UNIVAC 9700 System

Processor

Programmer Reference

This document contains the latest information available at the time of publication. However, the Univac Division reserves the right to modify or revise its contents. To ensure that you have the most recent information, contact your local Univac Representative.

UNIVAC is a registered trademark of the Sperry Rand Corporation.

Other trademarks of the Sperry Rand Corporation in this publication are:

UNISCOPE

PRINTED IN U.S.A.

PAGE STATUS SUMMARY

ISSUE: Update C to UP-7936

Part/Section	Page Number	Update Level
Cover/Disclaimer		C
PSS	1	C
Contents	1 thru 7	C
1	1 2 thru 6 6a 7 thru 9 10 11	Orig. C C* C Orig. C
2	1 thru 3 4 thru 12 13, 14 14a 15, 16 16a 17, 18 19 thru 21 22 23, 24 25, 26 27 thru 30 31 32 33 34 thru 41	C Orig. C C* C C* C Orig. C Orig. C Orig. C Orig. C Orig.
3	1 thru 3 4 4a 5, 6 6a 7, 8 8a 9 thru 11 12 13 thru 22 23, 24 24a 25 thru 27 28 thru 30 30a 31 thru 37	Orig. C C* C C* C C* Orig. C Orig. C C* Orig. C C* Orig.
4	1 2 3, 4 4a 5 6 thru 15	C Orig. C C* C Orig.
5	1 thru 19	Orig.
6	1 thru 15	Orig.

Part/Section	Page Number	Update Level
Appendix A	1 2 thru 7 8 thru 10	C Orig. B
Appendix B	1 thru 3	Orig.
Index	1 thru 11	C
User Comment Sheet		
Total: 189 pages including covers		

Part/Section	Page Number	Update Level

*New pages

CONTENTS

3. INPUT/OUTPUT SECTION

5. SYSTEM CONSOLE

6. INCREMENTAL PRINTER

APPENDIXES

FIGURES

TABLES

I. INTRODUCTION

1.1. GENERAL

This reference contains the information for programming the UNIVAC 9700 System (Figure 1-1). It describes the internal operation of the processor, main storage, and the I/O channels. The formats of the system status and control words are detailed. Also provided is a brief description of the optional features available for expansion of the processor and main storage.

The UNIVAC 9700 System consists of a processor, console, expandable plated-wire main storage, one multiplexer channel, up to four selector channels, and UNIVAC 9000 Series byte-oriented peripheral devices. Information pertaining to individual peripheral devices may be obtained from their respective programmer and operator references.

Appendix A lists machine instructions with their associated hexadecimal operation (op) code. The instructions are in alphabetical order according to instruction name. Also listed are the five instruction-type formats. A detailed description of these codes and instructions is given in the *UNIVAC 9700 System OS/4 Assembler Programmer Reference, UP-7935* (current version).

Appendix B, the glossary, is an alphabetic listing of all mnemonics described in this reference.

Figure 1–1. UNIVAC 9700 System

1.2. FUNCTIONAL DESCRIPTION

The UNIVAC 9700 Processor (processor) is a medium scale, microprogrammed, general purpose processor that may be used for commercial, scientific, or real time data processing. The processor operates in word parallel with a repertoire of fixed-point binary, hexadecimal floating-point, decimal, arithmetic, logical, status switching and branching instructions. It has a flexible and safe storage protection scheme. Comprehensive storage relocation and protection, with eight levels of indirect addressing algorithm, provide dynamic relocation and rollin/rollout of portions of the operating software and user (worker) programs.

One multiplexer channel and up to four selector channels permit the attachment of a wide variety of control units and peripheral devices. Main storage has a minimum capacity of 131K bytes, expandable to more than 1048K bytes.

The two basic types of internal processor operations are the data path logic, which is data manipulations; and control, which is control signal generation. The I/O channel operates as an independent data handling system which operates in parallel with the processor.

The processor operates under microprogram control and contains a recovery timer, storage protection, and register stack. Microprogramming refers to the technique whereby certain logic functions, such as incrementing the instruction address field of the current program status word, accessing the instruction operands in storage and general registers, and fetching the next instruction, are controlled by series of microinstructions. These microinstructions are located in the control storage (COS) which is a storage medium dedicated to the microprogram. A microprogram actually controls the execution of all program instructions. Figure 1–2 illustrates the basic functions of the processor and its interrelationship with all essential data and control paths.

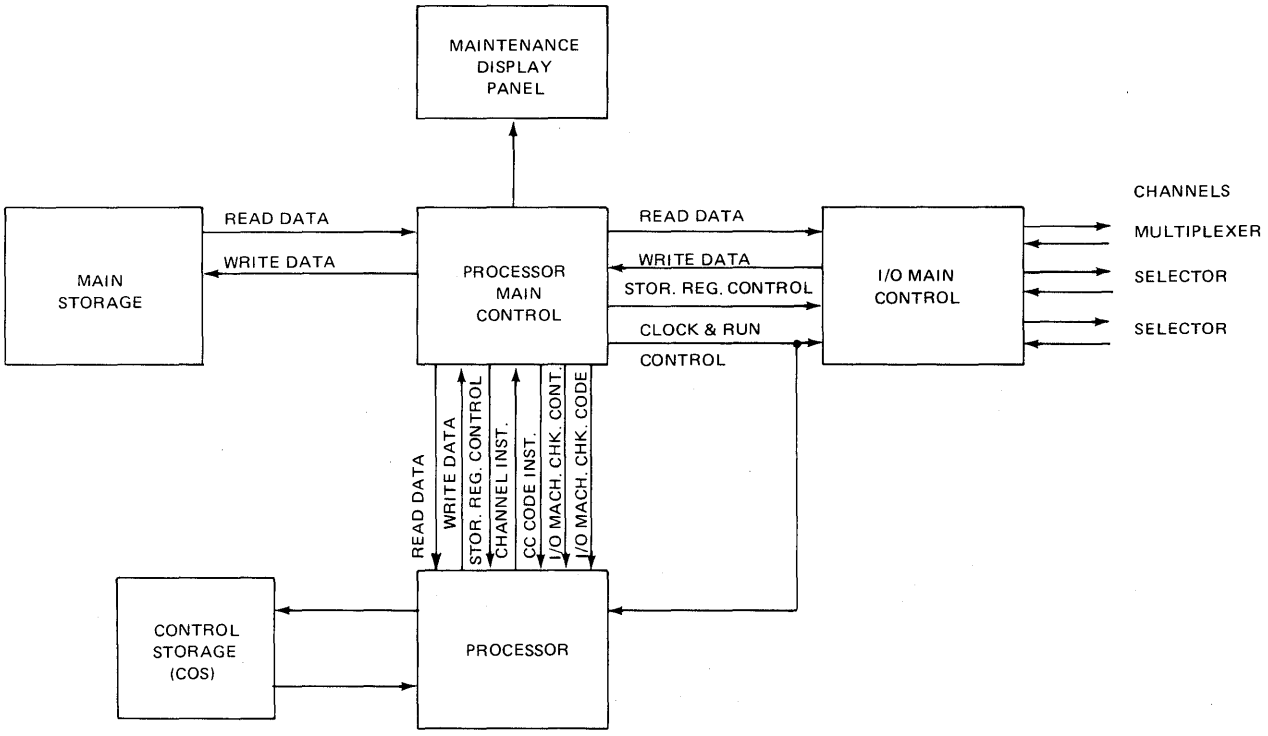


Figure 1–2. UNIVAC 9700 Processor, Functional Block Diagram

1.2.1. Arithmetic Section

The arithmetic section performs all logical operations, arithmetic operations, data comparisons, and shifting. Fixed-point binary arithmetic uses twos complement number representations. Floating-point arithmetic and decimal arithmetic use signed absolute-value number representation. The arithmetic section also performs single and double indexing of operand addresses, together with address relocation and indirect addressing. This section contains a register stack comprised of:

