precision of the long word is combined with the faster operation and economy of the short word.

1-6. Priority

The UNIVAC III central processor automatically allots memory access to the central processor and to the input-output (I-O) devices in a way that makes best use of computer operating time. This is a feature that is normally found only in the most expensive computing systems.

1-7. Multiply Instructions

The multiply instruction uses a special technique called short cut multiply which decreases by one-third the maximum number of additions necessary for a multiply problem.

1-8. Initiate Input-Output Instruction

The instruction which initiates the input-output operations, supplies the designated input-output synchronizer with all necessary control information, and then transfers control of the input-output operation to the synchronizer. After the initiate I-O instruction has been performed, the only central processor time that is required for an input-output operation is the time necessary to transfer a word between a synchronizer and memory. Thus the high-speed central processor is not limited by the slower speed input-output devices.

1-9. Automatic-Programming Aids

Automatic field selection and indirect addressing, which are invaluable aids in data processing applications, are built into the central processor. Also, the extensive use of the sense flip-flop and automatic-interrupt features provide the programmer and the operator with maximum control over the system.

1-10. Scatter-Read/Gather-Write Control Word

The use of the scatter-read/gather-write control word allows one central-processor instruction to control the transfer of data between magnetic tape and various segments of memory. This transfer is done by the automatic reading of sequential scatter-read/gather-write control words from memory. These control words contain the address of the first memory location of a memory segment and the number of words in that segment. When the data from one segment are processed, the next scatter-read/gather-write control word is automatically read from memory. Thus, information can be transferred between a tape unit and various segments of memory with one central-processor instruction. This represents a great saving in program time.

1-11. Number-System Compatibility

The central processor performs arithmetic operations with either binary or decimal numbers.

1-12. Checking Features

All transfers between the central processor and memory are checked both for correct transfer and for address decoding. Also, all addition and subtraction operations are automatically checked.

1-13. FUNCTIONAL COMPONENTS OF CENTRAL PROCESSOR

Figure 1-1 shows the functional components of the central processor and the major data-flow paths between these components. The purpose of each functional component will be explained in the text that follows; the data flow will be explained later in this section (heading 1-32).

1-14. Computer-Control Register

The computer-control register (CCr) is the main working register in the central processor. This register performs all the shifting and data manipulation that is required by the central processor. The CCr also receives all the memory inputs which contain central-processor data or instructions.

1-15. Instruction-Control Section

The instruction-control section receives each central-processor instruction, partly from memory and partly from the CCr, and then decodes these instructions into various instruction lines, program counts, and register selection outputs as specified by the instruction. These signals then are used by other sections of the central processor to control the execution of the instruction.

1-16. Storage-Address Section

The storage-address section contains a maximum of 30 storage-address registers (SAr). These registers include a maximum of 15 index registers (Ir) which are used to modify the instruction address; and a maximum of 15-memory-address counters (MAC), which contain input-output memory addresses, an instruction memory address, and an operand memory address. The MAC that contains the instruction memory address is called the control counter (CC), and it stores the address of the last instruction that was read from memory. The address of the last operand that was read from memory is stored in the MAC called the memory-address register (MAr). All other MAC's contain memory-address information for input-output operations. There is one MAC for each input-output operation and one MAC for each input-output channel.

The MAC's and Ir's contain 15 bits each; correspondingly numbered MAC's and Ir's occupy one 36-bit register (6 bits of this register are not used).

In addition to the storage-address registers, the storage-address section also contains four 24-bit tape-control-word registers (TCWr) which hold control and address information for tape operations, and which occupy one 36-bit register each.

1-17. Field-Select Section

The field-select section, which contains 10 flip-flops and control logic, performs the following functions in the central processor:

Provides signals that control field selection or indirect addressing;

Provides a path for memory-address information during the transfer of the contents of a SAr or a TCWr to memory;

Provides temporary storage of arithmetic-carry information during multiprecision operations;

Provides a means of deleting zeros and commas during zero-suppress operations;

Stores and modifies the shift count; and

Provides a means of shifting during multiply and divide operations.

1-18. Memory-Address Section

The memory-address section performs all memory-address modification and supplies addresses to memory for both read and write operations. The section contains an adder and a register. The adder, called the memory adder, performs all address modification, provides a transfer path for storing data during certain instructions, and provides a means of comparing data during other instructions. The register, called the memory-address selector (MAS), converts the address format from serial to parallel, supplies an address to memory, and provides a path for loading either the SAr's or the TCWr's.

1-19. Arithmetic Section

The arithmetic section performs all the arithmetic operations and the data modification for the central processor and provides temporary storage for information. The arithmetic section contains four accumulator registers (Ar1 through Ar4) and an adder (called the main adder). The accumulator registers are addressed either singly or in combination. When addressed, they supply one word at a time to either the write register for storage into memory, or to the computer-control register for data manipupation, or to the main adder for modification.

When more than one Ar is addressed, the register designated by the smaller number contains the most significant digits, and the register designated by the larger number contains the least significant digits. For example, if Ar1 and Ar4 are designated, Ar1 contains the most significant digits, and Ar4 contains the least significant digits.

The main adder performs the addition process for all arithmetic instructions. Besides its function in the arithmetic instructions, the main adder is used in logical operation instructions (erase and superimpose), is used in comparison instructions, and is the only input path to the Ar's. In all these uses the main adder performs the addition process; the different results are achieved by varying the inputs to the adder.

1-20. Input-Output Control Section

The input-output control section coordinates data transfer between memory and the synchronizers, and checks all data transfers between memory and the central processor. The section contains one register, the synchronizer-control register (SCr), which provides temporary storage for all information that is read from memory. Control logic in the section checks data transfers, arranges data in the SCr into the input-output formats, and coordinates the input-output operations.

1-21. Priority Section

The priority section receives memory requests from the central processor and from the various input-output synchronizers and allots memory access according to a predetermined sequence. The outputs from the priority section control the reading and restoring of memory addresses in the storage-address section. The priority-section outputs are also encoded to produce an address during input-output operations.

1-22. Write Register

The write register supplies data to memory during all store operations. The register receives data in different formats from the arithmetic section, the synchronizers, the memory-address section, and the computer-control register. The write register then arranges these formats into the memory format and supplies the data to memory.

1-23. Sense-Flip-Flop Section

The sense-FF section contains eight sense FF's and an inhibit-input-output-interrupt FF, all of which are set, reset, or tested by an instruction; and three comparison FF's, which represent equal to, greater than, or less than conditions. The appropriate high, low, or equal flip-flop is set during specific instructions if the minuend input is either equal to, greater than, or less than the subtrahend input. The comparisons are made either in the memory adder or the main adder. All the flip-flops in the sense-FF section can be tested by an instruction. If the flip-flop is set when tested, program control is transferred to a subroutine.

1-24. Interrupt Section

The interrupt section receives inputs from the central processor and the synchronizers, and it interrupts the program when there are certain predetermined conditions. When the program is interrupted,

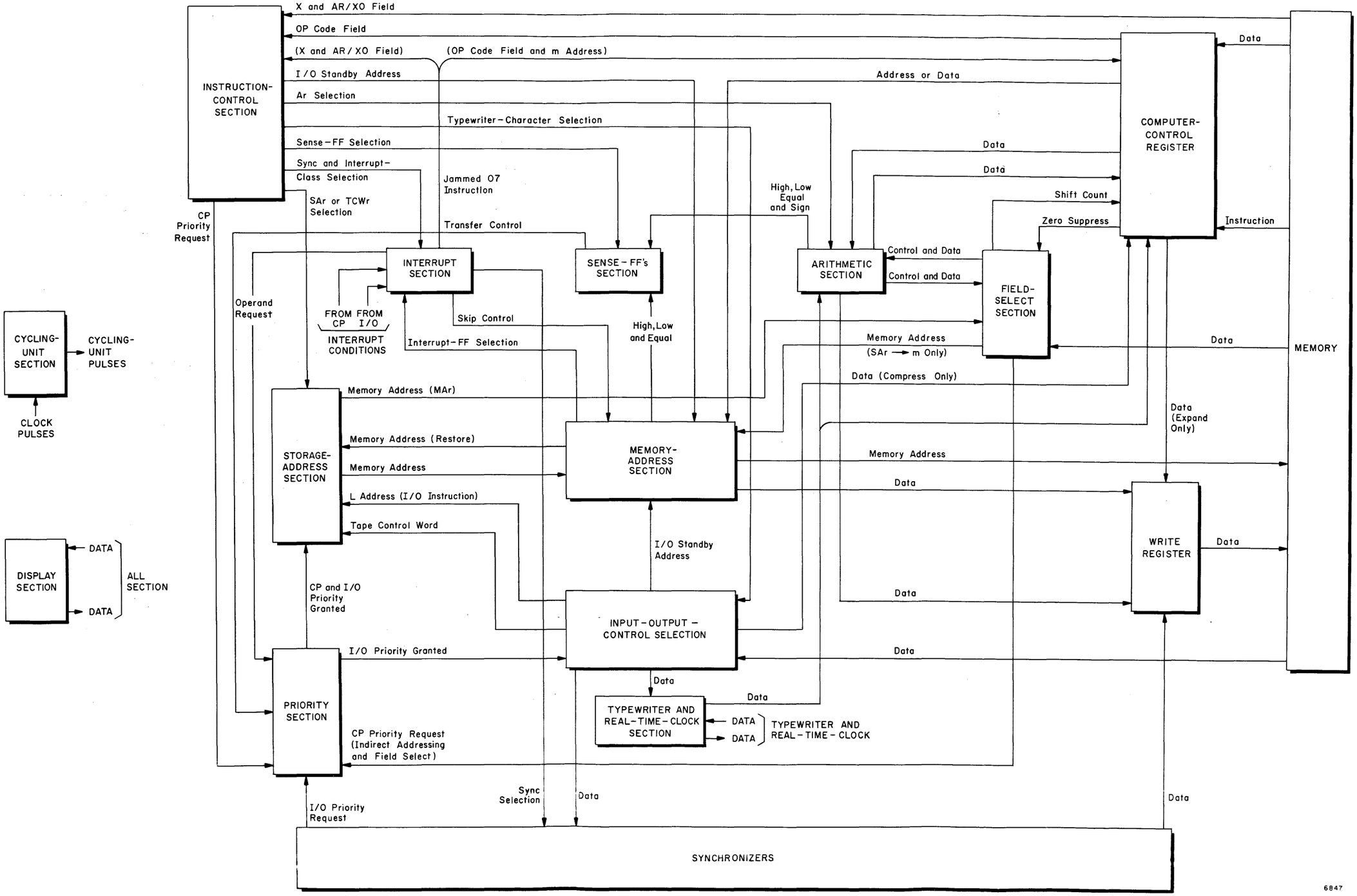


Figure 1-1. Central Processor, Functional Block Diagram

a transfer instruction is jammed into the CCr and into the instruction-control section. This instruction is then executed.

1-25. Typewriter and Real-Time-Clock Section

The typewriter and real-time-clock section controls all data transfers between the central processor and either the typewriter or the real-time clock. The section contains a 6-bit buffer register which stores one character at a time; a translator, which translates the 6-bit character into typewriter code; and control logic.

1-26. Cycling-Unit Section and Display Section

The cycling unit supplies all sections of the central processor with timing signals. These signals are produced and controlled by the central clock. The display section contains control logic which lights the indicators on the engineer's console and which permits the manual control of computer operations from the engineer's console.

1-27. UNIVAC III WORD FORMATS

There are four types of word formats used in the UNIVAC III central processor. These types are as follows:

Data words, which represent three types of data;

Instruction words, which represent all instructions;

Input-output specification words, which control the operation of the input-output devices; and

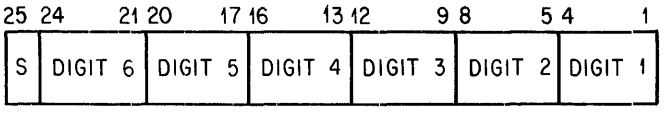
Control words, which perform special functions during specific instructions.

All word formats contain check bits in bit-positions 26 and 27. These check bits are not included in the descriptions that follow.

1-28. Data Words

There are three types of data words: the decimal data word, the alphanumeric data word, and the binary data word.

The format of the decimal data word is as follows:

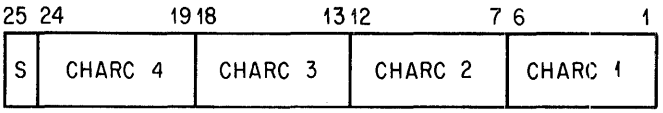


where

DIGIT (1 through 6) represents six decimal digits in XS-3 code.

S represents the sign bit (0 for positive, 1 for negative).

The format of the alphanumeric data word is as follows:

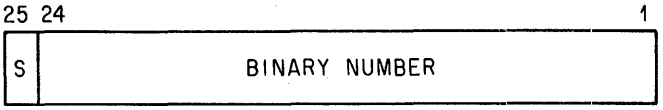


where

CHARC (1 through 4) represents four alphanumeric characters.

S is the same as in the decimal data word.

The format of the binary data word is as follows:



where

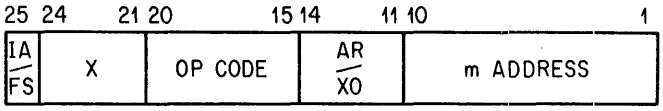
BINARY NUMBER represents one 24-bit binary number.

S is the same as in the decimal data word.

1-29. Instruction Word

There is one general instruction word which is used for all instructions.

The format of the instruction word is as follows:



where

m ADDRESS represents the first ten bits (unmodified) of a memory address; a shift count; or an interrupt FF. (The unmodified address is referred to as the m address and the modified address is referred to as the m' address. The contents of the m-ADDRESS field are automatically modified by the contents of the index register specified by the contents of the X-field.)

AR/XO represents the following:

The precision of an arithmetic operation, and the Ar's specified in instructions that use Ar's;

The sense FF that is to be addressed during a sense-FF instruction;

The SAr designated during an SAr-load or -store instruction;

The TCWr that is specified during a TCWr-store instruction;

The type of interrupt FF specified during a test, or reset interrupt-FF instruction; or

The synchronizer and standby location to be used during an initiate I-O instruction.

OP CODE is the octal representation of the instruction to be performed. X represents the index register which modified the m-ADDRESS field. IA/FS represents the indirect-addressing or field-selection control bit.

1-30. Input-Output Specification Words

Bit positions 1 through 15 of the input-output specification word contain the L address which is the address of the first data word, and bit positions 17 through 24 contain information which controls the operation of the synchronizer. The specific function performed by the contents of bit positions 17 through 24 varies with each input-output unit and these functions are described in the synchronizer manuals.

Bit position 16 of the I-O specification word causes a program interrupt if it contains a 1-bit.

1-31. Control Words

The four types of control words are as follows:

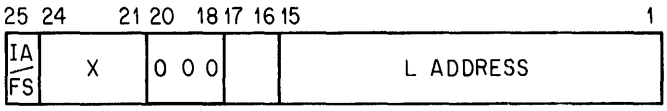
Indirect-addressing control word, which specifies the address of the next operand during indirect addressing;

Field-selection control word, which specifies the address of an operand and selects a specified field within that operand;

Index-register-modification control word, which specifies the number with which an index register is to be modified and compared; and

Scatter-read/gather-write control word, which specifies the address of the first data word and the number of words to be transferred during a tape operation.

The format of the indirect-addressing control word is as follows:



where

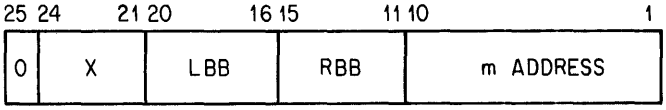
L ADDRESS represents the address of the next operand. (This address is indexable.)

Bits 18 through 20 must be all 0's.

X represents the index register which modifies the L ADDRESS field.

IA/FS represents the field-selection or indirect-addressing control bit.

The format for the field-selection control word is as follows:



where

m ADDRESS represents the least significant 10 bits (unmodified) of the next operand address.

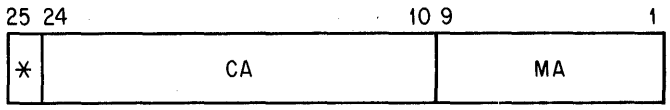
RBB (right-boundary bit) represents the bit position of the least-significant bit (LSB) of the field that is selected.

LBB (left-boundary bit) represents the bit position of the most-significant bit (MSB) of the field that is selected.

X represents the index register which modifies the m address.

Bit 25 must be a 0.

The format for the index-register-modification control word is as follows:



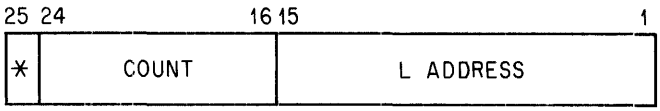
where

MA represents the amount by which the index register is modified (increased or decreased).

CA represents the number with which the modified index register is compared.

(*If bit position 25 contains a 0, the contents of the MA-field is added to the contents of the Ir; if bit position 25 contains a 1, the MA-field is subtracted.)

The format for the scatter-read/gather-write control word is as follows:



where

L ADDRESS represents the address of the first data word. (This address is not indexable.)

COUNT represents the number of words to be processed.

(*If bit 25 is a 1, it means that this is the last scatter-read/gather-writer control word, and bits 1 through 24 are ignored.)

1-32. DATA FLOW IN THE UNIVAC III CENTRAL PROCESSOR

The major data flow in the central processor is as follows:

Data flow during an instruction call (when an instruction is read from memory);

Data flow during multiprecision instructions;

Data flow during instructions which use the main adder;

Data flow during instructions which transfer data between the SAr and memory;

Data flow during shift instructions;

Data flow during miscellaneous instructions such as transfers, sense-FF instructions, expand instructions, and zero-suppress instructions; and

Data flow during the input-output operations.

1-33. Data Flow During an Instruction Call

The data flow during an instruction call (figure 1-2) is divided into three parts: the reading of the instruction from memory; the reading of the first operand from memory, called the first operand call; and the performance of field selection or indirect addressing. The first two parts are performed for all instructions, and the third part is performed only when specified by a 1 in the 25th bit position of the instruction word.

1-34. DATA FLOW WHEN READING THE INSTRUCTION FROM MEMORY. A memory request for an instruction is produced at the end of the previous instruction by the instruction-control section. The request is routed to the priority section where memory access is granted. When memory access is granted, the contents of the CC (in the storage-address section) are transferred to the adder in the memory-address section where they are increased by 1. The contents of the CC-plus-1 are then sent to the memory-address selector, and from there to memory. The location specified by the contents of CC-plus-1 contains the next instruction. When the instruction is read from memory, the X- and AR/XO-fields are read directly into the instruction-control section and the complete instruction word is read into the CCr.

1-35. DATA FLOW DURING THE FIRST OPERAND CALL. The contents of the X- and AR/XO-fields are decoded by the instruction-control section. The X-field selects an index register and the AR/XO-field selects an accumulator register. The contents of the selected Ir and the m address from the CCr then are added in the memory adder. At this time, the operation code is read from the CCr to the instruction-control section. As a result of this decoding process, the index-register and accumulator-register selections are made before the operation code is decoded.

The m' address is sent to the MAS and, from there, to memory and to MAr. Also, the contents of the specified Ar are sent to the write register at this time. If the instruction is a store instruction, the contents of the write register are written into memory. Otherwise, the contents of the memory location specified by the m' address are read into the CCr. Many instructions do not require an operand. In these cases, the CCr is cleared during the next minor cycle.

The 25th-bit position of the instruction is transferred into the field-select section. If the 25th-bit position contains a 1 bit, the first operand is either an indirect-addressing or field-selection control word.

1-36. DATA FLOW DURING FIELD SELECTION AND INDIRECT ADDRESSING. If an indirect-addressing or field-selection control word is specified by a 1 bit in the 25th-bit position of the instruction word, the contents of the X-field of the control

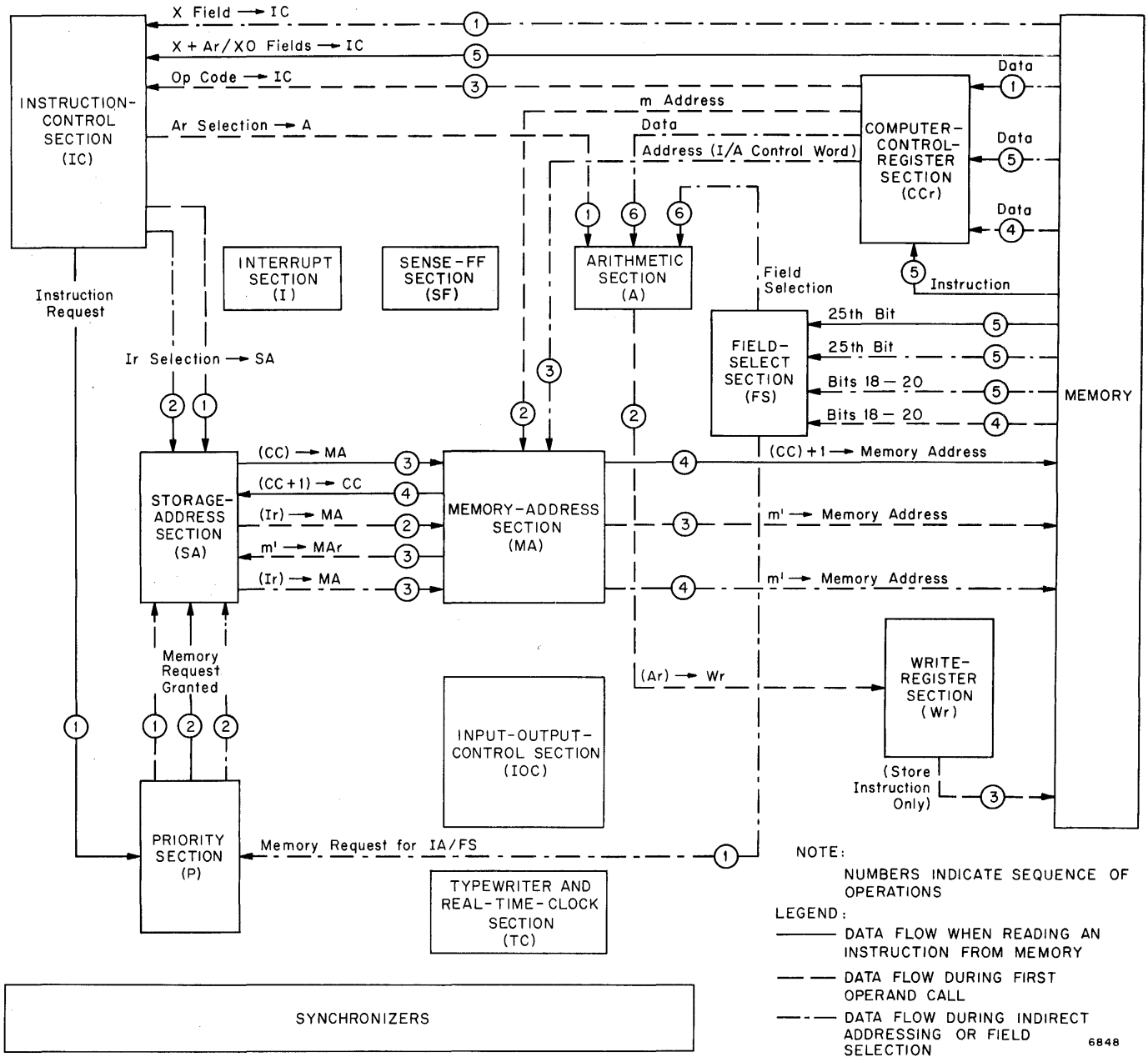


Figure 1-2. Data Flow During an Instruction Call, Functional Block Diagram

word (which is the first operand) is read into the instruction-control section at the same time that the control word is read into the CCr. The contents of the Ir selected by the X-field then modifies the address field of the control word in the same way as in the first operand call. The operand specified by the new m' address then is read into the CCr.

When the first operand is read into the CCr, bits 18, 19, and 20 are also read into the field-select section. If these bit positions do not contain all 0's, the contents of the operand specified by the new m' address (second operand) are transferred to the arithmetic section and only the fields selected by the field-selection control word are transferred to the specified Ar. If bit positions 18, 19, and 20 contain all 0's and bit position 25 does not contain a 1 bit, all bits in the second operand are processed. If bit-position 25 contains a 1, however, the operand is treated as another indirect-addressing or field-selection control word.

NOTE

In all data-flow descriptions that follow, the description begins at the conclusion of the first operand call.

1-37. Data Flow During Multiprecision Instructions

During multiprecision instructions, an operand request is produced by the instruction-control section (figure 1-3) for each additional operand. When this request is granted, the contents of MAr are transferred to the memory-address section where they are decreased by 1. This modified address is then transferred to memory and also back to the MAr. The contents of the memory location, represented by the modified address, are read from memory into CCr during a read operation, or the contents of the write register are written into the memory location during a write operation.

1-38. Data Flow During Instructions Which Use the Main Adder

The data flow is similar for all instructions which use the main adder. These instructions include the add, subtract, compare, logical operations, and extract instructions (figure 1-4). The operand in the CCr is transferred to the arithmetic section and is combined with the contents of the specified accumulator registers. The specific differences among

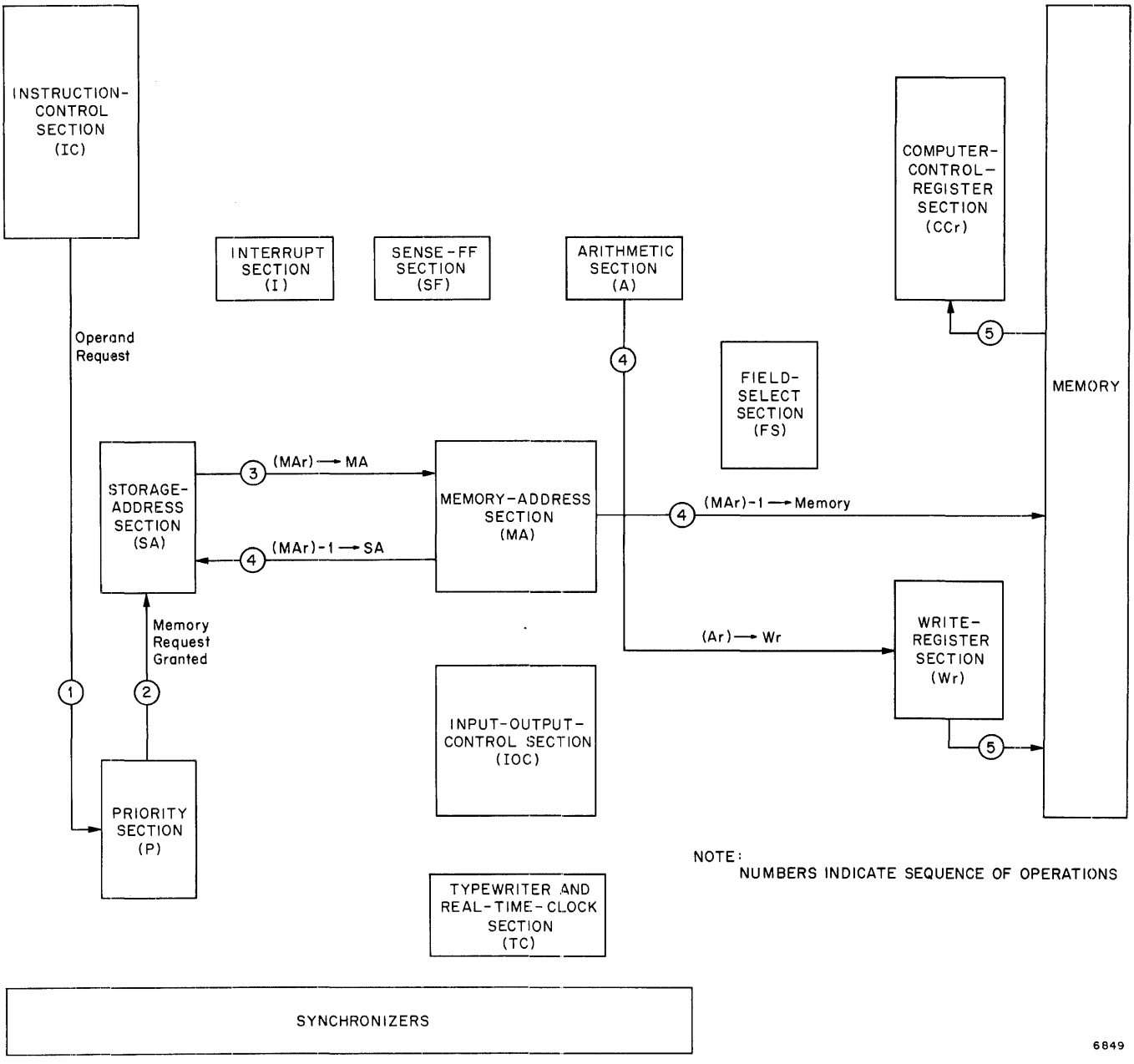


Figure 1-3. Data Flow During Multiprecision Instructions, Functional Block Diagram

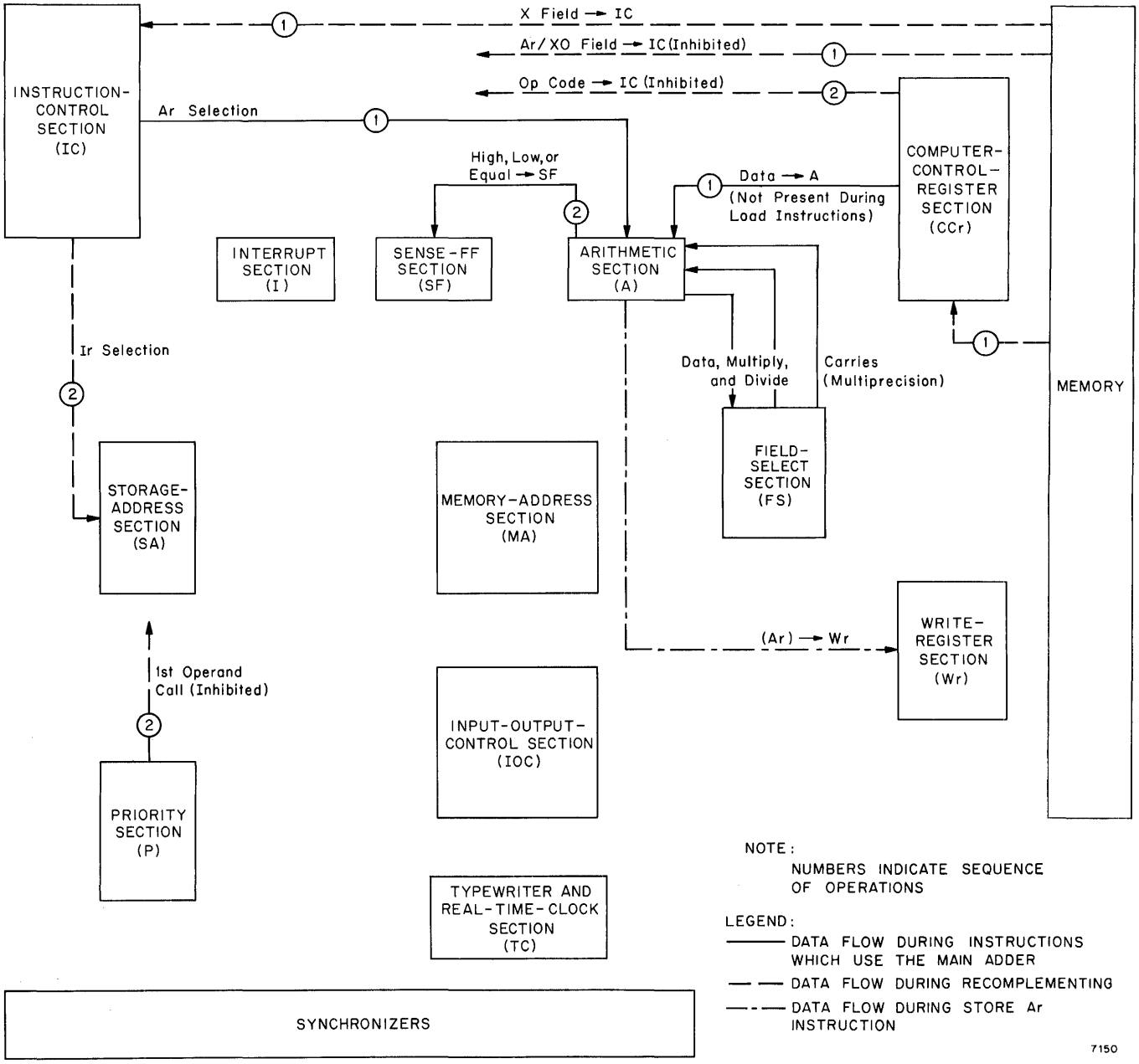


Figure 1-4. Data Flow During Instructions Which Use the Main Adder, Functional Block Diagram

instructions are produced through different methods of gating the operands into the main adder. During these instructions, the memory request for the next instruction is made while the last operand is being read into the arithmetic section.

When multiprecision arithmetic operations are specified, the word containing the least significant bits is transferred to the main adder first, and the word containing the most significant bits is transferred last. The carry from the most-significant-bit position of one word is stored in the field-select section during a multiprecision operation. The carry is then supplied to the least-significant-bit position of the next word (figure 1-5).

1-39. RECOMPLEMENTING. The result of the add or subtract instructions may produce a negative answer. If the answer is negative, it is expressed in complement form and must be recomplemented. Recomplementing is performed after the addition of the last operand, and after the next instruction is read from memory into the CCr. During the time that the recomplementing is performed, the next instruction is held in the CCr until the recomplementing is completed. The contents of the X-field of the next instruction, however, are transferred to the instruction-control section, the index register is selected, and the memory request for the operand is made. When recomplementing is completed, the memory request is granted. The contents of the

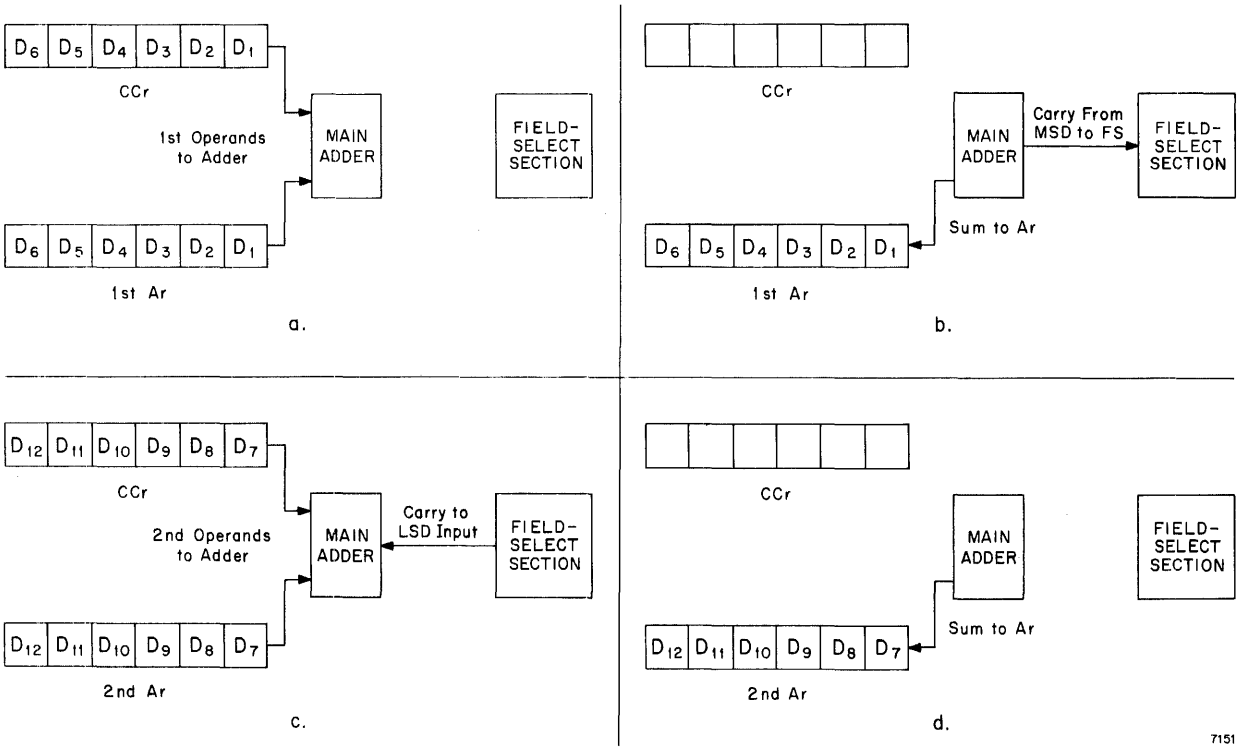


Figure 1-5. Example of Double-Precision Add with Carries, Functional Block Diagram

operation-code field and the AR/XO-field then are transferred to the instruction-control section, and the first operand is read from memory.

1-40. HIGH, LOW, AND EQUAL FLIP-FLOPS. During the add, subtract, and compare instructions, there is an output from the arithmetic section which sets flip-flops in the sense-FF section. The add and subtract instructions set the equal FF if the result equals zero, and reset it if the result does not equal zero. The compare instructions set the appropriate flip-flop (high, low, or equal).

1-41. EXTRACT INSTRUCTIONS. The contents of the fields to be extracted during an extract instruction are determined by the field-select FF's in the same way that the fields are determined during field selection. Therefore, field selection must be designated for an extract instruction; otherwise, the extract instruction will function in the same way as a load-Ar instruction.

1-42. MULTIPLY AND DIVIDE INSTRUCTIONS. The multiply and divide instructions are done by successive additions and subtractions. The 48-bit dividend occupies Ar1 and Ar2 and the divisor is contained in the CCr. The quotient is placed in Ar2 and the remainder is placed in Ar1. The multiplier occupies Ar1 and the multiplicand is contained in the CCr; the 48-bit product is placed in Ar2 and Ar3. Shifting in both the multiply and divide operations is done by routing the data through the field-select section.

1-43. STORE- AND LOAD-ACCUMULATOR INSTRUCTIONS. During the execution of the accumulator store instruction, data are transferred from the specified Ar's directly to the write register, and from there into memory. During the execution of the multiword load instructions, the contents of adjacent memory locations may be transferred to any designated set of Ar's. During the execution of the multiword store instructions, the contents of any set of Ar's can be transferred to adjacent memory locations. The only restriction is that the contents of the lowest numbered Ar occupy the last memory location during a store instruction and that the lowest numbered Ar receives the input from the last memory location during a load instruction.

Figure 1-6 shows the storing of three words from adjacent memory locations into three non-adjacent Ar's, and figure 1-7 shows the storing of the contents of two Ar's in adjacent memory locations.

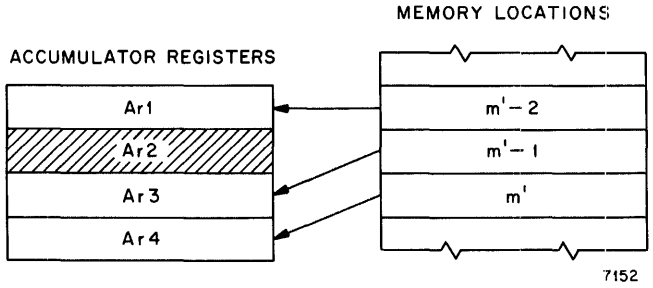


Figure 1-6. Loading Accumulator Registers from Adjacent Memory Locations, Diagram

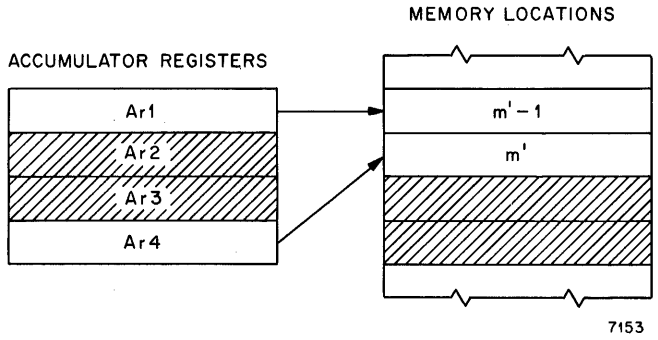


Figure 1-7. Storing Contents of Accumulator into Adjacent Memory Locations, Diagram

1-44. Data Flow During Instructions Which Transfer Data Between the Storage-Address Registers and Memory

The instructions which transfer data between the storage-address registers and memory are these:

Instructions which store the contents of a SAr (figure 1-8);

Instructions which load a SAr from memory (figure 1-8); and

Instructions which either modify, or modify and compare the contents of an Ir (figure 1-9).

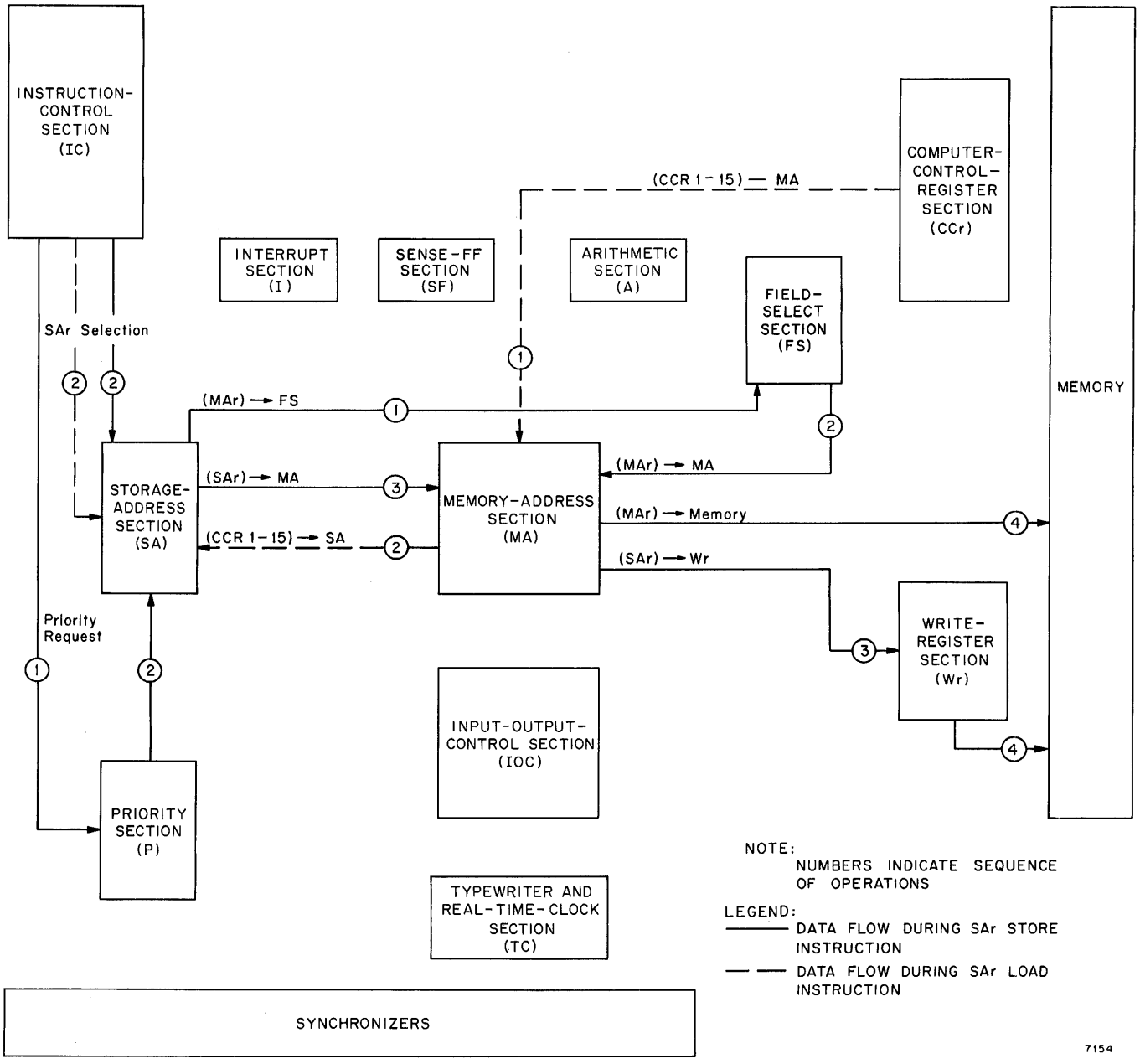


Figure 1-8. Data Flow During Storage-Address-Register Load and Store Instructions, Functional Block Diagram

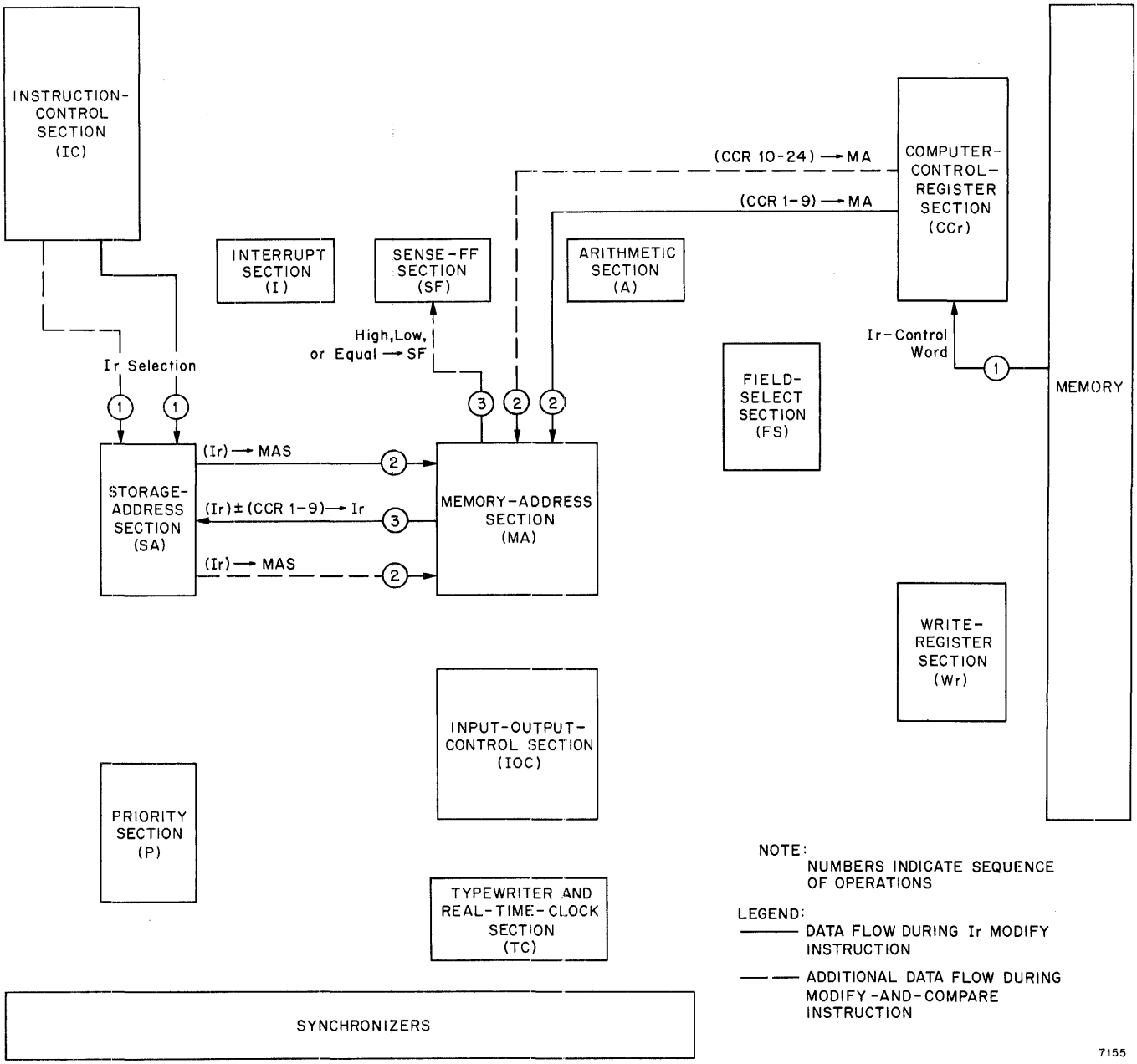


Figure 1-9. Data Flow During Index-Register Modify and Compare Instructions, Functional Block Diagram

1-45. DATA FLOW DURING INSTRUCTIONS WHICH STORE THE CONTENTS OF A STORAGE-ADDRESS REGISTER. When the first operand is read from memory during a store-SAr instruction, another memory request is made. When memory access is granted, the contents of the SAr specified by the AR/XO-field of the instruction are transferred through the memory adder to the write register. The m' address is sent from MAr through the field-select section to the MAS. The contents of the MWr, which contain the contents of the SAr, then are transferred to the memory location specified by the MAr.