- General purpose registers
- Floating-point registers
- Working registers

1.2.1.1. GENERAL PURPOSE REGISTERS

A register stack contains the general registers in two sets of 16 each. These registers can be used for fixed-point arithmetic, logical operations, and the indexing (except register) of instructions and operand addresses.

- Fixed-Point Arithmetic

  Fixed-point numbers have a fixed, predetermined binary point (or power) and are held in the twos complement form. When the sign bit is set to 1, the integer represents a negative value; when the sign bit is 0, the integer represents a positive value. When held in one of the 16 general registers, a fixed-point number is generally treated as a 32-bit operand. When a half-word fixed-point number is called from main storage and loaded into a register, the sign is extended to the left to fill the full-word register. The contents of the register are then handled as a full-word operand in fixed-point arithmetic operations.

  Double-word operations use a 64-bit operand consisting of one sign bit followed by a 63-bit integer field. The 64-bit operand is located in two adjacent general registers, and it is addressed by an even address referring to the lower-numbered register of the pair.

  When fixed-point data is located in storage, it may be stored as a half word, full word, or double word. This data must be located on the integral storage boundary of its associated format.

- Logical Operations

  Logical operations such as comparing, translating, editing, bit setting, and bit testing are also performed by the arithmetic section of the processor on data stored in the general purpose registers or in main storage and the results are placed back into the general purpose registers or main storage.

1.2.1.2. FLOATING-POINT REGISTERS

Four floating-point registers are available. Each floating-point register has a capacity of 64 bits to accommodate either the short format (one word) or the long format (one double word). A short operand occupies the high order bit positions of the register. The low order half of the register is ignored and remains unchanged in short precision floating-point operations. ←

The floating-point registers are addressed by various 4-bit fields in the instruction being executed by the processor. The operation code being executed determines whether a given 4-bit field addresses the floating-point registers or the general registers. The floating-point registers have addresses 0, 2, 4, and 6.

1.2.1.3. WORKING REGISTERS

Decimal arithmetic operations are performed in eight working registers. Decimal number fields can be variable in length and can exist in two formats: packed decimal numbers and unpacked decimal numbers.

The packed decimal format is used for all decimal arithmetic operations. In the packed format, each byte contains two digits. The least significant four bits of the least significant byte provides the sign of the number.

In the unpacked decimal format, each byte contains one digit of a multidigit number. The byte is divided into two equal fields: zone and digit. A zone value is represented in the most significant four bits, and the digit is represented in the least significant four bits. The zone portion of the least significant byte specifies the sign of the number. The unpacked format must be used when data is to be processed by certain I/O devices, such as a printer.

Instructions are provided for converting decimal numbers from packed to unpacked and from unpacked to packed format.

1.2.2. I/O Section

The I/O section initiates, directs, and monitors the transfer of data between main storage and the peripheral subsystems. After the I/O instruction has been initiated, the data transfer is performed concurrently with other processor functions. The I/O section contains multiplexer and selector channels that have a standard UNIVAC 9000 Series I/O interface.

1.2.2.1. MULTIPLEXER CHANNEL

The multiplexer channel controls the transfer of data between the processor or main storage and I/O subsystems. Connections between the multiplexer channel and the I/O subsystems are by way of the 8-bit compatible I/O interface. When an operation is initiated, the channel maintains control of data transfers between the I/O subsystem and main storage. This control allows the processor to proceed concurrently with I/O operations.

Basic characteristics of the multiplexer channel are:

- Up to 175-KB transfer rate.
- One to two UNIVAC 9000 Series compatible interfaces, each with eight drops.
- → Up to 63 subchannels.
- Elaborate error detection and reporting capability.
- Complete command and data chaining capability.

There are two types of subchannels associated with I/O operations: standard and data communication. The multiplexer channel is able to distinguish between these two types of subchannels by the format of the device address presented to the channel during an I/O instruction or by way of a request initiated by the control unit.

The multiplexer channel can operate in either the multiplex mode or the control-unit-forced-burst mode. In multiplex mode, the channel facilities are shared by a number of concurrently operating I/O devices, with the I/O interface being assigned to a control unit only long enough to transfer one byte of data. Upon completion of this data exchange, the I/O interface is assigned to another control unit requesting service, and the operation proceeds in a similar manner. In the control-unit-forced-burst mode, the control unit stays connected to the I/O interface until as many bytes of data as required for the duration of the operation are transferred. Termination is accomplished either by the channel having the byte count expire or by the control unit.

1.2.2.2. SELECTOR CHANNEL

One selector channel is provided with the basic system configuration; three additional selector channels are available as optional features. Eight standard control units may be attached to each selector channel. Up to 16 I/O devices can be attached to each of the eight control units, depending on the particular subsystem selected. The devices attached to a selector channel are serviced on a burst mode (one-at-a-time) basis; that is, once transfer of data is initiated between a particular device and main storage, that transfer must be completed before another device on that channel can transfer data.

Because a subchannel is required to maintain data transfer with only one subsystem at a time, there is only one subchannel in the selector channel. Thus, the channel and subchannel in the selector channel may be considered to be identical entities. Also, another I/O device may be executing a previously initiated operation which does not require communication with the channel such as rewinding a tape. An operation may be terminated by either the channel or the control unit; thus, the selector channel is free to monitor requests for status presentation from another I/O device or another processor instruction. Basic characteristics of the selector channel are:

- UNIVAC 9000 Series compatible interface
- Up to 1,111-KB transfer rate
- Up to four selector channels
- Complete command and data chaining capability

1.2.3. Main Storage

The main storage consists of high speed plated wire. Main storage is contained in freestanding cabinets with a 600-nanosecond read or write cycle time for a full word (2.3). Minimum storage size is 131,072 bytes expandable to a maximum main storage size of 1,048,576 bytes. Addresses and data are checked for odd parity. Each main storage cabinet is equipped with seven address switches. These switches are used as the seven most significant bits of the 24-bit cabinet starting address. If it becomes necessary to place a 131,072-byte storage offline, the switches on the cabinet with the highest addresses can be set to the starting address of the cabinet removed from service. The remaining online cabinets then can have sequential addresses. Only the three lower switches are used for the cabinet starting address. The remaining four top switches are not used but are provided to accommodate expansion beyond 1048K main storage.

Basic characteristics of main storage are:

Characteristic	Description
Type of storage	Plated wire
Capacity	131,072 bytes (minimum) 196,608 bytes 262,144 bytes 393,216 bytes 524,288 bytes 655,360 bytes 786,432 bytes 917,504 bytes 1,048,576 bytes (maximum)

Characteristic	Description
Cycle Time	600 nanoseconds to read or write one word (four bytes). 600 nanoseconds to write either one, two, or three bytes.
Operating Modes	Nondestructive read/write.
Storage Data Path	36 bits wide; consisting of four bytes (where a byte is defined as eight bits) plus a parity bit associated with each byte. The state of the parity bit is such that the parity of the 9 bits (byte plus parity bit) is odd.
Checking	Storage protection, write or read/write. (The storage protection check is performed in the processor.) Address and data parity. Nonexistent main storage location is referenced.

1.2.4. Control Storage

Control storage is the hardware element that holds the microinstructions microprogrammed for control logic. It is a high speed nondestructive read/write storage (RWM). The COS consists of two independent and separately addressable sections: an address calculation (AC) section, and an operand manipulation (OM) section. Each section contains a program nonalterable (PN) memory and a program alterable (PA) memory. The former type of memory is utilized to store the basic instruction set and floating-point microinstructions while the PA type memory is used to store microdiagnostics, programmable emulators, etc.

A map of the COS is shown in Figure 1–2A.

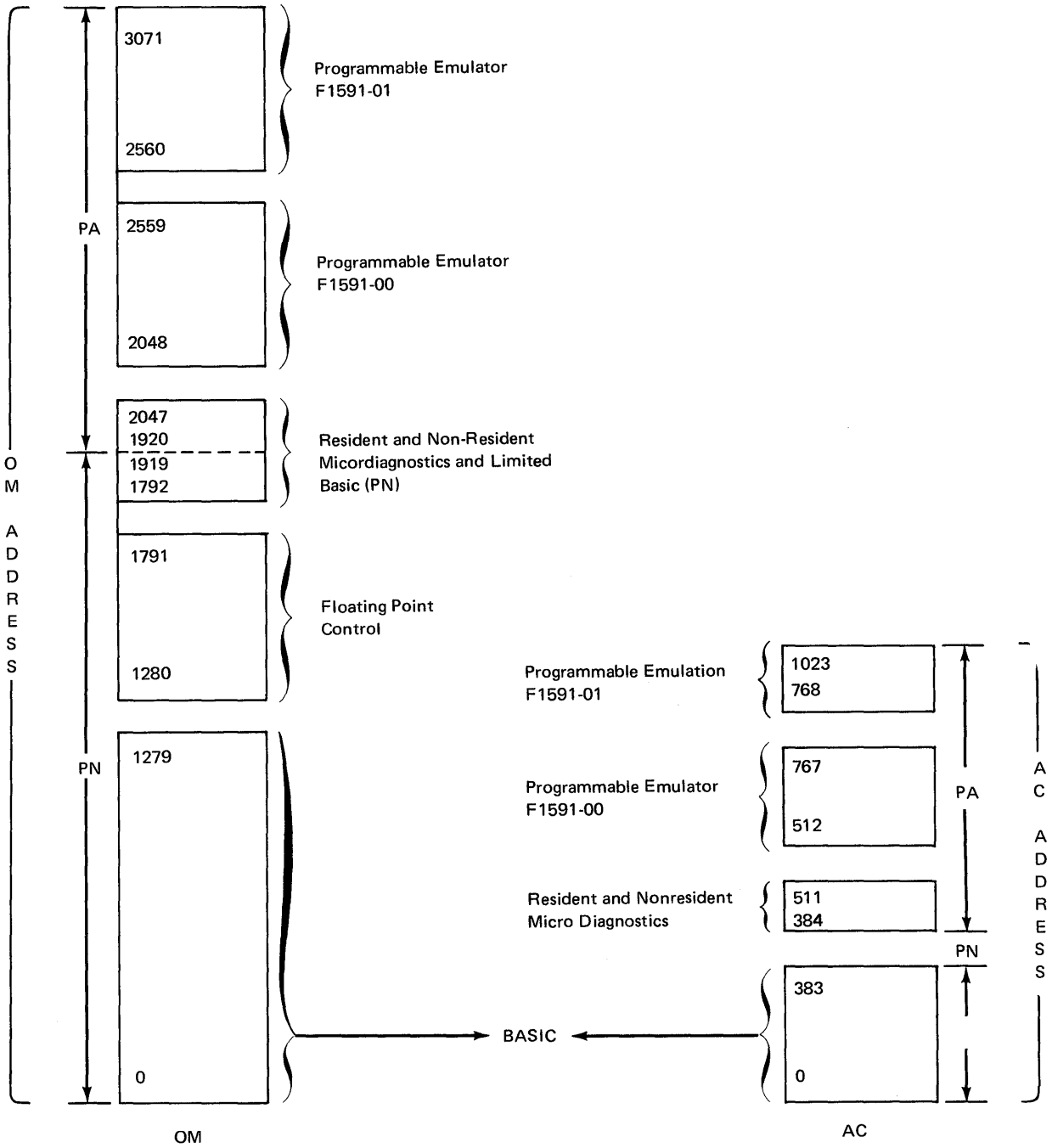
1.3. CONFIGURATION

The configurations and optional features for the processor and the operator console are described in the following paragraphs.

1.3.1. Processor

The configuration for the processor is shown in Figure 1–3. The solid-line boxes define the basic components, and the dashed-line boxes define the optional features available.

The optional features available for the processor are listed and described in Table 1–1.



NOTES:

1. OM denotes operand manipulation section
2. AC denotes address calculation section
3. PN denotes program nonalterable
4. PA denotes program alterable
5. OM address is out of bounds at OM address 2048_{10} if F1591-00 is not installed and at 2560_{10} if F1519-01 is not installed.
6. AC address is out of bounds at AC address 512_{10} if F1591-00 is not installed and at 768_{10} if F1519-01 is not installed.