1-46. DATA FLOW DURING INSTRUCTIONS WHICH LOAD A STORAGE-ADDRESS REGISTER. During the execution of the load-SAr instructions, the first 15 bits of the operand in the CCr are transferred through the memory-address section to the SAr specified by the instruction-control section. The request for the next instruction is made while data are being transferred to the register.

1-47. DATA FLOW DURING INSTRUCTIONS WHICH MODIFY AND COMPARE THE CONTENTS OF AN INDEX REGISTER. For instructions which either modify, or modify and compare the contents of an Ir, the first operand is the index-register-modification control word. The contents of bit positions 1 through 9 of this word are transferred to the memory adder where they are added to or subtracted from the con-

tents of the Ir specified by the AR/XO-field of the instruction. The modified contents of the Ir are then returned to the same Ir.

If the instruction calls for a comparison, the clearing of the CCr is inhibited and the contents of the CA-field of the control word are then subtracted from the modified contents of the Ir. The result of the subtraction sets the appropriate high, low, or equal indicator in the sense-FF section.

1-48. Data Flow During Shift Instructions

During the execution of the shift instruction, the shift count, which is contained in the four least significant bits of the contents of the MAr, is transferred to the field-select section after the m' address has been placed in the MAr. The contents of the Ar's specified by the AR/XO-field are then transferred to the CCr. The contents of the CCr are shifted according to the type of shift instruction and then returned to the Ar's. The shifting is terminated by an end-of-shift signal from the field-select section.

The CCr performs only right decimal shifts (four bits at a time) and right binary shifts. Thus, all shifting must be performed by combinations of the two types of right shifts. Alphanumeric shifts are performed by combining decimal shifts and binary shifts. Left shifts are performed by translating the left shift counts into right shift counts.

For example, the 2-digit left shift of the six digits represented by A through F in figure 1-10 could be done by either of two methods. The first method would consist of transferring all the digits from one register to another register and then performing a 2-digit left shift (a and b of figure 1-10). The second method would consist of shifting one digit at a time from the four least-significant-bit positions of one register into the four most-significant-bit positions of another register, and repeating this shifting four times (c of figure 1-10). The latter method performs the left shift by using the right-shift operation and is the method used in the UNIVAC III system.

Figure 1-11 shows the data flow for shifting right two digit positions, and figure 1-12 shows the data flow for shifting left two digits during double-precision shift instructions. The data flow for single-precision shift instructions is shown in e, f, and g of figure 1-11 and in e and f of figure 1-12. The letters A through L represent 4-bit decimal digits.

1-49. Data Flow During Miscellaneous Instructions

The miscellaneous instructions are these:

The sense-FF instructions (figure 1-13);

The test-interrupt instructions (figure 1-13);

The expand instruction (figure 1-14);

The compress instruction (figure 1-14);

The transfer instructions;

The zero-suppress instruction; and

The typewriter instructions (figure 1-13).

1-50. SENSE - FLIP - FLOP INSTRUCTIONS. The sense-FF instructions include all instructions which set, reset or test the flip-flops in the sense-FF section. A specific flip-flop is selected by the decoded output of the AR/XO-field. If the instruction requires testing the flip-flop output, and if the flip-flop is set, the m' address is placed in the CC rather than in the MAr. This transfer which is controlled from the priority section, causes the program to jump to the instruction in the location specified by the address of the test instruction.

1-51. INTERRUPT INSTRUCTIONS. During the execution of an instruction which tests or resets the interrupt FF's, the specific flip-flop is addressed through a combination of the AR/XO-field and the m-ADDRESS field. The AR/XO-field specifies a class-1 or -2 interrupt, or a priority channel in a class-3 interrupt; the m-ADDRESS field specifies the specific flip-flop. If the instruction calls for testing the flip-flop, and if the flip-flop is set, the next sequential instruction is performed; if the flip-flop is reset, 2 is added to the contents of the CC when the next instruction is read from memory. Thus, one instruction is skipped if the flip-flop is reset.

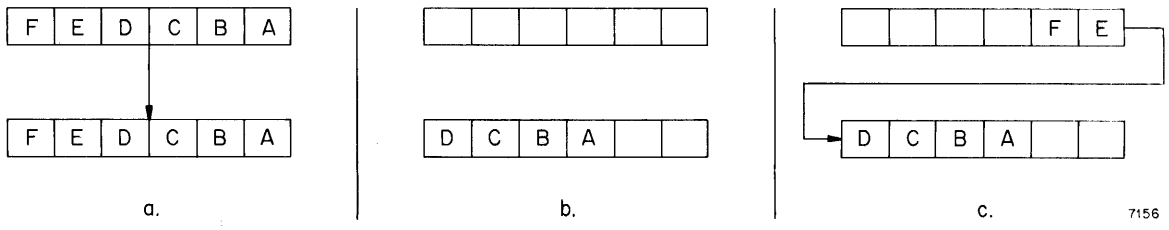


Figure 1-10. Shifting Right with Right Shifts or with Left Shifts, Diagram

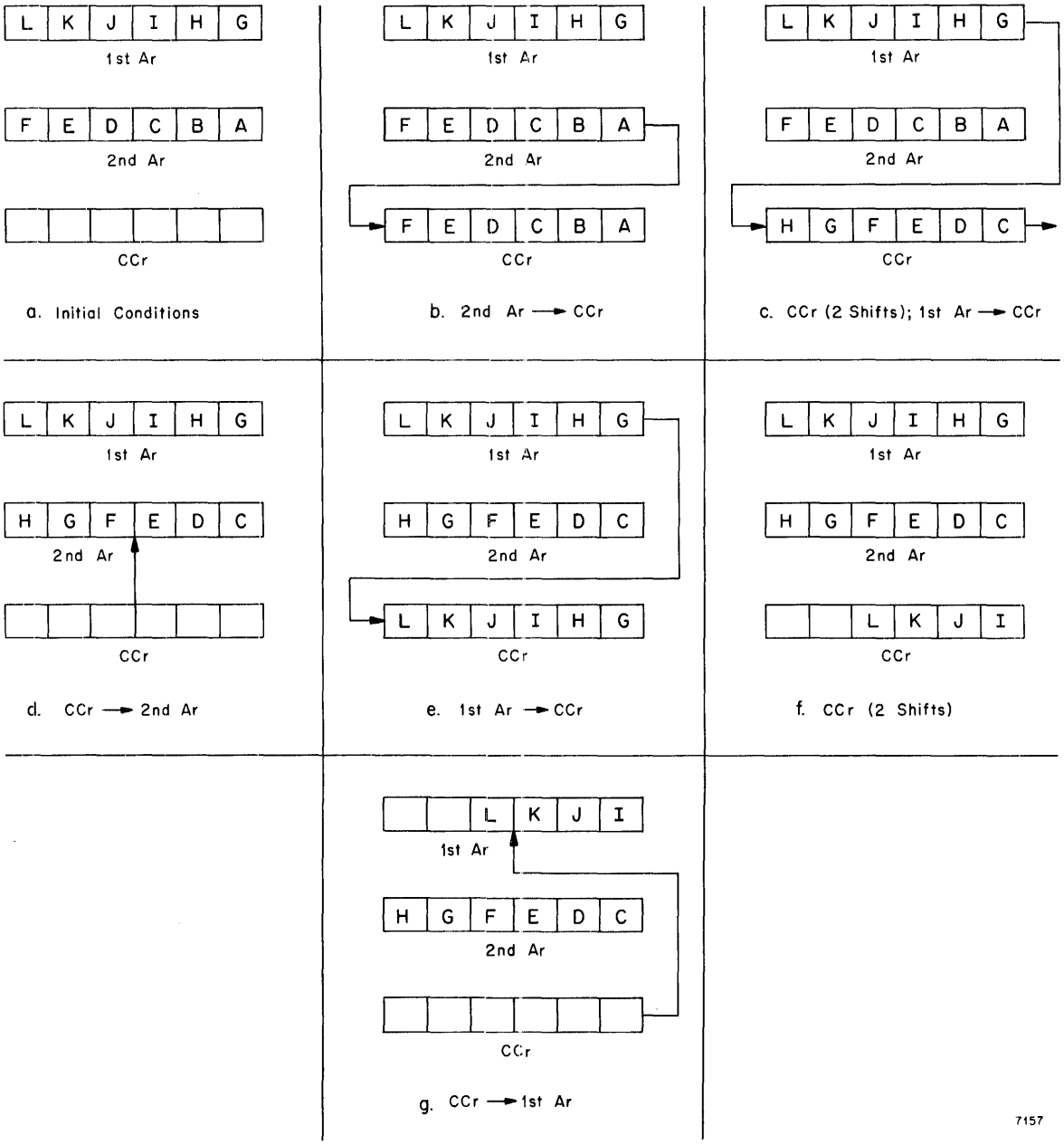


Figure 1-11. Data Flow During Double-Precision Two-Digit Right Shift, Diagram

1-52. DATA FLOW DURING THE EXPAND INSTRUCTION. The expand instruction converts 12 4-bit decimal digits (2 words contained in the Ar's) into 12 6-bit alphanumeric characters which are stored in 3 words in memory. When memory access is granted, the first word to be converted is transferred from the specified Ar to the CCr, and a memory request is made in order to write a word into memory. When memory access is granted, the first four digits of the first word are transferred to the write register where the two zone bits are added. The word then is written into the memory location specified by the address in MAr. For the storing of subsequent words, a new memory request is made and the process is repeated. The contents of MAr are decreased by 1 after the first word is read from memory.

Three memory request are made during the expand instruction; the first to store the first four digits of the first decimal word; the second to store the last two digits of the first decimal word and the first two digits of the second decimal word; and the third to store the last four digits of the second decimal word.

1-53. DATA FLOW DURING A COMPRESS INSTRUCTION. The compress instruction reads

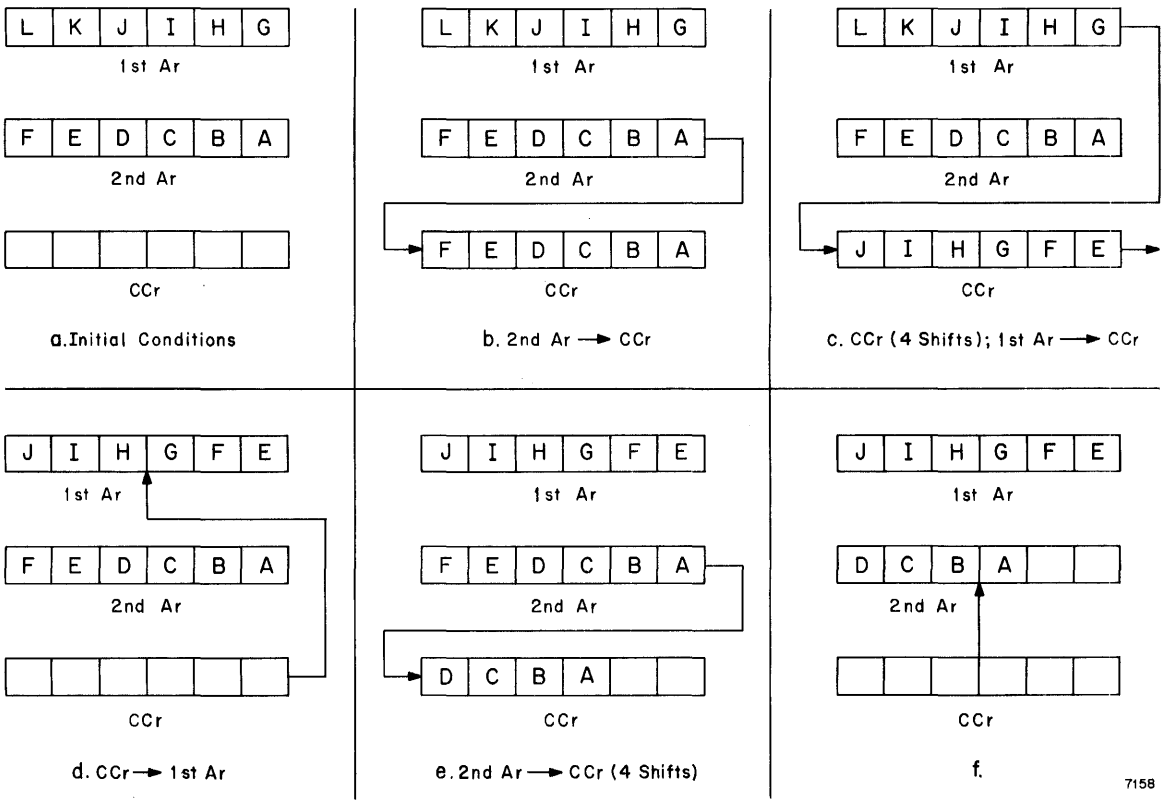


Figure 1-12. Data Flow During Double-Precision Two-Digit Left Shift, Diagram

three alphanumeric words from memory, converts these into two decimal data words, and transfers the two words to two Ar's. The word to be compressed is transferred from memory to the input-output control section, and only the numeric bits are transferred from the input-output control section to the CCr. The data are then transferred from the CCr to the Ar's specified by the AR/XO-field of the instruction.

Three memory accesses are required for the compress instruction. During the execution of the first memory access, the first alphanumeric word is read from memory and the four digits are read into the first four digit positions of the first Ar. During the execution of the second memory access the second alphanumeric word is read from memory and the four decimal digits are transferred to the fifth- and sixth-digit position of the first Ar and to the first- and second-digit position of the second Ar. During the execution of the third memory access, the third alphanumeric word is read from memory and the four decimal digits are transferred to the last four-digit positions in the second Ar.

1-54. DATA FLOW DURING TRANSFER INSTRUCTIONS. The transfer instructions cause the program to execute the instruction contained in the m' address location rather than the next sequential instruction in the program. This is done by decoding the first operand as an instruction and storing the m' address in the CC rather than the MAr.

1-55. DATA FLOW DURING ZERO-SUPPRESS INSTRUCTION. During the execution of the zero-suppress instruction, the output of the CCr is decoded to produce signals which represent decimal 0's or commas. These signals then are transferred to the field-select section which uses the same process to eliminate the zeros as is used in selecting the fields during field selection. This process is continued until all leading zeros (0's to the left of the most significant digit) are removed. The unique feature of the zero-suppress instruction is that the m' address is increased, rather than decreased, with additional operand calls. This is necessary because the most-significant-digit positions must be sampled first to eliminate the leading zeros.

1-56. DATA FLOW DURING TYPEWRITER INSTRUCTIONS. During the execution of the instructions which transfer one alphanumeric character from memory to the typewriter, the word containing the character is transferred to the input-output control section. One character, which is designated by the AR/XO-field, is transferred from the input-output-control section to the typewriter- and real-time-clock section. From there it is transferred in parallel to the typewriter.

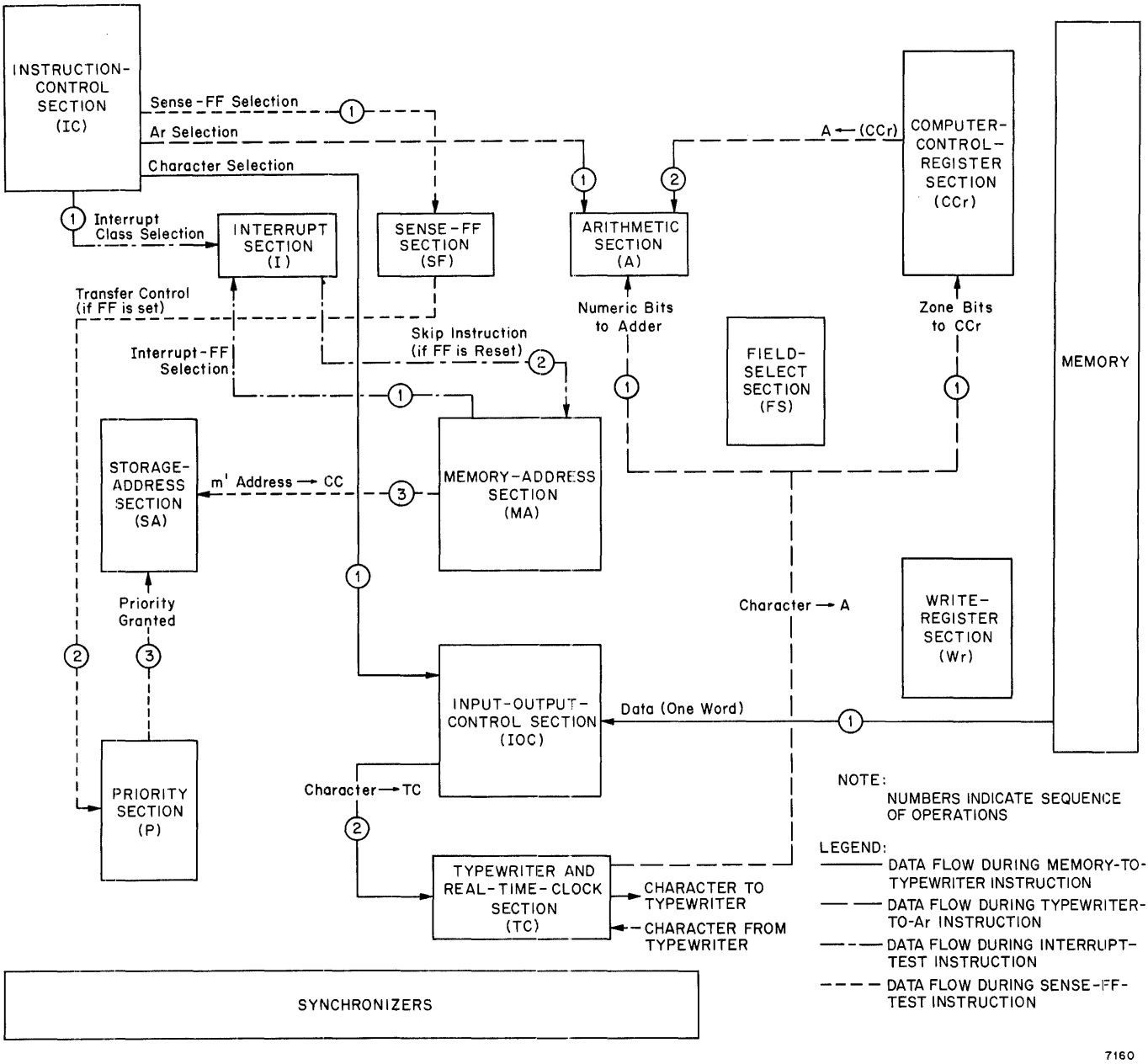


Figure 1-13. Data Flow During Typewriter Interrupt, and Sense-Flip-Flop Instructions, Functional Block Diagram

During the execution of the instructions which transfer one alphanumeric character from the typewriter to the Ar, the character is first transferred from the typewriter to the typewriter-buffer register. The four numeric bits are then transferred to the main adder and the two zone bits are transferred to bit-positions 5 and 6 of the CCr. The complete word in the CCr is transferred through the adder to the Ar specified by the AR/XO-field.

1-57. Data Flow During Input-Output Operations

During the time interval of the input-output operations, the central processor controls memory addressing and the transferring of all data, specification, and control words between memory and the synchronizers. Each input-output synchronizer is assigned one MAC and one memory location. The MAC supplies the memory address of either a data word, or a control word (during tape operations only). The specification word is produced by the initiate-I-O instruction and is stored in a specified memory location, called the I-O standby location, between the time that the input-output instruction is initiated and the time that the synchronizer is ready to perform the operation.

The input-output operations are divided into three parts for all I-O devices except the tape units for which they are divided into four parts (figure 1-15).

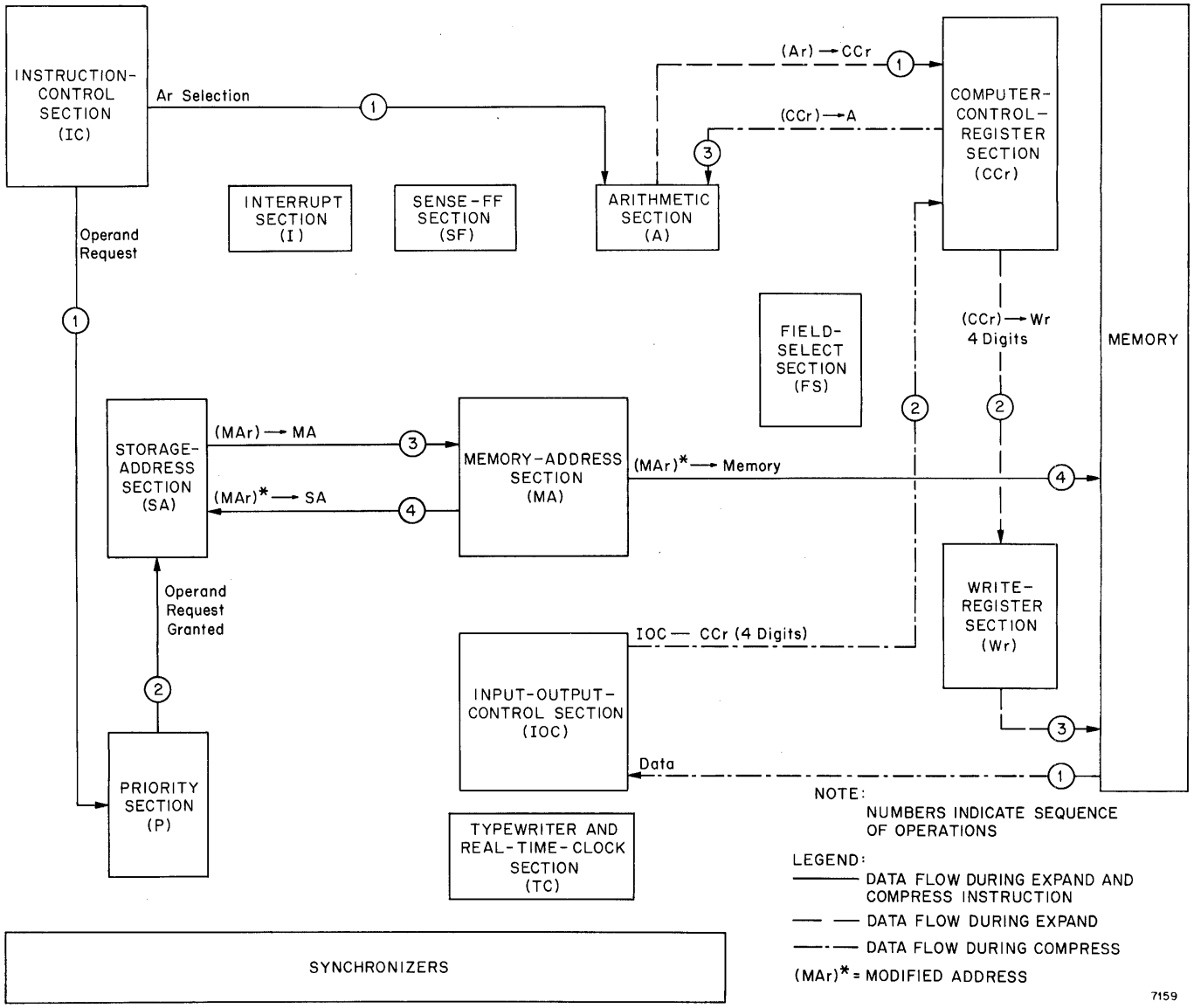
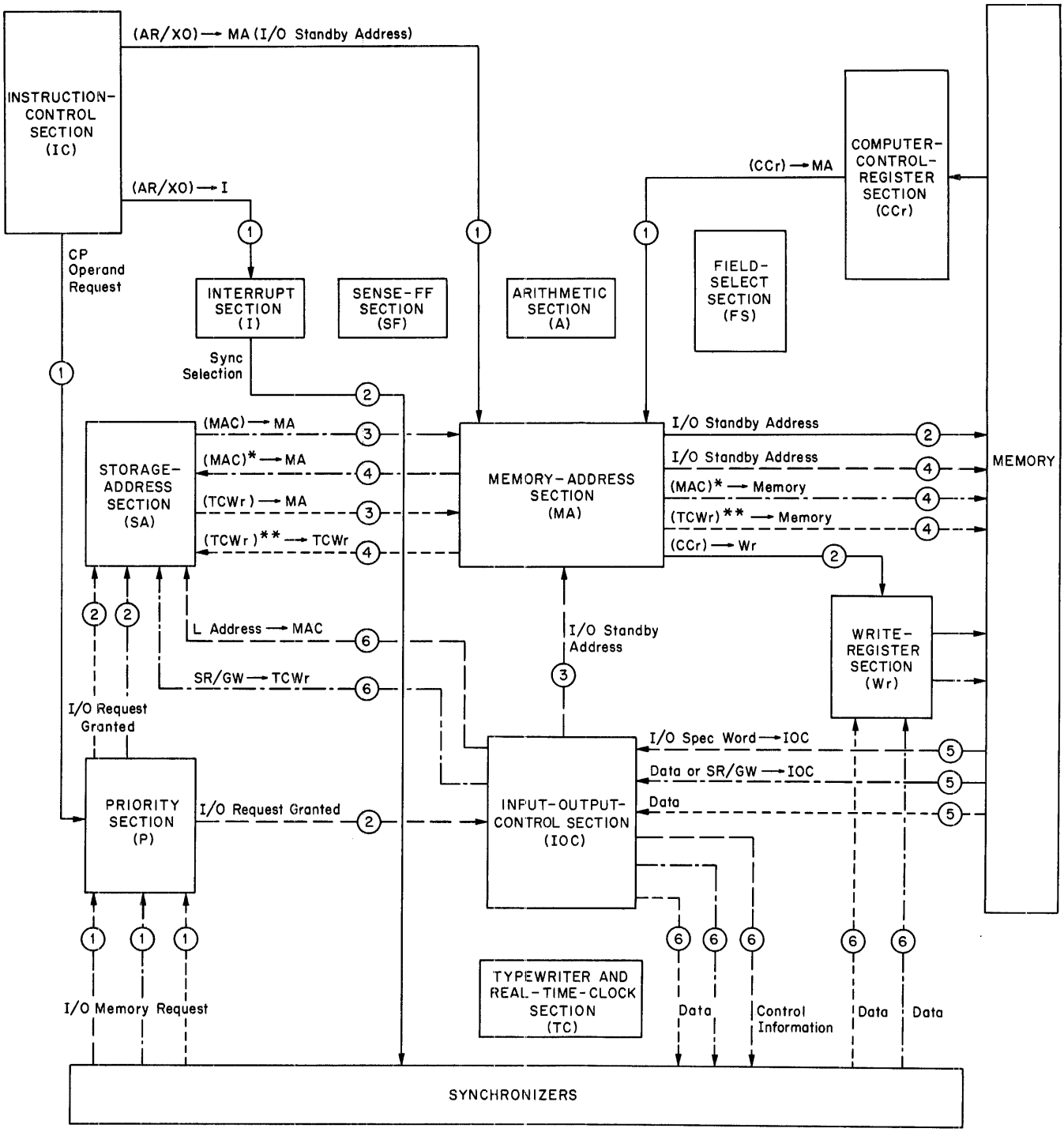


Figure 1-14. Data Flow During Expand and Compress Instructions, Functional Block Diagram

Operation	Description
Initiate-I-O instruction.	Transfers input-output specification word to I-O-standby location.
I-O-instruction call.	Transfers input-output specification word from memory to input-output control section, decodes the word, and loads a MAC.
I-O data call.	Transfers data words between synchronizer and memory.
First data call for tape.	Transfers scatter-read/gather-write control words from memory to tape-control-word register during tape operations.

1-58. DATA FLOW WHEN TRANSFERRING THE SPECIFICATION WORD TO MEMORY. The first operand that is read into the CCr, as a result of the initiate-I-O instruction automatic operand call, is the input-output specification word. The contents of the CCr are transferred through the memory adder to the write register. The I-O standby address, which is contained in the AR/XO-field of the instruction, is transferred directly to the MAS. The specification word is then read into memory.

The contents of the AR/XO-field of the instruction are also decoded to select a desired synchronizer. The synchronizer is selected by setting a flip-flop in the interrupt section, which informs the synchronizer that a specification word is in the standby location.



NOTES:

NUMBERS INDICATE SEQUENCE OF OPERATIONS

(MAC)* = MODIFIED ADDRESS (EXCEPT ON FIRST DATA CALL)

(TCWr)** = (COUNT) −1; (ADDRESS) +1

LEGEND:

——— DATA FLOW WHEN WRITING THE SPECIFICATION WORD INTO THE STANDBY LOCATION

— — DATA FLOW WHEN READING THE SPECIFICATION WORD FROM THE STANDBY LOCATION

—·— DATA FLOW FOR DATA WORDS FOR ALL DEVICES EXCEPT TAPE, AND FOR THE SCATTER-READ/GATHER-WRITE CONTROL WORD FOR TAPE

– – – – – DATA FLOW FOR DATA WORDS FOR TAPE

7161

Figure 1-15. Data Flow During Input-Output Operations, Functional Block Diagram

1-59. DATA FLOW WHEN TRANSFERRING THE SPECIFICATION WORD FROM THE STANDBY LOCATION. When a synchronizer is able to accept the specification word, it makes a memory request. When the request is granted, a signal is sent from the priority section to the input-output control section where it is encoded to produce the address of the standby location. The specification word then is read from the standby location to the input-output control section. The input-output control section then transfers the L address to the MAC that is assigned to the input-output device. The other fields of the input-output specification word are transferred to the synchronizer.

1-60. DATA FLOW FOR DATA WORD FOR ALL DEVICES EXCEPT TAPE UNITS. After the L address is stored in the MAC, the synchronizer requests memory access as soon as the synchronizer is able to accept a data word. When the request is granted, the contents of the MAC (unmodified) are sent through the memory-address section. During the first memory access, the address is not modified. During all other memory accesses, the address is increased or decreased, depending on the operation. The address then is sent to memory. If the operation is a read operation, the data word is transferred from the synchronizer through the write register into the specified memory location. If the operation is a write operation, the word is transferred from memory to the input-output control section and from there to the synchronizer. This process is repeated until all words have been transferred.

1-61. DATA FLOW FOR TRANSFERRING THE SCATTER - READ / GATHER - WRITE CONTROL WORD. After the L address is stored in the specified MAC during a tape input-output operation, which uses control words, the UNISERVO* III synchronizer requests memory access for the scatter-read/gather-write control word. When the request is granted, the address in the MAC is sent through the memory-address section and the control word at that address is read into the input-output control section. The word then is sent from the input-output control section to a TCWr in the storage-address section.

1-62. DATA FLOW FOR WORDS DURING TAPE OPERATIONS. When the UNISERVO III synchronizer is able to process a data word, it requests memory access. When the request is granted, the address portion of the control word in the TCWr is increased by 1 or it is decreased by 1 during backward read, and sent to memory. The COUNT field of the control word is decreased by 1, and the modified control word is returned to the TCWr. if the operation is a tape-read operation, the word is transferred from the synchronizer through the write register to the specified memory location. If the operation is a tape-write operation, the word is read from memory through the input-output control section to the synchronizer.

If the scatter-read/gather-write control word does not contain a 1 in bit position 25, another control word is read from memory when the count in the present control word is decreased to 0. If the control word contains a 1 in bit position 25, the control word is not processed and the operation stops.

1-63. UNIVAC III INSTRUCTIONS

The programmer has available a total of 61 central-processor instructions in SALT code. Forty-five of these instructions are designated only by the contents of the operation field and the remaining 16 are designated by a combination of the operation field and the AR/XO-field. The 16 instructions that are desisgnated by the operation code and the AR/XO-field are the instructions which set, reset, and test the flip-flops in the sense FF section, and the instructions which test and reset the interrupt FF's. The instructions which address the flip-flops in the sense-FF section have operation codes 60, 61, and 62; and the instructions which address the interrupt FF's have operation codes 64 and 65.

Table 1-1 lists all instructions that are designated by operation codes only, and table 1-2 lists the instructions that are designated by a combination of operation codes and the contents of the AR/XO-field.

Table 1-1. UNIVAC III Instructions

Operation Code in Octal	Instruction	SALT Code	Description
00	No operation	NOP	Perform next instruction.
01	Read typewriter	RT	Add the contents of the typewriter-buffer register to bit positions 1 through 6 of the Ar specified by the AR/XO-field.

*Trademark of the Sperry Rand Corporation.

Table 1-1. UNIVAC III Instructions (cont)

Operation Code in Octal	Instruction	SALT Code	Description
02	Write typewriter	WT	Transfer the character specified by the AR/XO-field of the instruction from the m' location to the typewriter-buffer register.
03	Display	DIS	Transfer the contents of the m' location to the engineer's console display.
04	Store memory address	STMC	Store the contents of the MAC in the first 15-bit positions of the m' location.
05	Store tape-control-word register	STCR	Store the contents of the specified TCWr in the m' location.
06	Transfer unconditionally	TUN	Transfer the program to the m' address.
07	Transfer, return	TR	Store the address of the next instruction in the m' location and transfer to the instruction in the m'+1 location.
10	Store	ST	Transfer the contents of the specified Ar's to the m' location.
11	Store with changed sign(s)	STCS	Same as 10 except that signs are changed.
12	Load	L	Transfer the contents of the m' location to specified Ar's.
13	Load with changed sign	LCS	Same as 12 except that the sign is changed.
14	Extract	EXT	Replace bits in specified Ar's by bits in designated field of the operand in the m' location.
15	Superimpose	SUP	Place a 1 bit in each bit position of the specified Ar's if a 1 bit appears in the corresponding bit position of either the specified Ar or the m' location.
16	Erase	ERS	Place a 0 bit in each bit position of the specified Ar if a 0 bit appears in the corresponding bit position of either the specified Ar's or the m' location.
20	Add decimal	A	Add algebraically the contents of a specified Ar to the contents of the m' location and return the sum to the register from which the operand was taken.
21	Subtract decimal	S	Subtract algebraically the contents of the m' location from the contents of a specified Ar and return the difference to the same register.
22	Decimal add, higher	AH	Add algebraically the contents of the m' location to the contents of specified Ar's and return the sum to higher numbered Ar's.
23	Decimal subtract, higher	SH	Subtract algebraically the contents of the m' location from the contents of specified Ar's and return to higher numbered Ar's.
24	Binary add	BA	Same as 20 except that operands are in binary.
25	Binary subtract	BS	Same as 21 except that operands are in binary.
26	Binary add, higher	BAH	Same as 22 except that operands are in binary.
27	Binary subtract, higher	BSH	Same as 23 except that operands are in binary.
30	Multiply	M	Multiply the six decimal digits in m' by the six decimal digits in the Ar1 location and return the product to Ar2 and Ar3.
31	Divide	D	Divide the contents of Ar1 and Ar2 (decimal) by the six decimal digits in the m' location and return the quotient to Ar2 and the remainder to Ar1.

Table 1-1. UNIVAC III Instructions (cont)

Operation Code in Octal	Instruction	SALT Code	Description
40	Shift right	SR	Shift the contents of the specified Ar's right the number of decimal-digit positions specified by the m' address.
41	Shift left	SL	Shift the contents of the specified Ar's left the number of decimal-digit positions specified by the m' address.
42	Shift alphanumeric right	SAR	Shift the contents of the specified Ar's right the number of alphanumeric-character positions specified by the m' address.
43	Shift alphanumeric left	SAL	Shift the contents of the specified Ar's left the number of alphanumeric-character positions specified by the m' address.
44	Shift binary circular	SBC	Shift the contents of the specified Ar right the number of bit positions specified by the m' address.
50	Store index register	STX	Transfer the contents of the specified Ir to the first 15-bit positions of the m' location, and fill the remainder of the memory location with binary 0's.
51	Load index register	LX	Load the specified Ir with the first 15 bits in the m' location.
52	Modify index register	IX	Add or subtract the contents of bit positions 1 through 9 of the m' location to the specified Ir and store the result in the specified Ir.
53	Modify index register and compare	ICX	Add or subtract the contents of bit positions 1 through 9 of the m' location to the specified Ir; compare the new contents of Ir with bit positions 10 through 24 of the m' location and set the appropriate high, low, or equal indicator.
54	Compare	C	Compare algebraically the contents of the specified Ar's with the contents of the memory location specified by the m' address and set the appropriate high, low, or equal indicator.
55	Compare absolute	CA	Compare without regard to sign the contents of the specified Ar's with the contents of the m' location and set the appropriate high, low, or equal indicator.
56	Compare 0 bits	CZRO	Compare the 1 bits in the specified Ar's with the 0 bits in the m' location. If every bit position in the memory location contains a 0 for every bit position in the Ar that contains a 1 bit, set the equal indicator; otherwise set the high indicator.
57	Compare 1 bits	CONE	Compare the 1 bits in the specified Ar's with the 1 bits in the m' location. If corresponding bit positions of the memory location and the Ar's contain 1 bits, set the equal indicator; otherwise set the high indicator.
66	Actuate keyboard	ACT	Enable one character to be transferred from the keyboard to the typewriter-buffer register when character key is depressed.
70	Initiate input-output	IOF	Perform the input-output operation specified by the specification word in the m' address.
71	Decimal to alphanumeric (expand)	DTA	Convert decimal data in two adjacent Ar's into three alphanumeric words and store the result in three memory locations beginning with the m' location.

Table 1-1. UNIVAC III Instructions (cont)

Operation Code in Octal	Instruction	SALT Code	Description
72	Alphanumeric to decimal (compress)	ATD	Convert three alphanumeric words into decimal data words and place the result in two designated Ar's.
73	Zero suppress	ZUP	Load the contents of the specified Ar's with the contents of the m' location and eliminate all alphanumeric 0's and commas.
76	Load time	LT	Transfer the output of the real-time clock to the Ar specified by the AR/XO-field of the instruction.
77	Wait	WAIT	Transfer the m' address to the CC and stop the central processor, and when the START KEY is depressed, transfer to the instruction in the CC.

Table 1-2. Sense-Flip-Flop Instruction Designations

Operation Code in Octal	Instructions	SALT Code	AR/XO Contents	Description
60	Test high	THI	0011	Transfer the program to the m' address if the high flip-flop is set.
60	Test low	TLO	0101	Transfer the program to the m' address if the low flip-flop is set.
60	Test equal	TEO	0110	Transfer the program to the m' address if the equal flip-flop is set.
60	Test positive	TPOS	0001 = Ar1 0010 = Ar2 0011 = Ar3 0100 = Ar4	Transfer the program to the m' address if the sign of the Ar specified by the AR/XO-field is positive.
60	Transfer if Input-output interrupt is prevented	TIOP	0000	Transfer the program to the m' address if the input-output interrupt flip-flop is set.
60	Test sense indicator	TSI	1000 = Si1 1001 = Si2 1010 = Si3 1011 = Si4 1100 = Si5 1101 = Si6 1110 = Si7 1111 = Si8	Transfer the program to the instruction in the m' address if the sense indicator (Si) specified by the AR/XO-field is set.
61	Allow input-output interrupt.	AIO	0000	Reset the prevent I-O interrupt flip-flop.
61	Reset sense indicator	RSI	1000 = Si1 1001 = Si2 1010 = Si3 1011 = Si4 1100 = Si5 1101 = Si6 1110 = Si7 1111 = Si8	Reset the sense indicator (Si) specified by the AR/XO -field.
62	Set sense indicators	SSI	1000 = Si1 1001 = Si2 1010 = Si3	Set the sense indicator (Si) specified by the AR/XO-field.

Table 1-2. Sense-Flip-Flop Instruction Designations (cont)

Operation Code in Octal	Instructions	SALT Code	AR/XO Contents	Description
			1011 = Si4 1100 = Si5 1101 = Si6 1110 = Si7 1111 = Si8	
62	Prevent input-output interrupt.	PIO	0000	Set the prevent input-output flip-flop.
64	Test central-processor error	TPE	0001	Skip one instruction if central-processor-error indicator specified by the m address is reset.
64	Test contingency indicator	TCI	0010	Skip one instruction if the contingency indicator specified by the m address is reset.
64	Test input-output indicator	TIO	0011—1111 priority channels 3—15	Skip one instruction if the input-output-error indicator specified by the m address is reset.
65	Reset central processor-error indicator	RPE	0001	Reset the central-processor-error indicator specified by the m address.
65	Reset contingency indicator	RCI	0010	Reset the contingency indicator specified by the m address.
65	Reset input-output-error indicator	RIO	0011—1111 priority channels 3—15	Reset the input-output-error indicator specified by the m address.

Section 3
INSTRUCTION-CONTROL SECTION

3-1. INTRODUCTION

This section describes the logic of the central-processor instruction control (figure 3-1). The logic areas of the central-processor instruction control and their functions are as follows:

Logic Area	Function
Instruction decoder	Translate operation code of the instruction into control signals.
Program counter	Provide signals that control the steps within the instruction.
Accumulator-register selector	Determines the precision of multiprecision instructions, and selects accumulator registers, tape-control-word registers, storage-address registers, interrupt FF's, typewriter characters, or synchronizers during various instructions.
Index-register selector	Selects either the index register that modifies the first operand address, or a storage-address register during instructions which load or store a storage address register.
Memory-request network	Produces memory requests for the next instruction or for an operand.

The instruction-control section also produces four signals which affect the decoding of an instruction. These signals are the instruction ending pulse (CVEP), the instruction inhibit signal (CVRINH), the recomplement signal (CVRECP), and the instruction-decode-inhibit signal (CCVA).

3-2. Instruction Decoder

The instruction decoder consists of an instruction register, an octal decoder, a group encoder, and an instruction-line generator.

The instruction register contains six flip-flops which receive the operation code from bit positions 15 through 20 of the computer-control register (CCr). This code is read into the instruction register at T4. The outputs from the instruction register, CVR(1—6) are the inputs to the octal decoder and, under special conditions, are control signals for the central processor.

The octal decoder, an AND-gate decoding network, translates the output of the instruction register into 64 separate signals, called CVRD signals. Fifty of these signals, which represent valid instructions, are transferred to the group encoder. The remaining 14 signals are transferred to an invalid-code-error FF.

The group encoder combines CVRD signals to produce outputs representing 34 single instructions and 14 instruction groups. The CVRD signals are combined according to specific instruction types; some instructions are represented by single-instruction signals only, others by group signals only, and others by single and group signals. The outputs from the group encoder are designated as CVRG signals. Thirty-eight of these signals are transferred to the intruction-line generator, and 10 are used directly as control signals.

The instruction-line generator combines CVRG signals with the outputs from the program counter to produce signals which control the steps within an instruction. The outputs of the instruction-line generator (CV lines) are control signals for all parts of the central processor.

3-3 Program Counter

The program counter produces 10 outputs that control the sequence of operations within an instruction. The 10 outputs, designated CVCP0 through CVCP9, are produced one at a time. The complement of the program-counter outputs ($\overline{\text{CVCP0}}$ through $\overline{\text{CVCP9}}$) are also used as program-counter signals. The program counter is reset to CVCP0 at T3 when a new instruction is read from memory and is advanced at each successive T3.

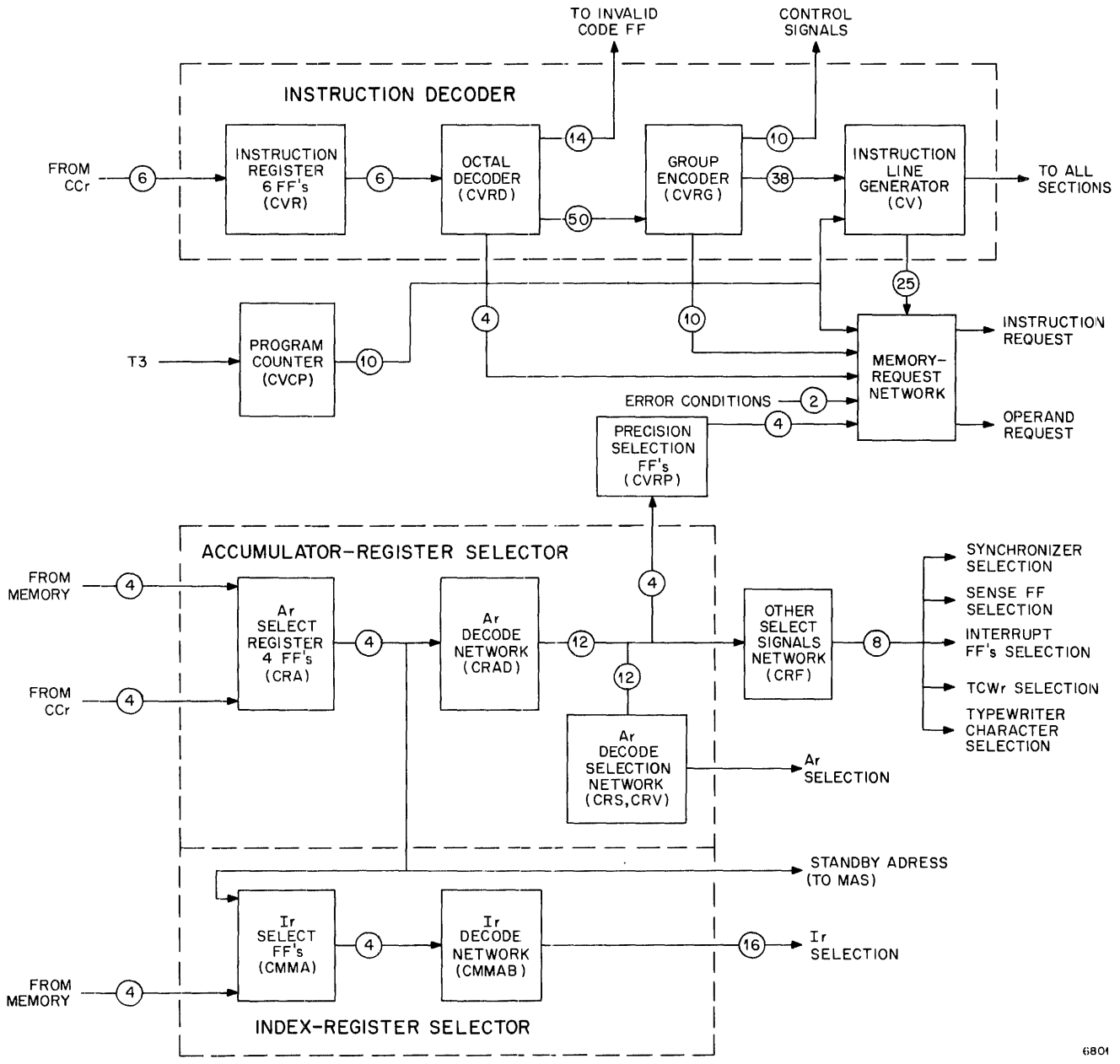


Figure 3-1. Instruction-Control Section, Functional Block Diagram

3-4. Accumulator-Register Selector

The accumulator-register selector contains four flip-flops, a decoding network, and three selection networks. The four flip-flops, called the AR-select FF's, receive at T3 the contents of bit positions 11 through 14 of an instruction word either from memory or from the CCr during a recomplement cycle. The output of the four flip-flops goes to the IR-select FF's, memory-address selector, or decoding network. The decoding networks decode the output of the four flip-flops to determine the precision of multiprecision operations and to select accumulator registers, sense FF's, tape control-word registers, storage-address register interrupt FF's, typewriter characters, or synchronizers.