Figure 1–2A. Control Storage Map

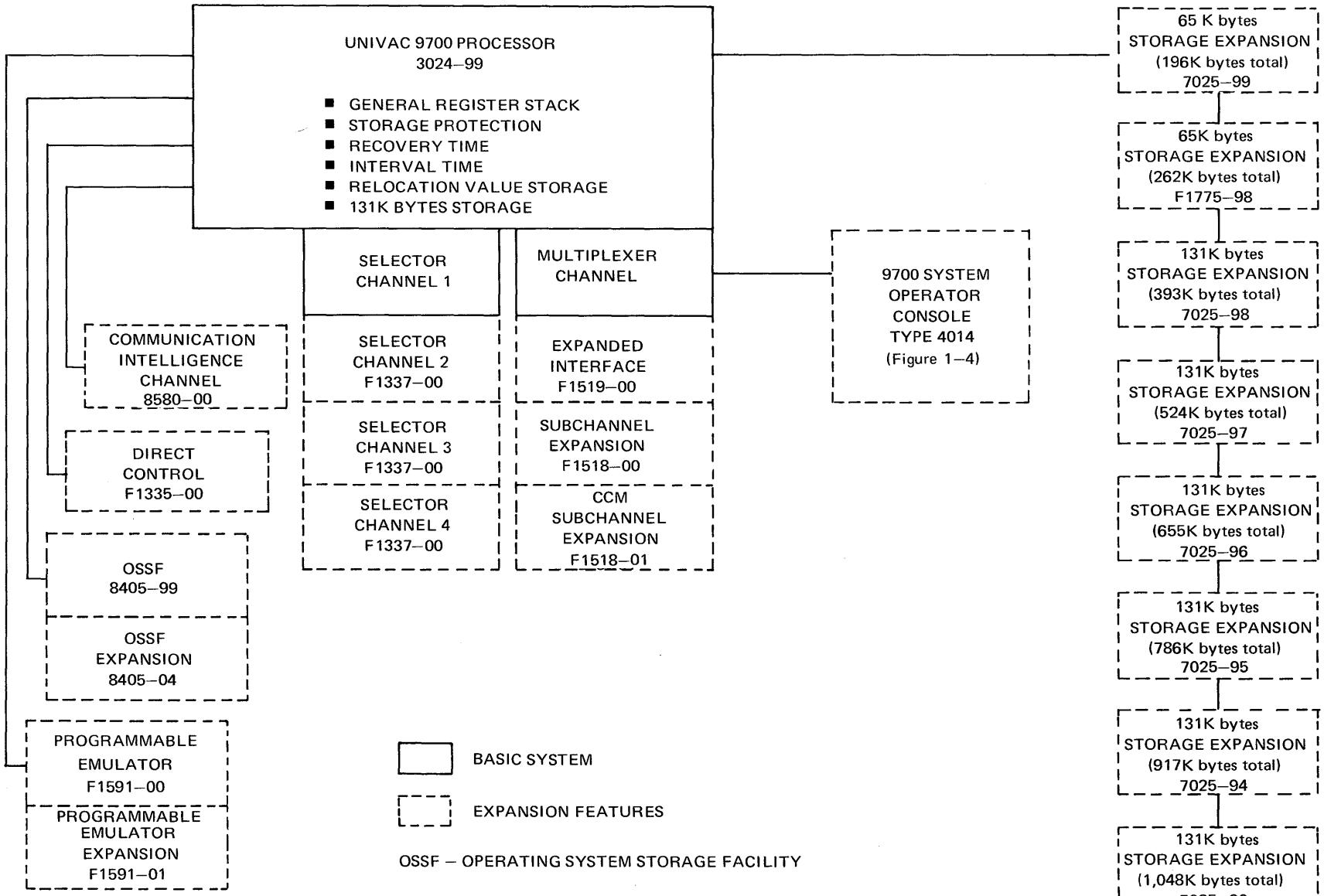


Figure 1–3. System Configuration for UNIVAC 9700 Processor

Feature/Type Number	Name	Description
F1335-00	Direct Control	Provides a special interface between the 9700 processor and another central processing unit (for example, another 9700 processor) equipped with the same or compatible capability, plus two instructions for transfer of control information, and generation of external interrupts.
F1337-00	Selector Channel	Provides a 1.11 megabyte (833-KB) channel with up to 8-subsystem capability. Includes channel programming and main storage protection. Maximum of three F1337-00 per system. (Maximum of four selector channels including basic.) Selector channel 2 housed in processor cabinet. Channel expansion cabinet (type 1916-00) is prerequisite to house selector channels 3 and 4.
F1591-00	Programmable Emulator	Provides programmable control for emulation of: IBM 1401, 1440, and 1460 programs; Series 70 301 and 501 programs, and Series 70 mode programs.
*F1591-01	Programmable Emulator Expansion	Provides programmable control, using special hardware instructions, for concurrent operation of any two of the emulators listed under F1591-00 which is prerequisite to this feature.
F1519-00	Expanded interface	Expands multiplexer channel interface to provide up to 15-subsystem capability (up to 16 subsystems if F1518-00 is installed).
F1518-00	Subchannel Expansion	Expands the number of subchannels per multiplexer channel to 31.
*F1518-01	Communications Controller Multichannel (CCM) Subchannel Expansion	Expands the number of subchannels per multiplexer channel to 63 by providing 32 additional subchannels. F1518-00 (subchannel expansion) and type 0973-00 (standard interface adapter) are prerequisites to this feature.
8405-99	Operating System Storage Facility	Includes a special channel, control, and one disc file (type 8405-04) with a minimum of 3 million bytes of direct access storage. Average latency is 8.34 milliseconds. May be expanded by up to seven additional disc files (type 8405-04).
8405-04	Operating System Storage Facility Expansion	Provides 3 million bytes additional direct access storage to the operating system storage facility. Average latency is 8.34 milliseconds.
8580-00	Communications Intelligence Channel	Provides a flexible, communications dedicated, input/output channel.
7025-99	Storage Expansion – 65K	Provides additional 65,536 bytes of main storage. Expands main storage from 131,072 bytes to 196,608 bytes.
F1775-98	Storage Expansion – 65K	Provides additional 65,536 bytes of main storage. Expands main storage from 196,608 bytes to 262,144 bytes.
7025-98	Storage Expansion – 131K	Provides additional 131,072 bytes of main storage. Expands main storage from 262,144 bytes to 393,216 bytes.

*Feature or type is utilized in UNIVAC 9700/Series 70 mode of operation

Table 1–1. Optional Features (Part 1 of 2)

Feature/Type Number	Name	Description
7025-97	Storage Expansion – 196K	Provides additional 131,072 bytes of main storage. Expands main storage from 393,216 bytes to 524,288 bytes.
7025-96	Storage Expansion – 262K	Provides additional 131,072 bytes of main storage. Expands main storage from 524,288 to 655,360 bytes.
7025-95	Storage Expansion – 131K	Provides additional 131,072 bytes of main storage. Expands main storage from 655,360 bytes to 786,432 bytes.
7025-94	Storage Expansion – 131K	Provides additional 131,072 bytes of main storage. Expands main storage from 786,432 bytes to 917,504 bytes.
7025-93	Storage Expansion – 131K	Provides additional 131,072 bytes of main storage. Expands main storage from 917,054 bytes to 1,048,576 bytes.
4014-00/01	System Console	Provides operator controls, CRT display with entry keyboard.
0772-00/01	Console Printer	Provides hard copy printing of selected console output data at up to 30 characters per second.
2519-00/01	Multi-Channel Switch (MCS)	Provides one 2 x 1 switch and space for up to six switches for switching a subsystem or subsystem string between two UNIVAC 9000 Series I/O channels.
F1541-00	MCS Expansion	Provides one additional 2 x 1 switch to type 2519. Up to five of these switches may be added to type 2519.

*Feature or type is utilized in UNIVAC 9700/Series 70 mode of operation

Table 1–1. Optional Features (Part 2 of 2)

1.3.2. Operator Console

The minimum operating configuration for the UNIVAC 9700 System Operator Console (operator console) consists of a modified UNISCOPE 100 Display Terminal, an operator panel, and a control unit (Figure 1–4).

Optional features available for the UNIVAC 9700 System Console (system console) consist of the console printer, channel adapter, and multi-channel switch (MCS) control. These features are listed in Table 1–2.

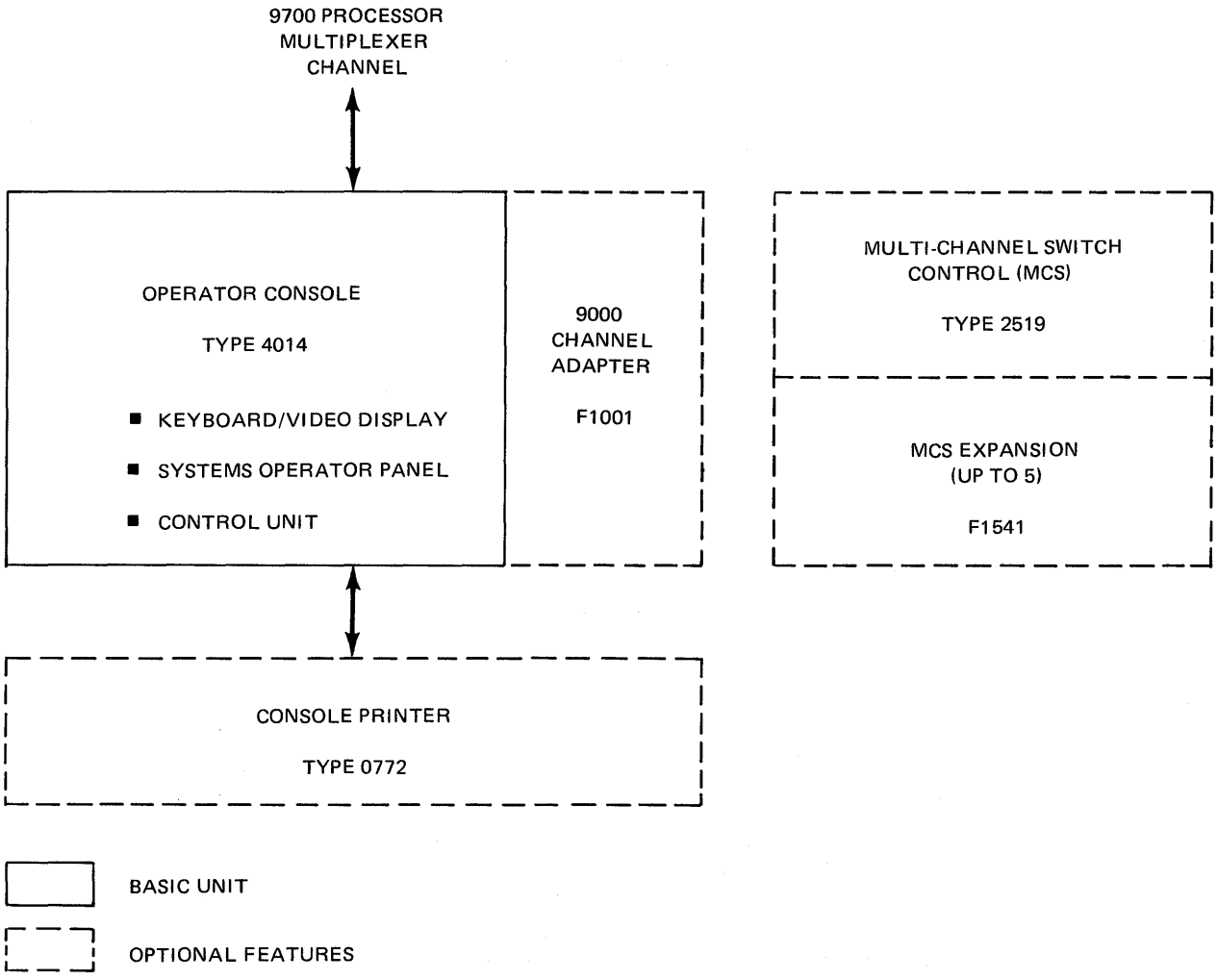


Figure 1–4. UNIVAC 9700 System Console Configuration

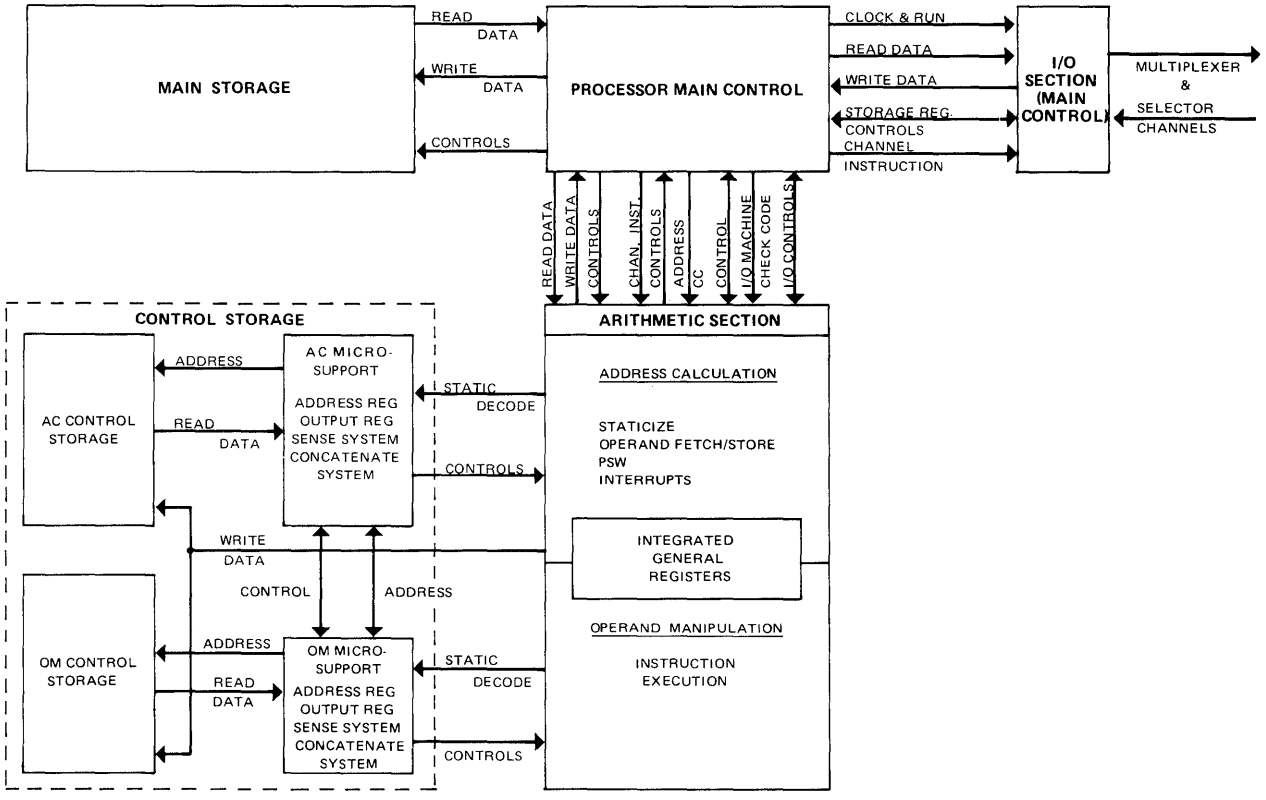
Feature/Type Number		Name	Description
60 Hz	50 Hz		
4014-00	4014-01	Operator Console	
0772-00	0772-01	Console Printer	A cabinet providing a printer mechanism and control, power, cooling, and interface to the system operator control. The printer has the full 94-character ASCII set plus space and is capable of printing at a 30-character-per-second rate.
F1001-01	F1001-01	9000 Channel Adapter	This feature provides for the control and transfer of data between the UNIVAC 9700 System channel and one of the following UNIVAC systems: 9200, 9200 II 9300, 9300 II, and 9400.
2519-00	2519-01	Multi-Channel Switch (MCS) Control	A freestanding cabinet containing one basic 2 x 1 switch and providing power and space for up to five F1541-00 switches. Operator control is provided in a separate housing located on top of the console. One switch assembly is supplied with provisions for five additional switches at the operator console panel. The MCS permits a subsystem or series of subsystems to be switched (statically) between I/O channels on the same or different UNIVAC 9000 Series processors.
F1541-00	F1541-00	Multi-Channel Switch Expansion (MCSE)	Provides expansion of one additional switch for the multichannel switch control.

Table 1–2. UNIVAC 9700 System Console, Basic Components and Optional Features

2. ARITHMETIC SECTION

2.1. GENERAL

The arithmetic section of the UNIVAC 9700 Processor (processor) is comprised of two independent sections: the address calculation (AC) section and the operand manipulation (OM) section. Each section is part of the control storage and has an associated address, output, and decoding logic. These sections can operate independently and in parallel to each other.



The AC section contains the hardware for the performance of instruction staticizing, operand fetch and storeback, maintenance of the program status word (PSW), and interrupt processing to include the interrupt initialization sequence (IIS).

The OM section contains the hardware to perform the execution of the instructions to manipulate the data (operands) to be processed. The OM section is the more powerful of the two sections and has the most addressing flexibility. The AC section normally acts as a slave to the OM section.

The splitting of the arithmetic section into two sections allows staticizing of the next instruction to be overlapped with the completion of the execution of the current instruction. Also, operand fetches of a storage-to-storage (SS) instruction are overlapped with the execution.

2.2. REGISTERS

The processor contains a register stack with a capacity of 48 words. This stack contains three types of registers: general, floating point, and working. One additional software identifiable register called the current relocation register is in the arithmetic section but is not part of the register stack. See Figure 2–1 for processor register allocations.

2.2.1. General Registers