3-5. Index-Register Selector

The index-register selector contains four flip-flops and a decoding network. The four flip-flops, called IR-select FF's, normally receive the contents of bit positions 21 through 24 of the instruction word. The IR-select FF's also receive information from the AR-select FF's during storage-address-register load or store instructions. The four outputs from the IR-select FF's are decoded to select 1 of the 15 index registers. This decoding is completed

one pulse time before the outputs are available from the instruction register (CVr). Thus, the index register is selected before the instruction to be used is determined.

If the IR-select FF's contain all 0's, no index register is selected and the m address is not modified during the automatic operand call.

3-6. Memory-Request Network

The memory-request network receives inputs from the instruction decoder, from the precision-selection network, from the program counter, and as a result of error conditions. These inputs are processed to produce both instruction and operand requests. The instruction requests are initiated at the end of each instruction or when certain error conditions are present. The operand requests are initiated by instructions which require more than one operand call (first operand call is automatic).

3-7. INSTRUCTION DECODER

The instruction-decoder decodes and combines signals to produce all the control signals necessary to perform an instruction. Figure 3-2 shows the instruction register and octal decoder.

3-8. Instruction Register

At T3 the instruction ending pulse signal (CVEP) resets all flip-flops in the CVr to 1, and at T4 FS

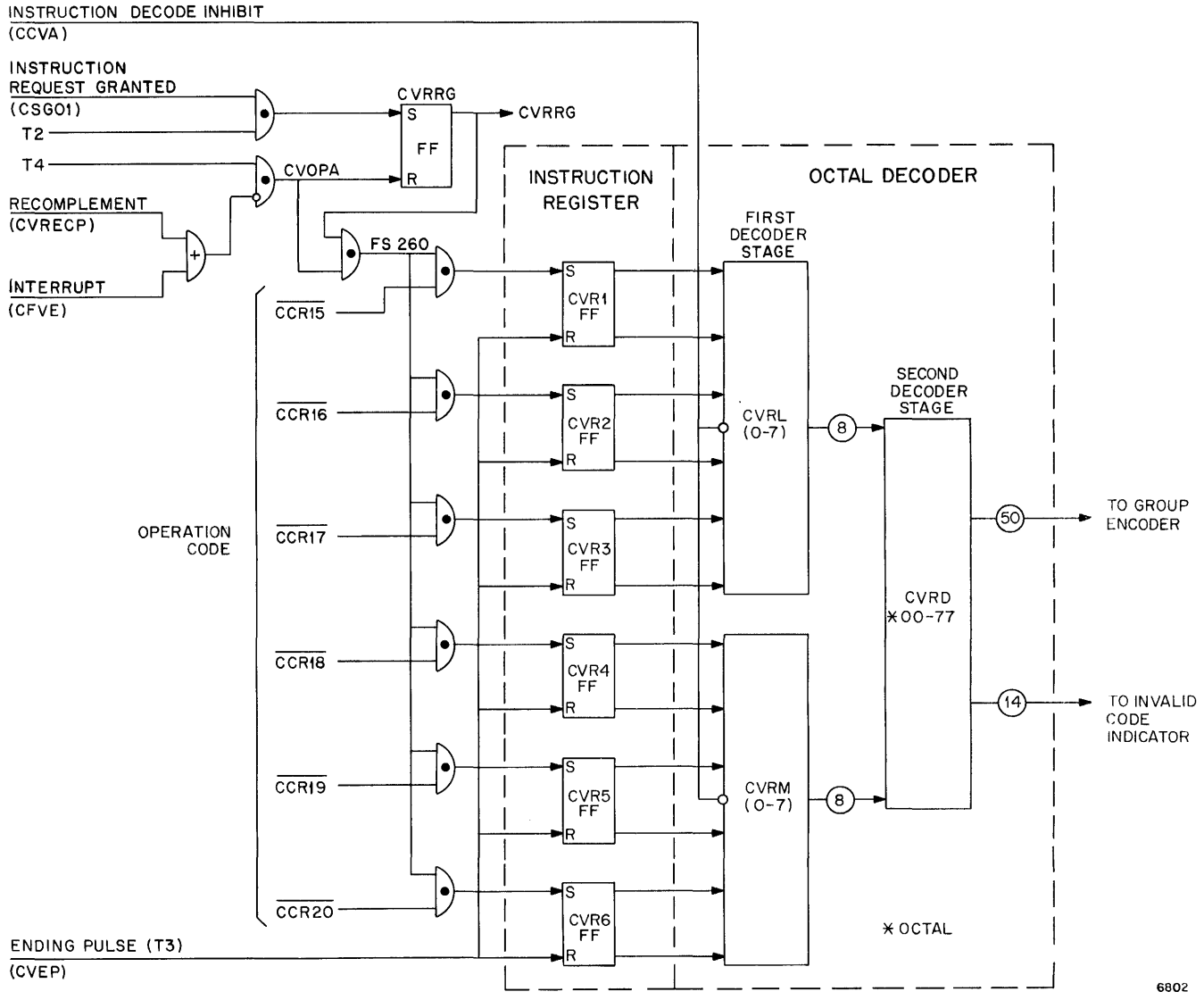


Figure 3-2. Instruction Register and Octal Decoder, Detailed Block Diagram

260A transfers the 0's from bit positions 15 through 20 in the CCr, $\overline{CCR(15-20)}$, to the CVr. Function signal 260 is normally present when an instruction is available in the CCr. The production of FS 260A is inhibited during recomplementing and when the program has been interrupted.

Function signal 260A is produced by the CVOPA signal and the set output of the CVRRG FF. CVOPA is present at T4 except when recomplementing is performed or when the computer program is interrupted. The CVRRG FF is set at T2 after the instruction memory request is granted (CSG01) and is reset by CVOPA. Thus, function signal 260A is produced at T4 after the instruction has been read from memory, after recomplementing, or after the program has been interrupted.

3-9. Octal Decoder

The contents of the CVr are decoded in two stages. The first stage of the octal decoder decodes the outputs of FF's CVR1 through CVR3 and the outputs of FF's CVR(4—6) into eight outputs for each group of three flip-flops. The outputs from CVR(1—3) are designated CVRL(0—7), representing the least-significant octal digit of the operation code, and the outputs from CVR(4—6) are designated CVRM(0—7), representing the most significant octal digit of the operation code. The second octal decoding stage combines the CVRL and CVRM outputs to produce an output for each possible combination of these signals (total of 64 outputs). The 64 outputs are designated as CVRD (00—77 in octal code). These signal designations correspond to the operation code.

The decoding is inhibited by the presence of the instruction-decode-inhibit signal (CCVA) which is present during indirect addressing, field selection, program interrupt, the computer stop cycle, and recomplementing.

3-10. Group Encoder

The group encoder has two types of outputs. The final number in the signal designation for one type of output is an octal number. This type of output represents a single instruction and the number in the output designation is the same as the operation-code number. The final number for the other type is a nonoctal number. This type of output represents the 14 instruction groups. Their functions are as follows:

Output	Representation and Function
CVRG08	Add instructions which return the results to Ar's from which the data came
CVRG09	Add instructions which return the results to Ar's other than the registers from which the data came
CVRG18	Instructions in the CVRG08 group and the 2 logical operation instructions (superimpose and erase)
CVRG19	Both logical operation instructions (superimpose and erase)
CVRG28	Instructions which transfer data from memory to Ar's
CVRG29	Decimal-add instructions
CVRG38	Binary-add instructions
CVRG39	Ar-comparison instructions
CVRG48	Instructions which transfer data from Ar's to memory
CVRG49	Two instructions which require only 1 minor cycle
CVRG58	Instructions which transfer information to Ir's
CVRG59	Instructions which transfer data from memory
CVRG68	Instructions which transfer the contents of a MAC to memory
CVRG69	Instructions which compare the contents of an Ar with the contents of a memory location (instructions 54, 55, and 57 only)

3-11. Instruction-Line Generator

Instruction lines are produced by combining the outputs from the program counter and the group encoder. The first two numbers in the instruction line designation represent the CVRG signal and the third number represents the program count. For example, CV590 signifies the CVRG59 group of instructions and program count CVCP0. If only two numbers are present in the CV-signal designation, it usually signifies that the CVRG signal is combined with $\overline{CVCP0}$. For example, CV59 is the combination of CVRG59 and $\overline{CVCP0}$.

There are seven exceptions to the CV-signal designation system. They are as follows:

CV48 represents CVRG48 and $\overline{CVCP4}$ (rather than $\overline{CVCP0}$).

CV304 and CV303 require an additional input from the multiply flip-flop in the reset condition.

CV313 and CV314 require two additional inputs from the divide flip-flops in the reset condition.

CV402S, CV411S, CV412S, CV422S, CV430S, and CV431S each requires one additional input which represents a single-precision shift.

CV401D, CV402D, CV411D, CV412D, CV430D, and CV431D each requires an additional input which represents a double-precision shift.

CV40, CV41, CV42, CV30, and CV31 are produced by CVRG signals alone.

CV581 and CV531 are inhibited by the CVRINH signal.

3-12. PROGRAM COUNTER

The program counter contains 9 flip-flops which produce the 10 program-count signals. The first seven program-count signals CVCP(0—6), are produced by one flip-flop for each signal and the last three program-count signals, CVCP(7—9) are produced by two flip-flops (represented by A and B on figure 3-3) whose outputs are combined in a matrix.

3-13. General Operation

During normal operation of the program counter, the flip-flops which produce CVCP(1—9) are set by the output from the previous flip-flop (the CVCP2 FF is set by CVCP1, the CVCP3 FF is set by CVCP2, and so forth), the output CVCPSA, and T3. The CVCP0 FF is set by the CVEPCPS signal which is produced by CVEP except during recomplementing. During recomplementing the CVCP0 FF is set at T2 by CVRECPJ. All flip-flops in the program counter are reset by T3; the CVCP1 through CVCP4 FF's are also reset by the CVRECPS signal during recomplementing.

The setting and resetting of all flip-flops in the program counter, except the CVCP0 FF, is inhibited by the instruction ending pulse (CVEP) and by the instruction-inhibit signal (CVRINH) except during recomplementing. The setting and resetting of the CVCP1 through CVCP3 FF's is inhibited by the instruction-decode-inhibit signal (CVVA). The setting and resetting of the CVCP3 through CVCP9 FF's is inhibited during all minor cycles, except the last, of PC4 through PC9 of the multiply and divide instructions. Also, the resetting of the CVCP0 FF through the CVCP2 FF and the setting of the CVCP3 FF is inhibited by the binary-shift FF (CVBRS).

3-14. Program-Counter Cycles

The five program-counter cycles shown in figure 3-4 are:

The normal cycle, basic for all instructions;

The shift cycle, used during the binary-shift instruction;

The recomplement cycle, used when an addition or subtraction requires recomplementing;

The multiply cycle, used during the multiply instruction only; and

The divide cycle, used during the divide instruction only.

3-15. NORMAL CYCLE. The CVEP signal sets the CVCP0 FF at T3 at the beginning of the normal cycle. The flip-flops which produce CVCP(1—9) are set successively at each T3. As each flip-flop is set in the program counter, the preceding flip-flop is reset. This process continues until another CVEP signal is present.

A CVRINH signal inhibits the setting and resetting of all flip-flops. Therefore, when this signal is present, the program counter produces the same program count until the signal is removed. The CVRINH signal has no effect on the program counter during the recomplement cycle.

3-16. SHIFT CYCLE. This shift cycle is similar to the normal cycle until PC2. At T3 after the CVCP2 FF is set, the output from the CVBRS FF inhibits the setting of the CVCP3 FF and the resetting of the preceding flip-flops in the program counter. The CVCP2 FF remains set until the CVBRS FF is reset which occurs when the shift operation is completed. Thus, the program process initiated by PC2 during a binary shift is repeated until the shift is completed.

The CVBRS FF is set at the beginning of PC2 of the binary-shift instruction and remains set until it is reset by an output from the end-of-shift FF.

3-17. RECOMPLEMENT CYCLE. The recomplement cycle is necessary when an arithmetic operation produces a result which must be complemented. The process is as follows:

(1) At T2 the CVCP0 FF is set by the CVRECPJ signal; the program-counter FF that was set at the end of the arithmetic operation is reset by the CVREPCS signal. The CVCP1 FF is reset if the operation is single precision; the CVCP2 FF is reset if the operation is double precision as shown in figure 3-4; the CVCP3 FF is reset if the operation is triple precision; and the CVCP4 FF is reset if the operation is quadruple precision. Also, at T2, the recomplement FF (CVRECP) is set.

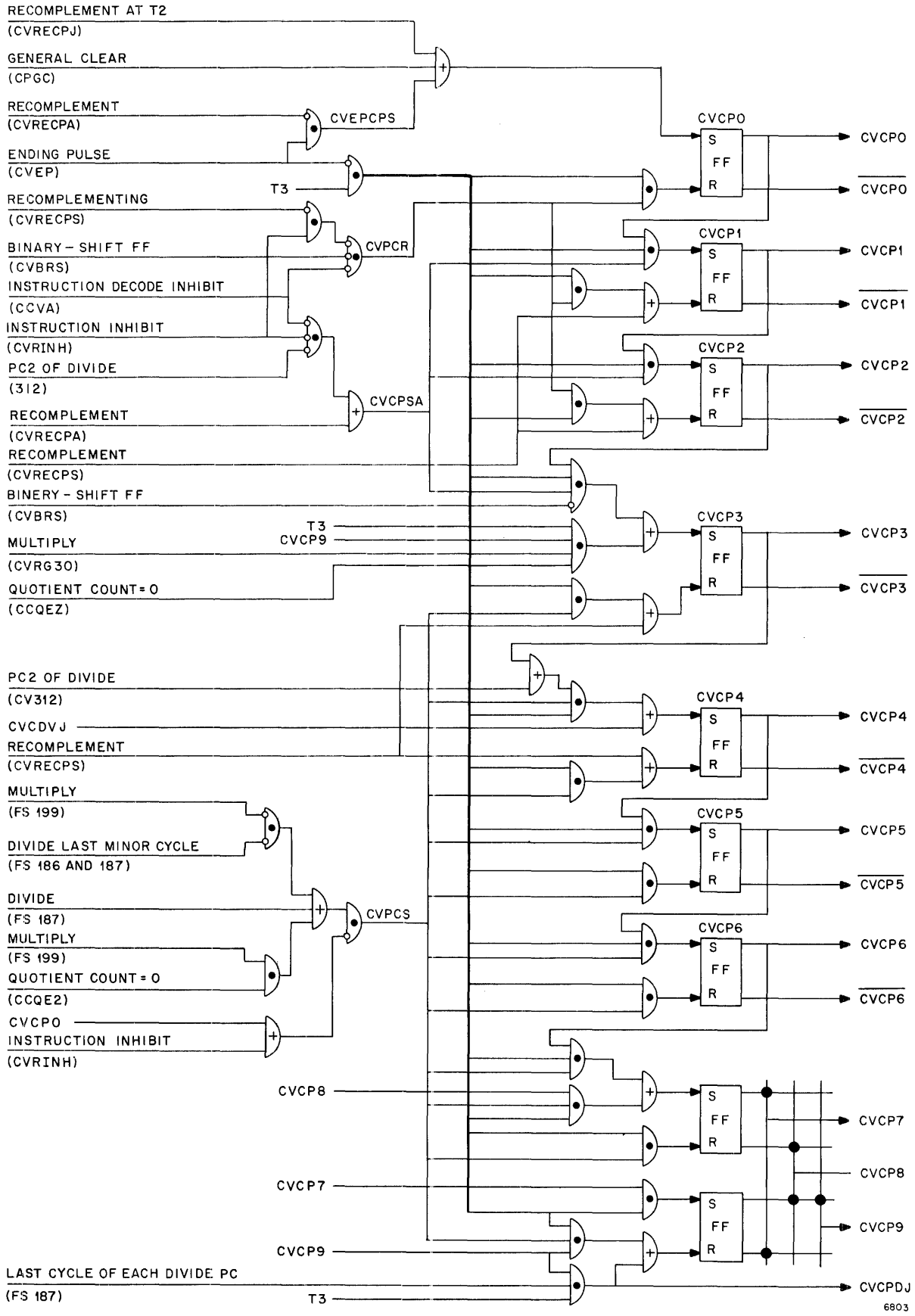


Figure 3-3. Program Counter, Detailed Block Diagram

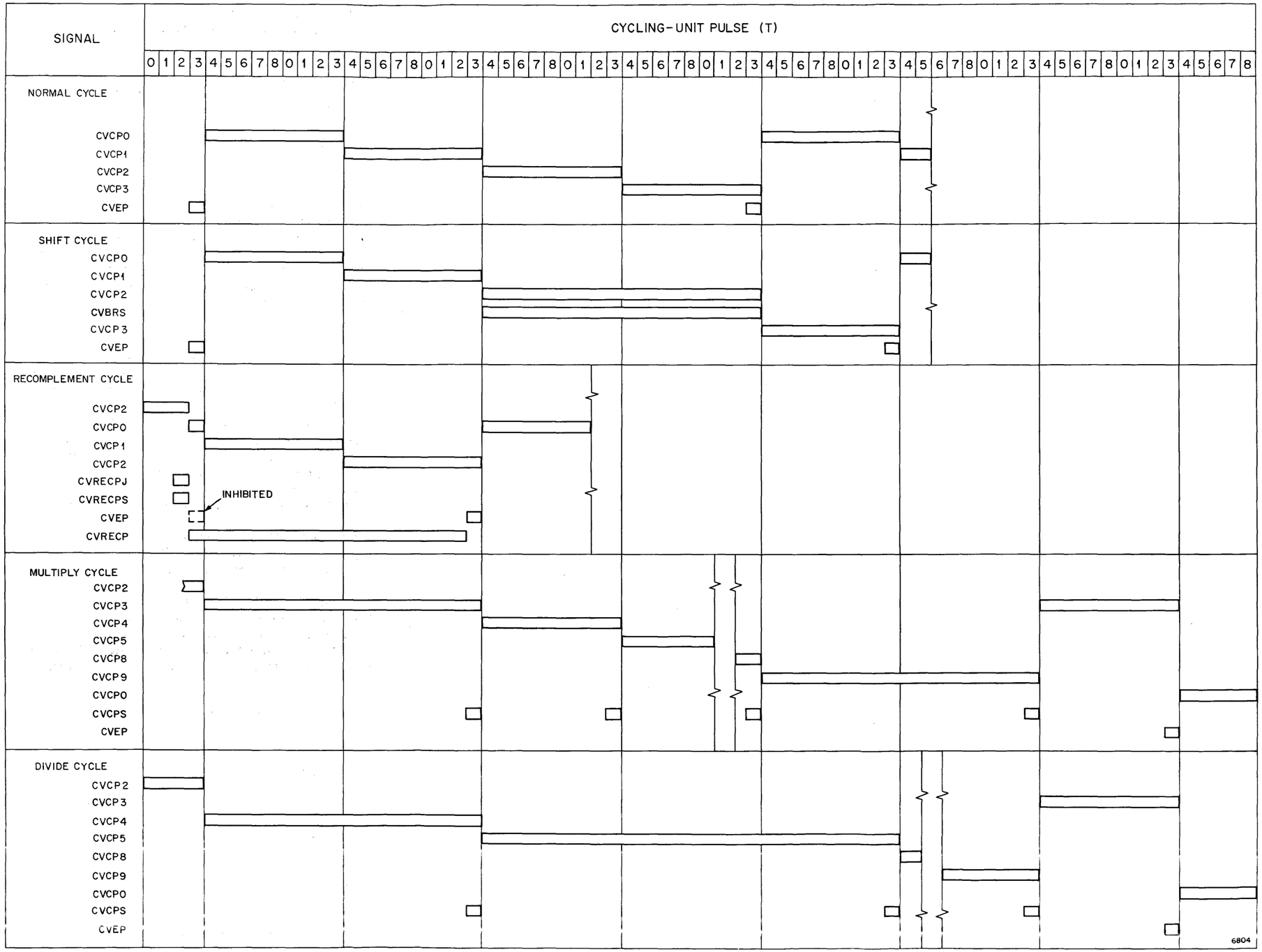


Figure 3-4. Timing Diagram for Typical Program Cycles

(2) At T3 the setting of the CVCP0 FF by the CVEPCPS signal is inhibited and the CVCP1 FF is set.

(3) When the recomplementing is completed the CVEP signal is produced and the normal program-counter cycle is initiated.

During the recomplement cycle, the inhibit produced by the CVRINH signal is not in effect.

3-18. MULTIPLY CYCLE. The central processor multiplies by successive additions. One program-counter output is provided for each XS-3 multiplier digit. The number of additions of the multiplicand and, consequently, the duration of the program count depend on the size of the multiplier digit (refer to section 10 for a description of multiplication). The multiply cycle is performed as follows:

(1) PC0 to PC3 function as in the normal cycle.

(2) The CVCP3 FF is reset and the CVCP4 FF is set by the CVPCS signal. This signal is present only when all the additions of the multiplicand for PC3 are completed. This completion is indicated when the quotient counter equals 0 (CCQEZ). The flip-flops which control PC4 through PC9 are also reset and set by the CVPCS signal. Consequently, each program count may contain more than one minor cycle since the next program-counter FF is not set until the quotient counter equals 0.

(3) When PC9 is completed the flip-flops which produce CVCP9 are reset, the CVCP3 FF is set, and the CVEP signal is produced.

3-19. DIVIDE CYCLE. Division requires at least two minor cycles for each digit, therefore, the program counter must provide variable length program counts during division (refer to section 10 for description of division). The divide cycle is performed as follows:

(1) PC0 through PC2 function as in the normal cycle.

(2) The CVCP2 signal (CV312) sets the CVCP4 FF; CVCP3 is not produced.

(3) The flip-flops which control PC4 through PC9 are successively set and reset by the CVPCS signal. This signal is only present during the final minor cycle for each digit in the divide operation.

(4) After PC9 is completed, the CVCP3 FF is set and the CVEP signal is produced.

3-20. ACCUMULATOR-REGISTER SELECTOR

The AR-register selector contains the following four logical units:

The AR-select FF's which receive and store AR-select information either from memory or from the CCr;

The AR-decode network which decodes the contents of the AR-select FF's into 12 signals;

The AR-selection network which selects either the Ar from which information is read and into which information is written; or the TCWr, sense FF, interrupt class, or synchronizer during various instructions; and

The precision network which provides one signal for each of the four possible levels of precision during multiprecision instructions.

3-21. AR-Select Flip-Flops

The AR-select FF's (figure 3-5), CRA1 through CRA4, are set by the zeros in the AR/XO-field, bit positions 11 through 14, of an instruction either from memory or from the CCr. These flip-flops are reset by the CVEP signal. The outputs from the AR-select FF's are CRA(1—4) which represent bits 14 through 11, respectively.

Function signal 127, produced at T3 by a set output from the CVRRG FF, gates the contents of a memory location (bit positions 14 through 11) into the AR-select FF's. This signal is present whenever an instruction is read from memory and is inhibited by the 127INH signal, which is present at the end of the recomplement cycle and during a program interrupt. Function signal 122 is present at the end of the recomplement cycle and gates the contents of bit positions 11 through 14 of the CCr into the AR-select FF's. This signal is inhibited when the program is interrupted.

3-22. AR-Decode Network

The AR-decode network (figure 3-5) decodes the CRA1 and CRA2 outputs in one quaternary decoder and the CRA3 and CRA4 outputs in another. Four separate outputs are produced for each pair of flip-flops. These outputs are labeled CRAD(0—7). The AR-decode network produces the following additional outputs which are formed by combinations of individual CRAD outputs. The combination outputs are these:

CRAD12, which is a result of either CRAD1 or CRAD2;

CRAD13, which is a result of either CRAD1 or CRAD3;

CRAD56, which is a result of either CRAD5 or CRAD6; and

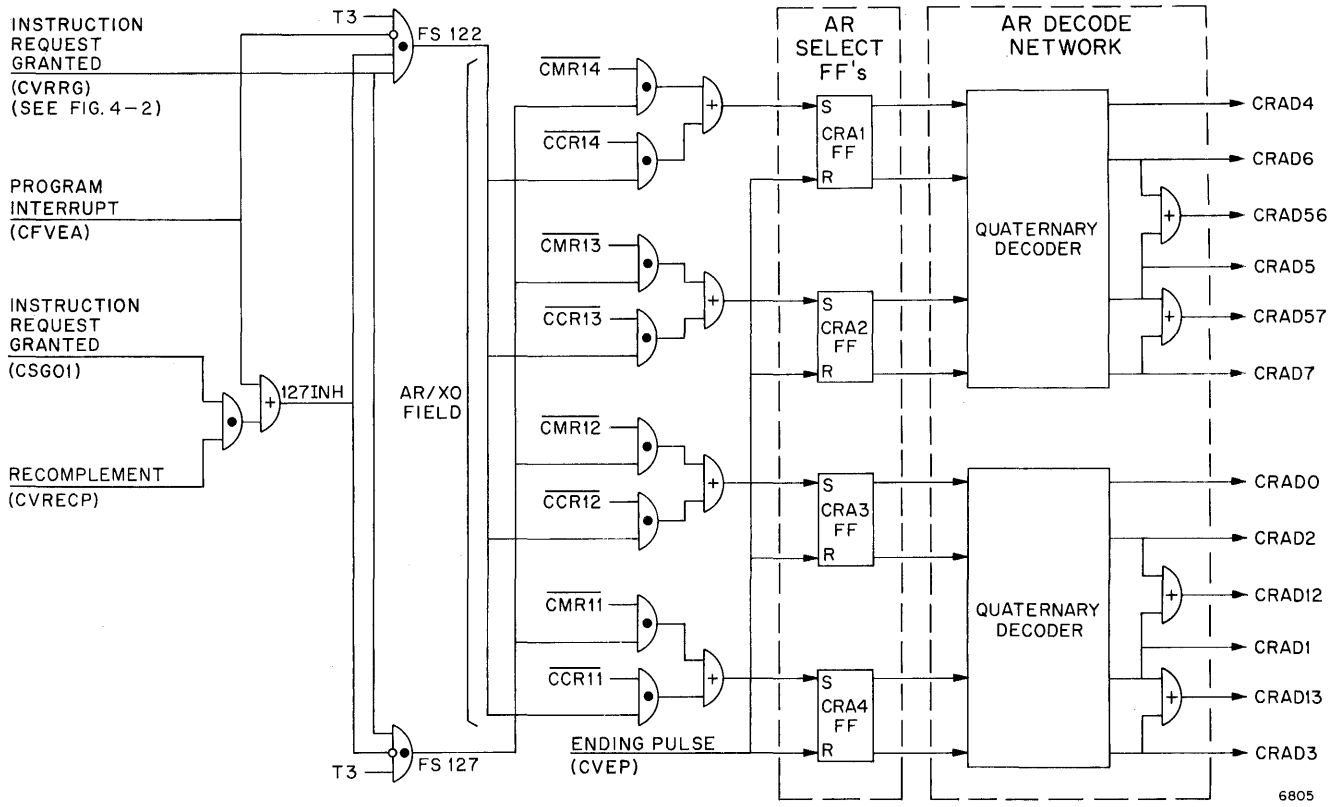


Figure 3-5. AR-Select FF's and Decode Network, Detailed Block Diagram

CRAD57, which is a result of either CRAD5 or CRAD7.

Table 3-1 lists the decoded outputs for all bit combinations in the AR-select FF's.

3-23. AR-Selection Network

The AR-selection network selects Ar's in an order specified by the contents of the AR/XO-field of the instruction, and by the type of instruction being performed. There are three types of instructions which select Ar's through the AR-selection network. They are as follows:

Instructions which select specific Ar's during specific program counts. These instructions include the multiply, divide, expand, compress, typewriter, and real-time-clock instructions.

The shift instructions which select specific Ar's based on the program count and the amount of shifting involved.

The multiprecision instructions which select the Ar's in a fixed sequence.

The typewriter and real-time-clock instructions require only one Ar, and the expand and compress instructions require two Ar's. The multiply instruction uses three Ar's and the divide instruction requires two. The selection of these registers during multiply and divide is described in section 10. The shift instructions require the selection of one or two Ar's depending upon the precision of the instruction.

There are two types of multiprecision instructions that use the output of the AR-selection network: the 08-type instruction (CVRG08, CVRG58, CVRG59, and CVRG73); and the 09-type instruction (CVRG09). The 08-type instruction selects the same Ar to supply the input to the adder and receive the output from the adder. The 09-type instruction selects one Ar to supply the input to the adder and another Ar to receive the output from the adder.

The general selection process for the 08- and 09-type instruction is as follows:

Single-precision 08-type instructions always select the designated Ar during PC1.

Multiprecision 08-type instructions (except the zero-suppress instruction) select the Ar's containing the least significant digits first, and then the Ar's containing the most significant digits.

Single-precision 09-type instructions first select the Ar represented by the larger number to supply the adder input, and then the Ar represented

Table 3-1. Decoding of AR-Select Flip-Flop Outputs

Contents of AR-Select FF's CRA 1	2	3	4	Signals Produced from Quaternary Decoders: CRA3 and CRA4	CRA1 and CRA2
0	0	0	0	CRAD0	CRAD4
0	0	0	1	CRAD1, CRAD12, CRAD13	CRAD4
0	0	1	0	CRAD2, CRAD12	CRAD4
0	0	1	1	CRAD3, CRAD13	CRAD4
0	1	0	0	CRAD0	CRAD5, CRAD56, CRAD57
0	1	0	1	CRAD1, CRAD12, CRAD13	CRAD5, CRAD56, CRAD57
0	1	1	0	CRAD2, CRAD12,	CRAD5, CRAD56, CRAD57
0	1	1	1	CRAD3, CRAD13	CRAD5, CRAD56, CRAD57
1	0	0	0	CRAD0	CRAD6, CRAD56
1	0	0	1	CRAD1, CRAD12, CRAD13	CRAD6, CRAD56
1	0	1	0	CRAD2, CRAD12	CRAD6, CRAD56
1	0	1	1	CRAD3, CRAD13	CRAD6, CRAD56
1	1	0	0	CRAD0	CRAD7
1	1	0	1	CRAD1, CRAD12, CRAD13	CRAD7
1	1	1	0	CRAD2, CRAD12	CRAD7
1	1	1	1	CRAD3	CRAD7

by the smaller number to receive the adder output (for example, if Ar1 and Ar3 are designated, the adder input register is Ar3 and the adder output register is Ar1).

Double-precision 09-type instructions select Ar4 as the first adder input register, Ar2 as the first adder output register, Ar3 as the second adder input register, and Ar1 as the second adder output register.

3-24. FUNCTION SIGNALS 235, 236, 237, AND 238. The combination of function signals 235 through 238 (figure 3-6) and the AR-selection network inputs produces the accumulator-register selections. The flip-flops that produce the function signals are set by CV lines for the multiply, divide, expand, compress, typewriter, and real-time-clock instructions.

During multiprecision instructions, one flip-flop is set by a CV line, and the other flip-flops are set in sequence at T3 by the output of the previous flip-flop. Function signal 235 is produced first by all 08-type instructions and function signals 236 through 238 are produced in sequence for 08-type multiprecision instructions. Function signal 236 is produced first by single-precision 09-type instructions (CV090S). Function Signal 237 is produced during PC1 by all double-precision 09-type instructions (CV090D) and FS 238 is produced during the next minor cycle.

Function signals 235 through 238 are produced simultaneously at T3 when an add or subtact instruction results in a sum or difference of 0. The presence of these function signals inserts a positive sign bit (0) and two check bits into all Ar's that were used during the arithmetic operation.

The 73 instruction reverses the normal sequence of the production of function signals 235 through 238 in order that the most significant word can be processed first. The first function signal produced depends on the precision of the operation, and the last function signal produced is always FS 235. For example, if the 73 instruction is triple precision, FS 237 is produced first, FS 236 is produced next, and FS 235 is produced last. The selection of FS 235 through FS 238 during a 73 instruction is performed by gating the output of the precision FF's (which is changed each program count during a 73 instruction) with CVRD73.

The CVEP and CVRINH signals inhibit function signals 236 through 238. The CVEP signal, which is present at T3, inhibits the production of the next function signal when the instruction is completed. The CVRINH signal inhibits the setting of the next flip-flop and keeps the flip-flop that is set, set.

3-25. FUNCTION SIGNAL 128. Function signal 128, produced during all 08-type instructions by a pulseformer, gates the output of the main adder into the Ar from which the adder input came. The pulseformer is set each pulse time between T5 and T2 during the 08-type instructions. The output from the pulseformer is available from T6 through T3.

The CVRINH signal inhibits the setting of the FS 128 pulse former except during the recomplement cycle. The output of the pulseformer is inhibited at specific times during the multiply and divide instructions (refer to section 10).

3-26. FUNCTION SIGNAL 129. Function signal 129 selects the register into which data are returned

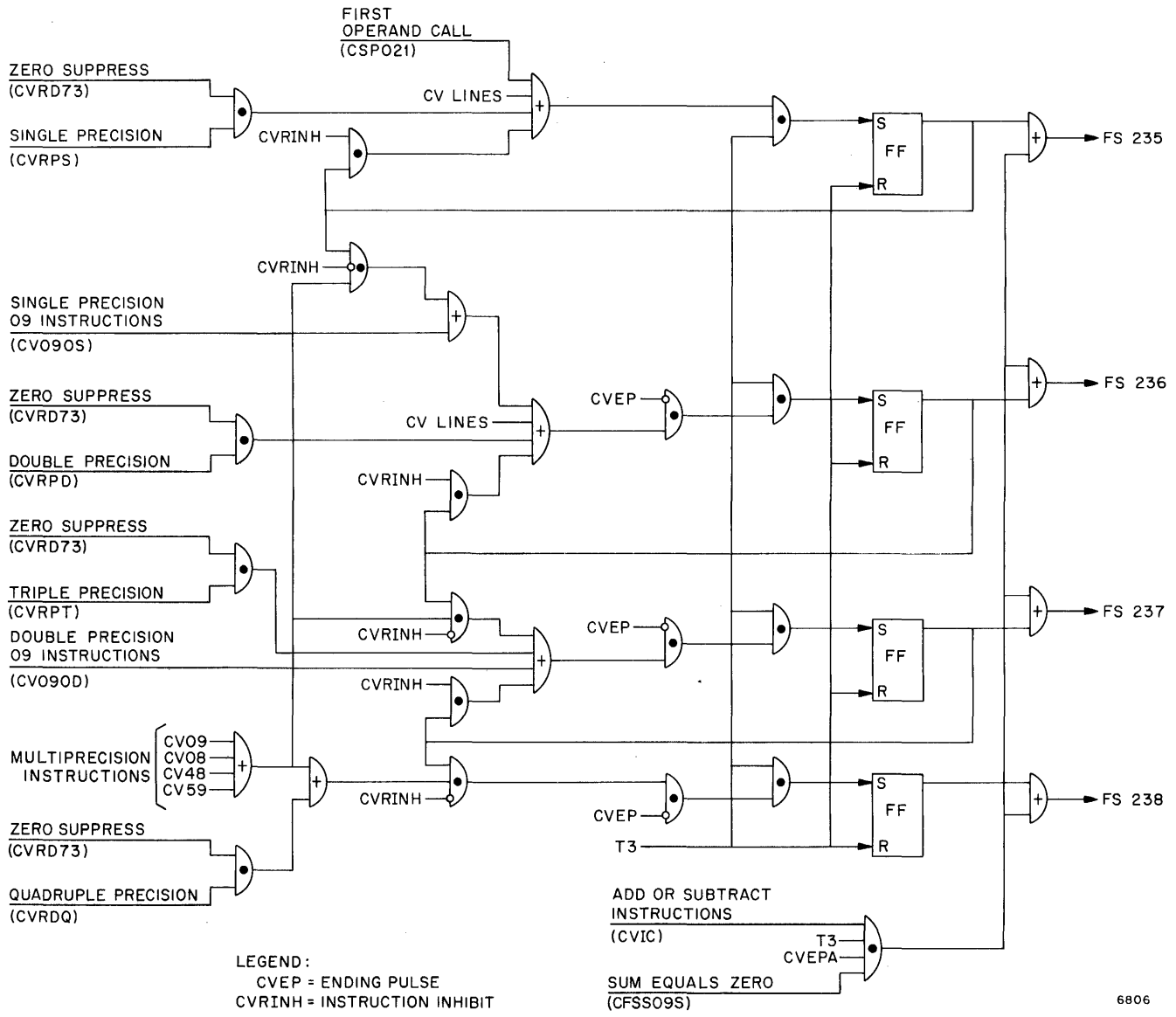


Figure 3-6. Production of Function Signals 235, 236, 237, and 238, Detailed Block Diagram

during an 09-type instruction. This function signal is produced by gating CVRG09 with T6 through T3. Thus, FS 129 is present between T6 and T3 for all program counts of the 09-type instructions. Function signal 129 is used with function signals 236, 237, and 238 during PC1 and PC2 of the 09-type instructions. Function signal 129 is inhibited during recomplementing.

3-27. Selection of Accumulator Registers

The selection of accumulator registers is performed by combining the outputs of the AR-decode network with FS 235 through FS 238. The combination of these signals produces CRS(1—4) which gate information out of Ar 1 through 4, and CRV(1—4) which gate information into Ar 1 through 4. During the 08-type instructions, the CRS and CRV signals select the same Ar; during 09-type instructions, the CRS and CRV signals select different Ar's.

The details of the Ar-selection gating are listed in tables 3-2 and 3-3. Table 3-2 lists all bit combinations in the AR-select FF's, the function signals, the accumulator registers to be selected, and the selection sequence for all 08-type instructions. Table 3-3 lists the same information for the 09-type instructions. In table 3-3 all invalid bit configurations of the AR-select FF's are labeled not valid (NV). All single-precision-09 instructions require 1's in two of the AR-select FF's; double-precision-09 instructions require 1's in each of the AR-select FF's. In both tables the dashes signify that the decoded contents of the AR-select FF's do not select an Ar.

Table 3-2. Selection of Accumulator Registers During 08-Type Instructions by Function Signals 235 through 238

CRA 1	2	3	4	Ar's Selected By FS 235	FS 236	FS 237	FS 238
0	0	0	0	—	—	—	—
0	0	0	1	4	—	—	—
0	0	1	0	3	—	—	—
0	0	1	1	4	3	—	—
0	1	0	0	2	—	—	—
0	1	0	1	4	2	—	—
0	1	1	0	3	2	—	—
0	1	1	1	4	3	2	—
1	0	0	0	1	—	—	—
1	0	0	1	4	1	—	—
1	0	1	0	3	1	—	—
1	0	1	1	4	3	1	—
1	1	0	0	2	1	—	—
1	1	0	1	4	2	1	—
1	1	1	0	3	2	1	—
1	1	1	1	4	3	2	1

3-28. Precision Selection

The precision of an arithmetic operation (figure 3-7) (single, double, triple, or quadruple) is determined by the decoded outputs of the two precision-selection FF's. The outputs of these flip-flops (called A and B in figure 3-7) are decoded by a diode matrix to produce one signal for each of the four possible precisions. Table 3-4 lists all possible bit combinations of the AR-select FF's, and the precision signals they produce.

Table 3-3. Selection of Accumulator Registers During 09-Type Instructions by Function Signals 235 through 238

CRA 1	2	3	4	Ar's Selected by FS 236 Output	FS 236 Input	FS 237 Output	FS 237 Input	FS 238 Output	FS 238 Input
0	0	0	0	[1]NV	NV	NV	NV	NV	NV
0	0	0	1	NV	NV	NV	NV	NV	NV
0	0	1	0	NV	NV	NV	NV	NV	NV
0	0	1	1	4	3	—	—	—	—
0	1	0	0	NV	NV	NV	NV	NV	NV
0	1	0	1	4	2	—	—	—	—
0	1	1	0	3	2	—	—	—	—
0	1	1	1	NV	NV	NV	NV	NV	NV
1	0	0	0	NV	NV	NV	NV	NV	NV
1	0	0	1	4	1	—	—	—	—
1	0	1	0	3	1	—	—	—	—
1	0	1	1	NV	NV	NV	NV	NV	NV
1	1	0	0	2	1	—	—	—	—
1	1	0	1	NV	NV	NV	NV	NV	NV
1	1	1	0	NV	NV	NV	NV	NV	NV
1	1	1	1	—	—	4	2	3	1

[1] Not Valid

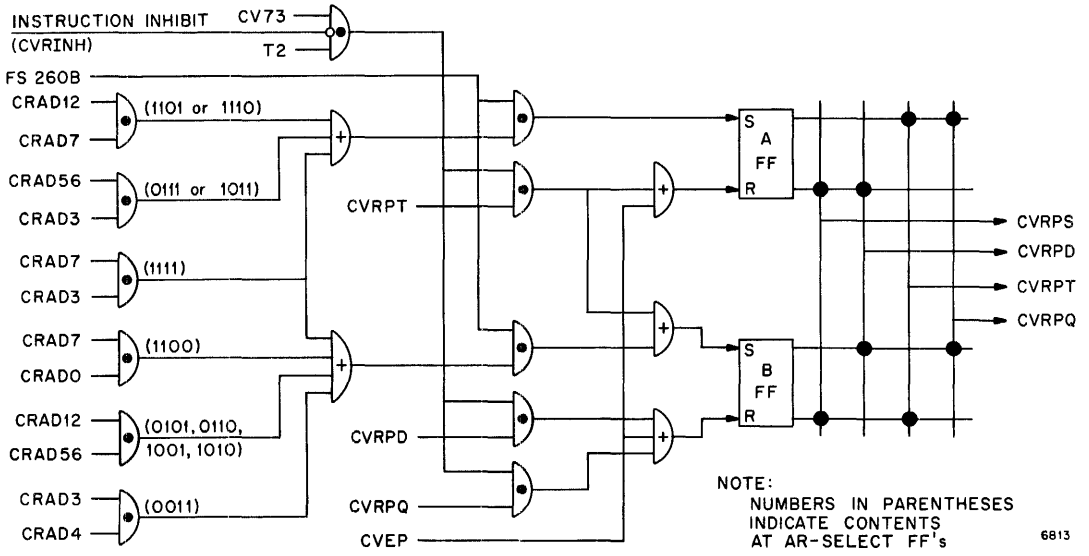


Figure 3-7. Production of Precision Signals, Detailed Block Diagram

Table 3-4. Precision Signals for All Bit Combinations in the AR-Select Flip-Flops

CRA1	CRA2	CRA3	CRA4	A	B	Signal	Precision
0	0	0	0	R	R	CVRPS	Single
0	0	0	1	R	R	CVRPS	Single
0	0	1	0	R	R	CVRPS	Single
0	0	1	1	S	R	CVRPD	Double
0	1	0	0	R	R	CVRPS	Single
0	1	0	1	S	R	CVRPD	Double
0	1	1	0	S	R	CVRPD	Double
0	1	1	1	R	S	CVRPT	Triple
1	0	0	0	R	R	CVRPS	Single
1	0	0	1	S	R	CVRPD	Double
1	0	1	0	S	R	CVRPD	Double
1	0	1	1	R	S	CVRPT	Triple
1	1	0	0	S	R	CVRPD	Double
1	1	0	1	R	S	CVRPT	Triple
1	1	1	0	R	S	CVRPT	Triple
1	1	1	1	S	S	CVRDQ	Quadruple

During the 73 instruction, the precision designation produced by the precision-selection FF's is decreased by one each program count. For example, if the instruction is triple precision, CVRPT is produced during PC0, CVRPD is produced during PC1, and CVRPS is produced during PC2. This gating arrangement is necessary in order to process the most significant word first.

3-29. Other Selection Functions of the AR-Select Flip-Flops

The CRAD(0—7) outputs are inputs to eight high-speed drivers which produce outputs CRF(1—8). The CRF(1—8) outputs represent CRAD(0—7), respectively. These outputs select the sense FF's, interrupt FF's, TCWr's, and synchronizers during various instructions.

The outputs of the AR-select flip-flops, CRA(1D—4D), are gated directly into the MAS to select the input-output standby address during an initiate I-O instruction.

3-30. INDEX-REGISTER SELECTOR

The IR selector consists of four FF's (CMMAA1 through CMMAA4), which store the X-field of an instruction when this instruction is read from memory and the AR/XO-field during SAr-load or -store instructions, and a decode network which decodes the contents of the four IR-select FF's (figure 3-8).

The IR-select FF's are set by a combination of the 0's in bit-positions 21 through 24 of an operand or instruction and FS 267. This function signal is present at T3 when the instruction memory request is granted, or during indirect addressing or field selection when the operand is read from memory.

The other input which sets the IR-select FF's comes from the AR-select FF's ($\overline{CRA1}$ through $\overline{CRA4}$) at T3 during instructions which store or load the contents of an SAr. This input is gated by FS 266 which is present at T3 during PC0 of all SAr-load or -store instructions.

The outputs of the IR-select FF's, CMMAA1 through CMMAA4, are decoded in two separate decoding networks. Signals CMMAA1 and CMMAA2 are decoded in one network to produce four outputs, CMMAB0 through CMMAB3; signals CMMAA3 and CMMAA4 are decoded in the other network to produce four outputs, CMMAB4 through CMMAB7. The decoded outputs are then combined with FS 100 to produce an output for each of the 15 Ir's, and 1 output which produces 0's if the IR-select FF's contain all 0's.

3-31. INSTRUCTION MEMORY REQUESTS

The request for the next instruction is made at the conclusion of each instruction. This instruction request occurs when an ending pulse sets the instruction-request FF (CSR01) at T7 during the next-to-last program count of an instruction. The instruction-request signal (CSP01S) is produced by instruction-control signals during most instructions, and by special gating for multiprecision instructions, the zero-suppress instruction, shift instructions, and by specific error conditions.

The instruction memory requests are inhibited during recomplementing by CVRECP, during a computer stop cycle by CCSTPD, and when the instruction-inhibit signal (CVRINH) is present.