The processor can address one of the two sets of general register stacks (supervisor and problem) under control of the PR (bit 18) of the current program status word (PSW, 2.4). Each stack contains 16 registers that may be used for indexing, fixed-point arithmetic, logical operations, and temporary storage. Each register has a capacity of 32 bits. The registers are addressed by particular 4-bit fields in the instruction being executed by the processor.

Certain instructions use a double-word operand which is held in two adjacent registers. The register address in the instruction field must be an even number. The register pair consists of the addressed register and the next higher-numbered register. The even-address register contains the most significant half of the double word, and the odd-address register contains the least significant half of the double word.

2.2.2. Floating-Point Registers

Four floating-point registers are provided, each having a capacity of 64 bits. They can accommodate either the short format (one word) or the long format (a double word). A short operand occupies the high order bit position of the register. The low order bit positions of the register are disregarded and remain unchanged in short precision floating-point operations.

The floating-point registers are addressed by various 4-bit fields in the instruction being executed. The operation code being executed determines whether a given 4-bit field addresses the floating-point registers or the general registers. The floating-point registers have addresses 0, 2, 4, 6. Data is in floating-point format.

2.2.3. Working Registers

Eight working registers are used for temporary storage of operands and intermediate results during decimal arithmetic operations. Each register is 32 bits in length. The registers used are dictated by the operation (op) code; therefore, they are not directly specified in related user-level instructions.

2.2.4. Current Relocation Registers

The current relocation register is considered the fourth general register. This register is automatically loaded from low order main storage whenever the key in the current PSW is changed. The new key is used as a pointer to access the appropriate relocation register (4.6.1.3). The availability of the relocation address factor in a register accelerates the calculation of the required storage addresses for operand or instruction access. This register provides the original address relocation from the relocation registers located in low order main storage.

Hardware Register Number	Program Specified Register Number	Program Status Word PR Bit 18	Size	Register
0	0	0	WORD (32 BITS)	SUPERVISOR GENERAL REGISTERS (16)
1	1	0		
2	2	0		
3	3	0		
4	4	0		
5	5	0		
6	6	0		
7	7	0		
8	8	0		
9	9	0		
10	10	0		
11	11	0		
12	12	0		
13	13	0		
14	14	0		
15	15	0		
16	0	1	WORD (32 BITS)	PROBLEM (USER) GENERAL REGISTERS (16)
17	1	1		
18	2	1		
19	3	1		
20	4	1		
21	5	1		
22	6	1		
23	7	1		
24	8	1		
25	9	1		
26	10	1		
27	11	1		
28	12	1		
29	13	1		
30	14	1		
31	15	1		
32	0		DOUBLE WORD (64 BITS)	FLOATING-POINT REGISTERS (4)
33				
34	2			
35				
36	4			
37				
38	6			
39				
40	(not directly specified by program)		32 BITS	WORKING REGISTERS (8)
41				
42				
43				
44				
45				
46				
47				
			20 BITS	CURRENT RELOCATION REGISTER (1)

Figure 2–1. Processor Registers

2.3. INFORMATION FORMATS

Data and instructions are transmitted in single or multiple 8-bit increments called bytes. Up to four bytes of information may be transmitted in parallel between various sections of the system. Instructions are made up of half-word lengths and can be one, two, or three half words long.

A half word is defined as a field containing two consecutive bytes and is basic for the instructions. A word is a field containing four consecutive bytes; a double word is a field containing two words. Figure 2-2 illustrates the various word formats and bit numbering.