3-32. Instruction-Control Signals

The following instructions produce an ending pulse during PCO:

Typewriter instructions;

No operation instruction;

Display instructions;

Load-time instruction;

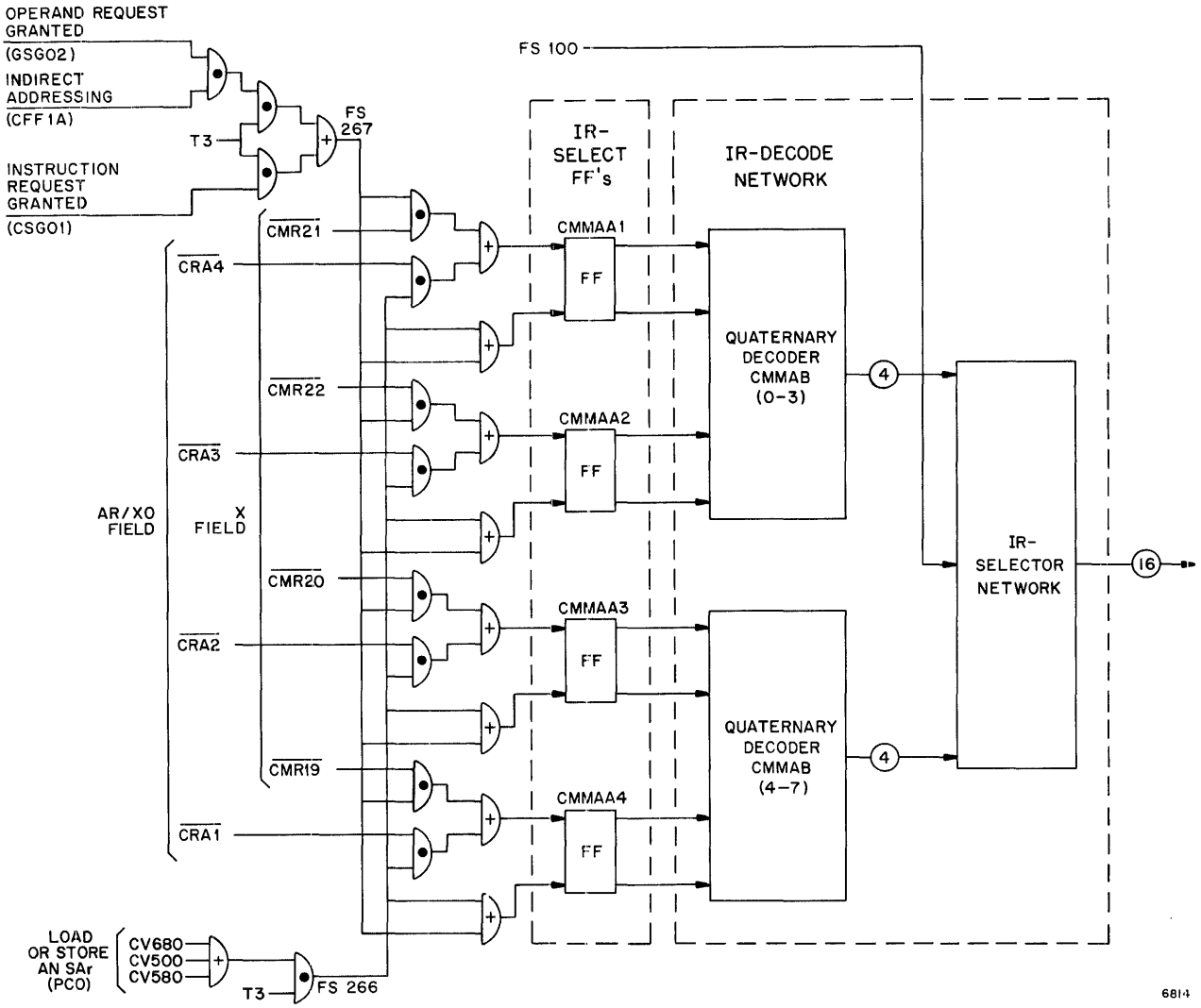


Figure 3-8. Index-Register Selector, Detailed Block Diagram

Test-sense-FF instructions; and

Wait instruction.

The instruction-request signal is produced either by gating the CVRD signal with CVCP0, or by using the CV line that is produced by gating a CVRG signal with CVCP0. There are six instructions that produce the instruction-request signal by direct gating of CV lines for program counts other than PC0. The instructions and the program counts that produce the instruction-request signal are as follows:

PC1 for all instructions which transfer data between memory and the Ir's, MAC's, or TCWr's:

PC1 for the initiate I-O instruction;

PC3 for the divide instruction;

PC4 for the multiply instruction;

PC2 for the modify-and-compare-Ir instruction; and

PC6 for the expand and compress instructions.

3-33. Multiprecision Instructions

The multiprecision instructions include all 08-type and 09-type instructions. The instruction-request signal for the 08-type instruction is produced by the combination of the CVRG signals (CVRG08, CVRG59, CVRG73, and CVRG48) and the CVRPEA signal. The CVRPEA signal is produced by combining the output from the precision-selection FF's with the outputs from the program counter, thus producing an instruction memory request at different program

counts. The instruction-request signal for single-precision operations is produced by combining CVCP0 and CVRPS; the ending pulse for double precision is produced by combining CVCP1 and CVRPD; the instruction-request signal for triple precision is produced by combining CVCP2 and CVRPT; and the instruction-request signal for quadruple precision is produced by combining CVCP3 and CVCPQ.

The instruction-request signal for the 09-type instructions is produced by the combination of the CVRG09 and CVRPEB signals. The CVRPEB signal is produced by combining the output from the precision-selection FF's with program-counter outputs. Only the outputs representing double precision and quadruple precision are represented. This is because single-precision 09-type instructions are represented by two 1 bits in the AR-select FF's, and double-precision 09-type instructions are represented by four 1 bits in the AR-select FF's.

3-34. Zero-Suppress Instruction

The instruction-request signal for double-, triple-, and quadruple-precision zero-suppress (73) instructions is produced by FS 236. Function signal 236 is present during the next-to-last program count of all multi-precision 73 instructions. The instruction-request signal, during PC0 for a single-precision 73 instruction, is produced by the combination of the CVRPEA and CVCP0 signals.

3-35. Shift Instructions

During shift instructions, the instruction-request signal is produced by the following conditions: a shift which involves two registers, a shift which involves three registers, and a shift which causes one of the registers to be cleared.

3-36. Error Conditions

An instruction-request signal is produced when an invalid operation code is detected, or when the dividend is larger than the divisor in a division problem. Both these conditions result in a program interrupt. An instruction call is made because the program interrupt requires that another instruction be called before the interrupt can take place. In the case of the division error, the instruction-request signal is produced to eliminate the need for going through the long-division instruction when an error is detected during PC1. In the case of the invalid operation code, the instruction-request signal is produced because no ending pulse would otherwise be produced by the invalid operation code.

The error conditions are represented by the CSP01A and CSP01C error signals for the invalid instruction codes, and CSP01B error signal for the division error. These error signals set the instruction-error FF (CSP01E) at T6, and the flip-flop is reset every T7.

3-37. OPERAND MEMORY REQUESTS

The first memory request for an operand is made automatically for each instruction. Additional operands are requested only when necessary. Additional operands are requested by the following:

All instructions which transfer data to or from SAr's produce an operand request during PC0.

The initiate I-O instruction which produces an operand request during PC0.

The expand instruction produces an operand request during PC1, PC3, and PC5.

The compress instruction which produces an operand memory request during PC1 and PC4.

All multiple-precision instructions which produce an operand request during each program count except when the CVRPEA or CVRPEB signals are present.

The operand memory requests are inhibited by the output of the instruction-request FF to ensure that an operand request is not made if the instruction request is not granted.

3-38. INSTRUCTION-CONTROL SIGNALS

The instruction-control signals and their functions are as follows:

The CVEP signal which represents the ending of one instruction and the beginning of the next;

The CVRINH signal which inhibits all program decoding if the memory request for an operand is not granted;

The CVRECP signal which controls the functioning of the recomplement cycle; and

The CCVA signal which inhibits the decoding of the operation code.

3-39. The CVEP Signal

The CVEP signal is produced by the output of the ending-pulse-storage FF (figure 3-9) which is set

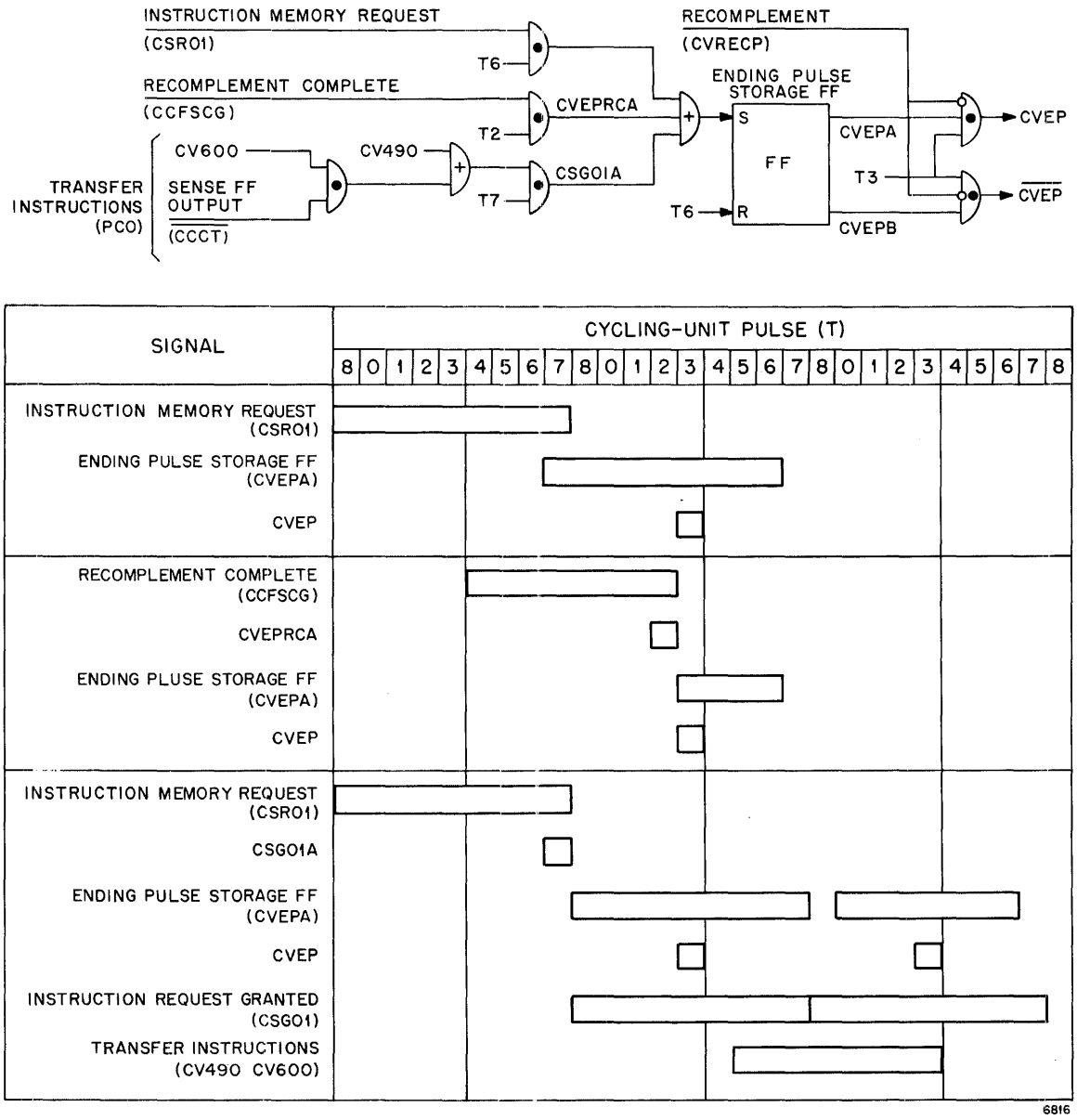


Figure 3-9. Production of the CVEP Signal, Detailed Block Diagram and Timing Diagram

under the following conditions: at T6 after a memory request has been made for an instruction; at T7 during PC0 of a transfer instruction (CSG01A); and at T2 following a recomplement cycle (CVEPRCA). The outputs of the ending-pulse-storage FF are CVEPA and CVEPB. These outputs are gated with T3 to produce the CVEP and $\overline{\text{CVEP}}$ signals. The CVEP and $\overline{\text{CVEP}}$ signals are inhibited during the recomplement cycle.

The CVEP signal performs the following functions:

Resets the CVr;

Sets the CVCP0 FF and inhibits the setting of all other CVCP FF's;

Sets instruction-decode-inhibit (CCVA) FF;

Inhibits the generating of function signals 236, 237, and 238;

Resets the AR-select FF's; and

Resets the precision FF's.

3-40. The CVRINH Signal

The CVRINH signal is produced by the CVRINH FF (figure 3-10) which is set at either T2 or T3 by the set output from the operand request FF (CSR02) and the absence of the operand-request-granted signal

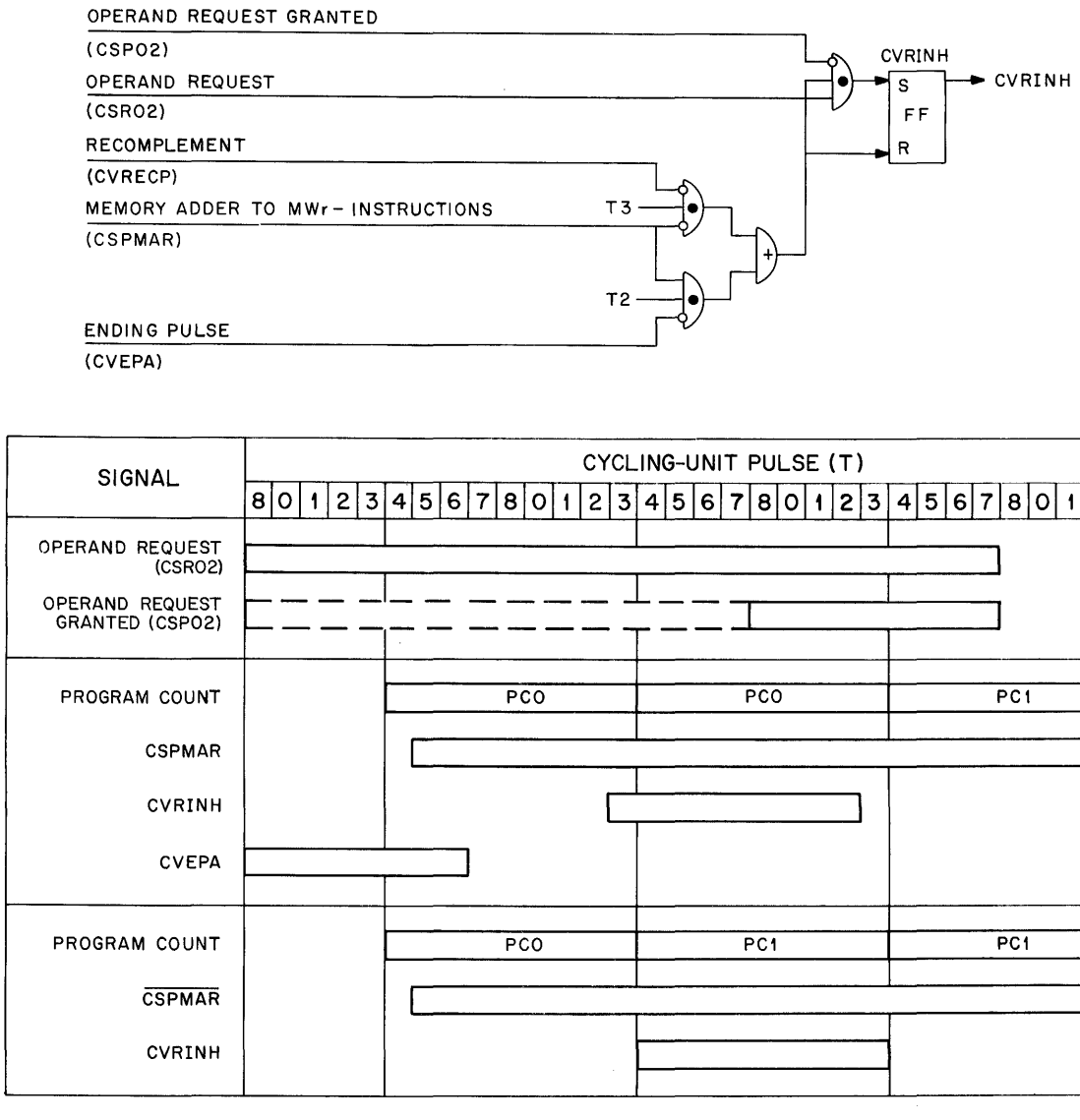
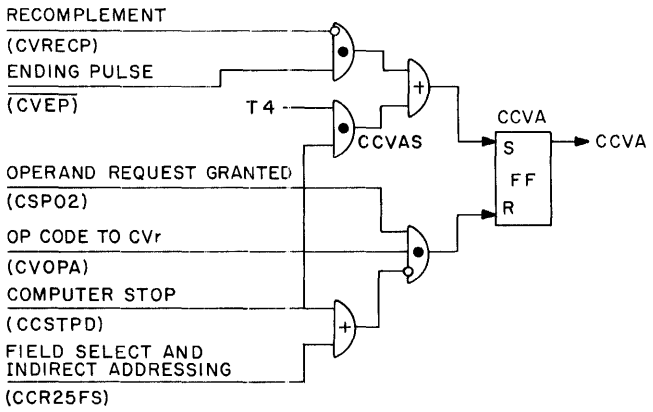


Figure 3-10. Production of the CVRINH Signal, Detailed Block Diagram and Timing Diagram

($\overline{\text{CSP02}}$). Thus, this flip-flop is set if the operand request is not granted.

The CVRINH FF is set at T2 during the instructions which transfer the memory-adder output to the MWr and during the load-MAC, and load-Ir instructions; and at T3 during all other instructions. When the flip-flop is set at T2, the output is available at T3 which enables the CVRINH signal to hold the program counter at its present count and the process associated with this count is repeated until the flip-flop is reset. When the CVRINH FF is set at T3, the output is available at T4. At T4 the next program count has been produced, and the process associated with this program count is performed. The use of the two timing pulses is necessary because the memory-addressing procedure for the instructions which transfer the memory-adder output to MWr and for the load-Ir, and load-MAC instructions can not be interrupted at the end of a program count without destroying the address information. The program count must be repeated to produce the address again. The other instructions store the address in an SAr at the end of a program count, and the previous program count does not have to be repeated.

The CVRINH signal is inhibited during recomplementing by CVRECP for instructions which set the CVRINH FF at T3, and by the CVEPA signal for instructions which set the CVRINH flip-flop at T2. The inhibit during recomplementing is necessary because the recomplement signal inhibits the production of CSP02 which normally produces CVRINH. The CVEPA signal, which is present while the ending-pulse-storage FF is set, inhibits the CVRINH



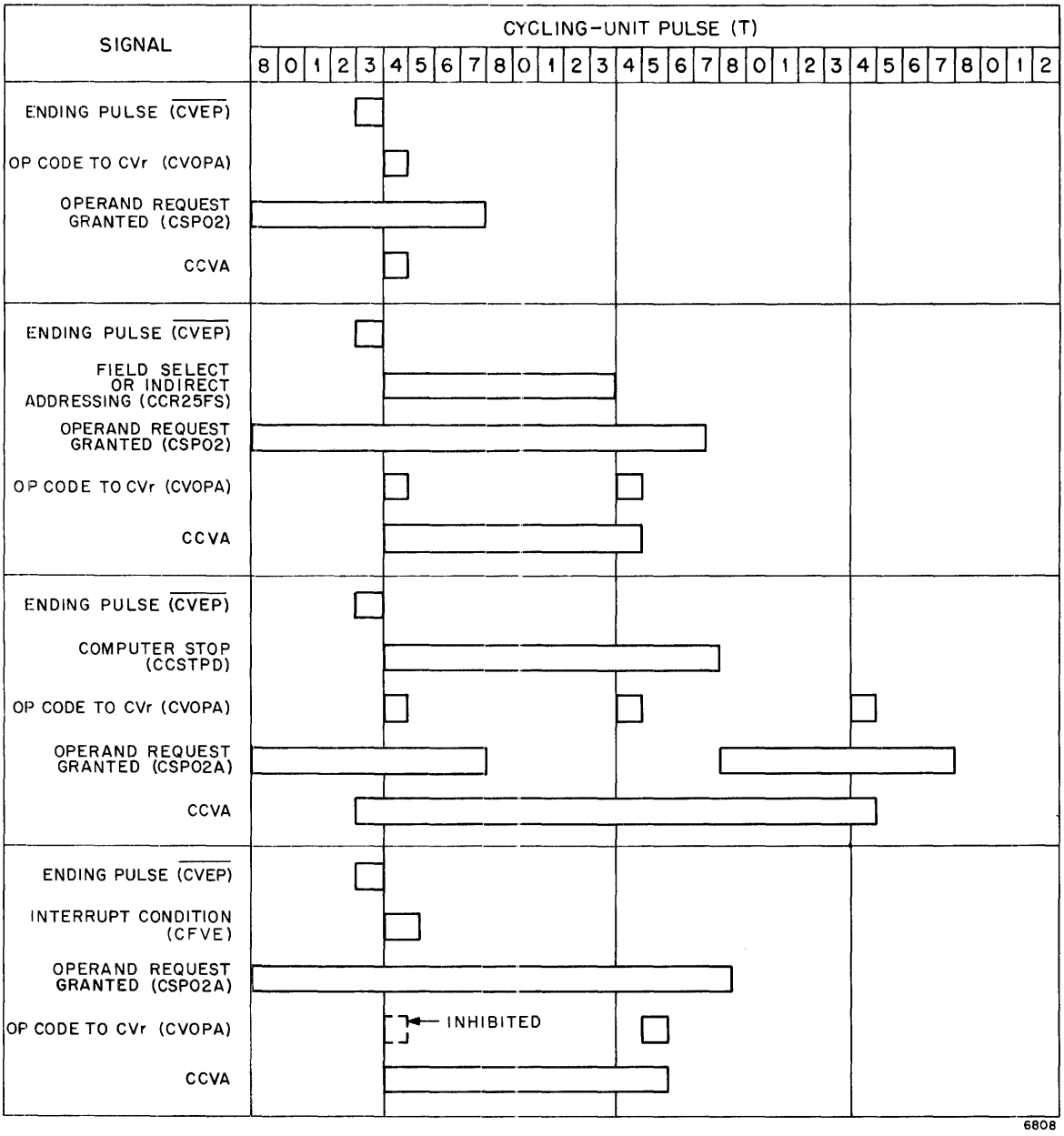


Figure 3-11. Production of the CCVA Signal, Detailed Block Diagram and Timing Diagram

signal during the first operand call of the next instruction. The inhibit by the CVEPA signal is necessary because at T2, when the CVRINH flip-flop is set, the new instruction is not decoded.

The CVRINH signal performs the following functions:

Inhibits the setting and resetting of the program-counter flip-flops;

Inhibits the production of FS 128; and

Inhibits the stepping of FS 235 and FS 238 (for example, if FS 236 is present, the CVRINH signal causes FS-236 FF to remain set, and inhibits the setting of FS-237 FF.)

3-41. The CVRECP Signal

The CVRECP signal is produced by the CVRECP FF. This flip-flop is set at T2 by the CVECPJ signal which is present when recomplementing is necessary. The CVRECP flip-flop is reset by CVEPRCA which is present at T2 when the recomplement cycle is complete.

The CVRECP signal performs the following functions:

Enables normal operation of the program counter when a CVRINH signal is present;

Enables normal production of FS 128 when the CVRINH signal is present;

Inhibits the production of CVEP;

Inhibits the production of an instruction request;

Inhibits the production of FS 260;

Inhibits the production of FS 127;

Inhibits the resetting of the instruction-decode-inhibit FF; and

Inhibits the production of CSP02.

3-42. The CCVA Signal

The CCVA signal (figure 3-11) inhibits the transfer of the instruction-register contents to the octal decoder. This signal is produced by the CCVA FF which is set by the $\overline{\text{CVEP}}$ signal and reset at T4 except under the following conditions:

During a computer stop cycle;

During field selection or indirect addressing; and

During interrupt conditions.

Under normal conditions, the CCVA FF is set at T3 and reset at T4. However, when any of the conditions which inhibit the resetting of the CCVA FF are present, this flip-flop remains set until the conditions are removed.

Section 4
PRIORITY SECTION

4-1. INTRODUCTION

The purpose of the priority section is to grant memory access in a sequence which makes possible the most efficient use of computer time. The priority section receives memory requests from the central processor and the input-output equipment. Several of these sources may request memory access simultaneously. The priority section grants memory access to only one source at a time. The source selected depends on comparisons of various input-output data-handling speeds. An I-O synchronizer that processes information at a fast rate is granted memory access more often than a slower operating synchronizer. Thus, a synchronizer with a fast word-assembly time requires access to the memory more often than a synchronizer with a slow word-assembly time.

For example, the tape synchronizer receives data faster than the card-reader synchronizer. The tape synchronizer assembles a word in 30 microseconds. The card-reader synchronizer assembles a word in 112 microseconds. If both the tape and card-reader synchronizer request memory access at the same time, the tape synchronizer gets priority first.

The central processor (CP) has the lowest priority because the execution of a CP instruction can be interrupted and completed at a later time without loss of information.

4-2. PRIORITY ORDER

The priority section controls up to 15 requests, each from a different source. Each source is assigned 1 of 15 priority channels. The priority channels are UNISERVO tape units, five channels; general purpose, eight channels; and the central processor, two channels, one for the instruction, and one for the operand. Table 4-1 lists the priority channels and order of priority.

Each general-purpose (GP) channel can accept an input from one synchronizer. This synchronizer may be for a card reader, card punch, high-speed printer, or any other peripheral unit, except tape, that is compatible with the UNIVAC III system. In addition, GP-channel 8 has an input from a memory-request-simulator flip-flop (CCMRS) which is set by a maintenance console (engineer's maintenance panel) switch.

Four of the five UNISERVO channels are for UNISERVO III tape units and one is for UNISERVO II tape units. Each channel receives inputs from the tape synchronizers which request access to the memory. Other functions of these signals are explained in the synchronizer descriptions, manual 2, part 3, section 1, table 1-6. All inputs are related to tape operations but perform the same function, that is, request memory access. The purpose of the different inputs is described in the synchronizer descriptions.

The central-processor instruction channel has two inputs: one input, CSP01S, is produced by the central processor to request an instruction from memory; the other, CSP01ST, is manually inserted from the maintenance console.

The operand channel has four inputs. Each requests memory access for an operand.

Table 4-1. Order of Priority Channel

Source	UNISERVO III Synchronizers				UNISERVO II Synchronizer	General Purpose Synchronizers								Central Processor	
	HA	HB	FA	FB		1	2	3	4	5	6	7	8	Operand Call	Instruction Call
Order of Priority	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Priority Channels	3	4	14	15	13	5	6	7	8	9	10	11	12	2	1

4-3. OPERATION OF PRIORITY CHANNELS

The basic functions of the priority channels are to:

Store the memory request until memory access can be granted;

Compare requests to determine if the request can be granted;

Initiate the readout of address data from a storage-address register (SAr) or tape-control-word register (TCWr) when memory access has been granted; and

Control the storage of address data in a specified SAr or TCWr.

The general operation of a priority channel is as follows:

A memory-request signal sets the memory-request flip-flop, producing the memory-request signal (CSR) as shown in figure 4-1.

A decoder compares the CSR signal with memory requests from devices with higher priority. If no higher priority device is requesting memory, a priority-granted signal (CSP) is produced. This signal controls the readout of address information from a SAr.

At the end of one minor cycle, the CSP signal produces the CSRG signal which resets the memory-request flip-flop and sets the memory-granted flip-flop.

A granted flip-flop produces the CSG signal which returns the address into the specified SAr.

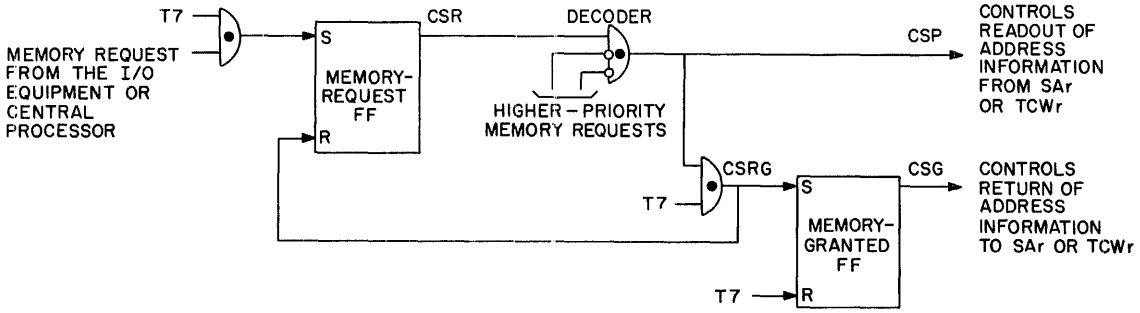
The memory-request flip-flop is set at T7 and is not reset until the associated granted flip-flop is set at the next T7. The granted flip-flop remains set for one minor cycle and is reset at T7.

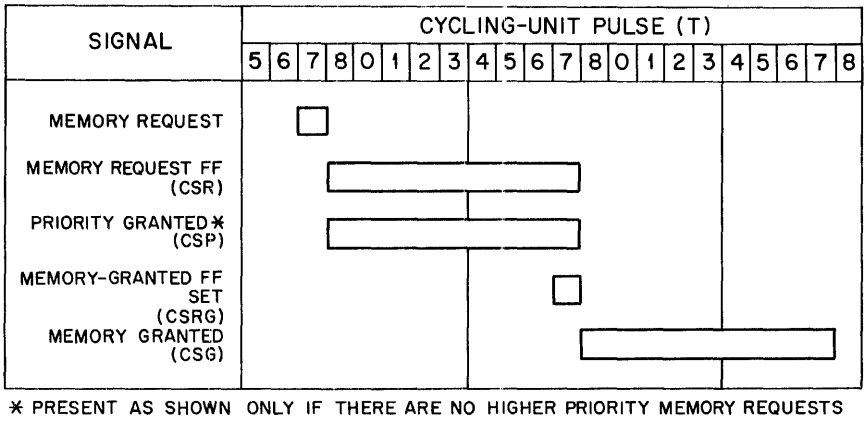
4-4. OPERATION OF CENTRAL-PROCESSOR CHANNELS

A signal from the instruction channel sets the operand-request FF for an automatic operand call. The operand channel also produces operand calls during the unconditional-transfer instruction (06), the automatic interrupt, and for instructions requiring multi-operand operation.

4-5. Automatic Operand Call

The automatic-operand-call operation is present





* PRESENT AS SHOWN ONLY IF THERE ARE NO HIGHER PRIORITY MEMORY REQUESTS

6834

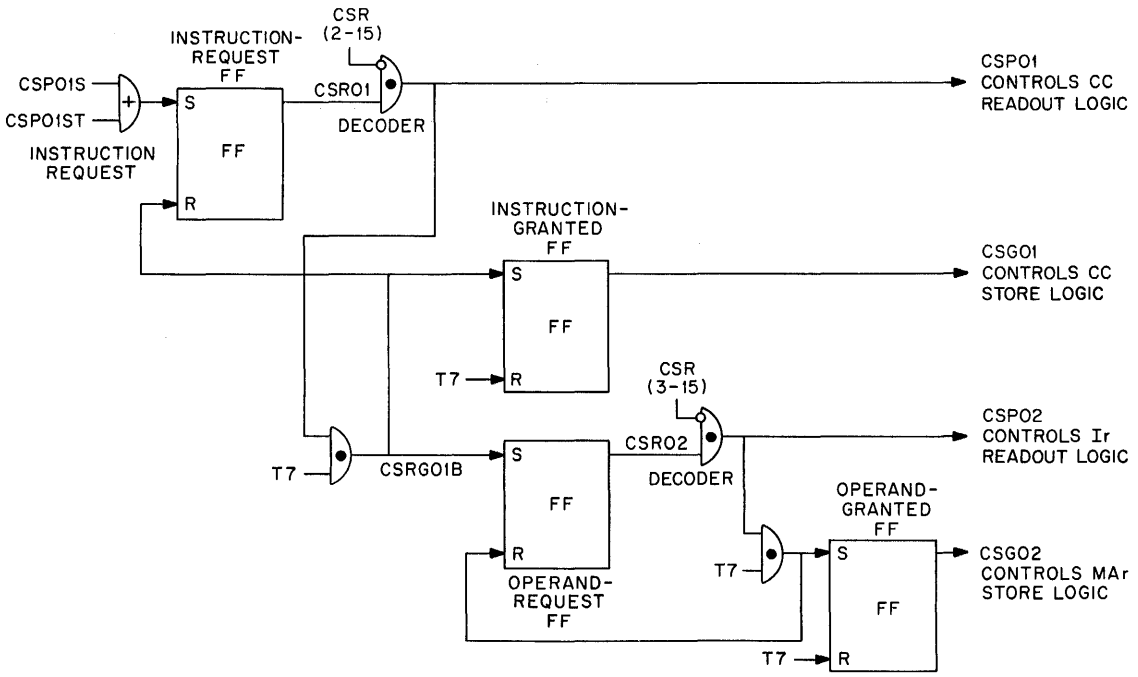
Figure 4-1. Typical Priority Channel, Detailed Block Diagram

for all instructions and requests memory either to read an operand from memory or write data into memory. The channel outputs control the modification of the operand address (m) and the storing of the modified operand (m') address (figure 4-2).

4-6. Transfer Instructions

The two transfer instructions are the unconditional-transfer instruction and the test-ARi instruction. Either instruction replaces the contents of the control counter (CC) with the modified-operand address (m') of the instruction currently in the computer-control register (CCr). The unconditional-transfer instructions (06) causes the transfer whenever it is present; the test-ARi instruction causes the transfer only when a specified sense flip-flop is set.

Signal CV490 initiates the unconditional-transfer instruction, and signal CV600 (figure 4-3) combined with CCCCT (representing a sense-conditional-transfer-FF output) initiates the test-ARi transfer. Either of these signals sets the instruction-granted flip-flop for another minor cycle and inhibits the setting of the operand-granted flip-flop. Thus, the modified operand address (m') is read into the CC instead of MAr.



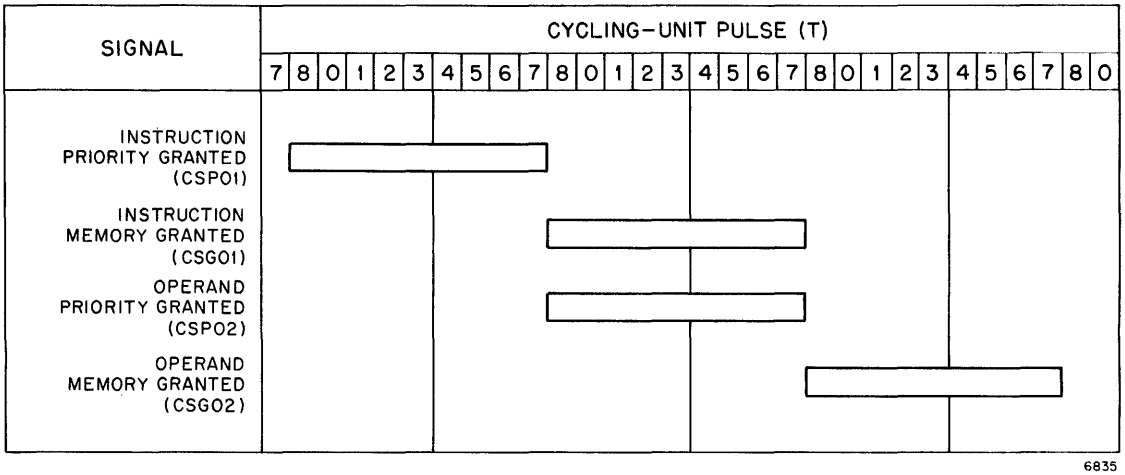


Figure 4-2. Automatic Operand Call, Detailed Block Diagram

4-7. Automatic Interrupt

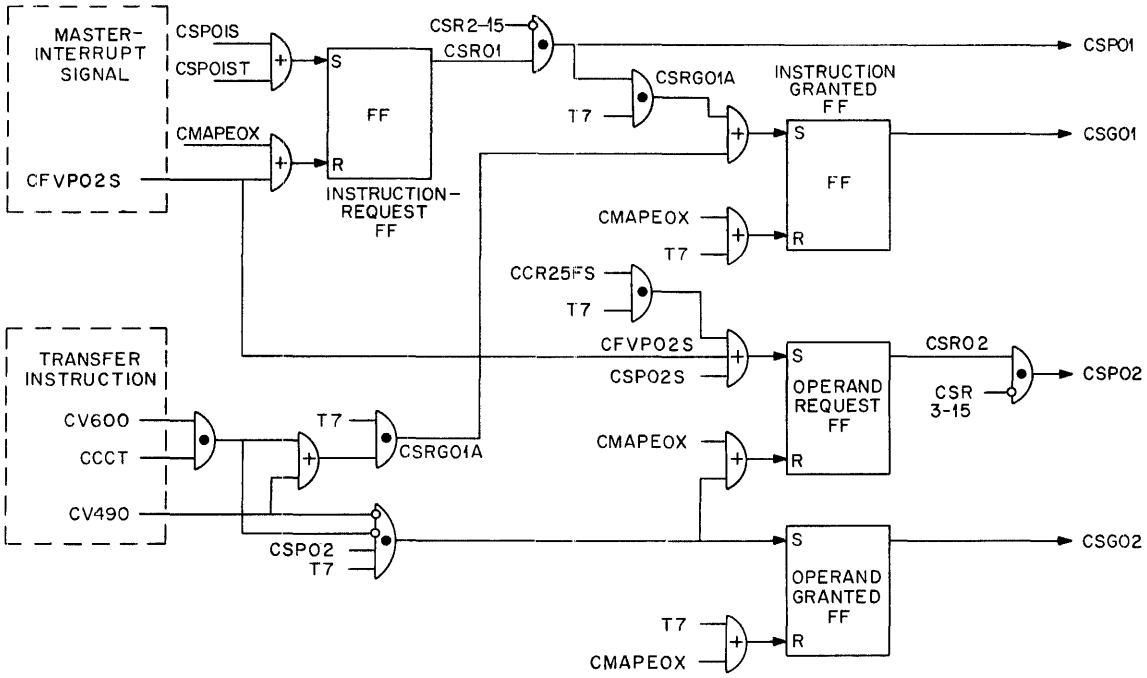
Signal CFVP02S, produced during an automatic interrupt (figure 4-3), resets the instruction-request FF and sets the operand-request FF. The uses of CFVP02S are explained in section 11.

4-8. Multioperand Calls

In addition to its functions associated with the instruction-call operation, the operand-call channel is used for multioperand operations and for successive operand calls when indirect addressing or field selection is used. When an extra operand is requested, the operand-request flip-flop is set, and the contents of MAr are read out. The signal for the multiprecision operand call is CSP02S, and the signal for indirect addressing is CCR25FS (figure 4-3). Either signal sets the operand-request flip-flop at T7.

4-9. Priority Inhibits

The production of the automatic-operand-call signal (CSP02) is inhibited during the first program count



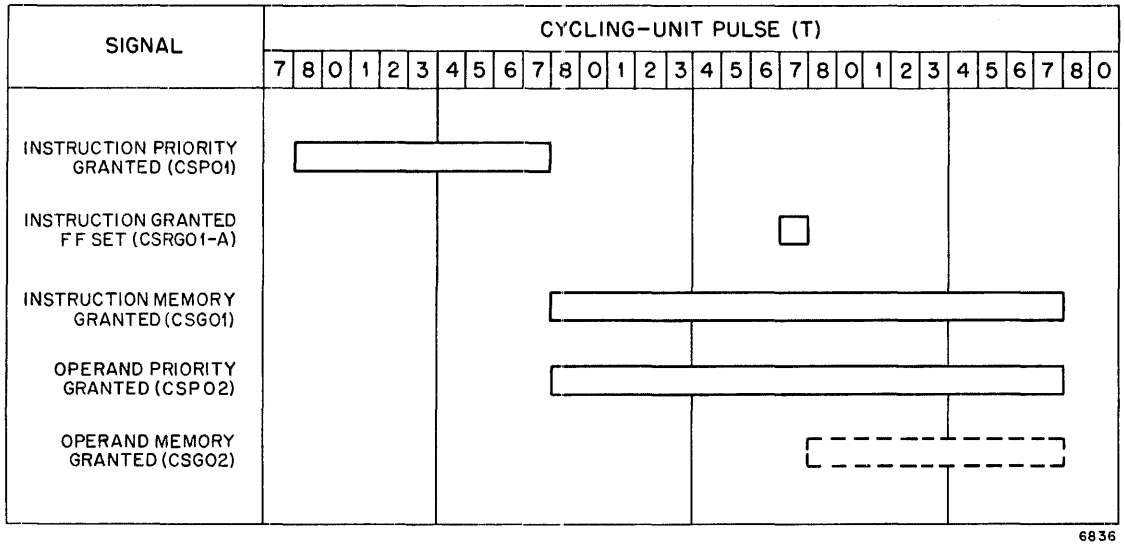


Figure 4-3. Transfer Instructions, Detailed Block Diagram

(PC0) of the store-SAr instructions and the store-TCWr instruction; during the recomplement cycle (CVRECP); and when the computer stop signal (CCCSTP) is present.

4-10. Priority Manual Reset

All priority-request flip-flops are reset by the CMAPEOX signal which is produced by the general-clear signal that originates from the maintenance console.

Section 5
COMPUTER-CONTROL REGISTER

5-1. INTRODUCTION

The computer-control register (CCr) is used for manipulating information within the central processor. The computer-control register operates during all of the central-processor instructions, often repeating operations, or performing different operations, during the same instruction.

The CCr consists of 28 flip-flops that are set by 0-information bits. The bits are either shifted from one flip-flop to another by function signals, or held by the flip-flops. Outputs from the CCr are taken from different flip-flops according to the operation. For example, the main adder uses 4-bit parallel outputs from CCr FF's 1, 2, 3, and 4; and the memory-write register uses the outputs from FF's 5, 9, 13, and 17.

Shifting within the CCr is either a right decimal shift or a right binary shift. Either decimal or binary shifting is made circular by transferring bits from the least-significant-bit positions to the most-significant-bit positions. Shifting is used to load the CCr four bits at a time, to rearrange bit positions within CCr, or to produce an output, four bits at a time.

5-2. GENERAL OPERATION

Figure 5-1 shows the formats of the different inputs handled by the CCr, and shows the patterns of decimal and binary shifting within the CCr. Each square in the figure represents a bit, and the numbers in the squares indicate the bit positions in a UNIVAC III word. The flip-flops of the CCr are numbered to correspond to the order of the bit positions of a word received from memory.

Paths for shifting, shown in figure 5-1, are the decimal-shift path (solid lines) and the binary-shift path (broken lines). Decimal shifting is the transfer of the four bits of each digit to the next less-significant-digit position. Digits (defined as four bits each) are circularly shifted by transferring the digit in the least-significant-digit (LSD) position (1—4) to the most-significant-digit (MSD) position (25—28). Binary shifting is the transfer of each bit to the next less-significant-bit (LSB) position in the CCr. Circular-binary shifting is done by transferring the bit from the LSB position (FF 1) to the most-significant-bit (MSB) position used for binary shifting (FF 25).

An example of information shifting within the CCr is the transfer of a data word to the main adder as shown in figure 5-2. The data word enters the CCr from memory, 27 bits in parallel. Bits, in consecutively numbered bit positions of a word from memory, enter correspondingly numbered flip-flops in the CCr at T3. The flip-flop outputs are available the next pulse time (T4) as inputs to the main adder and as inputs for shifting within the CCr. The data in the CCr shift one digit position each pulse time. All transfers to the adder are from the least-significant-digit position of the CCr: FF's 1, 2, 3, and 4. The first digit is transferred at T4. At T5, after the first decimal shift, the second digit (bits 5, 6, 7, and 8) is transferred to the adder. The shifting and transfer to the adder continues until the MSD, bits 25 (sign), 26 and 27 (check bits), and bit 28 enter the adder at T1.

5-3. LOGIC OF COMPUTER-CONTROL REGISTER

The CCr is explained in detail under the following three major headings: register functions, register inputs, and register outputs. The register functions are explained before register inputs because it is necessary to know the shifting functions before the register-input process can be understood. Under subheadings of register functions, shifts and zero-jamming are explained; under subheadings of register inputs, inputs from memory, typewriter, real-time clock, accumulator register, synchronizer-control register, and from interrupt FF's are explained; and under subheadings of register outputs, outputs that are related to other logic sections are explained.

5-4. Register Functions

Information in the CCr is manipulated by shifting or entering zeros. The functions performed within

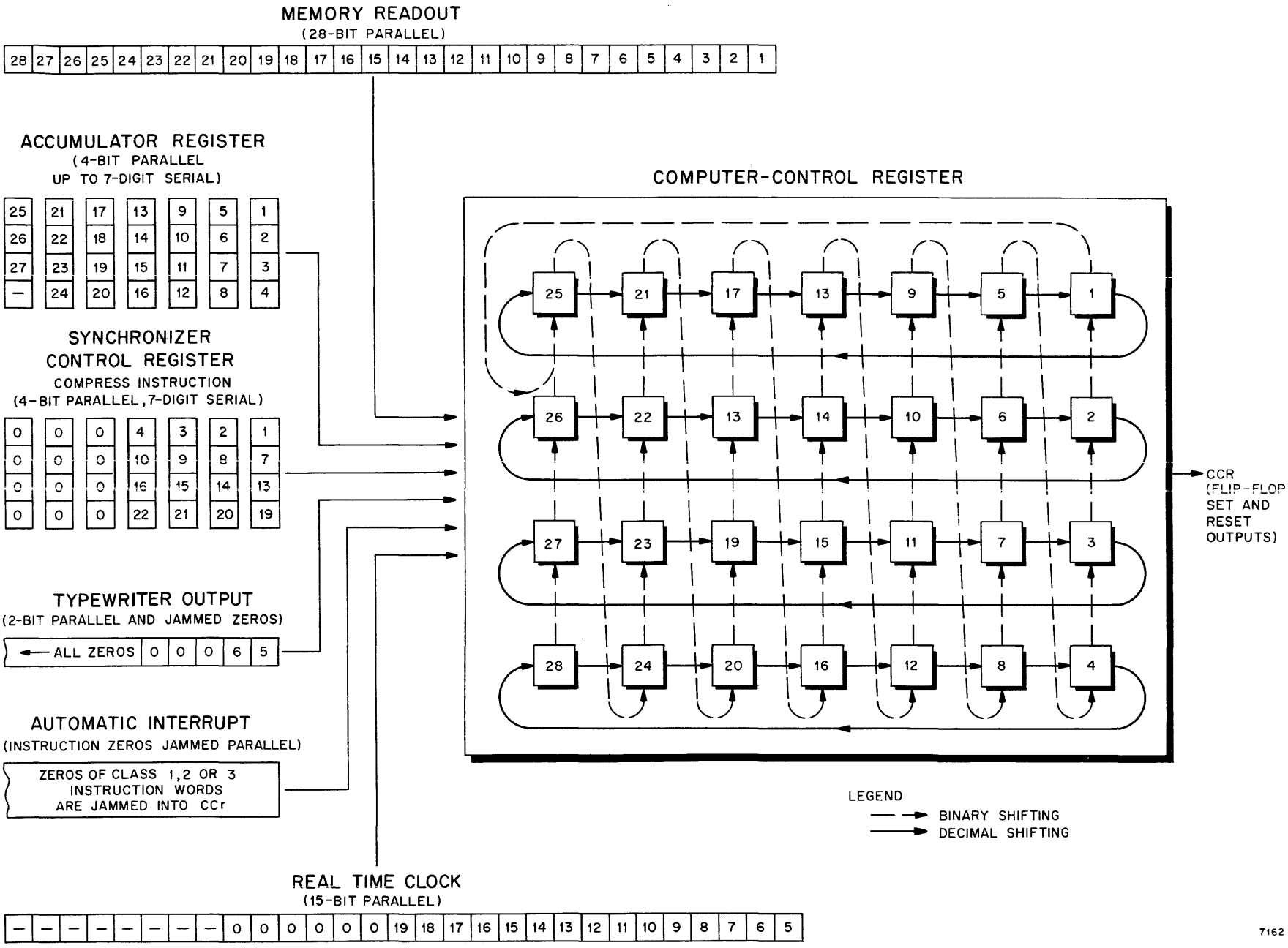


Figure 5-1. Computer-Control Register, Functional Block Diagram

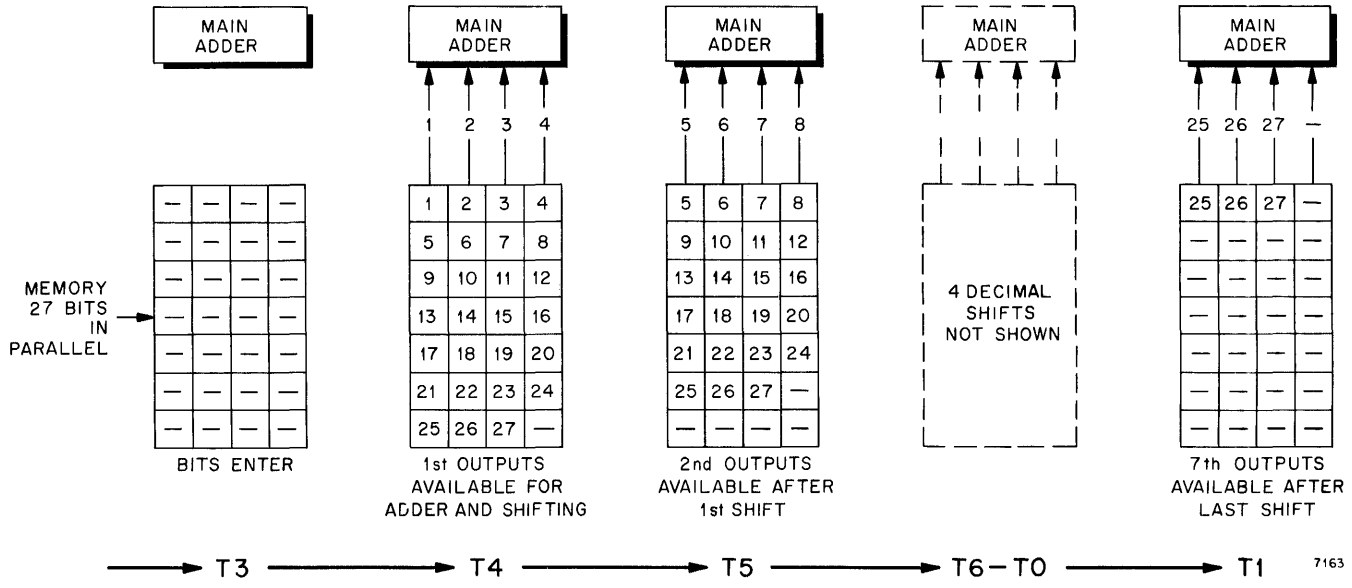


Figure 5-2. Example of the Transfer of a Word from Memory Through the Computer-Control Register to Main Adder, Functional Block Diagram

the CCr and the function signals that control the functions are as follows:

FS	Function
143	Resets FF's
144	Jams binary zeros
146	Controls decimal shifts
261	Controls circular-decimal shifts
249	Controls binary shifts
257	Controls circular-binary shifts

5-5. RESETTING. Each flip-flop in the CCr is reset by FS 143 during shifting, loading, or clearing operations unless the reset is overridden by a set-input signal (figure 5-3). Function signal 143 is present during all CCr shift functions, during an input from memory, during entry of interrupt-instruction words, and for general clear.

5-6. JAMMING BINARY ZEROS. All CCr flip-flops are jammed to binary 0's by FS 144 before information is entered at T3 during instructions that require zeros to be shifted ahead of information inputs, or when information in CCr is to be erased (figure 5-4). Function signal 144 is produced by CV-lines for program counts during which the CCr is to be cleared, or by the skip signal from the console (CDVRSK).

5-7. DECIMAL SHIFTING AND CIRCULAR-DECIMAL SHIFTING. Decimal shifting within the CCr is done as shown in figure 5-3. The four bits of each digit are shifted in parallel to the four flip-flops of the next LSD position when the input gates are alerted by FS 146.

The gates for shifting 0 bits into each flip-flop are alerted by FS 146, but FS 143 resets the flip-flops to 1 if it is not overridden by a set input. During each pulse time that FS 146 is present, all digits within the CCr shift one-digit position to the right. Circular-decimal shifting occurs only if FS 261 is present at the same time FS 146 is present. Binary 0's are entered into the MSD position during decimal shifting except during entry of information from an Ar or during circular-decimal shifting. The shifting of 0's into FF's 1 and 2 by FS 146 is inhibited by FS 274 during a read typewriter instruction so that typewriter information can enter these flip-flops during this instruction.

Function signal 146 is produced by either of two inputs, a set output from the FS-146 flip-flop, or by shift signal CCSCE that is used during a left alphanumeric-shift instruction. Function signal-146 FF is set when information in the CCr is to be decimally shifted and is reset when shifting is completed. Signal CSP021 sets the FS-146 FF when the m address of the instruction word is transferred from memory into the memory adder. During the multiply and divide instruction, FS-146 FF is set by CVMD1, representing the multiply and divide add cycles. (Refer to section 10.) All other FS-146-set signals are CV lines for instruction which require shifting of information into, out of, or within CCr during specific program counts. During a shift-circular-binary instruction (CVRG44), FS-146 FF is set during PC1 and is not set again until the end of the shift when CCFSCFD is generated.

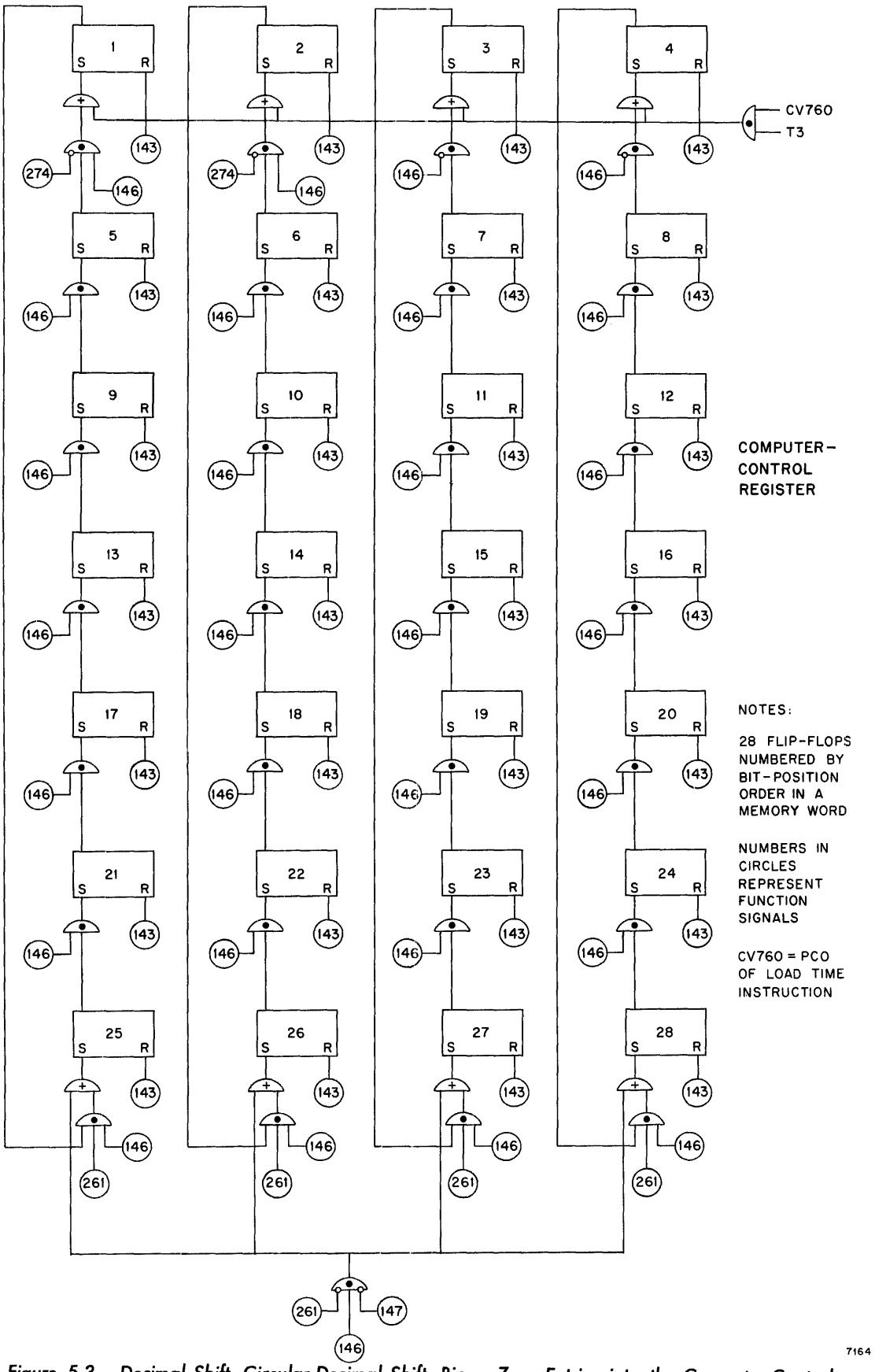


Figure 5-3. Decimal Shift, Circular-Decimal Shift, Binary-Zero Entries into the Computer-Control Register, Detailed Block Diagram

The FS-146 FF is set at T3 and FS 146 is normally available at the beginning of a program count (T4) for all instructions. During the expand (71) or compress (72) instructions, however, the FS-146 FF is set at different times to shift only certain digits. Function-signal-146 FF is normally reset at T1, but it is reset at different pulse times during expand, compress, shift, and modify-Ir instructions. During shift instructions, FS-146 FF is reset by signals CCFSCFA, CCFSCFB, or CCFSCFC when shifting within CCr is completed.

Function signal 261 is produced either during a multiply instruction, a divide instruction, or by an instruction-inhibit signal (CVRINH), or by a recomplement signal (CVRECPB).

5-8. BINARY SHIFTING AND CIRCULAR-BINARY SHIFTING. Binary shifting within the CCr is done as shown in figure 5-4. The input gates for binary shifting are alerted by FS 249. Each pulse time that FS 249 is present, information is shifted 1-bit position to the right. Function signal 143 is present during any shifting to reset flip-flops not set by a zero input.

Circular-binary shifting occurs only when FS 249, for binary shifting and FS 257, for circular-binary shifting, are present. Function signal 249 is present at the following times:

During the shift-binary-circular instruction (44) from the time that the CVBRS FF is set until the shift-count equals 0 (CCFSBE0);

At T6 during PC1 of the Ir-modify-and-compare instruction (CV531); (This use of FS 249 is necessary in order to place bit position 10 of the index-register-modification control word into FF1 of the CCr);

From T4 through T7 of PC1, PC3, and PC6 of the compress instruction (72); and

For two pulse times (T1 and T2) during alphanumeric-shift instructions when the shift count is an odd number (1 or 3); (This use of FS 249 is necessary because odd shift counts cannot be represented by decimal shifts alone; they require two extra binary shifts.)

Function signal 249 also sets FF 28 of the CCr in order to shift a 0 from bit position 28.

Function signal 257 is produced at T4 during a shift-binary-circular instruction and is present until the shift count is reduced to 0 (CCFSBE0).

5-9. Register Inputs

The input from each information source is gated into the CCr by a different function signal. Information bits are entered into the CCr, 27 bits in parallel, or are shifted into the register, 4 bits in parallel. Zero-information inputs set the flip-flops and, in some operations, binary or decimal XS-3 0's are entered before or after the information bits to ensure that all unused bit positions contain binary or decimal 0's. Information in the CCr is binary or decimally shifted, or is held until the flip-flops are reset before the next input.

5-10. MEMORY INPUT. One word from memory is entered into the CCr, 27 bits in parallel, into FF's 1—27 (figure 5-5). Memory input gates to the CCr, except the FF-28 input gate, are alerted by FS 141 and are enabled by 0 bits from memory. The FF-28 input gate is enabled by a 1 bit.

Function signal 141 is present at T3 when the memory granted signals for instructions or operands (CSG01 or CSG02) are present. Function signal 141 is inhibited during interrupts because the interrupt instruction is not read from memory but rather is produced by the central processor. Function signal 141 is also inhibited if a new operand from memory is not required during an operand call as occurs during PC2 of the Ir-modify-and-compare instruction (53), during PC2 and PC6 of the expand instruction (71), during program interrupt (CFVEA), and during the load-time instruction (CV760D).

5-11. TYPEWRITER INPUT. One alphanumeric character (six bits) from the typewriter enters the central processor during a read-typewriter instruction. This input is divided between the main adder and the CCr. The four LSB's of the character enter directly into the main adder, and the two MSB's CCK05 and CCK06, enter CCr, FF's 1 and 2, in parallel at T4 (figure 5-5). These MSB's, 5 and 6, are entered into the CCr to delay their entry into the main adder until T5. Before information is entered, binary 0's are jammed into the CCr by FS 144 and are decimally shifted during the read-typewriter instruction. At T4, the shifting of 0's into FF's 1 and 2 is inhibited so that the typewriter information can enter the CCr FF's. Inputs from the typewriter to the CCr, FF's 1 and 2, are entered from gates alerted by FS 274 and enabled by 0's from the typewriter. The function of the typewriter is described in section 14 of this manual.

5-12. Real-Time-Clock Input

The 19 bits from the real-time clock consists of 7 bits for the hour code, 8 bits for the minute code, and 4 bits for 0.1-minute code. Before the information is entered into the CCr, FF's 1, 2, 3, and 4 are jammed to 0 at T3 by signal CV760 (figure 5-3). The four bits of the 0.1-minute code, CCRCK(01—04), enter directly into the main adder, and the remaining 15 bits enter the FF's 1 through 15 of the CCr as shown in figure 5-5. In addition to the 15 bits

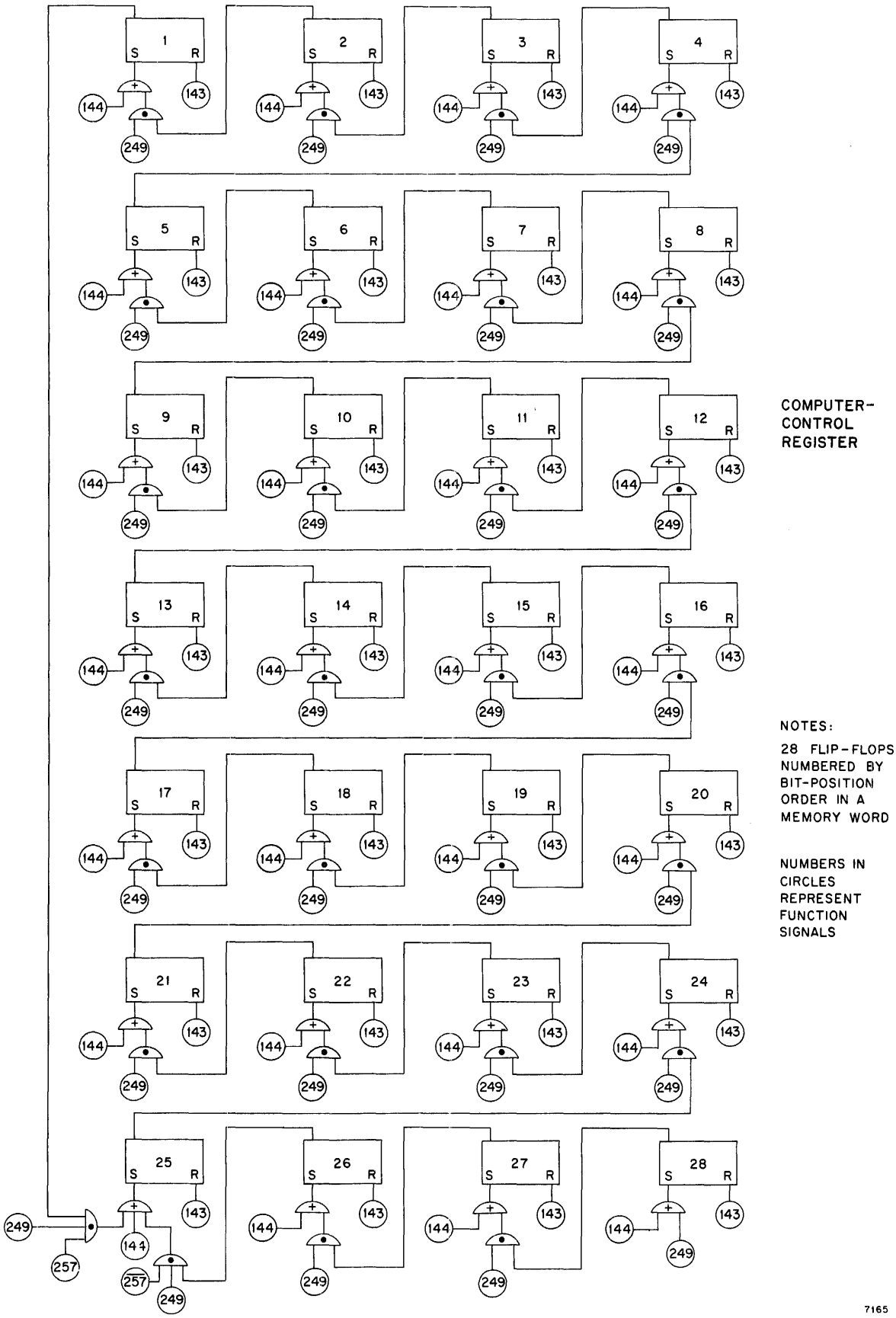


Figure 5-4. Binary Shift, Circular-Binary Shift, and Jammed Zeros into the Computer-Control Register, Detailed Block Diagram

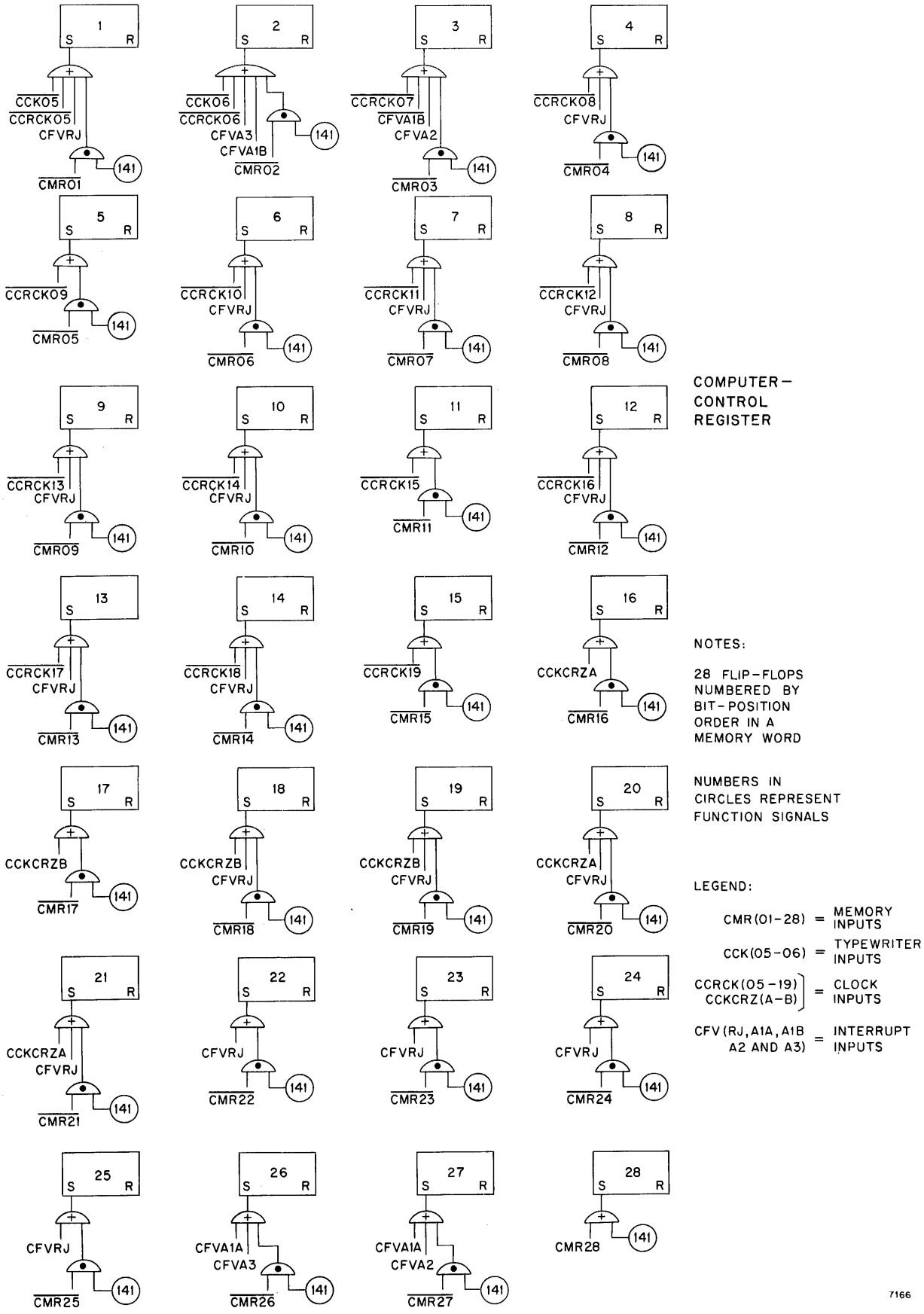


Figure 5-5. Parallel Inputs from Memory, Real-Time Clock, and Typewriter into the Computer-Control Register, Detailed Block Diagram

entering the CCr, 6 binary 0's are inserted in the next-most-significant FF's, 16 through 21, of the CCr by signals CCKCRZA and CCKCRZB. The input from the real-time clock enters the CCr from gates alerted by FS 275. The gates are enabled by 0 bits from the real-time clock. The information is transferred grom the CCr to the main adder, by decimal-shifting of the information to FF's 1 through 4 of the CCr, four bits each pulse time from T5 through T0. The digit including the sign (0) is transferred at T1. The function of the real-time clock is described in section 14.

5-13. Interrupt Inputs

During the execution of the program interrupt, an instruction word from the interrupt section is jammed into the CCr after it is cleared by FS 143. This word contains a transfer-and-return instruction (07).

Class 1, class 2, and class 3 interrupts jam different addresses into the CCr. The locations specified by these addresses contain instructions which transfer program control to different subroutines for different classes of interrupts. Interrupt signal CFVRJ is present for all three classes to set flip-flops in the CCr for bit positions that have 0's common to all three instruction words. Interrupt signals CFVA1A and CFVA1B set additional CCr FF's (to 0's) for a class 1 instruction word; CFVA2 sets additional CCr FF's (to 0's) for a class 2 instruction word; and CFVA3 sets additional CCr FF's (to 0's) for a class 3 instruction word, as shown in table 5-1.

5-14. Accumulator-Register Inputs

The Ar inputs are gated into the CCr one digit at a time. They then are decimally shifted before entry of the next digit as shown in figure 5-6. One decimal digit, signals CRB(01—04), is gated from the Ar into the CCr each pulse time from T4 through T1. Function signal 147 gates the digits into the most significant CCr FF's (25, 26, 27, and 28). The digits then are decimally shifted by FS 146 and FS 143 until the CCr is loaded. During the first program count of the multiplication or division instruction, CV301 or CV311, decimal XS-3 0's are entered into the CCr at T3 by setting FF's 28 and 27, and resetting FF's 26 and 25 (figure 5-6).

Function signal 147, produced by CV lines, gates the Ar outputs into the CCr from T4 through T3 during shift and expand instructions. During single- or double-precision shift instructions, FS 147 is produced for only one program count. However, for double-precision shifts of less than one word (less than four alphanumeric characters or less than six decimal digits), FS 147 is produced for either three or four program counts.

5-15. Synchronizer-Control-Register Inputs

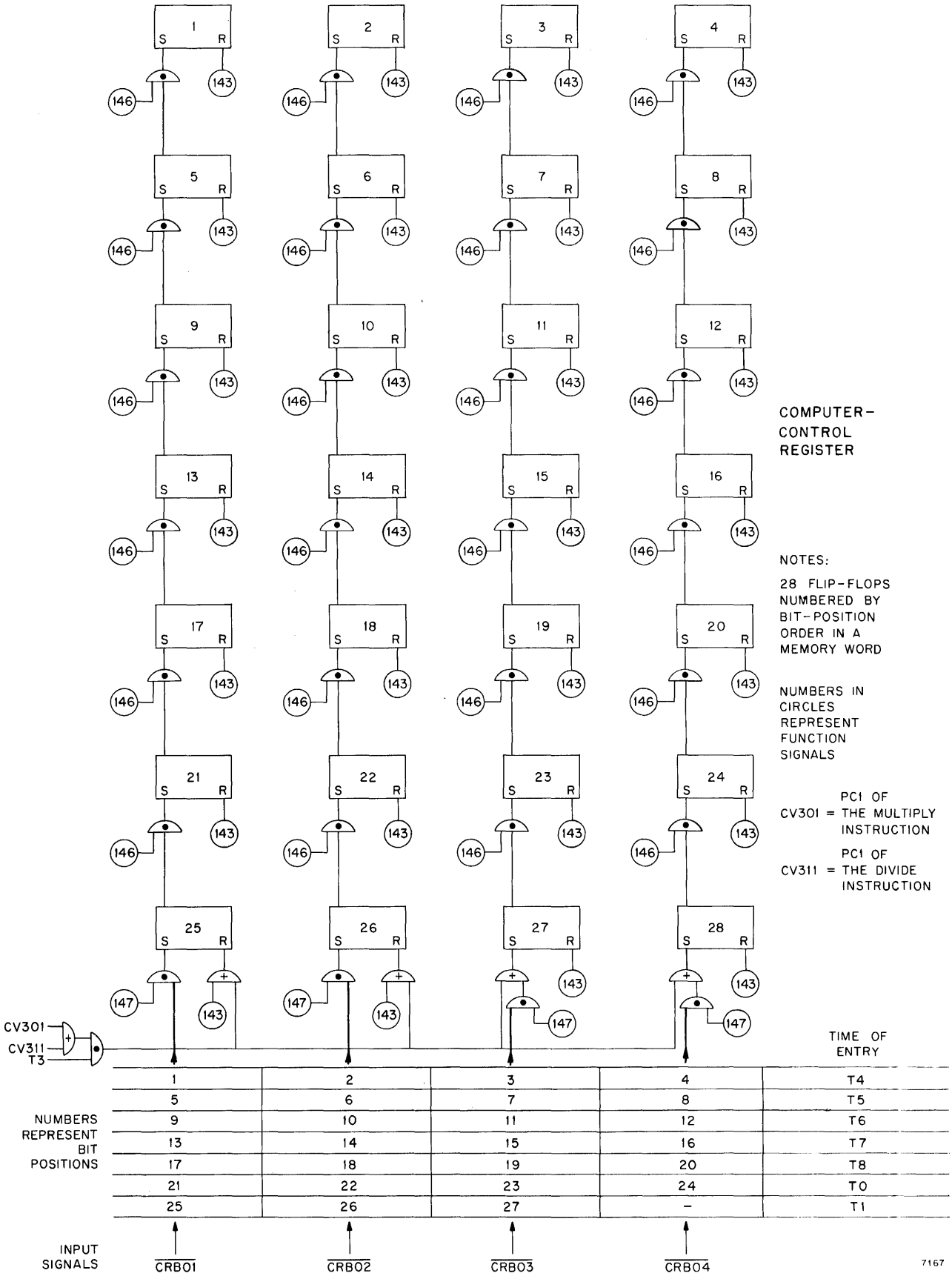
The SCr inputs are used during the compression of 6-bit alphanumeric characters into 4-bit decimal digits. The numeric portion of the four alphanumeric characters are transferred in parallel from the SCr to the CCr. The four bits representing the numeric portion of each of the four characters, are transferred to the CCr in series, 1 bit each pulse time from T4 through T7. Function signal 276 gates 1-bit outputs from the SCr, CCSCK(1—4), into the CCr instead of the 0 bits that are gated from the other input sources (figure 5-7).

Before entry of digits from the SCr, binary 0's are gated into the CCr by FS 144 and T3 (figure 5-4). These 0's then are gated by FS 249 and FS 143 for a binary shift during entry of the SCr digits. An output from any one of the four input gates, produced by a 1 bit from the SCr, inhibits a binary

Table 5-1. Interrupt Instruction Words

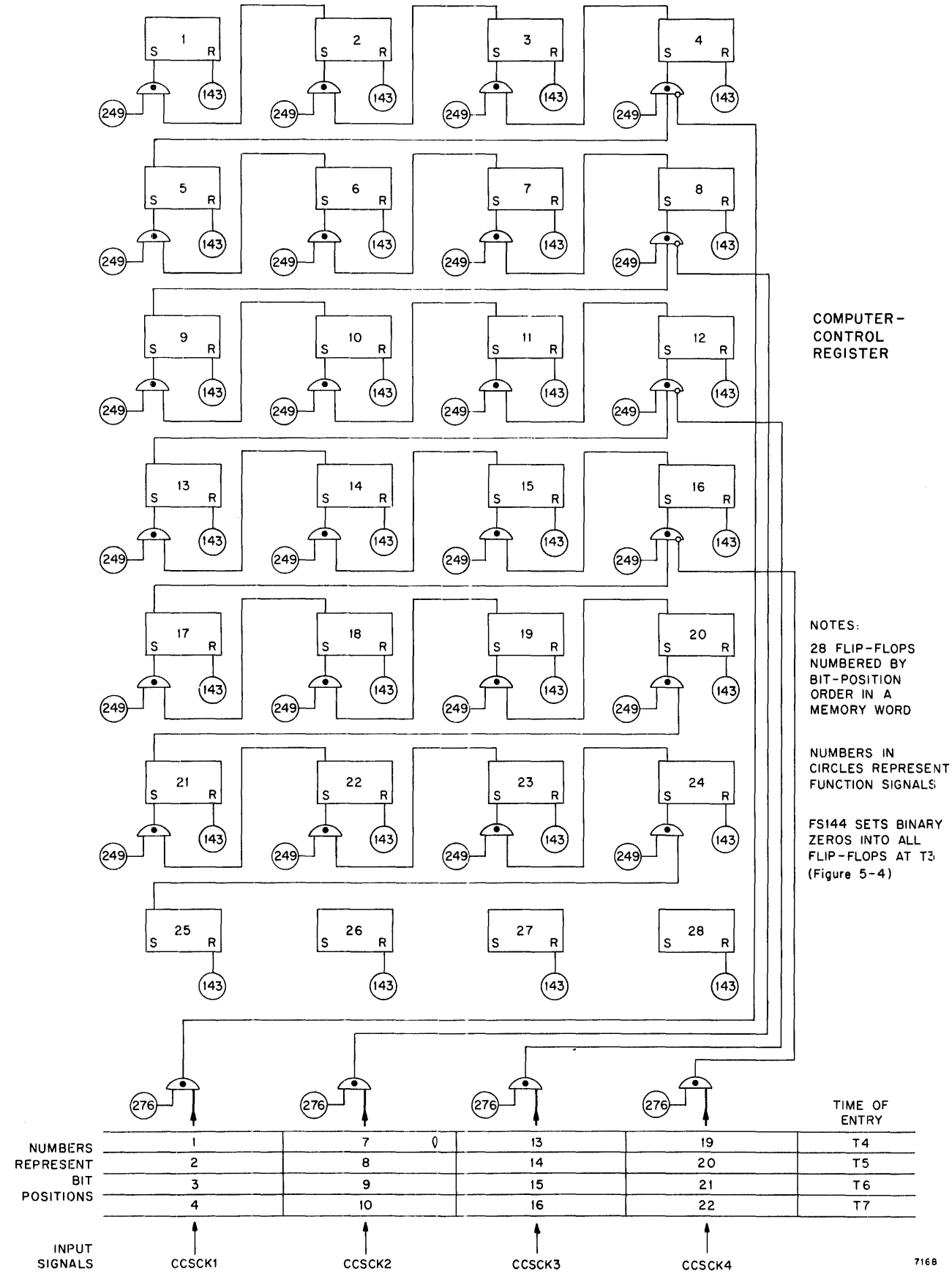
CCr FF's	Common 0's	Class 1	Class 2	Class 3
	CFVRJ	CFVA1A, CFVA1B, CFVRJ	CFVA2, CFVRJ	CFVA3, CFVRJ
1	0	0	0	0
2	1	0*	1	0*
3	1	0*	0*	1
4	0	0	0	0
5	1	1	1	1
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	1	1	1	1
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	1	1	1	1
16	1	1	1	1
17	1	1	1	1
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0
26	1	0*	1	0*
27	1	0*	0*	1

*Indicates 0's that are in addition to the common 0's entered by signal CFVRJ.



1	2	3	4	T4
5	6	7	8	T5
9	10	11	12	T6
13	14	15	16	T7
17	18	19	20	T8
21	22	23	24	T0
25	26	27	–	T1

Figure 5-6. Accumulator Register Loading Computer-Control Register and XS-3 Zeros Entered During Multiply and Divide, Detailed Block Diagram



NUMBERS REPRESENT BIT POSITIONS	CCSCK1	CCSCK2	CCSCK3	CCSCK4	TIME OF ENTRY
	1	7	13	19	T4
	2	8	14	20	T5
	3	9	15	21	T6
	4	10	16	22	T7

Figure 5-7. Synchronizer-Control Register Loading the Computer-Control Register, Detailed Block Diagram

shift of 0's into the CCr: FF's 4, 8, 12 or 16. The zeros, first gated into the CCr are loaded into FF's 1 through 21 except where 1 bits are in the input from SCr.

Function signal 276 is present from T4 through T7 during program counts 1, 3, and 6 of the compress instruction (72), unless inhibited by the instruction-inhibit signal.

5-16. REGISTER OUTPUTS

Outputs from the CCr are taken from different flip-flops according to the type of instruction. All logic circuits using CCr outputs are listed in table 5-2, with a listing of the CCr-FF outputs used by each.

Table 5-3 is a summary of function signals which control transfer operations. Table 5-4 is a summary of input sources and word formation.

Table 5-2. Computer-Control-Register Outputs

From CCr FF's	To Logic Circuit	Format	Remarks
CCR01—CCR04	Main adder	4-bit parallel up to 7-digit serial	Requires decimal shift within the CCr
CCR01—CCR04	Memory adder	4-bit parallel up to 7-digit serial	Requires decimal shift within the CCr
CCR22—CCR23	Modulo-3 checker in the arithmetic section or check-bit generator in the memory adder	2-bit parallel	During arithmetic and initiate I-O instructions
CCR05, CCR09, CCR13, CCR17	Memory-write register	4-bit parallel up to 4-digit serial	Requires binary shift within the CCr only during 71 instruction
CCR11—CCR14	AR-select FF's	4-bit parallel for 1 pulse time	Present only after recomplementing
CCR15—CCR21	Instruction register	6-bit parallel for 1 pulse time	Present during all instructions
CCR25	Sign circuits	1-bit transfer	
CCR01—CCR27	Suppress controls	27-bit parallel	Present only during 73 instruction

Table 5-3. Summary of Function Signals, Which Control Transfer Operations to and Within Computer-Control-Register Flip-Flops, and Interrupt Inputs

Function Signal	Symbol	Description	CCr Flip-Flops
141	Memory $\longrightarrow$ CCr	Enters one word from memory to CCr in parallel	1—27
143	CLEAR CCr	Resets all CCr flip-flops	1—28
144	BINARY ZEROS $\longrightarrow$ CCr	Sets all CCr flip-flops to 0	1—28
146	10^{-1} CCr $\longrightarrow$ CCr	Right-decimal shifts digits to the next LSD position in CCr	1—28
147	Ar $\longrightarrow$ CCr	Enters Ar outputs into CCr	25—28
249	2^{-1} CCr $\longrightarrow$ CCr	Right-binary shifts bits to next LSB position in CCr	1—28
257	$CCr_1 \longrightarrow CCr_{25}$	Right-circular-binary shifts from right end of register to reenter CCr flip-flop 25	25

Table 5-3. Summary of Function Signals, Which Control Transfer Operations To and Within Computer-Control-Register Flip-Flops, and Interrupt Inputs (cont)

Function Signal	Symbol	Description	CCr Flip-Flops
261	$CCr_{1-4} \longrightarrow CCr_{25-28}$	Right-circular-decimal shifts from LSD to reenter MSD of CCr	25—28
271	Typewriter ⟶ CCr	Enters 2 most significant bits of 6-bit alphanumeric character from typewriter into CCr	1—2
275*	Real-Time Clock ⟶ CCr and Main Adder	Enters 15 MSB's of 19 bits from real-time clock and inserts 0's in the next 6-MSB positions	1—21
276*	SCr ⟶ CCr	Enters output of SCr buffers into CCr	4, 8, 12, and 16
†	Interrupt ⟶ CCr	Enters common 0's of interrupt-instruction words	1, 4, 6—10, 12—14, 18—25
†, ‡	Class 1 Interrupt ⟶ CCr	Enters class 2 interrupt 0's	2, 3, 26, and 27
†, ‡	Class 2 Interrupt ⟶ CCr	Enters class 2 interrupt 0's	3 and 27
†, ‡	Class 3 Interrupt ⟶ CCr	Enters class 3 interrupt 0's	2 and 26

* These functions signals are explained in section 13.
† Initiated by an automatic interrupt.
‡ Class interrupt signal always occurs in conjunction with CFVRJ.

Table 5-4. Summary of Input Sources and Word Formation

Input Source	Inputs Controlled By	Input Format	Information Loaded in FF's	Check-Bit Source	Sign Source
Memory	FS 141	Parallel	1—27	Memory	Memory
Typewriter	FS 274	Parallel	1—2 (0 bits gated in 7—24)	Main adder	CCr, Jammed 0
Real-time clock	FS 275	Parallel	1—15 (0 bits gated in 16—21)	Main adder	CCr, Jammed 0
Interrupt section	Program-interrupt FF	Parallel	1—27	Interrupt section	Interrupt section
Accumulator register	FS 147	Four bits of digits in parallel; seven digits in series	1—27	Ar	Ar
Synchronizer-control register	FS 276	Four bits of the digits in series; four digits in parallel	1—16 (0 bits gated in 17—21)	Main adder	Sign circuits (for the most significant word only)

Section 6
STORAGE-ADDRESS SECTION

6-1. INTRODUCTION

The storage-address section consists of 30 storage-address registers (SAr's) and four tape-control word registers (TCWr's). The 30 SAr's consist of 15 memory-address counters MAC(1 through 15) and 15 index registers Ir(1 through 15). The 15 MAC's consist of a control counter (CC), which contains the address of the last instruction read from memory; a memory-address register (MAr), which contains the most recent m' address; and 13 I-O memory-address counters MAC(3 through 15), which contain addresses for input-output operations (figure 6-1).

Corresponding numbered MAC's and Ir's are contained in the same register and the four TCWr's occupy one register each. Each register consists of four recirculating delay loops with a capacity of nine bits per loop. A complete register contains 36 bits. Registers that contain a MAC and an Ir use 30 of the 36 bits. Registers that contain a TCWr use 24 of the 36 bits.

When information is read from registers which contain a MAC and an Ir, the time at which the register outputs are sampled determines whether the MAC or the Ir portion of a register is read. If the outputs of a register are sampled from T0 through T3, the MAC portion is read, and if the outputs are sampled from T4 through T7, the Ir portion is read. Information is written into the MAC's from T8 through T2, and into the Ir's from T3 through T6. The one pulse-

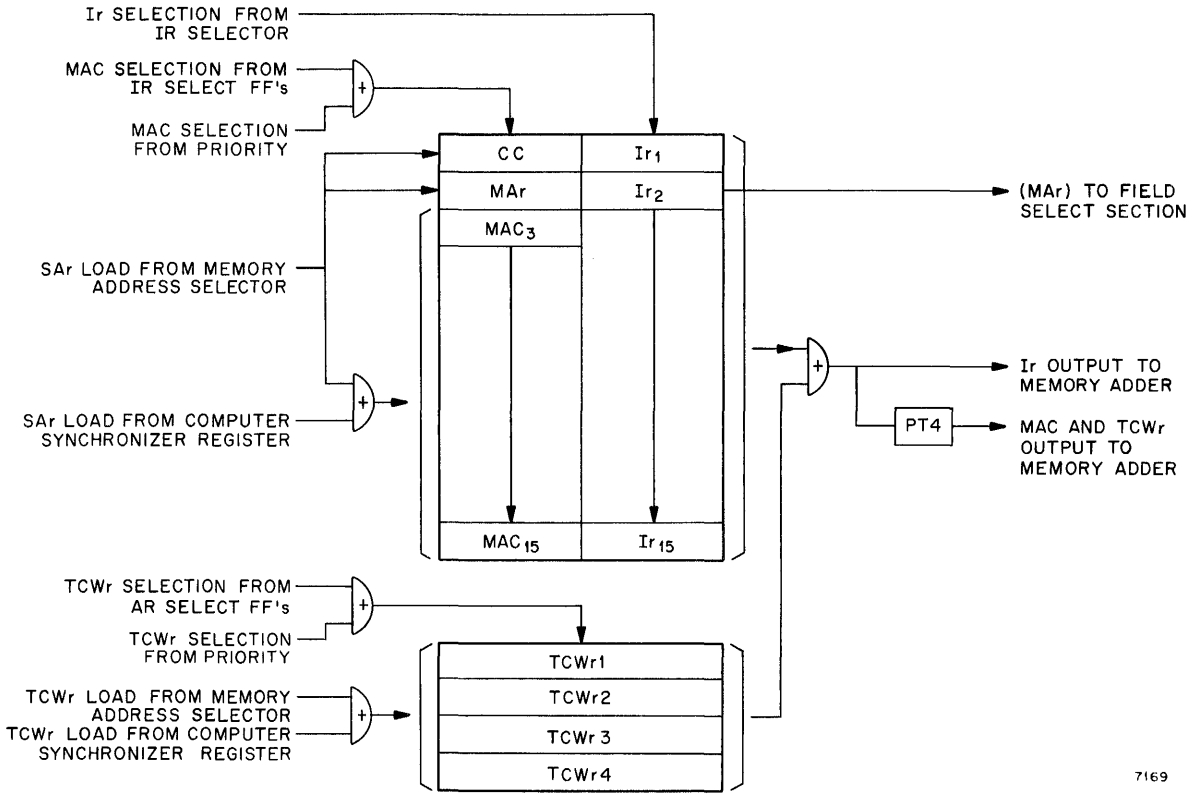


Figure 6-1. Storage-Address Section, Functional Block Diagram

time delay between the input and the output of the registers is produced by the pulseformer in the delay loop. Information is read into the registers that contain a TCWr so that the outputs are available from T0 through T5 (refer to table 6-3). The input and the output timing from the MAC's, Ir's, and TCWr's is shown in table 6-1.

The Ir's are used primarily to modify the m address or L address of an instruction or control word in the CCr. The secondary purpose is to provide program-control information through index-register modification and comparision. The MAC's are used for all central-processor instruction calls, central-processor operand calls, and all input-output memory requests except when tape control words are used. The TCWr's provide the address and control information when the tape control words are used.

Table 6-1. Input and Output Timing of Memory-Address Counter, Index Register, Tape-Control-Word Register

Time	MAC		Ir		TCWr	
	Input	Output	Input	Output	Input	Output
T0	5—8	1—4	—	—	5—8	1—4
T1	9—12	5—8	—	—	9—12	5—8
T2	13—16	9—12	—	—	13—16	9—12
T3	—	13—16	1—4	—	17—20	13—16
T4	—	—	5—8	1—4	21—24	17—20
T5	—	—	9—12	5—8	—	21—24
T6	—	—	13—15	9—12	—	—
T7	—	—	—	13—15	—	—
T8	1—4	—	—	—	1—4	—

6-2. Selection of SAr's and TCWr's

The Ir outputs are selected by the decoded outputs of the index-register selector, and the MAC outputs are normally selected by the priority-granted signal (CSP) from the priority section. The MAC outputs are also selected by the decoded output of the index-register selector during the 04 and 07 instructions, which store the contents of a MAC.

There is one MAC assigned for each CSP signal, with CSP01 selecting the CC, and CSP02 selecting the MAr. When tape control words are used, the MAC's used with the UNISERVO III tape-unit synchronizers (MAC3, MAC4, MAC14, and MAC15) are selected only during the first tape-data call when the gather-write/scatter-read control word is read from the memory. Subsequent tape-data calls, which produce CSP03, CSP04, CSP14, and CSP15, select TCWr's 1 through 4, respectively. The TCWr's are also selected by the output of the AR-select FF's during a store-TCWr instruction (05).

6-3. Inputs to SAr's and TCWr's

The two registers which contain the CC and IR1, MAr and Ir2 are loaded from the memory-address selector (MAS) only, and all other registers are loaded either from the MAS or from the synchronizer-control register (SCr). A MAC is loaded from the MAS each time the address contained in that MAC is processed by the memory adder. An Ir is loaded from the MAS only during the instructions which change the contents of an Ir (51, 52, and 53 instructions). A MAC is loaded by the SCr outputs when the L address of an input-output specification word is transferred from input-output control section to a MAC. The input to bit position 16 of the Ir's is inhibited at T6; the input to bit-position of the MAC's, however, is not inhibited.

The inputs to the TCWr's come from the SCr during the first tape-data call and from the MAS during other tape-data calls.

6-4. Outputs from SAr's and TCWr's

The principal output from the SAr's and TCWr's is to the memory adder. In order for the memory adder to supply an address to memory at the correct time, bit positions 1—4 of the SAr or TCWr must be transferred to the memory adder at T4. The contents of an Ir is transferred directly to the memory adder because bit positions 1—4 of the Ir are available at T4. The MAC and TCWr outputs, however, must be delayed four pulse times in order to transfer bit positions 1—4 to the memory adder at T4. This delay is done by routing the MAC or TCWr output through a four-pulse-time delay network to the memory adder. Thus, the MAC or TCWr input is supplied to the memory adder at the correct time.

6.5. LOGIC OF THE SAr AND TCWr SELECTION

The discussion of the SAr and TCWr input and output selection that follows is closely related to figures 6-2 and 6-3. Figure 6-2 shows the interrelation of the decoding networks which produce the SAr and TCWr selection, and figure 6-3 shows, in detail, the production of the function signals which control the selection.

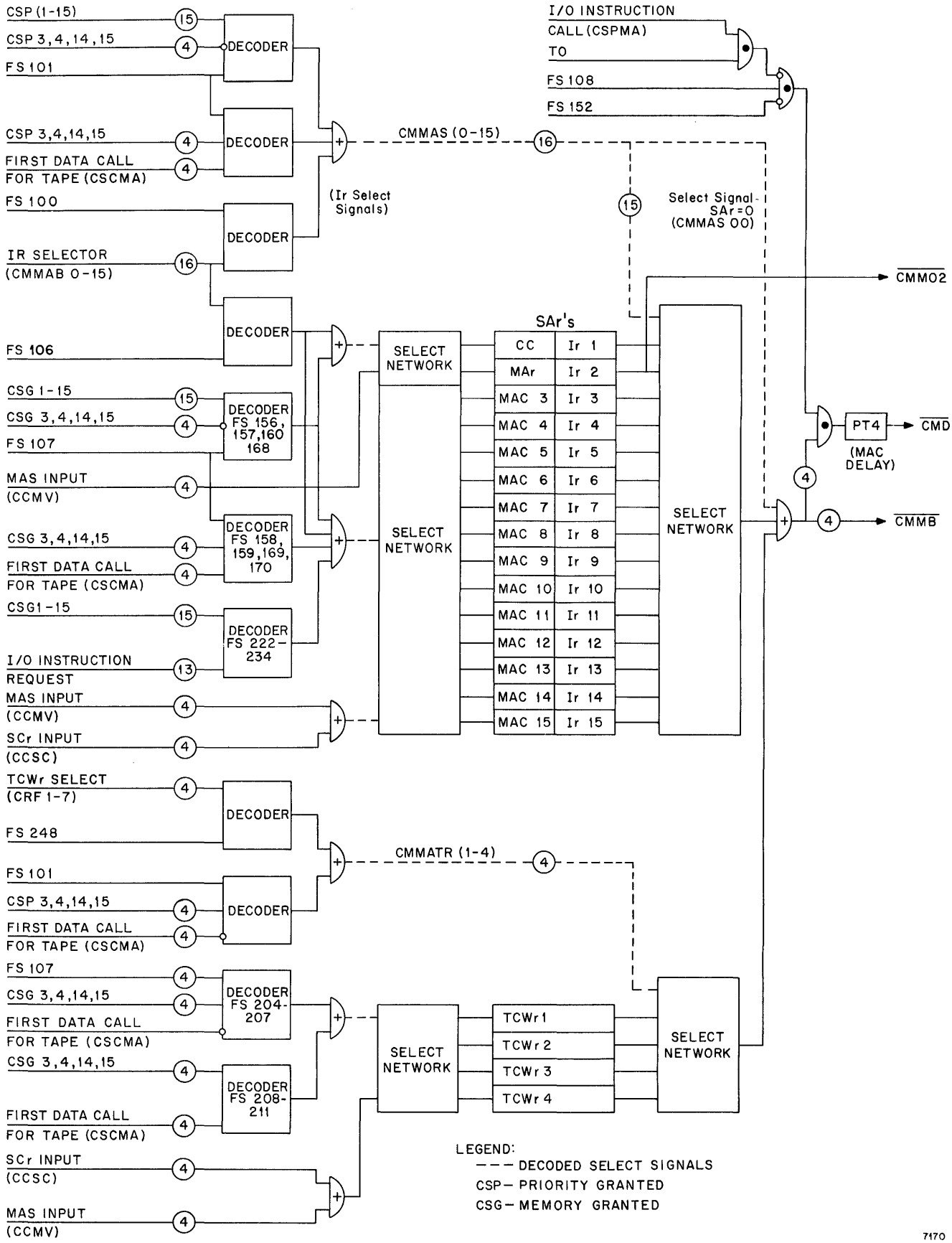


Figure 6-2. Storage-Address-Register and Tape-Control-Word-Register Selection, Detailed Block Diagram

6-6. Selection of SAr's

Either the Ir or MAC outputs are selected by the CMMAS(01—15) signals. These signals are produced at the different times in order to differentiate between the MAC or the Ir outputs. The selection signals gate the 0 bits, in 4-bit parallel, from the SAr through a buffer network. The outputs of the buffer network are $\overline{\text{CCMB(01—04)}}$.