Fixed-length fields such as half words, words, and double words are located in main storage on a integral boundary for the unit of information. Instructions must be located on half-word boundaries. An integral boundary is defined as main storage address for a unit of information which is a multiple of the length of the unit in bytes, as follows:

Half word (two bytes) – multiple of 2

Full word (four bytes) – multiple of 4

Double word (eight bytes) – multiple of 8

Storage addresses are expressed in binary form within the processor. Integral boundaries for half words, words, and double words are given as binary addresses in which one, two, or three of the low order bits are 0, respectively.

Variable-length fields are not limited to integral boundaries and can start at any byte address.

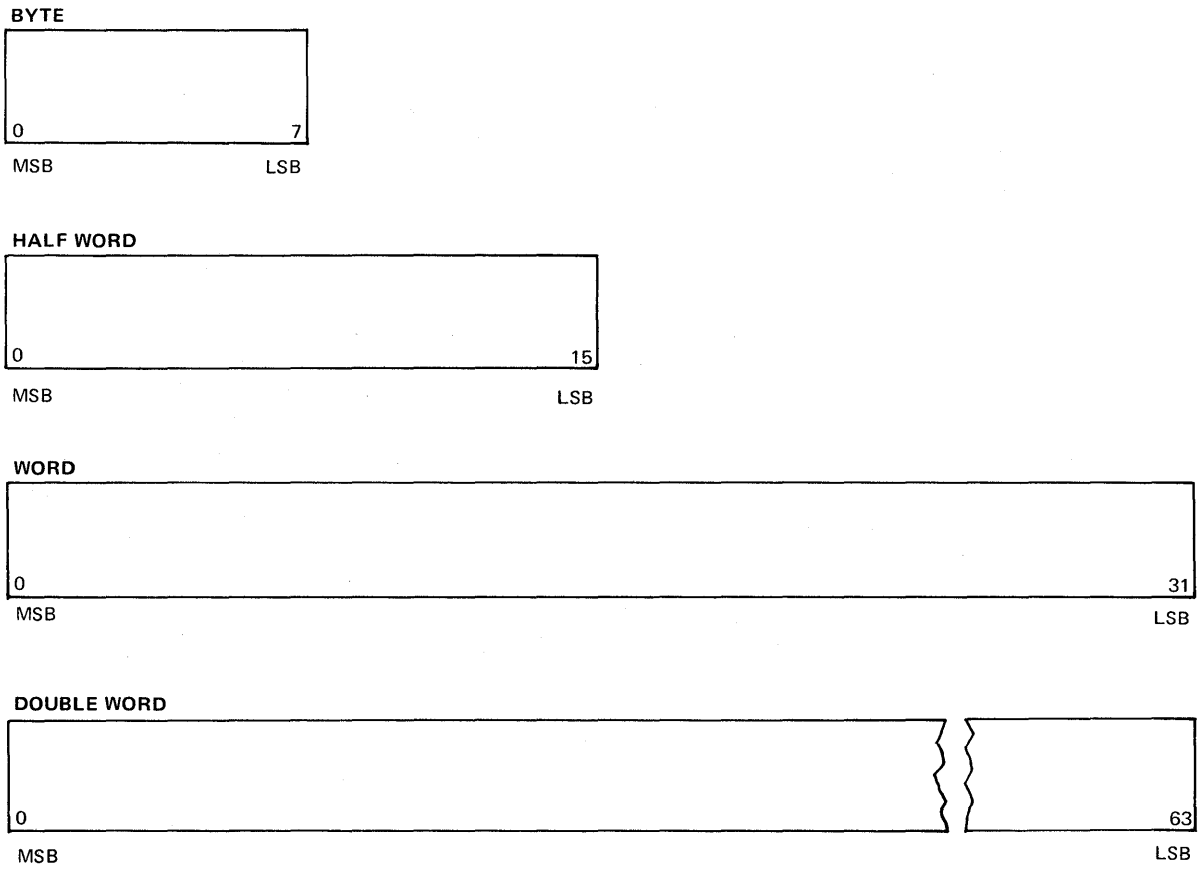


Figure 2–2. Information Formats

2.3.1. Data Formats

Data is represented in several different formats, depending on the type of instruction which is to manipulate the data.

2.3.1.1. FIXED-POINT NUMBERS

Fixed-point numbers are represented in one of threee fixed-length formats consisting of one sign bit (bit 0) followed by a binary integer field. When the sign bit is set to 1, the integer represents a negative value; when set to 0, the integer represents a positive value. Negative integers are represented in the twos complement notation.

The half-word, word, and double-word fomats are as follows:

HALF WORD

WORD

DOUBLE WORD

When held in one of the 16 general registers, a fixed-point number is treated as a 32-bit operand. Certain multiply, divide, and shift operations use a 64-bit operand consisting of one sign bit followed by a 63-bit binary integer field. A 64-bit operand is located in two adjacent registers and is addressed by referring to the even-numbered register of the even/odd pair.

2.3.1.2. FLOATING-POINT NUMBERS

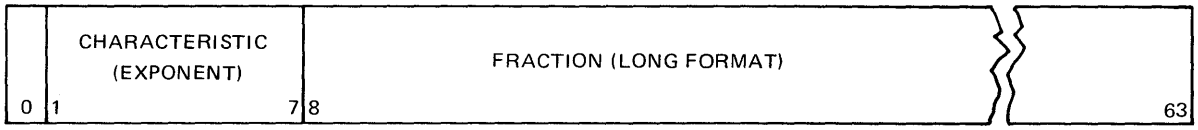
Floating-point numbers are represented in signed magnitude form and have a fixed-length format which is either a word (short format) or a double word (long format). Both formats may be used in storage or in floating-point registers. In either format, the sign bit (bit 0) is the sign of the fraction and bits 1 through 7 are the characteristic. The fraction field in the short form consists of bits 8 through 3l; in the long format, it consists of bits 8 through 63 (which represent 6 and 14 hexadecimal digits, respectively).

The characteristic is a biased exponent expressed in excess-64 binary notation. The fraction is expressed as a hexadecimal number having the radix point to the left of the high order fraction digit. The quantity expressed by the full floating-point number is the product of the fraction and the number 16 raised to the power of the characteristic minus 64.