The inputs to the SAr's are selected by function signals 107 or 106 and the memory granted (CSG) signals. The input information is supplied by the MAS, CMMV(1—4), or the CSr, CCSC(1—4).

6-7. SELECTION OF INDEX-REGISTER. The selection signals for an Ir are produced by a combination of FS 100 and the output from the IR-selector. Function signal 100 is present for an Ir selection from T4 through T7 during the first operand call, and during PC1 of the 51, 52, and 53 instructions, which load the Ir. Function signal 100 is also produced during a load-Ir instruction (51) but it is not used to read out the contents of the Ir. Also, during the 50 instruction, which transfers the contents of an Ir to memory, the MAC portion of the register is also read out by FS 100 but the contents are not transferred to the memory adder.

The inputs to the Ir's are selected by a combination of FS 106 and the output from the IR selector. Function signal 106 is present from T3 through T6 only during the 51, 52, and 53 instructions, which load the Ir's. During all other Ir readouts, the contents of the registers are not altered.

If the IR-select FF's contain all 0's, the output is decoded and gated by FS 100 to produce CMMAS00. This output is encoded into four signals, and these four signals are applied directly to the buffer network to produce $\overline{\text{CMMB(01—04)}}$. The output of the buffer network represents all 0's, which are transferred directly to the memory-address section. Thus, when the IR-select FF's contain all 0's, the m or L address is not modified.

6-8. SELECTION OF MEMORY-ADDRESS COUNTERS. The MAC outputs are normally selected by FS 101 and the CSP (priority granted) signal (figure 6-2). Function signal 101 is present from T0 through T6 when a memory request for any channel, except the operand-request channel (channel 2), has been granted. When a memory request for the operand-request channel is granted during the first operand call, FS 100 is produced. Function signal 101 is produced for channel 2, however, during most additional operand calls. During additional operand calls for instructions that load a MAC or an Ir and for instructions that transfer the memory-adder output to the MWr, FS 101 is inhibited by CSPMAR. If an SAr is required during these instructions, it is selected by FS 100.

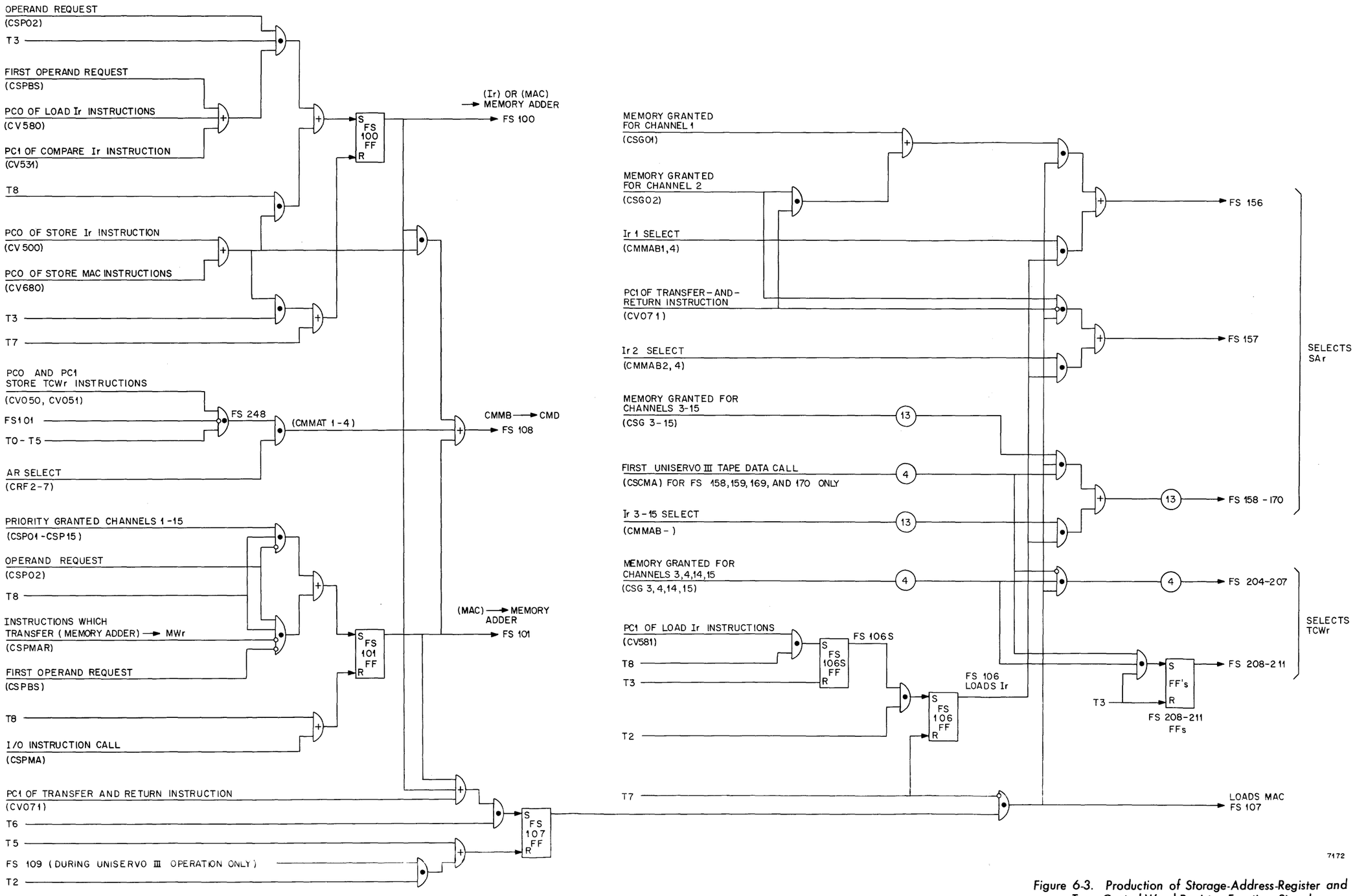
The production of FS 101 is inhibited during an input-output-instruction call because, during these calls, the memory address is supplied by an encoded output from the priority section. The inhibiting of FS 101 is performed by CSPMA which resets the FS-101 FF. Because a set signal overrides a reset signal in a pulseformer flip-flop, the FS-101 FF is set at T8 by the CSP signal and reset at T0 by CSPMA.

During an 04 or 07 instruction, either of which transfers the contents of a MAC to memory, the specific MAC is selected by the output of the IR-select network and FS 100. During these instructions, FS 100 is present from T0 through T3 in order to read out the MAC portion of the register.

The selection of MAC3, MAC4, MAC14, and MAC15 requires an additional input to represent a memory request for a scatter-read/gather-write control word. These inputs CSCMA(1—4) ensure that, when tape control words are used, information is read into the MAC's assigned to the UNISERVO III tape-unit channels only during the first tape call. This selection of a MAC is done by alerting the MAC selection gates and inhibiting the TCWr selection gates. Until the next scatter-read/gather-write control word is read from the memory, the TCWr's supply the memory address whenever either CSP04, CSP03, CSP14, or CSP15 is present.

The MAC inputs from the MAS are selected by FS's 156 through 170 and the inputs from the SCr to MAC3 through MAC13 are selected by FS's 222 through 234. The inputs from the MAS are the only inputs available to the CC or MAr, and are the inputs to MAC3 through MAC15 for all input-output data calls except tape data calls which use scatter-read/gather-write control words.

Function signals 156 through 170 are produced by a combination of a memory-granted signal (CSG) from 1 of the 15 priority channels and FS 107. Function signal 107 is present from T8 through T2 whenever information is read into a MAC or a TCWr. Function signals 156 through 170 are produced by CSG01 through CSG15 respectively, and this relation of function signal to memory-granted signal is altered only in the case of the transfer-and-return instruction (07). When CSG02 is present during this instruction, it is necessary to transfer the m' address to the CC instead of to the MAr. This is done by inhibiting the normal production of FS 157, which normally transfers the m' address into the MAr, and by producing FS 156 which transfers the m' address to the CC. Other transfer instructions that transfer the m' address to the CC, inhibit the production of CSG02 and produce CSG01 in the priority section.



7172

Figure 6-3. Production of Storage-Address-Register and Tape-Control-Word-Register Function Signals, Detailed Block Diagram

The production of FS's 158, 159, 169, and 170 requires the CSCMA(1—4) inputs, repsectively. These function signals return the output of the MAS to a MAC during the first data call for tape when control words are used.

Function signals 222 through 234 are produced during all input-output instruction requests and are used to transfer the L address to a specified MAC (figure 6-4). There is a function signal for each input-output channel (channels 3 through 15). These signals are produced by the combination of an instruction request from an input-output synchronizer, timing pulses T8 through T2, and an output from a combination encoder-decoder. The encoder-decoder produces one output at a time to represent the input-output channel that has been granted memory access. Thus, only the channel which has been granted memory access will accept the instruction request (figure 6-4).

The encoder-decoder consists of four flip-flops CSGVA(1—4) and a decoder. The flip-flops are set at T5 whenever information is being transferred between the central processor and memory. At T6 the four flip-flops are reset by the following CSG signals:

Flip-Flop	CSG Signal
CSGVA1	02, 04, 06, 08, 10, 12, or 14
CSGVA2	01, 04, 05, 08, 09, 12, or 13
CSGVA3	01, 02, 03, 08, 09, 10, or 11
CSGVA4	01, 02, 03, 04, 05, 06, or 07

The outputs of the flip-flops are then combined in the decoder which produces a single output to represent the channel that has been granted memory access. This output is present until the flip-flops are reset at the next T5. Thus, the encoded memory-granted signal for the channel is present from T7 through T6 which makes possible the transfer of information into MAC, from T8 through T2.

The encoded memory-granted signal from the decoder is produced by combining the set and reset outputs of the four flip-flops in an AND-gate decoding network. The decoding network contains 13 gates; 1 gate for each of the 13 channels. Each gate in the network combines the set and reset outputs of the four CSGVA FF's so that the gate produces an output for the presence of one CSG signal only. For example, table 6-2 shows how FS 222 is produced only when CSG03 is present. The lower case x's in the table represent the CSG signals which may be present when the flip-flop outputs as designated in the first column, are present. The upper case X's emphasize the condition in which there is an output from all four flip-flops. This condition is present only for CSG03. The other 12 decoder outputs are determined in a similar manner.

Table 6-2. Production of Function Signal 222 by CSG03

Flip-Flop Outputs	Memory Granted Signals (CSG) Which May be Present														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CSGVA1 Set	x		X		x		x		x		x		x		x
CSGVA2 Set		x	X			x	x			x	x			x	x
CSGVA3 Reset	x	x	X					x	x	x	x				
CSGVA4 Reset	x	x	X	x	x	x	x								

6-9. Selection of TCWr's

The outputs from the TCWr's are selected during tape operations which use control words by the following two processes:

The outputs from one of the four TCWr's are selected during all tape data calls except the first by a combination of FS 101, which is present from T0 through T6 during all tape data calls, and either CSP03, CSP04, CSP14, or CSP15. The selection of the TCWr is inhibited by CSCMA which is present during the first tape data call.

The outputs from one of the four TCWr's are selected during a TCWr-store instruction (05) by the decoded output of the Ar-select FF's and FS 248. Function Signal 248 is present from T0 through T5 of the TCWr-store instruction.

The AR-select-FF decoding for the TCWr selection is as follows:

AR-Select Flip-Flops				TCWr Selected
CRA1	CRA2	CRA3	CRA4	
0	0	0	1	1
0	0	1	0	2
0	1	0	0	3
1	0	0	0	4

The TCWr inputs from the MAS or from the SCr are selected as follows:

The MAS inputs CCMV(1—4) to one of the four TCWr's are selected by FS's 204 through 207,

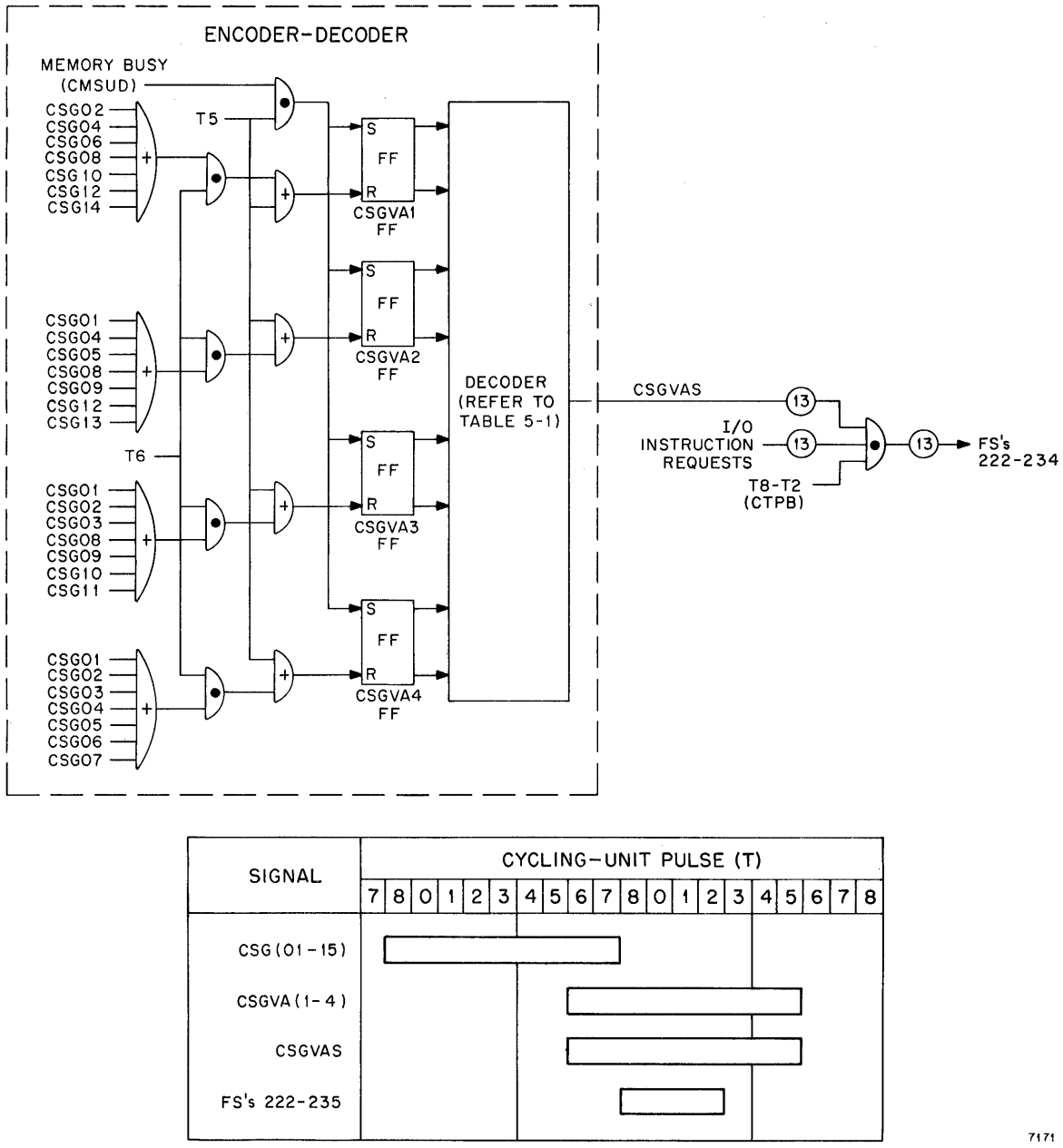


Figure 6-4. Production of Function Signals 222 through 234, Detailed Block Diagram

respectively. These function signals are produced by FS 107 and either CSP03, CSP04, CSP14, or CSP15 during all tape data calls except the first. Function signal 107 is present from T8 through T5 whenever information is to be read from MAS into a TCWr. The selection of TCWr(1—4) is inhibited by CSCMA(1—4) and FS 208 through FS 211, respectively, in order that the information is not gated into the TCWr during the tape-instruction call.

The CSr inputs, CCSC(1—4), to one of the four TCWr's are selected by FS's 208 through 211 during the first tape data call when scatter-read/gather-write control words are used. These function signals are present from T4 through T3.

The TCWr outputs must be available from T0 through T5 as shown in table 6-1. Function signals 208 through 211, however, are present from T4 through T3, and in order to have the outputs of the TCWr's available at the correct times the information is gated into the TCWr in the format shown in table 6-3.

Thus, the contents of bit positions, 01 through 24, of the TWCr's are available for readout from T0 through T5 (refer to table 6-1).

Table 6-3. Tape-Control-Word-Register Input Information

Time	Bit Positions			
T4	24	23	22	21
T5	—	—	—	—
T6	—	—	—	—
T7	—	—	—	—
T8	04	03	02	01
T0	08	07	06	05
T1	12	11	10	09
T2	16	15	14	13
T3	20	19	18	17

6-10. SAr AND TCWr OUTPUTS

The output from the Ir's goes directly to the memory adder and the output from the MAC's and TCWr's go to a delay network called the MAC delay. The MAC delay contains two pulseformers and a two-pulse-time-delay circuit for each of the four inputs. It delays a MAC or TCWr output for four pulse times so that the first four bits, which are gated from the register at T0, are available at the memory adder at T4. The outputs of MAC's and TCWr's are gated into the MAC delay by FS 108 which is present for all MAC and TCWr readouts.

The gating of a MAC or TCWr output to the MAC delay is inhibited by FS's 152 and 153 which are present to clear the first 6 bit positions of the memory address for the first data call and after the 20th data call for the card reader or punch (refer to section 12). The inhibiting, in this case, is done by jamming 0's into the MAC delay. The input to the MAC delay is also inhibited at T0 during an input-output instruction request.

The output of the register that contains MAr and Ir2 goes either to the field-select section or to the memory adder. The output to the field-select section is used to transfer the contents of MAr to the field-select FF's during the store-SAr instructions (04, 07, 50), and during the store-TCWr instruction (05).

Section 7
MEMORY-ADDRESS SECTION

7-1. INTRODUCTION

The memory-address section (figure 7-1) modifies, checks, and transfers address data. The memory-address section includes these:

A memory adder, which modifies and compares address information, provides an input path to a SAr or TCWr, and provides a transfer path from the computer-control register (CCr), storage-address register (SAr), or tape-control-word register (TCWr) to the memory-write register (MWr);

A modulo-3-check-bit generator, which produces the check bits for all information transferred from the memory adder to the MWr;

A memory-address-parity-bit generator, which produces a parity bit for each memory address processed by the memory-address section;

A memory-address selector (MAS), which converts an address from serial-parallel format to a parallel format, and which provides a transfer path for loading the SAr's or TCWr's;

A memory-cabinet selector, which determines the cabinet that will be used for a memory operation and the memory operation (read or write) that will be performed; and

A zero detector, which indicates a zero condition during the processing of the tape control word and during the compare portion of the 53 instruction.

The memory-address section handles information in groups of either 9, 10, 15, or 24 bits. This information comes from either the CCr, the storage-address-register section, or from the input-output-control section. The inputs from the CCr include:

The MA-field of the index-register-modification control word (9 bits), which is used to modify the contents of an index register (Ir) during a 52 or 53 instruction;

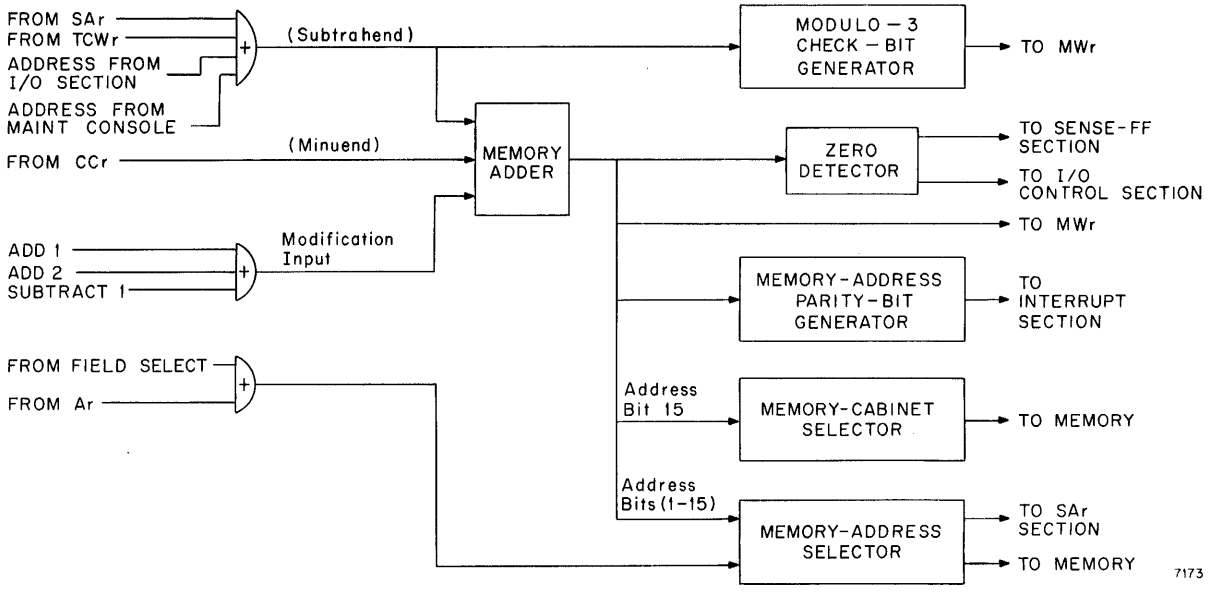


Figure 7-1. Memory-Address Section, Functional Block Diagram

The m address (10 bits), which is modified by the contents of a specified Ir either during the first operand call or during field selection;

The CA-field of the index-register-modification control word (15 bits), which is compared with the contents of a specified Ir during a 53 instruction;

The input-output specification word (24 bits), which is transferred through the adder to the MWr during a 70 instruction; and

The first 15 bits of the first operand, which are transferred through the adder to the Ir during a 51 instruction.

The inputs from the SAr section include:

The contents of the CC, which are increased by 1 for every instruction call except during the test-interrupt-FF instructions (64) when it may be increased by 2;

The contents of MAr, which are decreased by 1 for additional operand calls except during the zero-suppress instruction (73) when they are increased by 1;

The contents of a MAC, which are either increased or decreased by 1; (A 1 is normally added to the address for sequentially selecting memory locations, however, 1 may be subtracted from a MAC assigned to a tape unit when reading tape backwards. A 2 is added to the contents of a MAC for card-reader and card-punch addresses during the non-translate mode. Refer to section 12.)

The contents of a TCWr (24 bits), which contain a 15-bit address and a 9-bit tape word count; (During a forward-read or write operation of the tape unit, the address is increased by 1 and the tape word count is decreased by 1. During tape backward read both are decreased by 1.) and

The contents of a SAr or TCWr which are transferred through the adder to the MWr during a TCWr- or SAr-store instruction (04, 05, 07, 50), or an Ir-load instruction (51).

The memory-address section also receives a 4-bit address from the input-output section during the input-output instruction call. This address is transferred unmodified through the memory adder to the MAS.

7-2. MEMORY ADDER

The memory adder is a binary adder which adds a minuend digit to a subtrahend digit during one pulse time (a digit is defined as four bits). The adder adds up to seven digits in series, and the carry from the most significant bit (MSB) of one digit is the carry input to the least significant bit (LSB) of the next-most-significant digit.

The adder receives inputs from nine pulseformers; four minuend inputs CMAM(1—4), four subtrahend inputs CMAS(1—4), and one carry-input CMAC. The minuend pulseformers and the carry-input pulseformer are set for 1's and the subtrahend pulseformers are set for 0's.

7-3. Subtrahend Inputs

Each subtrahend pulseformer has three groups of inputs: one for the SAr's $\overline{\text{CMMB(01—04)}}$, one from the MAC delay $\overline{\text{CMD(01—04)}}$, one from the input-output control section CSPMA(1—4), and one from the maintenance console $\overline{\text{CDMS(00—14)}}$ (figure 7-2).

The $\overline{\text{CMMB(01—04)}}$ inputs normally represent an Ir output and are gated into the subtrahend pulseformers from T4 through T7 by FS 102. However, information from a MAC is gated into the pulseformers from T0 through T3 during the store-MAC instructions (04 and 07). (See figure 7-2.) Function signal 102 is present for the four following conditions:

When the contents of an Ir are added to an m or L address (indexing);

When the contents of FF's 1 through 9 of the CCr are algebraically added to the contents of an Ir during a 52 or 53 instruction;

When the contents of an Ir are stored during a 50 instruction; and

When the contents of an Ir are compared with the CA-field of the index-register-modification control word.

Function signal 102 also gates $\overline{\text{CMMB(01—04)}}$ to the memory adder during PC1 of the 04, 05, and 50 instructions. This information is not processed by the adder.

The $\overline{\text{CMD(01—04)}}$ represents the outputs from the CC, a MAC, or a TCWr, and is gated into the adder from T4 through T1 by FS 105 (figure 7-2). This function signal is present for the store-TCWr (05), store-MAC (04) and transfer-and-return (07) instructions, during either the readout of MAr for additional operand calls, or during the readout of MAC during input-output instructions calls.

The CSPMA(1—4) is the 4-bit address of the I-O standby location and is transferred from the input-output-control section directly to the subtrahend

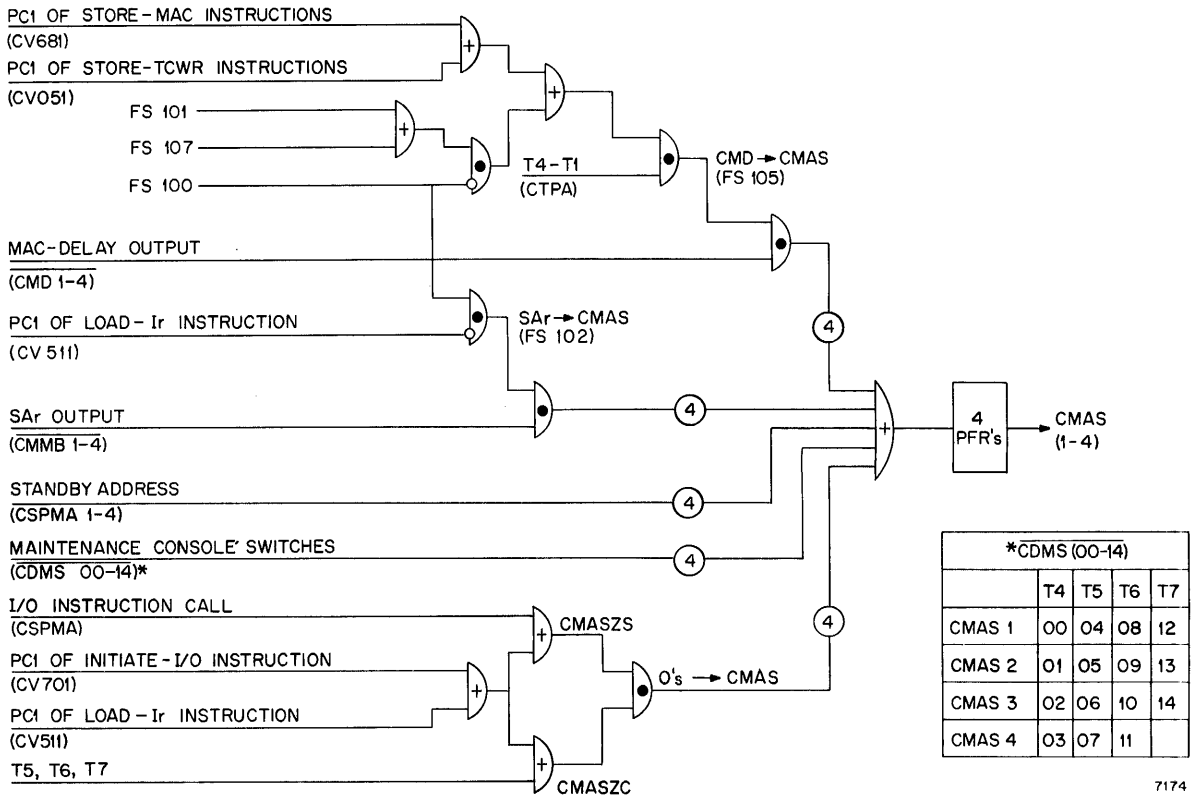


Figure 7-2. Production of Memory-Adder Subtrahend Inputs, Detailed Block Diagram

pulseformers at T4 during an input-output instruction call. Zeros are inserted into the subtrahend pulseformers at T5 through T7 by CMASZS and CMASZC in order to place 0's in the remaining 11 places of the address.

Zeros are also inserted into the subtrahend pulseformers by CMASZS and CMASZC during PC1 of the instructions which transfer, unmodified, information from the CCr to the minuend (instructions 51 and 70).

Signals $\overline{\text{CDMS(00—14)}}$ are gated into the subtrahend pulseformers from T4—T7 by FS 284 through 287. Function signal 284 gates CDMS(00, 04, 08, 12) to CMAS1, FS 285 gates CDMS(01, 05, 09, 13) to CMAS2, FS 286 gates CDMS(02, 06, 10, 14) into CMAS3, and FS 287 transfers CMDS(03, 07, 11) into CMAS4. The production of function signals is under maintenance-console control and will be discussed in detail in section 14.

7-4. Minuend Inputs

The minuend pulseformers have two types of inputs from the CCr, true and complemented. Function signal 103 (figure 7-3) gates the true contents of the CCr into the memory adder and FS 214 gates the complemented contents of the CCr into the adder.

Function signal 103 gates the CCr information into the memory-adder minuend pulseformers for the following five conditions:

When the m address (10 bits) of either the first operand or the field-select control word is modified by the contents of an Ir; (Function signal 103 is present from T4 through T6 during the first operand call or during field selection. During T6, FS 103A (figure 7-3) produces the CCRINH signal which inhibits the transfer of information to pulseformers CMAM3 and CMAM4. Thus, only 10 bits are transferred from the CCr to the memory adder: four bits during T4, four bits during T5, and two bits during T6. Signal CDCBMJ is produced at the maintenance console when an address, $\overline{\text{CDMS(00—14)}}$, from the console is inserted into the memory adder. The signal inhibits FS 103 and FS 103A because the address data are transferred from the console rather than from the CCr.)

When the MA-field of the index-register-modification control word modifies an Ir during a 52 or 53 instruction; (Function signal 103 is present from T4 through T6 of PC1 of these instructions, CV581, when complementing is not required. During T6, the CCRINH signal is produced by CV531 or CV532 to inhibit the transfer of information to pulseformers CMAM3 and CMAM4, and FS 220 is

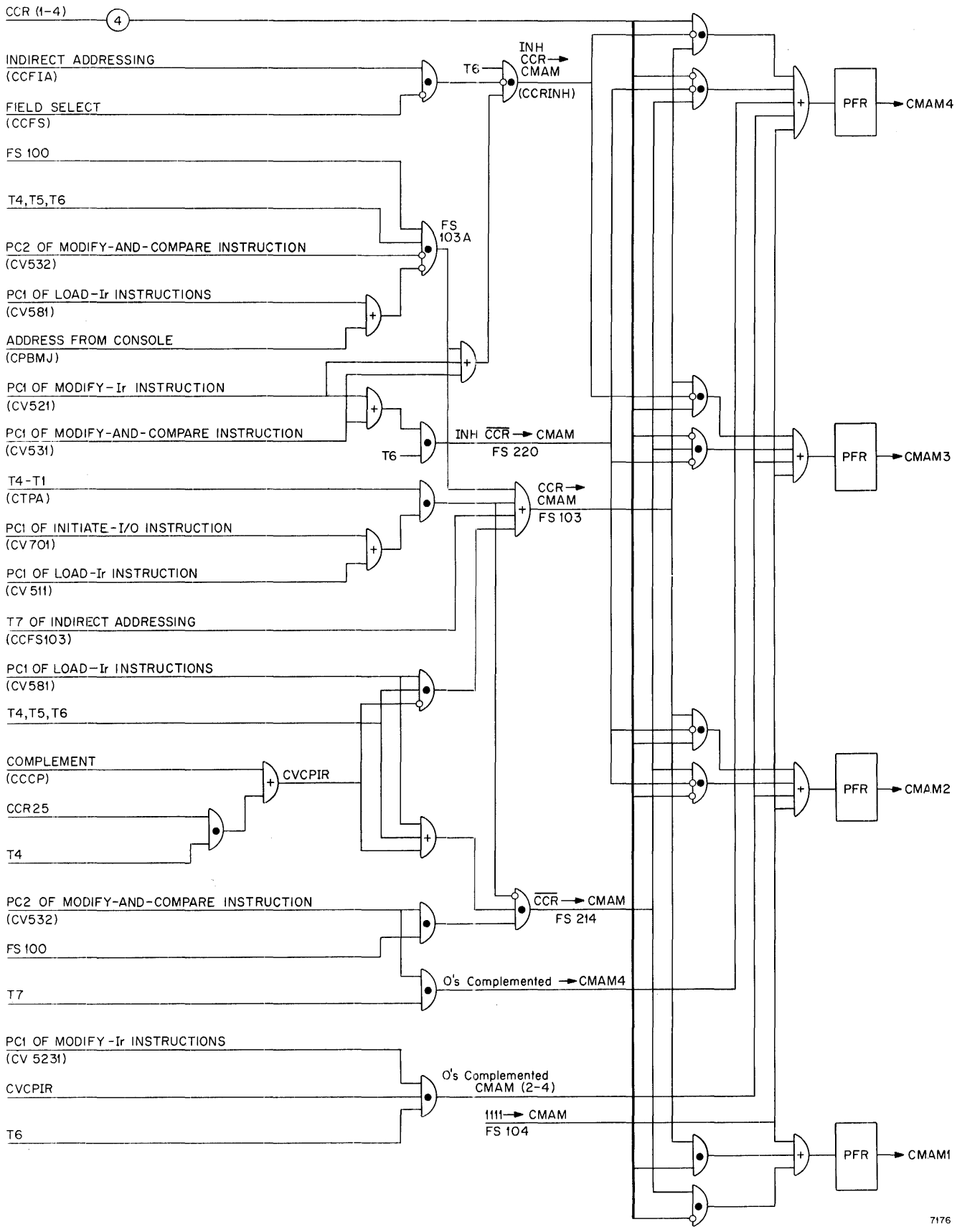


Figure 7-3. Production of Memory-Adder Minuend Inputs, Detailed Block Diagram

produced to inhibit the transfer of information to pulseformer CMAM2. Thus, only nine bits are transferred to the memory adder from the CCr.)

When the first 15 bits of the first operand are transferred from the CCr to the Ir during a 51 instruction; (Function signal 103 is present from T4 through T1 during PC1 of this instruction. Thus, the complete word in the CCr is transferred to the adder, but only the first 15 bits are transferred from the memory adder to the Ir.)

When the input-output specification word is transferred through the adder to the write register during a 70 instruction; (Function signal 103 is present from T4 through T1 of PC1 of the 70 instruction. Thus, a complete word (27 bits) is transferred through the memory adder.) and

When the 15-bit L address of the indirect-address control word is modified by the contents of an Ir; (Function signal 103 is produced in the same way that it is produced for the modification of the 10-bit m address except that the signal is present for one extra pulsetime, from T4 through T7. The production of FS 103 for the extra pulse time is done by the CCFS 103 signal which is produced by the field-select section. Also the production of the CCRINH signal is inhibited by the presence of the indirect-address signal, CCFIA, in order to transfer the 15 bits.)

Function signal 214 gates the complemented information into the adder for the following two conditions:

When the contents of an Ir are decreased by the magnitude specified in the MA-field of a 52 or 53 instruction. (Function signal 214 is present from T4 through T6 of PC1 of these instructions when complementing is indicated by the presence of CVCPIR. Signal CVCPIR is produced by combining CCR25 with T4. Signal CCR25 also sets a flip-flop which produces CCCP at T5. The gate producing CVCPIR is enabled by T4, T5, and T6. Function signal 220 is produced during T6 to inhibit the input to the CMAM2 through CMAM4 pulseformers in order that only nine bits will be accepted from the CCr. The CMAM2 through CMAM4 pulseformers are set to 1's at T6 by FS 269 and by FS 104 at T7 to represent 0's complemented in bit positions 10 through 15.); and

When comparing the contents of the CA-field of the Ir control word with the contents of an Ir during a 53 instruction. (Function signal 214 is present from T4 through T7 of PC2 of this instruction which allows the 15 bits to be transferred from the CCr.)

Another input to the four minuend pulseformers is FS 104 (Figure 7-4). This function signal sets all the minuend pulseformers to 1 and is present whenever 1 is subtracted from the output of a SAr or TCWr. The subtracting of 1 from a number is done through the addition of 1111 because subtraction is performed by adding the 2's complement of a number. The 2's complement of 0001 is 1111. Function signal 104 enables the subtraction process for the following functions:

The subtraction of 1 from the TCWr address during a magnetic-tape backward read;

The subtraction of 1 from the contents of MAr during additional operand calls; and

The subtraction of 1 from the tape-word counter during a magnetic-tape forward read or write operation.

The CMAM4 pulseformer has one input gate which is used only during the 53 instruction. The inputs to the gate are CV532 and T7 which are combined to set the CMAM4 pulseformer, the output of which represents a complemented 0 in bit position 16 of the CCr.

7-5. CMAC Pulseformer Inputs

The CMAC pulseformer adds 1 to the LSB of the memory-adder inputs for the following seven conditions:

When the contents of the CC are increased to read sequential instructions, FS 101 and CMMAS01, the CC select signal, set the CMAC pulseformer at T4; (The setting of the CMAC is inhibited by signal CSFMC01 for the first instruction call; by signal CCCT if an interrupt FF is reset during a test-interrupt FF instruction (64), and by signal CDCCR, when the same instruction is repeated under control of the maintenance console.);

When a 1 is added to MAC or TCWr address; (This operation is controlled by FS 213 which is present from T7 through T8 during input-output data calls. At T4, during the time that FS 213 is present, a 1 is added either to the address portion or the tape control word, or to the contents of a MAC. For a description of function of FS 213, refer to section 12.);

When a 1 is added to produce the 2's complement in order to subtract the contents of the MA-field from the contents of an Ir during a 52 or 53 instruction; (produced by CMAAICB which is present at T4 of PC1 of the 52 or 53 instruction when a complement operation is required, refer to section 10);

When a 1 is added to produce a 2's complement in order to compare the contents of the CA-field with the contents of an Ir; (produced by CMAAICA at T4 of PC2 of the 53 instruction);

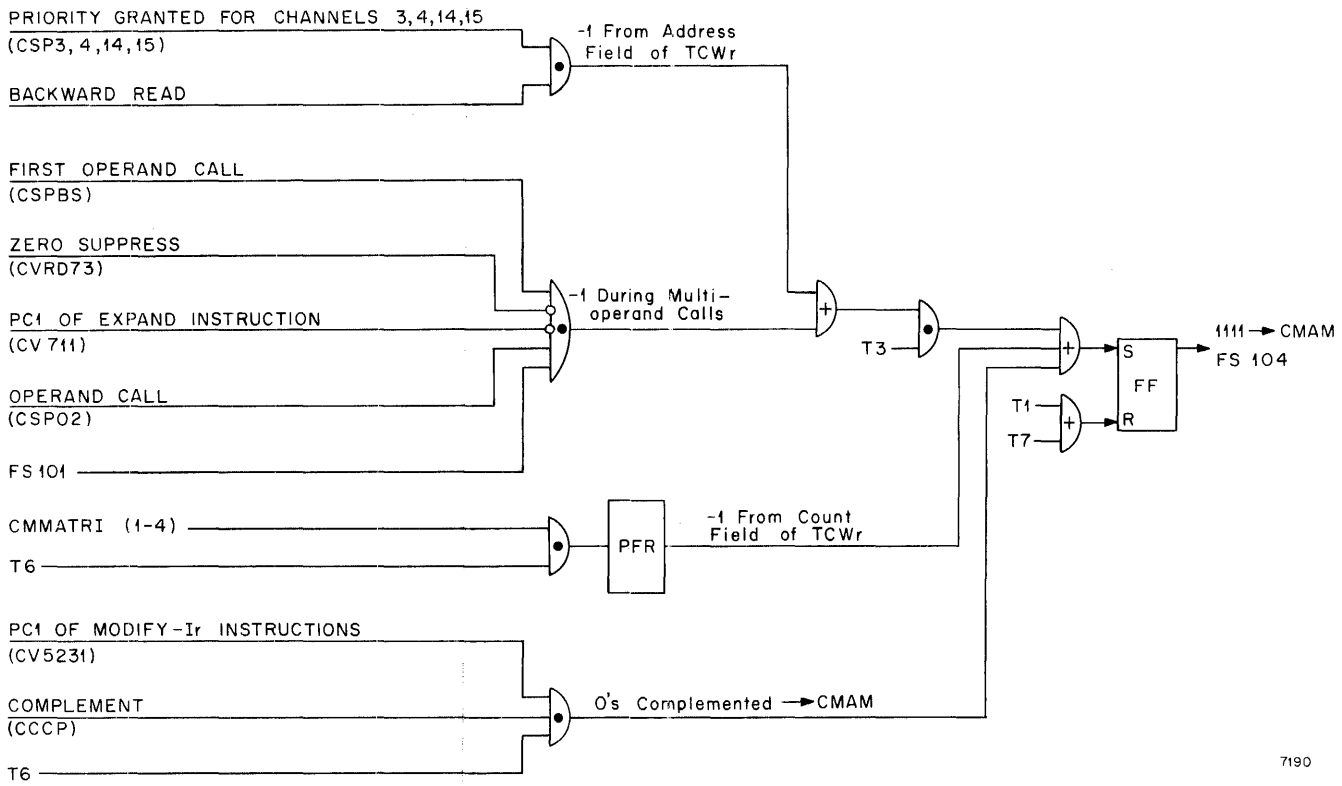


Figure 7-4. Production of Function Signal 104, Detailed Block Diagram

When a 1 is added to the contents of the CC during PC1 of the transfer-and-return instruction (07); produced by CV071);

When a 1 is added to the contents of MAr during the additional operand calls of a zero-suppress instruction (produced by 73B); and

When a carry-bit of 1 is added to the LSB of the memory adder input; (produced by a carry from the fourth bit of the preceding digit).

When FS 212 (figure 7-5) is present at T5, a 2 is added to the contents of a SAr for the following two conditions:

When an interrupt FF is reset during the test-interrupt instruction (64); and

When 2 is added to the card reader or card-punch address during a read or punch operation (refer to section 12).

7-6. Memory-Adder Addition Logic

The memory adder consists of a partial-add network and a final-add network. The partial-add network combines the minuend and subtrahend inputs of corresponding bit positions to produce a carry and sum for each bit position. The final-add network combines the partial sums and partial carries to produce a sum for each bit position.

The partial-add network for the first pair of bits (CMAS1 and CMAM1) is different from the partial-add network for the other bit positions. The partial-add network for the first bit position, shown in table 7-1, combines the minuend (CMAM1) and subtrahend (CMAS1) inputs with the carry input (CMAC), and the output of this combination is gated to produce the final sum for the first bit position (CMAF01). The partial-add network for the first bit position also produces a carry output for the first bit position (CMAK1). The partial-add networks for the other bit positions, shown in table 7-2, produce a sum output (CMA02 through CMA04) and a carry output (CMAK2 through CMAK4) for each subtrahend (CMAS) and minuend (CMAM) input bit position.

The final-add network combines the partial sums for bit positions 2 through 4 and the partial-carry outputs for bit positions 1 through 2, and produces final sum outputs for bit positions 2 through 4 (CMAF02 through CMAF04). The final-add network consists of a gating network which combines all combinations of carries and partial sums to produce a sum output for each bit position. The combinations are shown in table 7-3. The lines in the table represent partial-add network outputs that do not effect the final sum for that bit position.

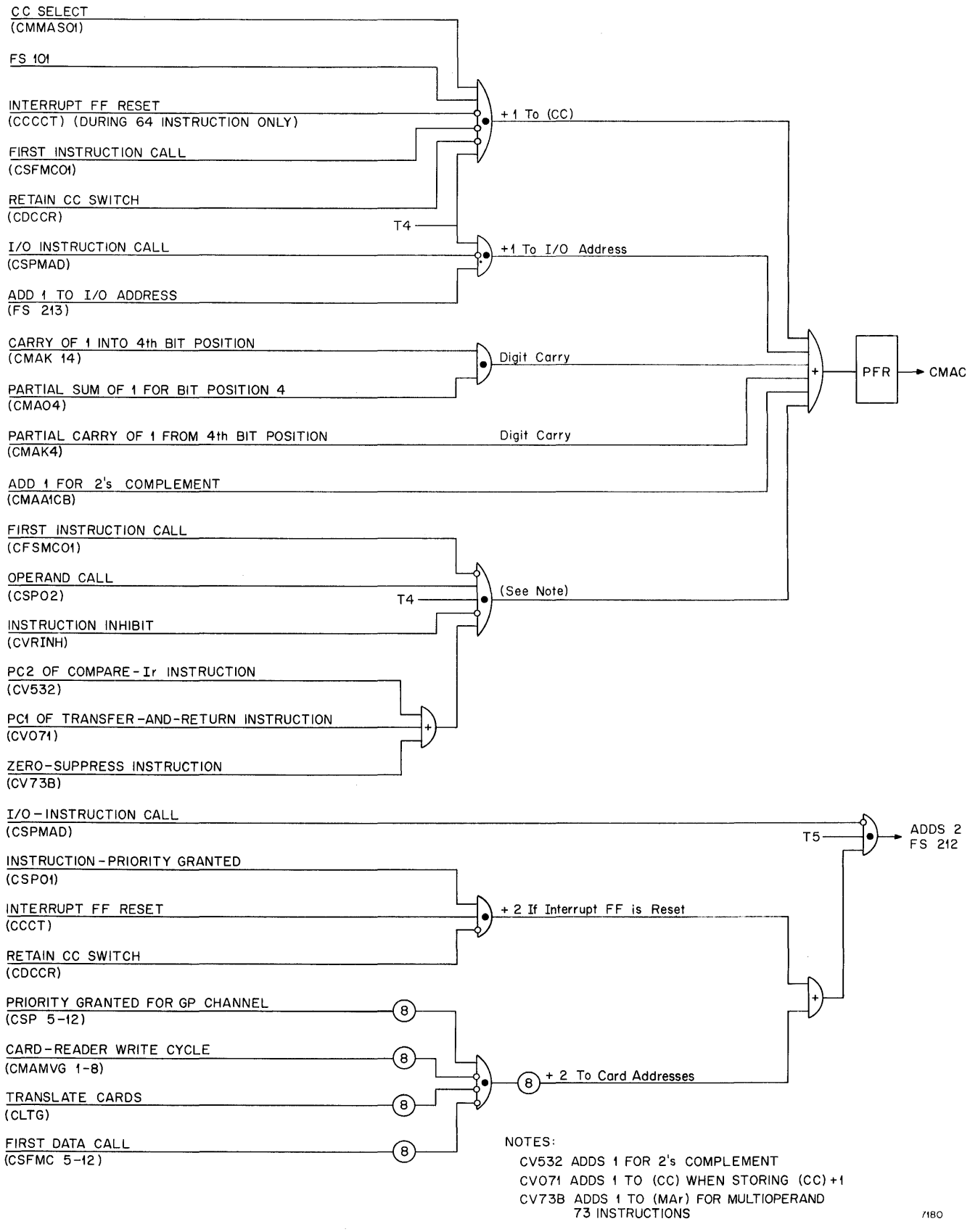


Figure 7-5. Production of Add-1 (CMAC) and Add-2 (Function Signal 212) Signals, Detailed Block Diagram

Table 7-1. Production of the Final Sum and Partial Carry for the First Bit Position

1st Bit Position Inputs		Carry Input	1st Bit Position Final Sum	1st Bit Position Partial Carry
CMAM1	CMAS1	CMAC	CMAF01	CMAK1
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Table 7-2. Production of Partial Sums and Partial Carries for Bit Positions 2 through 4

Inputs		Partial Sums	Partial Carries
CMAM(2-4)	CMAS(2-4)	CMA(2-4)	CMAK(2-4)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 7-3. Production of Final Sum Bits of 1 for Bit Positions 2 through 4

Partial Outputs Bit Position						Final Outputs Bit Position		
1	2		3		4	2	3	4
CMAK 1	CMA 02	CMAK 2	CMA 03	CMAK 3	CMA 04	CMAF 02	CMAF 03	CMAF 04
0	1	—	—	—	—	1	—	—
1	0	—	—	—	—	1	—	—
1	1	—	0	—	—	—	1	—
—	—	1	0	—	—	—	1	—
0	0	0	1	—	—	—	1	—
1	1	—	1	—	0	—	—	1
—	—	1	1	—	0	—	—	1
—	—	—	—	1	0	—	—	1
—	—	0	0	0	1	—	—	1

During normal operation of the memory adder, the partial carry from the third bit position (CMAK3) is inhibited at T8 in order to block the carry into the 16th bit position when address information is modified.

When a 1 is subtracted from the count field of a TCWr during UNISERVO III tape operations, a 1 is inserted in the carry from the third bit position (CMAK3) at T8. This production of CMAK3 is necessary because subtracting 1 is done by adding all 1's to the count-field of the tape control word, and CMAK3 must be produced to add a 1 to bit position 16, the first bit of the count field of the tape control word. Function signal 104 produces the 1 bits for subtracting 1 from bit positions 17 through 24.

Function signal 212 inserts a 1 in the partial carry for bit position 1 (CMAK1) in order to add a 2 to an address.

7-7. MEMORY-ADDRESS PARITY GENERATOR

The memory-address parity generator produces a memory-address parity bit (odd parity) for addresses that are transferred to the memory from the central processor. After the parity bit is generated, it is compared with a parity bit from memory. If the bits are not the same, there is an error (refer to section 11).

The memory-adder output, CMAF(01—04) is normally applied to the parity-bit generator to produce a parity bit. A parity bit is also produced from the output of the precision FF's during an initiate I-O instruction, or supplied directly from the MAS during a store-SAr or store-TCWr instruction.

7-8. Generation of Parity Bit From the Memory-Adder Output

The parity-bit generator consists of two gating networks which produce a parity bit from the output of the memory adder (figure 7-6). One network produces a parity bit (even parity) for each digit as it is transferred from the memory adder to the MAS, and the other adds the parity bits for each digit to produce a final parity bit.

The first gating network consists primarily of two pulseformers (PFR A and PFR B, figure 7-6). Pulseformer A is set if the CMAF(01 and 02) output equals 00 or 11, and pulseformer B is set if CMAF(03 and 04) equals output 00 or 11. The outputs of the pulseformers are combined so that an output is produced from the gating network if one pulseformer is set and the other is reset. Therefore, this gating produces an output if the total number of 1's (or 0's)

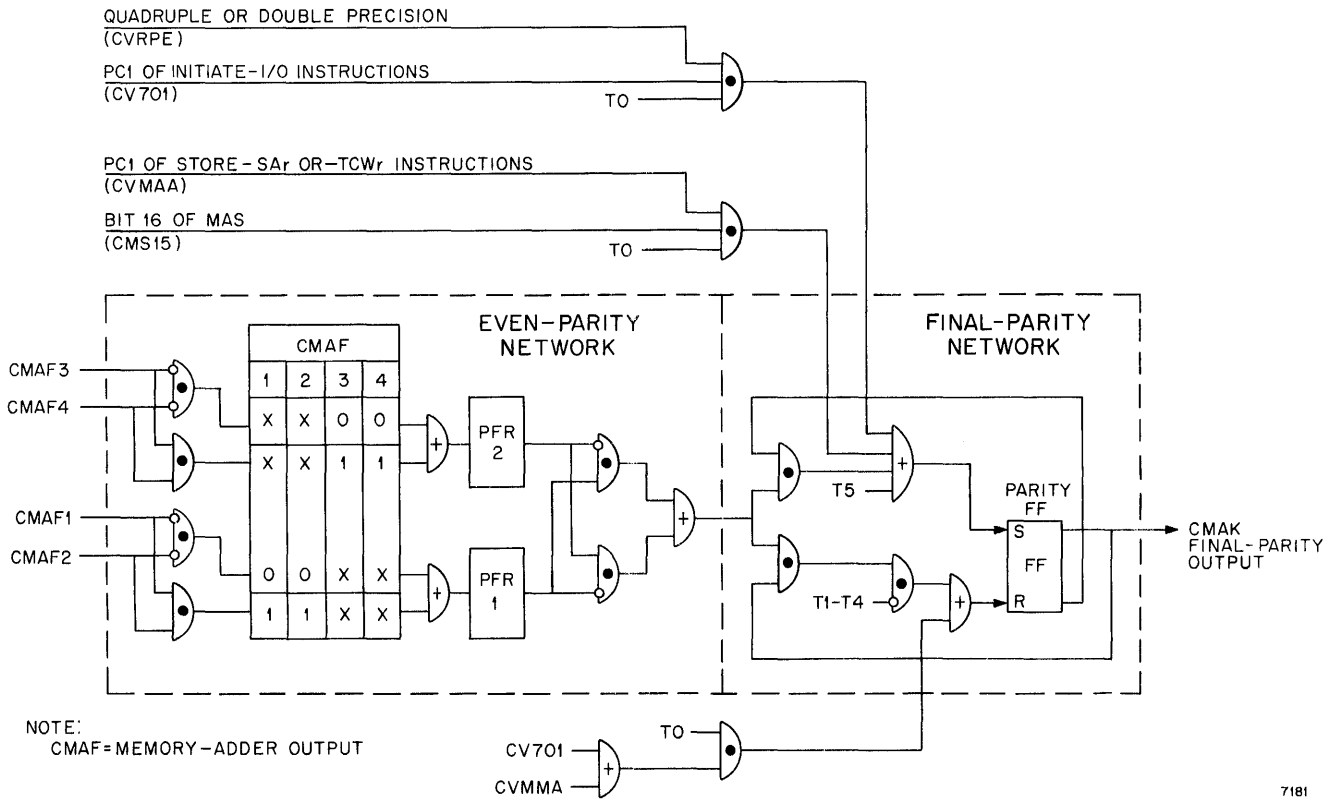


Figure 7-6. Memory-Adder Parity-Bit Generation, Detailed Block Diagram

is odd (table 7-4). The pulseformers receive inputs from T5 through T8. The memory adder transfers bits 15 and 16, CMAF(03—04), to the parity-bit generator at T8, but only a 14-bit address is significant for a parity check. Therefore, pulseformer B is set at T8, to simulate 0 from the memory adder.

The second gating network contains a parity FF, which stores the parity bits; and gates, which complement the parity FF whenever a parity bit of 1 is produced by the first gating network. The parity FF is initially set at T5, and from T6 through T0 the flip-flop is complemented by every 1 bit received from the first gating network. The complementing of the flip-flop for every 1 bit, as shown in table 7-5, produces an odd parity (final parity) bit at the output of the parity FF at T1. The parity FF stores the final-parity bit until required for comparison at T4.

Table 7-5 shows the production of the final parity bit at T1 for all combinations of inputs from the first gating network. The left column represents all possible combinations of parity bits from the four digits processed by the first gating network from T6 through T0; the right column shows the resulting outputs from the parity FF from T6 through T1. For example, if the parity bits produced by the first network were 0, 1, 0, and 0 (see arrow on table 7-5), the parity flip-flop is set at T5, and the first input,

Table 7-4. Parity Bit Production From First Gating Network

CMAF 01	02	03	04	State of PFRA	State of PFRB	Parity Bit Produced
0	0	0	0	S	S	0
0	0	0	1	S	R	1
0	0	1	0	S	R	1
0	0	1	1	S	S	0
0	1	0	0	R	S	1
0	1	0	1	R	R	0
0	1	1	0	R	R	0
0	1	1	1	R	S	1
1	0	0	0	R	S	1
1	0	0	1	R	R	0
1	0	1	0	R	R	0
1	0	1	1	R	S	1
1	1	0	0	S	S	0
1	1	0	1	S	R	1
1	1	1	0	S	R	1
1	1	1	1	S	S	0

a 0, does not complement the parity FF. The second parity bit, however, is a 1 and complements the flip-flop (resets it). All other parity bits are 0's and the flip-flop is not complemented again. Thus, the final parity bit is a 0.

Table 7-5. Production of Final Parity Bit

Parity Bit From First Gating Network				Parity Bit produced by Parity FF (Output at T1 is Final Parity Bit)				
T6	T7	T8	T0	T6	T7	T8	T0	T1
0	0	0	0	1	1	1	1	1
1	0	0	0	1	0	0	0	0
→0	1	0	0	1	1	0	0	0
1	1	0	0	1	0	1	1	1
0	0	1	0	1	1	1	0	0
1	0	1	0	1	0	0	1	1
0	1	1	0	1	1	0	1	1
1	1	1	0	1	0	1	0	1
0	0	0	1	1	1	1	1	0
1	0	0	1	1	0	0	0	1
0	1	0	1	1	1	0	0	1
1	1	0	1	1	0	1	1	0
0	0	1	1	1	1	1	0	1
1	0	1	1	1	0	0	1	0
0	1	1	1	1	1	0	1	0
1	1	1	1	1	0	1	0	1

7-9. Generation of Parity Bit During Store-SAr or Store-TCWr Instructions

During the execution of the store-SAr or store-TCWr instructions, the parity bit that is generated during the first operand call is stored in the 16th bit position of MAr. The storing is done by FS 298 (figure 7-9) which gates the parity bit (CMAKD) into pulseformer 6 of the MAS (CMS07) at T1 during PC0 of the store-Sar or store-TCWr instructions. The parity bit is then transferred from MAS to MAr during the normal MAr readin. When the contents of MAr are transferred through the field-select section into MAS, the 16th bit is gated into the parity FF at T0 from pulseformer 16 of the MAS (CMS15) (figure 7-6). The bit is stored in the parity FF until T4 when it is sent to the comparison circuits.

7-10. Generation of Parity Bit During Initiate I-O Instruction

During the execution of the initiate I-O instruction, the memory-address is obtained from the output of the AR-select FF's; therefore the parity bit cannot be generated from the output of the memory adder. The decoded output of the AR-select FF's always produces a precision signal (as described under heading 3-23), and this precision signal is used to produce the parity bit during a 70 instruction. The parity bit is produced by the output of the precision FF that is set to represent double- or quadruple-precision arithmetic. This flip-flop is set only when there are an even number of 1's in the AR-select FF's (either 2 or 4) and produces the CVRPE signal. Therefore, a parity bit of 1 is generated only when there is an even number of 1's in the memory address. This parity bit is gated into the parity FF at T0 of PC1 of the 70 instruction and is stored until T4.

7-11. MODULO-3 CHECK-BIT GENERATION

The modulo-3 check-bit generator in the memory-address section generates the check bit for all information that is transferred through the memory-address section to the MWr. After the check bits are generated, they are inserted in MWr pulseformers 26 and 27 and stored in memory along with the other 25 bits of the word (refer to section 8).

The purpose of the modulo-3 check-bit generation is to insert bits into bit positions 26 and 27 of a word in the MWr so that the magnitude of the word is a multiple of 3. This is done by dividing the word by 3, and deriving the check bits from the remainder, called the residue. If the residue is 1, a check bit of 1 is produced; if the residue is 2, a check bit of 2 is produced; and if the residue is 0, a check bit of 0 is produced. The check bits are stored in bit positions 26 and 27 of the word.

In order to generate the check bits each of the 27-bit positions is assigned a weight of either 1 or 2; odd bit positions have a weight of 1 and even bit positions have a weight of 2. The residue of a word is determined by adding the weighted values of the bit positions that contain a 1 bit, and reverting to 0 when the sum equals 3. When the 25 data-bit positions are evaluated, a residue of 0, 1, or 2 remains. The check-bit generator then inserts a 1 in bit position 26 and a 0 in bit position 27 for a residue of 1 (a check bit of 1,) a 1 in bit position 27 and a 0 in bit position 26 for a residue of 2 (a check bit of 2,) and a 0 in both bit positions for a residue of 0.

For example, if the binary representation of the decimal quantity 38 were transferred to memory through the memory-address section, the bit configuration would be as follows:

25 . . . 7 6 5 4 3 2 1

0 . . . 0 1 0 0 1 1 0

By applying a weight of 2 to the even bit positions and a weight of 1 to the odd bit positions, adding the weighted values of all bit positions containing a 1 bit, and reverting to 0 when the sum equals 3, a residue of 2 is obtained. The residue of 2 causes a 1 bit to be put in bit position 27 and a 0 to be put in bit position 26. The word stored in memory would be as follows:

27	26	25	. . .	7	6	5	4	3	2	1
1	0	0	. . .	0	1	0	0	1	1	0

The 27-bit word is now an even multiple of 3.

The check-bit generator produces the check bits for each digit as it enters the memory adder, and adds the residue of each digit to the accumulated residue from all preceding digits. The generator consists of two pulseformers, which store the accumulated residue; two decode networks, which determine the residue for each new digit; and a gating network, which adds the new residue to the accumulated residue (figure 7-7).

The residue for each digit is generated from the subtrahend inputs CMAS(1—4) to the memory adder for the store-SAr instructions (04, 07, and 50) or the store-TCWr instruction (50). The CCr provides the check bits during the initiate I-O instruction (70).

During the execution of the store-SAr or -TCWr instructions, the two pulseformers (CMAPF1 and CMAPF2) are set to 1 at T4, indicating a residue of 0. This T4 is one pulse time before the first adder input is received. The subtrahend inputs are present from T5 through T8 for store-SAr instructions, and from T5 through T1 for the store-TCWr instructions. Each subtrahend digit is decoded in two stages. The first stage determines the residue of each pair of bits in a partial digit, that is, the residue of bits 1 and 2, and the residue of bits 3 and 4. The second stage combines the outputs of the first stage to determine the residue for the complete digit.

A residue of 1 for the first pair of bits (CMAS1 and CMAS2) is represented by CMAP1 and a residue of 2 is represented by CMAP2. A residue of 1 from the second pair of bits (CMAS3 and CMAS4) is represented by CMAP3 and a residue of 2 is represented by CMAP4. A residue of 1 for a digit is represented by CMAPA and a residue of 2 is represented by CMAPB. A residue of 0 is represented by the outputs from the decode network, (CMAP1A through $\overline{\text{CMAP1C}}$ and $\overline{\text{CMAP2A}}$ through $\overline{\text{CMAP2C}}$), which are combined to produce the 0-residue indication. The final accumulated residue, which is obtained from the gating network, is samples at T1. A residue of 1 sets MWr pulseformer 27, and a residue of 2 sets MWr pulseformer 26. Because the MWr pulseformers are set to 0, the input is complemented and thus produces the correct check bits. Table 7-6 summarizes the generation of the check bits.

When the contents of the SAr are transferred to the MWr the final digit from the SAr enters the memory adder at T8, and the check bits are not entered into the MWr until T1. Therefore, 0's are inserted into the decoders at T0 and T1 in order not to alter the residue during these two pulse times. When the contents of a TCWr are transferred through the memory adder to the MWr, the final digit is entered at T1, therefore 0's are not inserted into the decoder at T0 and T1.

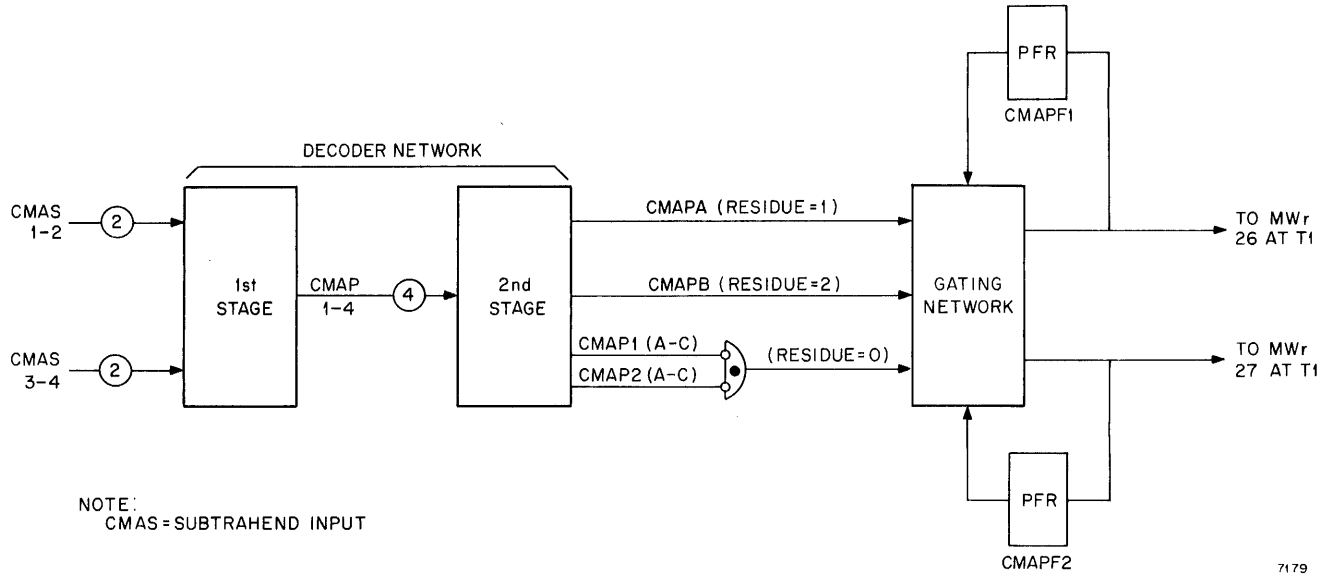


Figure 7-7. Modulo-3 Check-Bit Generation, Detailed Block Diagram

When the I-O specification word is transferred through the memory adder to the MWr during the initiate I-O instruction (70) the check bits are taken directly from the CCr and stored in the check-bit pulseformers until required by the MWr. The check bits are taken from CCr bit position 22 and CCr bit position 23 at T5 of PC1 of the 70 instruction, and during T5 through T1 0's are inserted into the subtrahend pulseformers which means that the residue is not altered. At T1 the residue is gated into the MWr to form the check bits.

Table 7-6. Summary of Modulo-3 Check-Bit Generation

CMAS 4 3 2 1	Output of the First Stage	Output of the Second Stage	Residue	MWr Pulseformers Set †
0 0 0 0	—	—	0*	26 and 27
0 0 0 1	CMAP1	CMAPA	1	27
0 0 1 0	CMAP2	CMAPB	2	26
0 0 1 1	—	—	0*	26 and 27
0 1 0 0	CMAP3	CMAPA	1	27
0 1 0 1	CMAP3, CMAP1	CMAPB	2	26
0 1 1 0	CMAP3, CMAP2	—	0*	26 and 27
0 1 1 1	CMAP3	CMAPA	1	27
1 0 0 0	CMAP4	CMAPB	2	26
1 0 0 1	CMAP4, CMAP1	—	0*	26 and 27
1 0 1 0	CMAP4, CMAP2	CMAPA	1	27
1 0 1 1	CMAP4	CMAPB	2	26
1 1 0 0	—	—	0*	26 and 27
1 1 0 1	CMAP1	CMAPA	1	27
1 1 1 0	CMAP2	CMAPB	2	26
1 1 1 1	—	—	0*	26 and 27

*Zero residue indicated by $\overline{\text{CMAP1(A—C)}}$ and $\overline{\text{CMAP2(A—C)}}$ from the second stage of the decoder.

† The output of the pulseformer in the MWr is a 0 when the pulseformer is set.

7-12. MEMORY-ADDRESS SELECTOR

The memory-address selector (MAS) converts a 15-bit address from a serial-parallel format to a parallel format. The MAS is also a transfer path for loading modified-address information into a MAC, for loading the Ir's during the 51, 52, and 53 instructions or for loading a TCWr during a tape data call.

In addition to receiving information from the memory adder, the MAS receives two unmodified-address inputs directly. One input from the AR-select FF's specifies the standby location where the 24-bit input-output specification word is to be stored. The second input, from the field-select section, specifies the location where information is to be stored during the execution of the store-SAr and store-TCWr instructions.

The MAS also has a recirculation input for the store-SAr and store-TCWr instructions, and for the 51, 52, and 53 instructions, which load information into an Ir. Recirculation delays the information in the MAS for four pulse times to supply the information at the correct time for memory addressing or for loading the Ir's.

7-13. Memory-Address Selector Operation

The MAS consists of 16 pulseformers (figure 7-8). These pulseformers are arranged in four groups with each group containing four pulseformers connected in series. This arrangement permits one digit (4 bits) to set one pulseformer of each group each pulse time. One group of pulseformers stores a digit for one pulse time and then transfers it to the next group of pulseformers. After one digit is transferred, the pulseformers can accept the next digit.

A total of 15 bits is transferred to the MAS during four consecutive pulse times. The MAS receives four bits in parallel for three pulse times and three bits in parallel for one pulse time. Therefore after four pulse times the 15 bits are stored in the MAS. At this time a memory pulse samples the outputs of the MAS and an address is transferred to memory in a parallel format.

The address digits are transferred to the MAS at T5, T6, T7, and T8. The T5 digit, bit positions 1 through 4, is delayed 4 pulse times; the T6 digit, bit positions 5 through 8, is delayed 3 pulse times; the T7 digit, bit positions 9 through 12, is delayed 2 pulse times; and the T8 digit, bit positions 13 through 15, is delayed 1 pulse time. All the bits are available from the MAS pulseformers (CMS00—CMS15) at T0 which is the only time that memory samples the MAS output.

7-14. Memory-Address Selector Inputs

The four inputs to the MAS (figure 7-9) are from the memory adder CMAF(01—04), from the AR-select FF's, CRA(1—4), or from the field-select section, CCFSB(0—3). A recirculating input CMS(0—3) is also used.

The inputs from the memory adder CMAF(01—04) are the main inputs to the MAS and are gated into

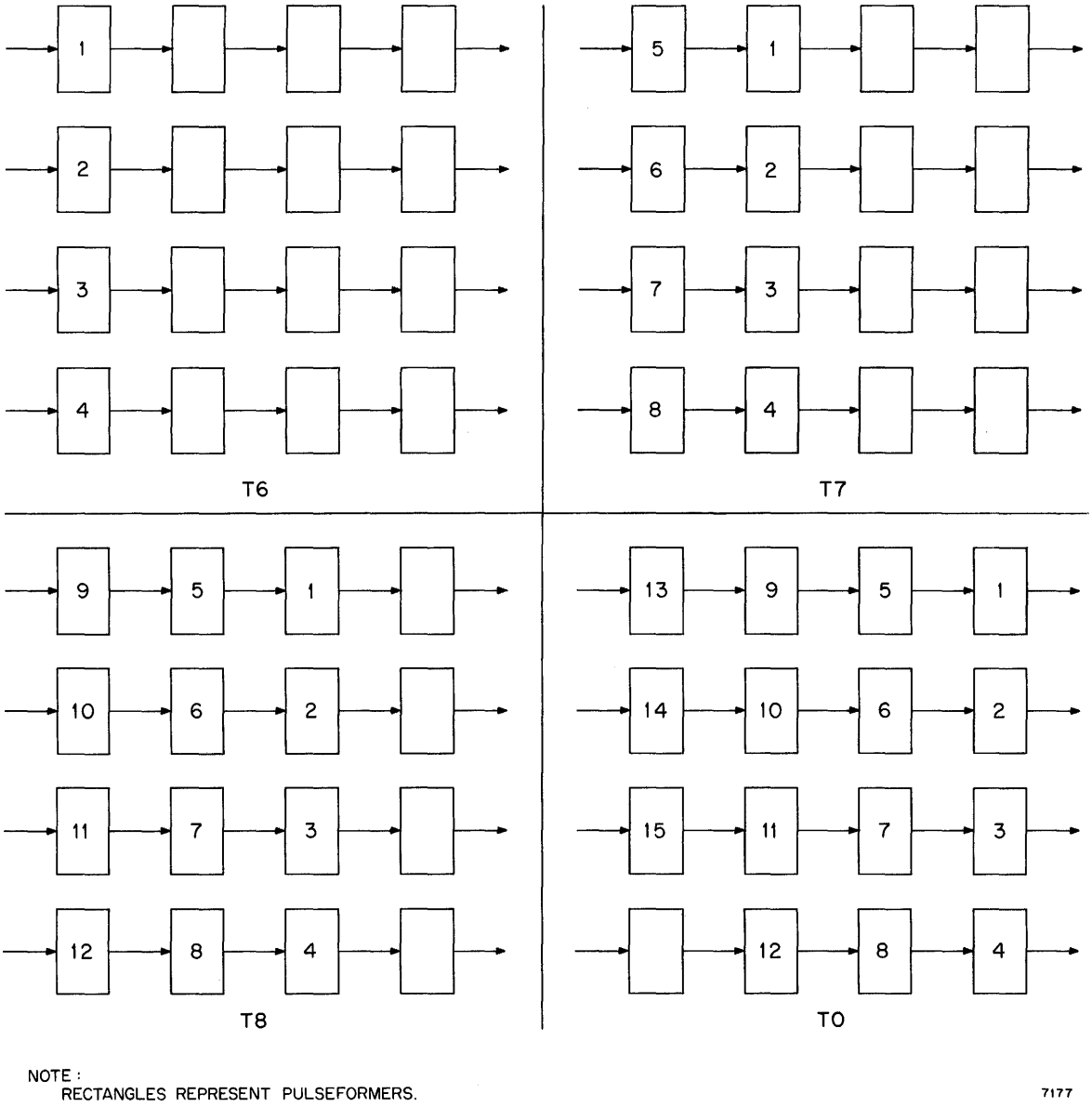


Figure 7-8. Memory-Address Selector, Functional Block Diagram

the MAS by FS 109. This function signal is present from T5 through T8 for all memory requests except the 70, 04, 07, 05, and 50 instructions, which transfer the output of the memory adder to the MWr. During the execution of the tape-data calls, the contents of the 24-bit TCWr must be transferred to the memory adder; therefore FS 109 is present during tape-data calls (represented by CSGTAS) from T5 through T2 rather than from T5 through T8. Function signal 109 is also present during PC1 of all instructions which load the Ir's. The MAS is used only as a transfer path to the Ir's during these instructions, and this is the only instance where information is transferred from the memory adder to the MAS without memory access being granted.

The inputs from the AR-select FF's are gated into the MAS by FS 250. This function signal, present at T5 of PC1 of the initiate I-O instruction, gates the address of the standby location into the MAS.

The inputs from the field-select FF's are gated into the MAS by FS 244. This function signal is present from T1 through T4 of the store-TCWr or store-SAr instructions.

The recirculation in the MAS is controlled by FS 245 which is present from T5 through T8 of the PC1 of the store-SAr or store-TCWr instructions, or from T8 through T3 of the store-Ir instructions.

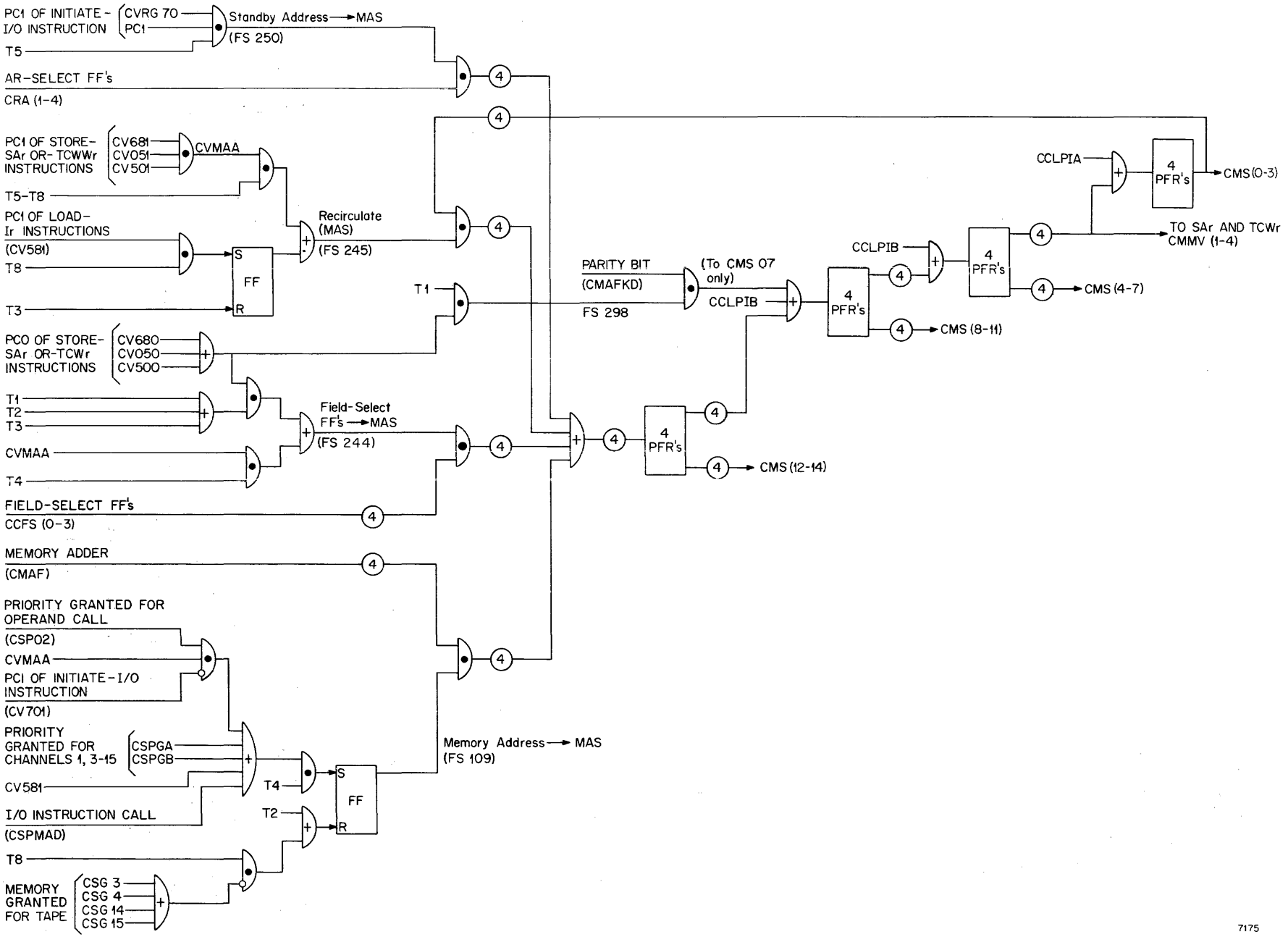


Figure 7-9. Memory-Address Selector, Detailed Block Diagram

The recirculation is necessary during the store instructions because the inputs from the field-select section to the MAS are supplied from T1 through T4 and the sampling must be delayed until T0. The recirculation during the load-Ir instructions is necessary to delay the first Ir input from T0 through T3.

7-15. MEMORY-CABINET SELECTOR

Each memory cabinet contains up to 16 384 locations. Cabinet 1 contains memory addresses from 1 through 16,384, and cabinet 2 contains addresses from 16,385 through 32,768. The cabinet selected depends upon the contents of the 15th bit position of the address information. If the 15th bit is a 1, the address must be above 16,385 and cabinet 2 is selected; if it is a 0, cabinet 1 is selected. The 15th bit alerts gates which are enabled by memory-read or memory-write signals, originating from input-output units or from central-processor instructions.

Normally the 15th bit is obtained from the memory-adder output (CMAF03 at T8). However, for instructions that recirculate the address in the MAS, the 15th bit is taken from the output of the second MAS pulseformer (CMS02E) at T8.

The memory read-write function is also controlled by the cabinet selection process. The write operation is determined either by function signals 130 through 139 which are initiated by the input-output units or by signals produced by an instruction requiring a store operation (04, 05, 07, 10, 11, 50, 70, or 71). The absence of signals produced by these instructions produces a read operation.

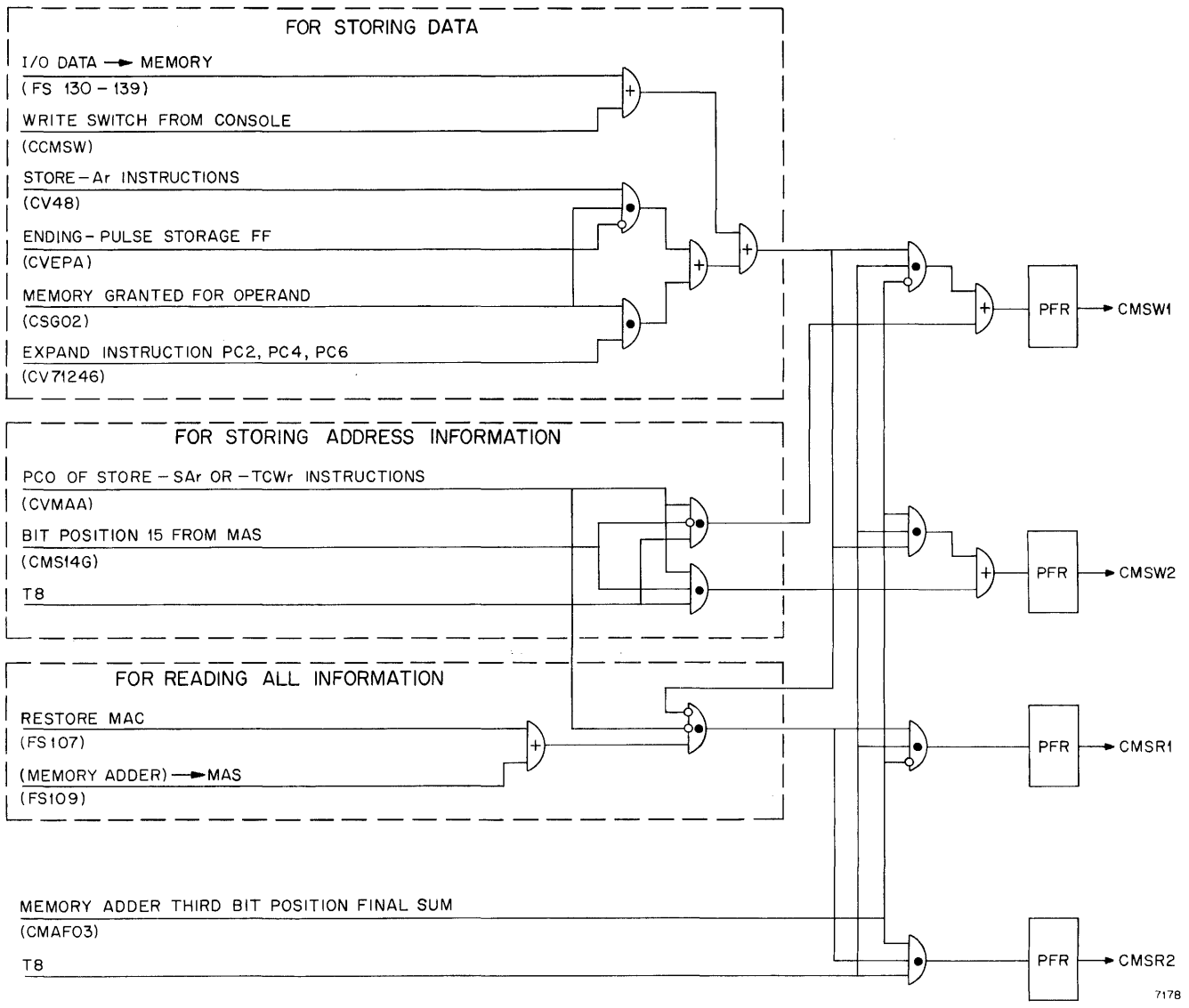


Figure 7-10. Cabinet Selector, Detailed Block Diagram

The memory read-write function and cabinet selection function are controlled by means of four pulseformers (figure 7-10). One pulseformer produces a read-from-cabinet-1 signal (CMSR1); one pulseformer produces a read-from-cabinet-2 signal (CMSR2); one pulseformer produces a write-into-cabinet-1 signal (CMSW1); and one pulseformer produces a write-into-cabinet-2 signal (CMSW2).

Section 1 of figure 7-10 shows the write logic for store operations. Function signals 130 through 139 are active for input-output operations that require writing into memory. Signal CVRG70 produces the write signal for writing the tape-specification word into the standby location. Signal CV48 (instructions 10 and 11) produces the write signals for the storing of the contents of an Ar into memory. Signal CV71246 produces the write signals for the storing of the expanded data during PC2, PC4, and PC6 of the 71 instruction. Signal CCMSW produces a write signal from the maintenance console.

In section 2 of figure 7-10 the write logic for the store-SAr and -TCWr instructions (CVMAA) is shown.

In section 3 of figure 7-10 the read logic for all memory requests is shown. An output is produced from this section at T8 during any memory-read operation by function signals 107 and 109. The read operations are inhibited by any write signals.

7-16. ZERO-DETECTOR FLIP-FLOP

The zero-detector FF (CSCC0) is set when there is a 0 output from the memory adder, and is reset when there is a nonzero output. The output of the flip-flop is sampled only during a UNISERVO III tape operation or during an Ir-modify-and-compare instruction (53) (see figure 7-11). During a UNISERVO III tape operation, the flip-flop is set when the tape-word-counter portion of the tape control word has been decreased to 0. During the 53 instruction the zero-detector FF is set if the results of a comparison indicate that the two quantities are equal (if the difference is zero).

7-17. Tape Operation

Bit 16 of the tape control word sets the CSCC0 FF if the bit is a 0, or resets the flip-flop if the bit is a 1. At T8, bit position 16, which contains the first bit of the tape-word counter, is the memory adder output, CMAF04. If CMAF04 is a 0, the CSCC0 FF is set, and if a 1 bit is present in any of the bit positions of the two digits that follow at T0 and T1, the flip-flop is reset. A 1 bit in bit position 16 means that the magnitude of the tape control word is greater than 0, therefore, the flip-flop is never set, if there is a 1 in bit position 16.

At T8, CMAF01 through CMAF03, containing bit positions 13, 14 and 15 of the tape control word, are also transferred to the zero-detector FF with CMAF04 (bit position 16). Signals CMAF(01--03) alert the reset input while CMAF04 alerts the set input. Therefore, regardless of the contents of the CMAF(01—03) outputs, the flip-flop is set if CMAF04 is a 0.

The reset input to the zero detector FF is inhibited at T3 and T4 because the memory-adder subtrahend pulseformers are reset to 1's at this time and would reset the CSCC0 FF before its output could be sampled during a UNISERVO III tape operation.

7-18. The 53 Instruction

The CSCC0 FF is set during PC2 of the 53 instruction. If the result of the subtraction, which takes place during PC2 of the 53 instruction, produces a 1 bit in the memory adder output, indicating that the compared quantities are not equal, the flip-flop is reset. The CMAF04 signal is inhibited at T8 because bit position 16 is not a part of the CA-field. The output of the CSCC0 FF is transferred to the sense-FF section, where it is combined with the memory-adder-carry output to produce a high, low, or equal signal (refer to section 11).

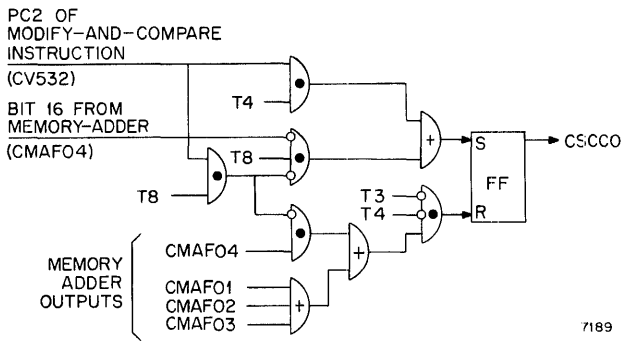


Figure 7-11. Zero Detector, Detailed Block Diagram

Section 8
MEMORY-WRITE REGISTER

8-1. INTRODUCTION

The purpose of the memory-write register (MWr) is to prepare one word at a time for writing into memory. Preparation of a word consists of altering an input format to the format required for writing into memory. Words enter the MWr in the format of the unit that is writing into memory. For example, words in the UNISERVO III format enter the MWr three bits in parallel for nine pulse times, and words in the general-purpose format enter the MWr as four bits in parallel for seven pulse times. Each format is arranged into the memory-word format by shifting the bits within the MWr and by transmitting the word to memory at T2.

The memory-write register consists of 28 pulseformers that are set by the 0's of the input words. During shifting, bits are shifted from one pulseformer to another according to the input format. Each shift delays a bit one pulse time. There are five formats each of which requires a separate shifting sequence to reach its output position by T2. At T2, bits of any one of the input formats are in the output pulseformers in bit positions that correspond to the memory format. This shifting sequence is called the serial-to-parallel conversion.

The five formats are: the central processor format, used by the Ar and memory-adder inputs; the UNISERVO III format; the input-output format, used by the UNISERVO II channel and general-purpose channels; the computer-control-register format (expand instruction only); and the maintenance-panel format.

Figure 8-1 shows the input formats, the MWr pulseformers, and the memory format. Each square represents a bit. The numbers in the squares indicate the positional relationship of each bit in a word. For example, bit 1 in the input format is bit 1 in the memory format and, at T2, is contained in pulseformer 1. The only exception is the CCr inputs which are spaced within MWr to include zero zone bits.

8-2. CENTRAL-PROCESSOR FORMAT

All instructions that store the central-processor information (except the expand instruction) use the central-processor format. This format enters the MWr in a parallel-serial format as 4-bit parallel digits, shifted serially once each pulse time from one of two sources, the accumulator register, or the memory adder.

8-3. Accumulator Register

The output signals, CRB(01—04), from the selected accumulator register (Ar) are gated into the MWr from T4 through T1 (figure 8-2). The output from the Ar consists of seven digits. Of the seven digits, the first six are information digits and the seventh digit is a combination of the sign and check bits.

Function signal 271 gates one digit each pulsetime into pulseformers 25 through 28. After each digit is entered into the MWr, it is shifted to the next least-significant-digit position of the memory format each pulsetime. Function signal 271 shifts the digit from the most-significant-digit position and FS 121 shifts the digits for the remaining shifts. Bits 25, 26, 27, and 28 require no shifting because they are entered into their correct position at T1. At T2 the MWr is loaded with each bit in its correct pulseformer for writing into memory.

Function signal 271 (figure 8-3) is present from T4 through T1. It is produced by a store-Ar instruction (CV48), or during the first operand call. The gates which transfer the contents of an Ar to the MWr are alerted during the first operand call because, if FS 271 were delayed until the store-Ar instruction was decoded, the loading of the MWr would have to be delayed until the following program count.

Function signal 121 is present from T4 through T1 when data, in the central processor format, are transferred to the MWr (figure 8-3). Function signal 121 is used for decimal shifting within the MWr. During the execution of the decimal shift, each digit

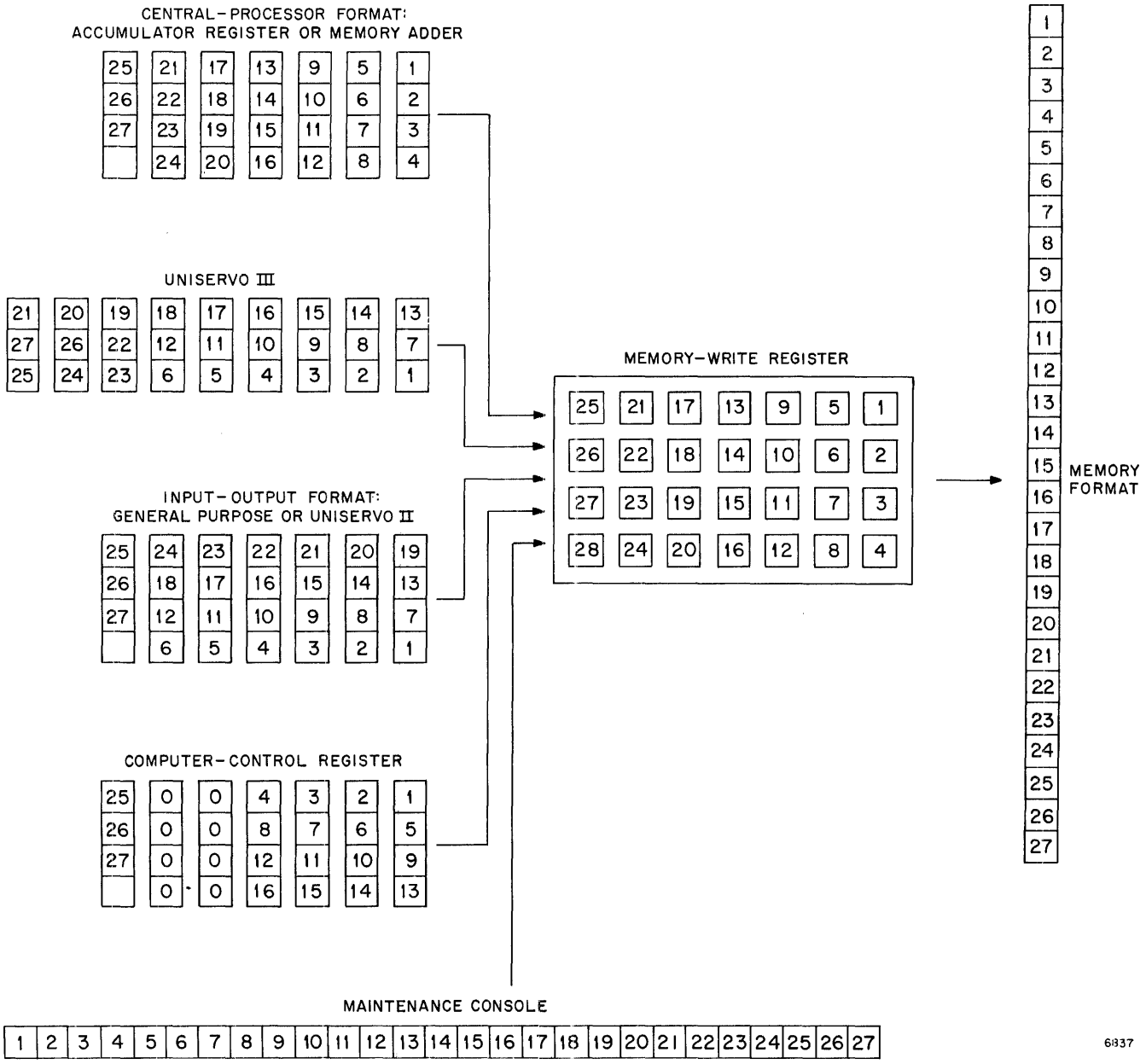


Figure 8-1. Memory-Write Register, Functional Block Diagram

is shifted into the next least-significant-digit position.

8-4. Memory Adder

Output signals CMAF(01—04) from the memory adder are gated in parallel by FS 270 into the four MWr pulseformers, 21 through 24, (figure 8-4). Each digit is shifted each pulsetime by FS 121 to the next least-significant-digit position of the MWr until T1. At T1, the plus sign, 0, is entered into pulseformer 25, and check bits (CMW26J and CMW27J) from the modulo-3 check-bit generator in the memory-address section enter pulseformers 26 and 27. At T2 the MWr is loaded with each bit in its correct pulseformer for writing into memory.

Function signal 270 (figure 8-3) is produced from T5 through T3 during PC1 of all the instructions (represented by signal CVRW1) that transfer information from the memory adder to the MWr. This includes all SAr- and TCWr-store instructions and the initiate-I-O instruction.

8-5. UNISERVO III FORMAT

Outputs from the selected UNISERVO III tape unit (HS21CP, HS22CP, and HS23CP) are gated into the

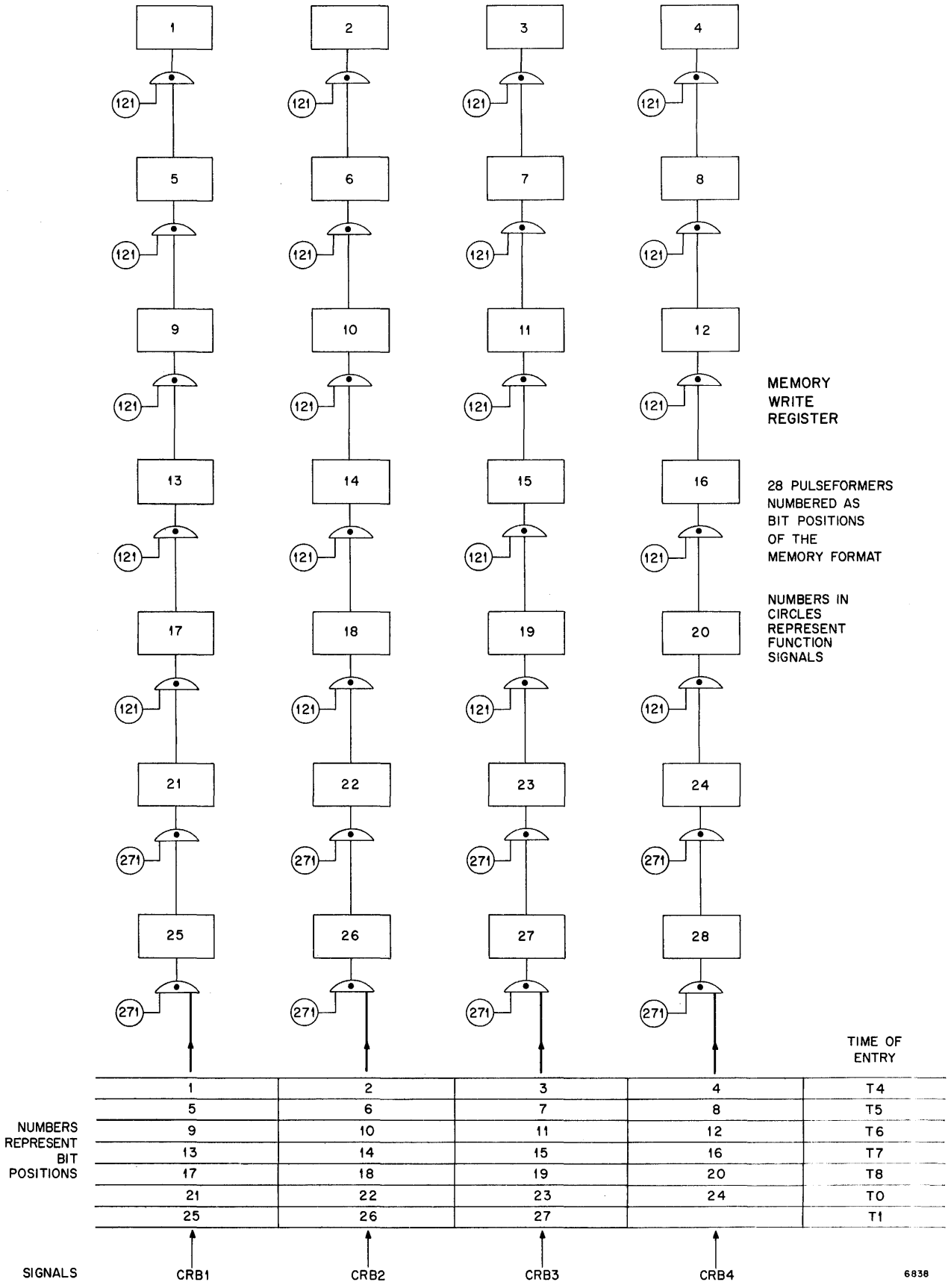


Figure 8-2. Memory-Write Register Loading from the Accumulator Registers, Detailed Block Diagram

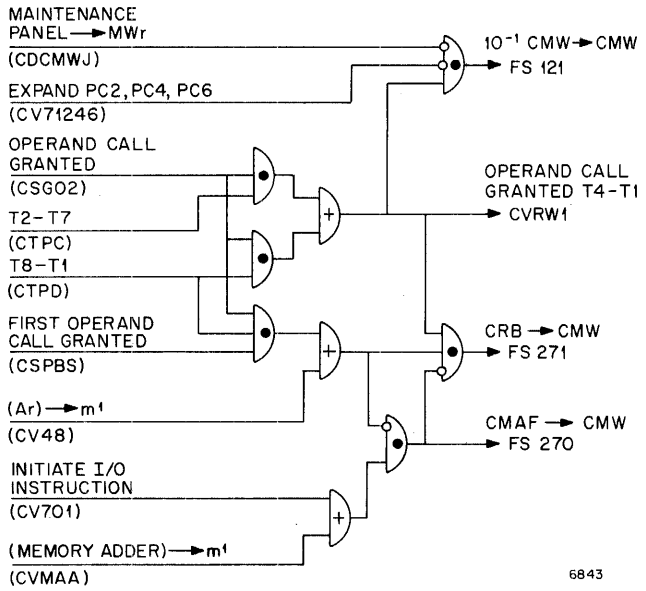


Figure 8-3. Production of CVRW1 and Function Signals 121, 270, and 271, Detailed Block Diagram

MWr for nine pulse times, T2 through T1. Function signal 131 gates the three UNISERVO III inputs in parallel into the MWr pulseformers 25, 27, and 21 (figure 8-5). Three bits enter the MWr each pulse time and then are shifted each succeeding pulse-time, T2 through T1. Bits move through the MWr in three paths and are shifted by FS 131 and FS 148. By T2 the MWr is loaded with each bit in its correct pulseformer for writing into memory.

Function signal 131 (figure 8-6) is produced by the set output from any one of the four write flip-flops, and is present for one minor cycle during entry of UNISERVO III formats into the MWr. Each flip-flop is set by a priority-granted signal for one of the four UNISERVO III channels (CSP03, CSP04, CSP14, or CSP15) and a write signal (HACPWT, HBCPWT, FACPWT, or FBCPWT) from a UNISERVO III synchronizer.

Function signal 148 (figure 8-6) is produced from T3 through T1 for all input-output formats that enter into the MWr, and for program counts 2, 4, and 6 of the expand instruction (71).

8-6. INPUT-OUTPUT FORMAT

All input-output devices, except the UNISERVO III tape unit use the same format. This format is used by the eight general-purpose channels and the one UNISERVO II channel. Inputs from these sources consist of four parallel bits which are entered in parallel into the MWr each pulse time, from T4 through T1, as shown in figure 8-7. General-purpose inputs have CMWG prefixes followed by two numbers. The first number represents the channel, 1 through 8, and the second number represents the input bit order, 1 through 4. For example, channel 2 inputs are CMWG21, CMWG22, CMWG23, and CMWG24. UNISERVO II inputs are UB2CP(1—4).

Table 8-1 shows the function signals that gate the four-parallel input signals of each unit into the MWr and the input signals for each channel. Function signals 132 through 139 gate the inputs from the general-purpose channels into MWr pulseformers 25 through 28, and FS 130 gates the inputs from the UNISERVO II channel into MWr pulseformers, 25 through 28.

Table 8-1. Memory-Write-Register Inputs from UNISERVO II and General-Purpose Channels, and Function Signals

Input-Output Source	Input Signals	Function Signal
UNISERVO II	UB2CP (1—4)	130
General Purpose Channel		
1	CMWG1 (1—4)	132
2	CMWG2 (1—4)	133
3	CMWG3 (1—4)	134
4	CMWG4 (1—4)	135
5	CMWG5 (1—4)	136
6	CMWG6 (1—4)	137
7	CMWG7 (1—4)	138
8	CMWG8 (1—4)	139

The bits are gated by FS 148 and 149 to shift within MWr through four paths. At T2, the MWr is loaded with each bit in its correct pulseformer for writing into memory.

Function signal 130, and FS's 132 through 139 (figure 8-8) are produced by separate gates and buffers that use similar logic. Each of these function signals is produced from T4 through T1 when priority is granted for the specified channel to write into the memory. Timing controls, CTPC and CTPD, alert gates which produce the input function signals. Signal CTPC alerts the function-signal gates from T4 through T7, and CTPD alerts the gates from T8 through T1. Each function signal is produced when either the CTPC signal and the priority-granted signal (CSP) are present, or when the CTPD signal and the memory-granted (CSG) signal are present. The MWr must be loaded while both the priority-granted and memory-granted signals are present, because the loading of the MWr must begin while

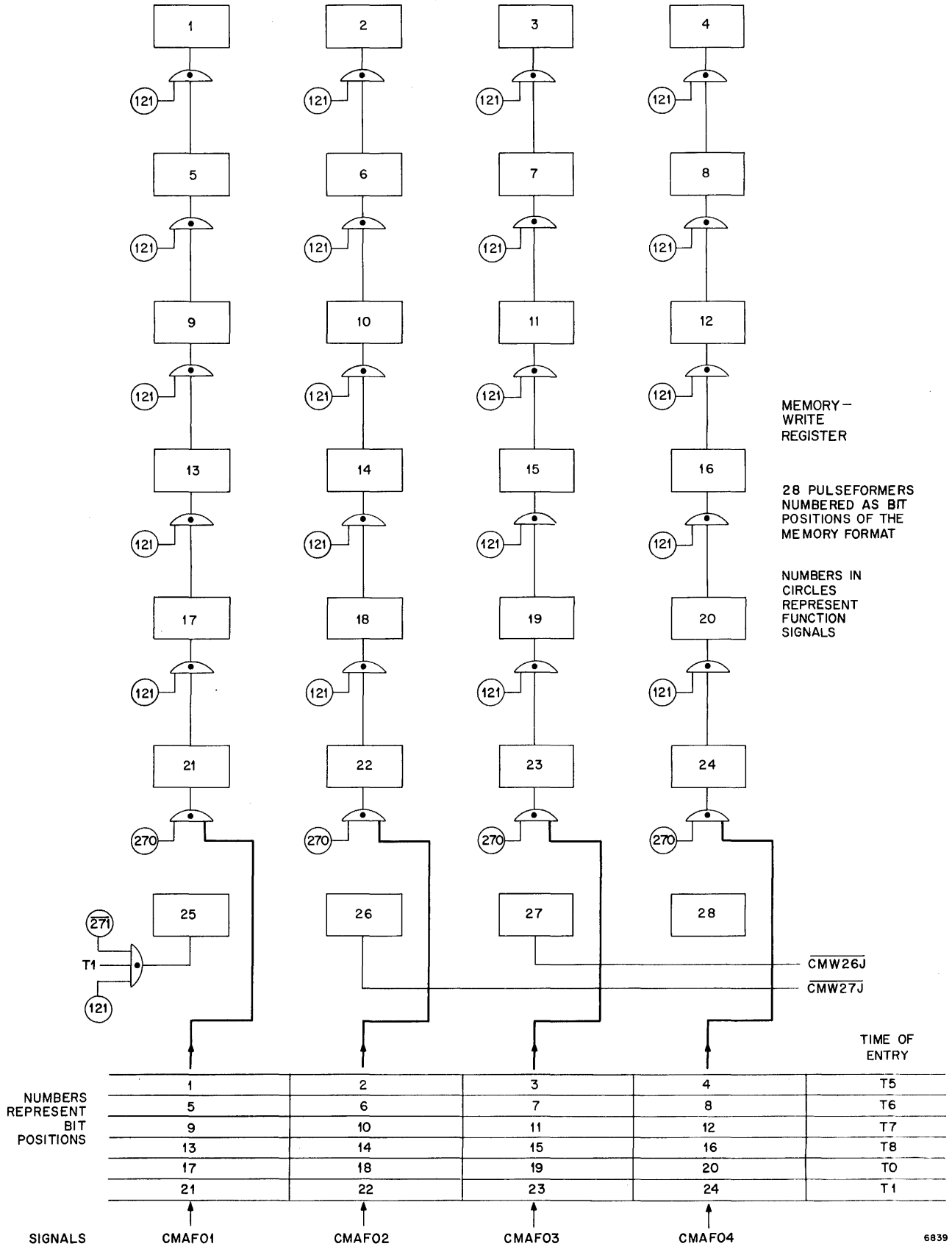


Figure 8-4. Memory-Write Register Loading from the Memory Adder, Detailed Block Diagram

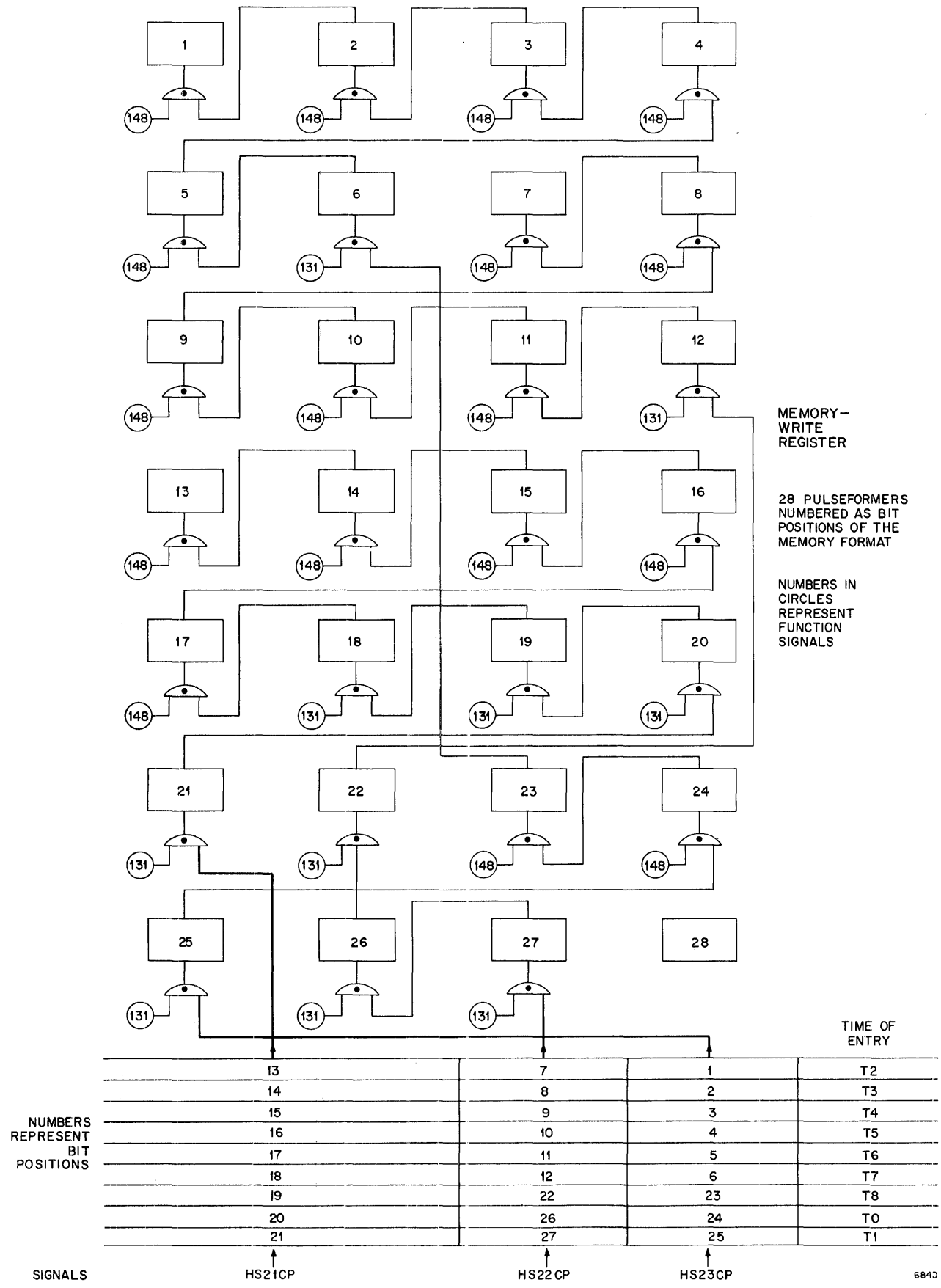


Figure 8-5. Memory-Write Register Loading from the UNISERVO III Channels, Detailed Block Diagram

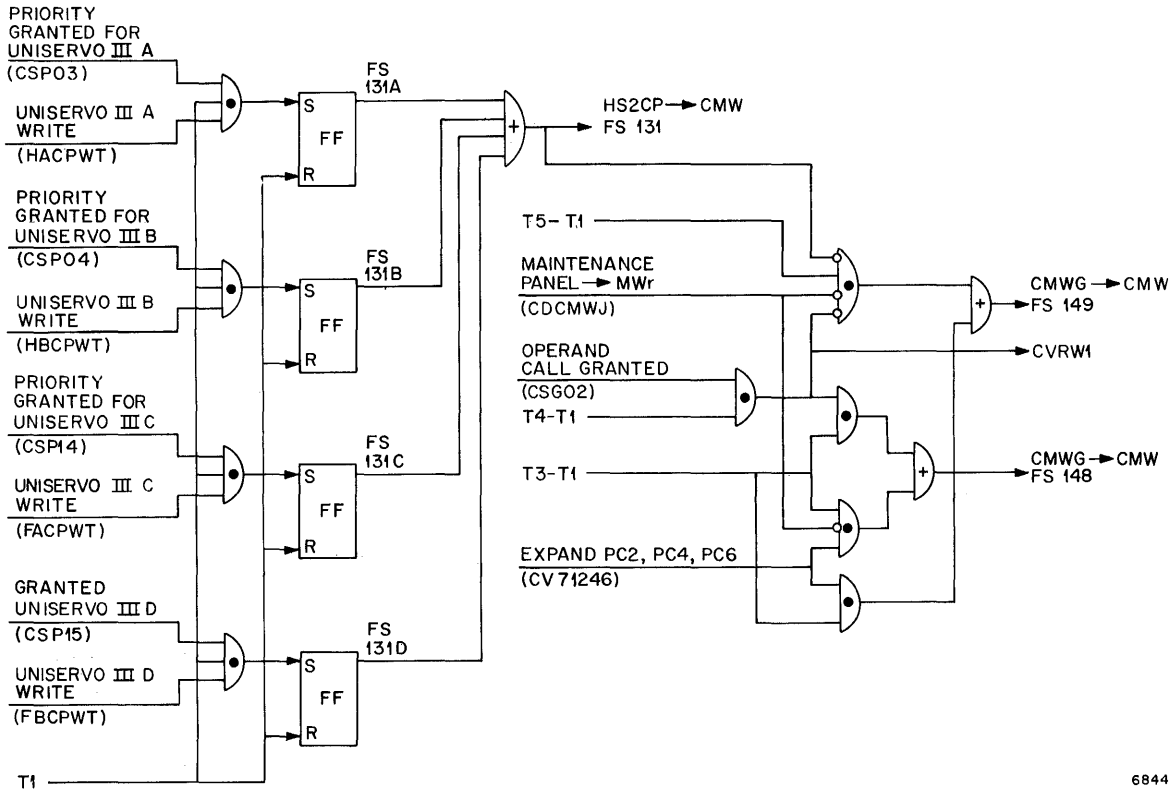


Figure 8-6. Production of Function Signals 131, 148, and 149, Detailed Block Diagram

the priority-granted signal is present and must end while the memory-granted signal is presented. See the timing chart in figure 8-8.

Both FS 148 and FS 149 (figure 8-6) are used for shifting bits through the MWr during input-output format entries into the MWr.

Function signal 149 (figure 8-6) is produced from T5 through T1 and is inhibited during the store-Ar instruction or during a UNISERVO III input. Function signal 149 is present during the entry of input-output-formats to the MWr and also during the expand instruction.

8-7. COMPUTER-CONTROL-REGISTER FORMAT, EXPAND INSTRUCTION ONLY

During the execution of the expand instructions, the two decimal data words contained in two Ar's are expanded into three alphanumeric data words and four decimal digits are positioned in the CCr for each expanded word before the word is transferred from the CCr to the MWr. When the four decimal digits are positioned in the CCr, the outputs from CCr pulseformers, 5, 9, 13, and 17 are gated into MWr from T5 through T8 (figure 8-9). Two zero zone bits are gated into MWr at T0 and T1 to form the alphanumeric characters. The check-bit logic of the main adder operates in such a manner so as to include only the check bits of decimal digits that are part of the expanded word (refer to section 10). The check bits for each word are stored in the quotient-counter register between the time that they are derived and the time that they are gated into the MWr. The sign of the least significant word from the Ar is stored in the sign flip-flop until it is gated into the MWr for each expanded word.

Function signal 277 gates four bits in parallel from the CCr into the MWr pulseformers 6, 12, 18, and 24 (figure 8-6) at T5, T6, T7, and T8. Each pulse time after a digit is entered, the digit is shifted one bit position within the MWr, gated by FS 148 and FS 149. At T0 and T1 the two zero zone bits are gated by FS 278 into the same MWr pulseformers as the CCr bits. Function signal 278 also gates the check bits, CMW26EJ and CMW27EJ, into pulseformers 26 and 27, and the sign bit, CMW25EJ into pulseformer 25 at T1. At T2, the MWr is loaded with each bit of an expanded word in its correct pulseformer for writing into memory.

At T1 during PC1, and T3 and T8 during PC4, function signal 280 gates the check bits from the modulo-3 check-bit generator in the arithmetic section for storage in the quotient-counter FF's, CCQ1 and

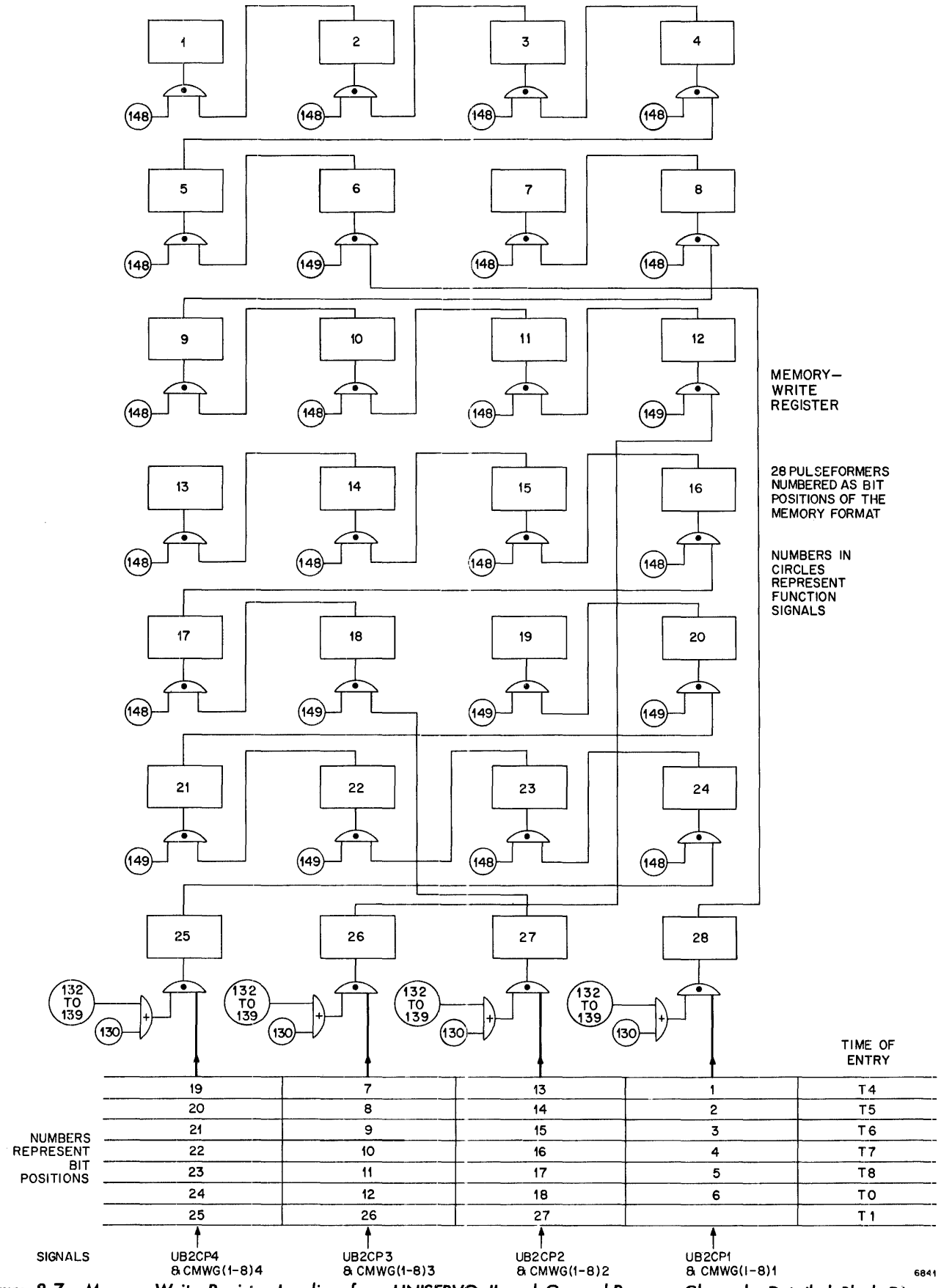


Figure 8-7. Memory-Write Register Loading from UNISERVO II and General-Purpose Channels, Detailed Block Diagram

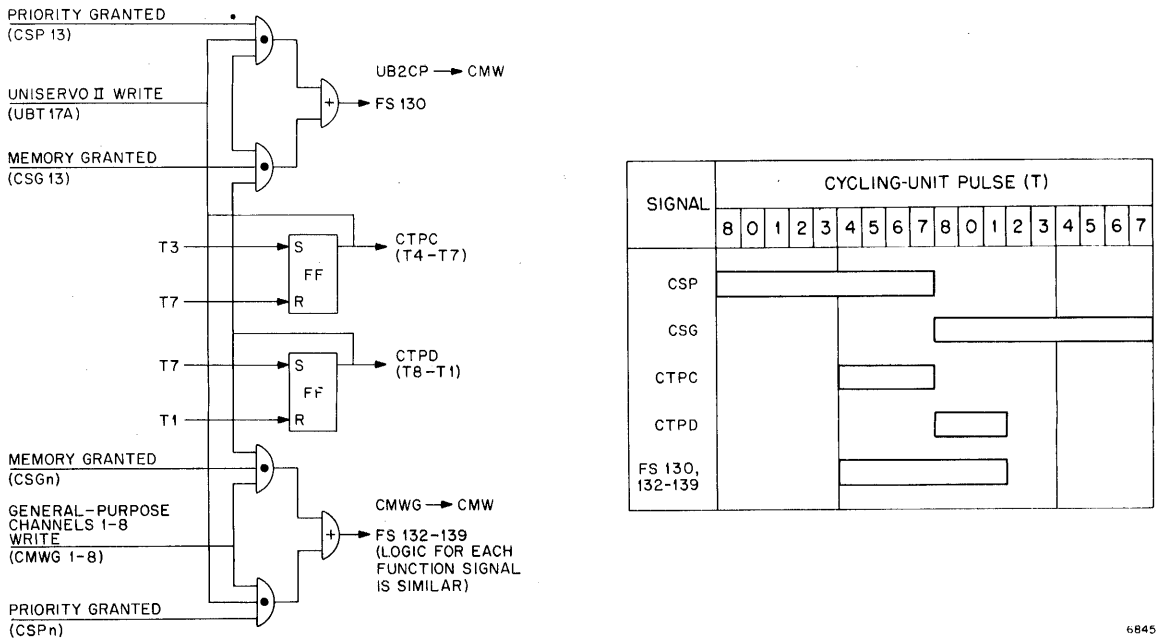


Figure 8-8. Production of Function Signals 130, 132 through 139, Detailed Block Diagram

CCQ2 (figure 8-10). Flip-flop CCQ1 is set by a 0 check bit in bit position 26 (both CAC03A and CAC03B are 0), and FF CCQ2 is set by a 0 check bit in bit position 27 (both CAC02A and CAC02B are 0). Function signal 280 resets the flip-flops to 1 unless overridden by a 0 set signal. Function signal 278 alerts gates to produce $\overline{\text{CMW26EJ}}$ or $\overline{\text{CMW27EJ}}$ to set the MWr for 0 check bits.

Function signal 278 (figure 8-10) is present at T3 and T4, and from T0 through T3 during PC2, PC4, and PC6 of the expand instruction, but is used only at T0 and T1 to enter zero-zone bits, and at T1 to enter check bits and a sign bit. Function Signal 277 (figure 8-10) is present from T5 through T8 of PC2, PC4, and PC6 of the expand instruction, and it is produced by the FS-277 FF.

8-8. MAINTENANCE-PANEL FORMAT

Each of the 27 bits in the word produced from the maintenance console is produced at a two-position switch. The word CDMW(01—27) is entered in parallel into the MWr at T1, and all of the 27 bits are in their correct pulseformers for writing into memory at T2.

8-9. SUMMARY OF FUNCTIONS

Table 8-2 lists all the input and transfer functions performed by the MWr, the function signals, and the pulseformers.

Table 8-3 shows the time each bit enters the MWr from different input formats.

Table 8-2. Summary of Function Signals Which Enter and Gate Inputs in the Memory-Write Register

Function Signal	Transfer Operation	Description	Pulseformers
121	10^{-1} CMW → CMW	Shifts the 4 bits of a digit to next least-significant-decimal position.	1—28
130	UB2CP → CMW	Enters output of UNISERVO II into MWr.	25—28
131	HS2CP → CMW	Enters output of UNISERVO III into MWr and shifts, with FS 148, the output through the MWr.	21, 25, and 27
132—139	CMWG → CMW	Enters output of general-purpose channels into MWr.	25—28
148	CMWG, UB2CP, or HS2CP → CMW	Shifts general-purpose, UNISERVO II, and UNISERVO III formats in binary shift patterns.	1—5, 7—11, 12—17, and 23—24

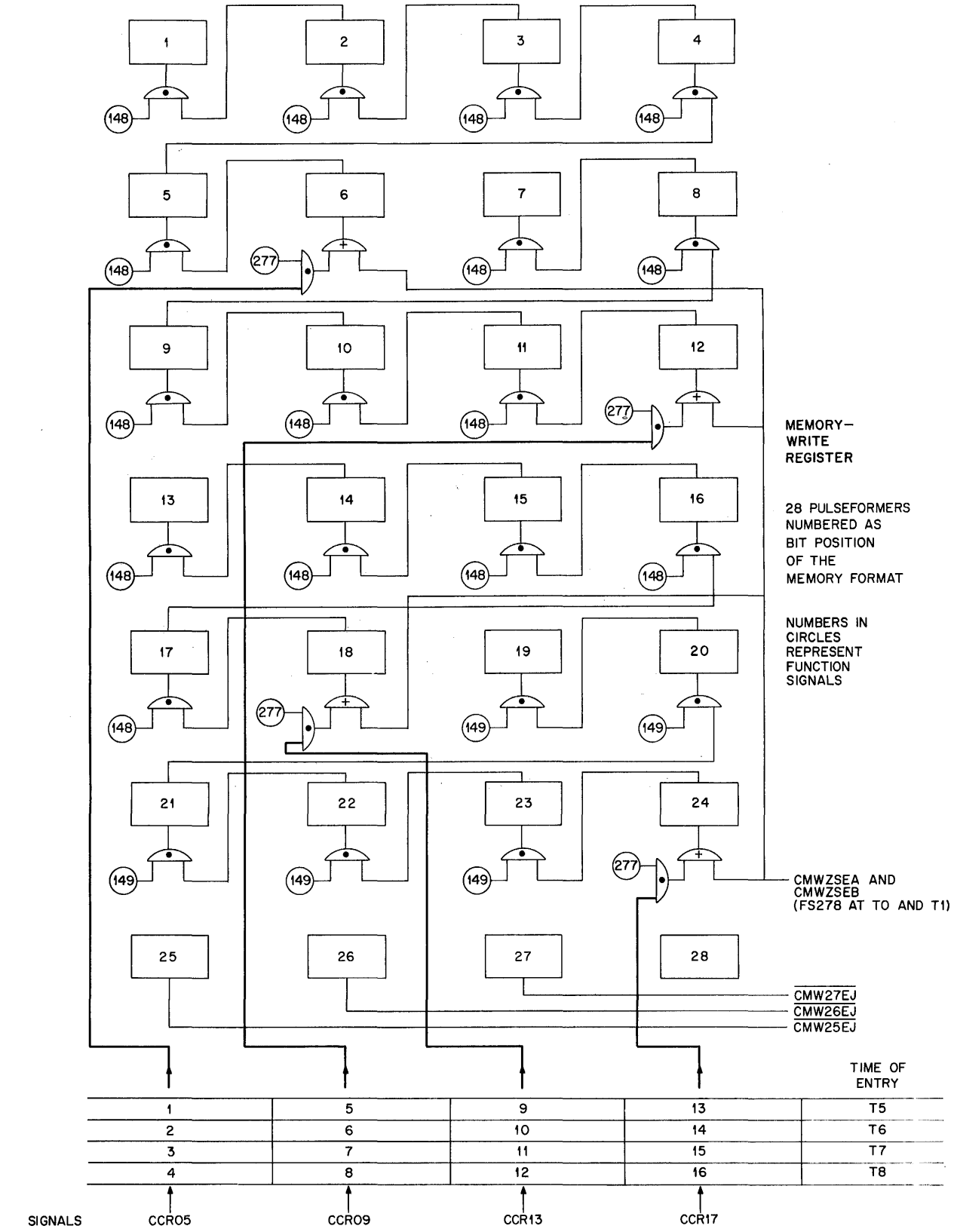
Table 8-2. Summary of Function Signals Which Enter and Gate Inputs in the Memory-Write Register (cont)

Function Signal	Transfer Operation	Description	Pulseformers
149	CMWG or UB2CP → CMW	Shifts general-purpose and UNISERVO II formats in binary shift patterns with FS 148	6, 12, and 18—22
270	CMAF → CMW	Enters output of memory adder into MWr	21, 22, 23, and 24
271	CRB → CMW	Enters output of Ar into MWr.	25, 28
277	CCR → CMW	Enters output of CCr into MWr.	6, 12, 18, and 24
278	Zeros → CMW	Enters 0's into MWr during an expand instruction.	6, 12, 18, and 24
280	Check bits → CMW	Enters check bits CMW26EJ and CMW27EJ into MWr during an expand instruction.	26 and 27
299	Zeros → CMW	Jams 0's into MWr during a SAr- or -TCWr-to-memory instruction.	21—24
121 and 271	Sign bit → CMW	Enters a + sign (zero) at T1.	25
	*CDMW → CMW	Enters maintenance console word into MWr.	1—27

*Enabled by a selection switch on the maintenance console.

Table 8-3. Bit-Entry Sequence of Input Formats Into Memory-Write Register

Input-Format Sources	Time of Bit Entry									
	T2	T3	T4	T5	T6	T7	T8	T0	T1	T2
Accumulator register CRB(1—4)			1	5	9	13	17	21	25	
			2	6	10	14	18	22	26	
or			3	7	11	15	19	23	27	
Memory adder CMAF(01—04)			4	8	12	16	20	24		
UNISERVO III HS 21CP, HS22CP, and HS23CP	13	14	15	16	17	18	19	20	21	
	7	8	9	10	11	12	22	26	27	
	1	2	3	4	5	6	23	24	25	
General-purpose channels CMWG(1—8) (1—4)			19	20	21	22	23	24	25	
			13	14	15	16	17	18	27	
or			7	8	9	10	11	12	26	
UNISERVO II UB2CP(1—4)			1	2	3	4	5	6		
Computer-control register CCR05, CCR09, CCR13, and CCR17			1	2	3	4	0	0	25	
			5	6	7	8	0	0	26	
			9	10	11	12	0	0	27	
			13	14	15	16	0	0		
Maintenance console CDMW01 through CDMW27									1—27	



NOTE:
NUMBERS INDICATE BIT POSITIONS IN THE FIRST WORD TO BE TRANSFERED DURING THE EXPANSION INSTRUCTION (71)

6842

Figure 8-9. Memory-Write Register Loading from the Computer-Control Register During Expand Instruction, Detailed Block Diagram

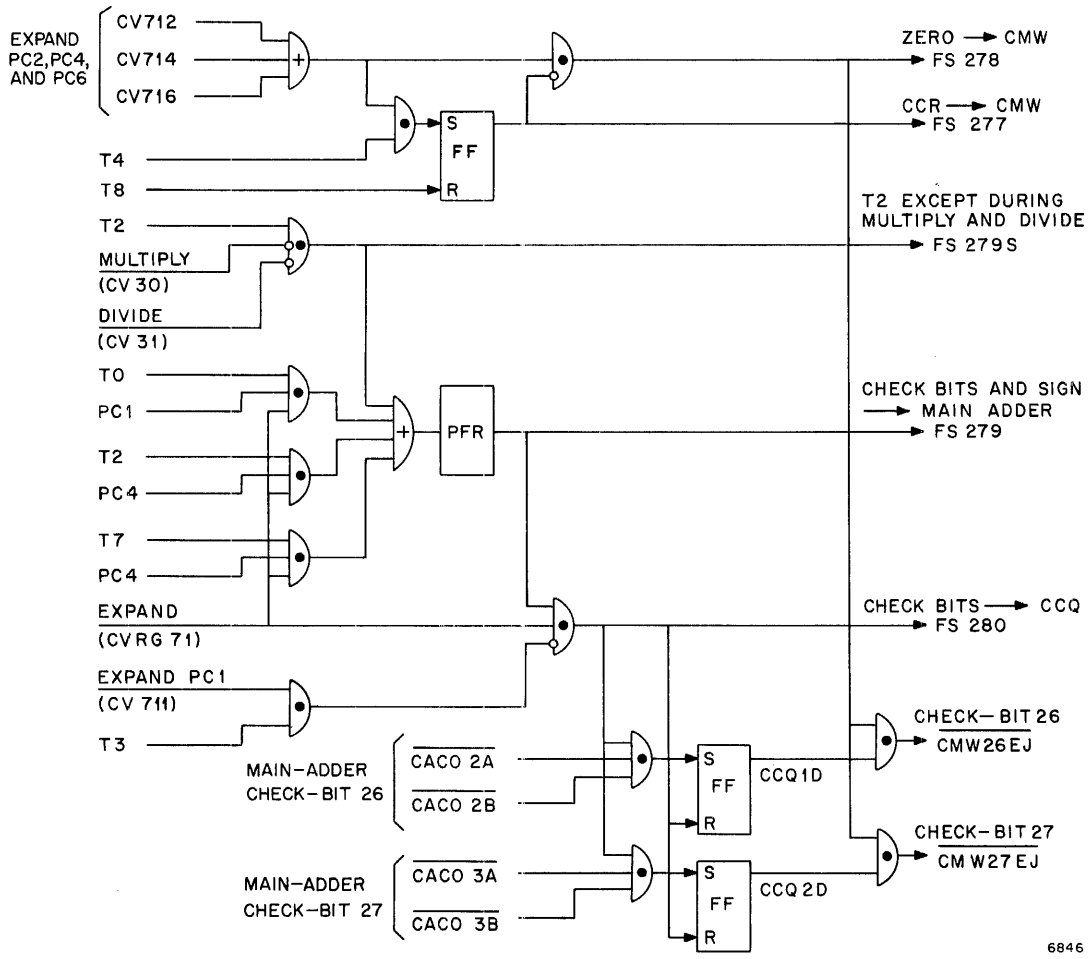


Figure 8-10. Production of Function Signals 277, 278, and 280, and the CMW26EJ and CMW27EJ Signals, Detailed Block Diagram

Appendix A
ENGINEERING-DRAWING SOURCES OF ILLUSTRATIONS

Table A-1. Figure References and Engineering Drawings for Section 2

To be supplied when available

Table A-2. Figure References and Engineering Drawings for Section 3

Text Reference		Engineering Drawing		Contents
Figure	Table	Number	Revision	
3-2	—	3812 001	H	CVr and first stage of octal decoding
3-2	—	3812 002	E	Second stage of octal decoding
—	—	3812 003	M	Group encoder
3-3	—	3812 004	N	Program counter
—	—	3812 005	K	Instruction-line decoder
—	—	3812 006	J	Instruction-line decoder
—	—	3812 007	L	Instruction-line decoder
—	—	3812 008	N	CVEP, instruction requests, and operand requests
3-2, 3-9, 3-10, 3-11	—	3812 009	M	CVRINH, FS 127, FS 266, FS 122, FS 267, FS 268, FS 260, CVRECP, CVRRG, and CCVA
3-6	—	3812 013	R	FS 235, FS 236, FS 238, FS 128, and FS 129
3-5	3-1, 3-2, 3-3	3812 026	K	AR selector
3-7	3-4	3812 027	L	Precision-select network
3-8	—	3812 108	J	Ir select-network and flip-flops

Table A-3. Figure References and Engineering Drawings for Section 4

Text Reference		Engineering Drawing		Contents
Figure	Table	Number	Revision	
4-1, 4-2, 4-3		3812 125	S	Priority request for priority granted FF's
4-1, 4-2, 4-3		3812 126	S	Priority-decode network
4-1, 4-2, 4-3		3812 127	F	Priority-decode network

Table A-4. Figure References and Engineering Drawings for Section 5

Text Reference		Engineering Drawing		Contents
Figure	Table	Number	Revision	
		3812 012	S	Production of FS 141
		3812 013	N	Production of CV4227
		3812 014	R	Production of CVBR5, FS 143, FS 144, FS 146, FS 249, FS 257, FS 274, FS 275, and FS 276
		3812 015	R	Production of FS 147 and FS 261
5-3, 5-4, 5-5, 5-6, 5-7		3812 070	L	CCr FF's 23, 24, 27, and 28
5-3, 5-4, 5-5, 5-6, 5-7		3812 071	R	CCr FF's 21, 22, 25, and 26

Table A-4. Figure References and Engineering Drawings for Section 5 (cont)

Text Reference		Engineering Drawing		Contents
Figure	Table	Number	Revision	
5-3, 5-4, 5-5, 5-6, 5-7		3812 072	K	CCr FF's 15, 16, 19, and 20
5-3, 5-4, 5-5, 5-6, 5-7		3812 073	K	CCr FF's 13, 14, 17, and 18
5-3, 5-4, 5-5, 5-6, 5-7		3812 074	K	CCr FF's 7, 8, 11, and 12
5-3, 5-4, 5-5, 5-6, 5-7		3812 075	H	CCr FF's 5, 6, 9, and 10
5-3, 5-4, 5-5, 5-6, 5-7		3812 076	M	CCr FF's 3 and 4
5-3, 5-4, 5-5, 5-6, 5-7		3812 077	M	CCr FF's 1 and 2

Table A-5. Figure References and Engineering Drawings for Section 6

Text Reference		Engineering Drawing		Contents
Figure	Table	Number	Revision	
6-3		3812 014	S	FS 106
6-2		3812 100	H	MAr, CC, MAC3, and Ir's 1—4
6-2		3812 101	F	MAC's 5—8 and Ir's 5—8
6-2		3812 102	G	MAC's 9—10 and Ir's 9—10
6-2		3812 103	F	MAC's 11—12 and Ir's 11—12
6-2		3812 104	G	MAC's 13—15 and Ir's 13—15

Table A-5. Figure References and Engineering Drawing for Section 6 (cont)

Text Reference		Engineering Drawing		Contents
Figure	Table	Number	Revision	
6-2		3812 105	E	TCWr's 1—2
6-2		3812 106	D	TCWr's 3—4
6-2		3812 107	H	MAC Delay
6-2		3812 108	J	MAC and Ir output selection
6-2		3812 109	F	MAC, Ir, and TCWr output selection
6-2		3812 110	L	MAC, Ir, and TCWr input selection (from MAS); FS's 156—170 and FS's 204—207
6-2		3812 112	E	MAC, Ir, and TCWr output buffer
6-3		3812 138	V	FS 100, FS 108, FS 107, and FS 106S
6-3		3812 139	H	Clearing of Ir, MAC, and TCWr; FS's 200—203 and FS's 171—185
6-4	6-2	3812 140	K	MAC input selection from SCr; FS's 222—234
6-3		3812 143	S	TCWr's input selection from SCr; FS's 208—211

Table A-6. Figure References and Engineering Drawings for Section 7

Text Reference		Engineering Drawing		Contents
Figure	Table	Number	Revision	
7-8		3812 015		FS 244 and FS 250
7-1		3812 113	U	Memory adder
7-5, 7-6	7-1, 7-2, 7-3	3812 114	N	Address-parity generator and check-bit generator
7-7, 7-8		3812 115	N	MAS
7-10		3812 129	K	Zero detector
7-2, 7-3, 7-4		3812 138	V	FS's 103, 103A, 105, 102, 109, 214, 245, 284—287, 220, and 269A
7-9		3812 142	S	Cabinet selector
7-8		3812 014	R	FS 298

Table A-7. Figure References and Engineering Drawings for Section 8

Text Reference		Engineering Drawing		Contents
Figure	Table	Number	Revision	
8-10		3812 001	M	Production of FS's 277, 278, 279S, 279, and 280
8-7		3812 014	R	Production of FS 270 and FS 271
8-10		3812 028	K	Production of CMW26EJ and CMW27EJ check bits
8-2, 8-3, 8-4, 8-5, 8-6		3812 116	K	MWr pulseformers 1—8 and production of CMWZSEA
8-2, 8-3, 8-4, 8-5, 8-6		3812 117	J	MWr pulseformers 9—16
8-2, 8-3, 8-4, 8-5, 8-6		3812 118	M	MWr pulseformers 17—24 and production of CMWZSEB
8-2, 8-3, 8-4, 8-5, 8-6		3812 119	J	MWr pulseformers 25—28
8-7		3812 138	U	Production of FS 121 and CVRW1
8-5, 8-8, 8-9	8-1	3812 142	P	Production of FS's 130—139, FS 148, and FS 149