The short and long formats are as follows:

WORD

DOUBLE WORD

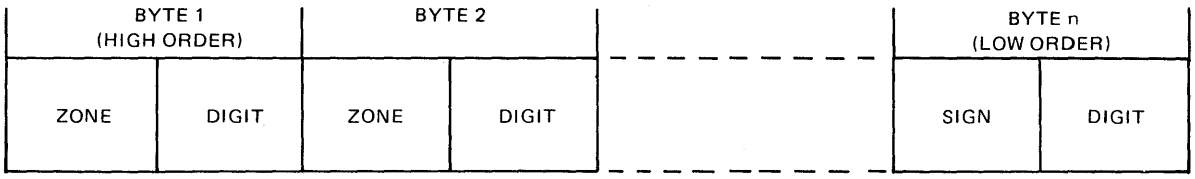


2.3.1.3. DECIMAL NUMBERS

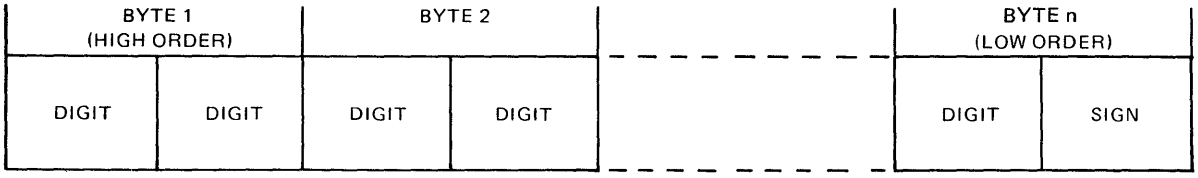
Decimal numbers are represented in signed absolute form and in either packed or unpacked formats with variable lengths. These are used in decimal arithmetic and logical operations. For decimal operations, the numbers are again processed from data stored in the general purpose registers or in main storage and the results are placed back into the general purpose registers or into main storage, requiring a field area of from 1 to 16 eight-bit bytes.

The packed and unpacked decimal formats are shown as follows:

UNPACKED DECIMAL FORMAT



PACKED DECIMAL FORMAT



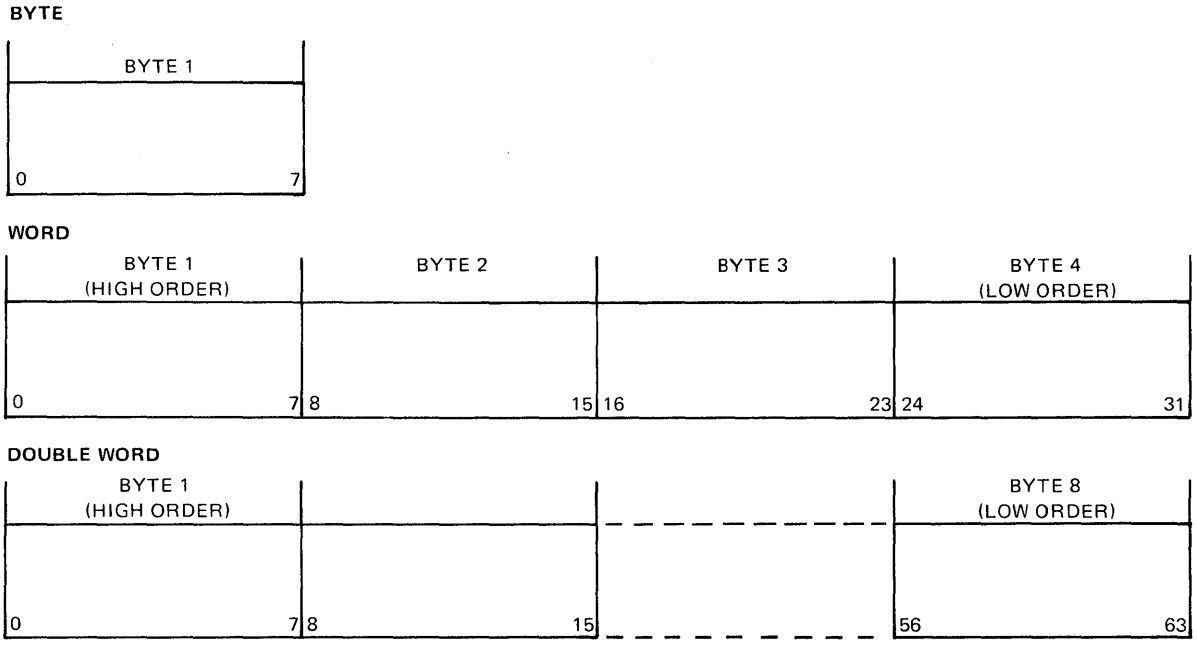
2.3.1.4. LOGICAL INFORMATION

Logical information is processed as fixed-length or variable-length data. Operations using this type of information include moving, comparing, translating, editing, bit testing, bit manipulation, and shifting.

Fixed-length data, which consists of one, four, or eight bytes, is processed from data stored in the general purpose register or in main storage and the results are placed back into these registers or into main storage.

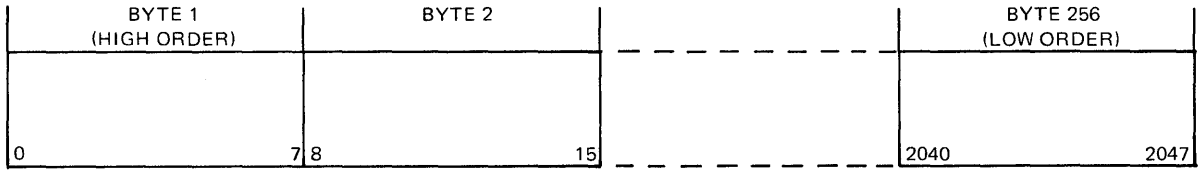
Data in the general registers normally occupies all 32 bits. In logical operations, all bits are treated uniformly, with no distinction made between sign and numeric bits. In some operations, only the low order eight bits of a register are used; the remaining 24 bits are unchanged. In some shift operations, all 64 bits of an even/odd pair of registers are used.

Logical data in fixed-length format is as follows:



Variable-length data has up to 256 bytes of alphabetic or numeric character codes (alphanumeric data). This information is again processed from data stored in the general purpose registers or in main storage and the results are placed back into the general purpose registers or into main storage, and occupies fields which may start at any byte address. The processing order of variable length data is from high order (most significant) byte to the low order (least significant) byte.

Logical data in variable-length format is as follows:



2.3.2. Instruction Formats

Instructions are identified by an op code field which is the most significant eight bits of the instruction. Two general categories of instructions related to a processor state are available to the programmer:

- Supervisor (Privileged) Instructions

  Privileged instructions can be executed only when the processor is in the supervisor mode. The PSW specifies the operating modes of the processor. Processor states may switch as a result of an interrupt condition that causes a new PSW to be obtained from storage or from a load-program-status-word (LPSW) instruction. If the program attempts to execute a privileged instruction, a program exception interrupt occurs. It should be noted that the supervisory routines use the privileged instructions and the problem programmer should avoid these instructions.

- Problem (Nonpriviledged) Instructions

  The problem instructions can be executed in either the supervisor or the program mode.

The format used for programmed instructions for directing peripheral devices and processing data may vary in format and length. The format used is dictated by the operation to be performed and the operand location. The length of the instruction is dictated by the format and is either one, two, or three half words.

The five instruction formats are shown in Figure 2–3 and are identified as follows:

Register to Register (RR)

Register to Indexed Storage (RX)

Register to Storage (RS)

Storage and Immediate Operand (SI)

Storage to Storage (SS)

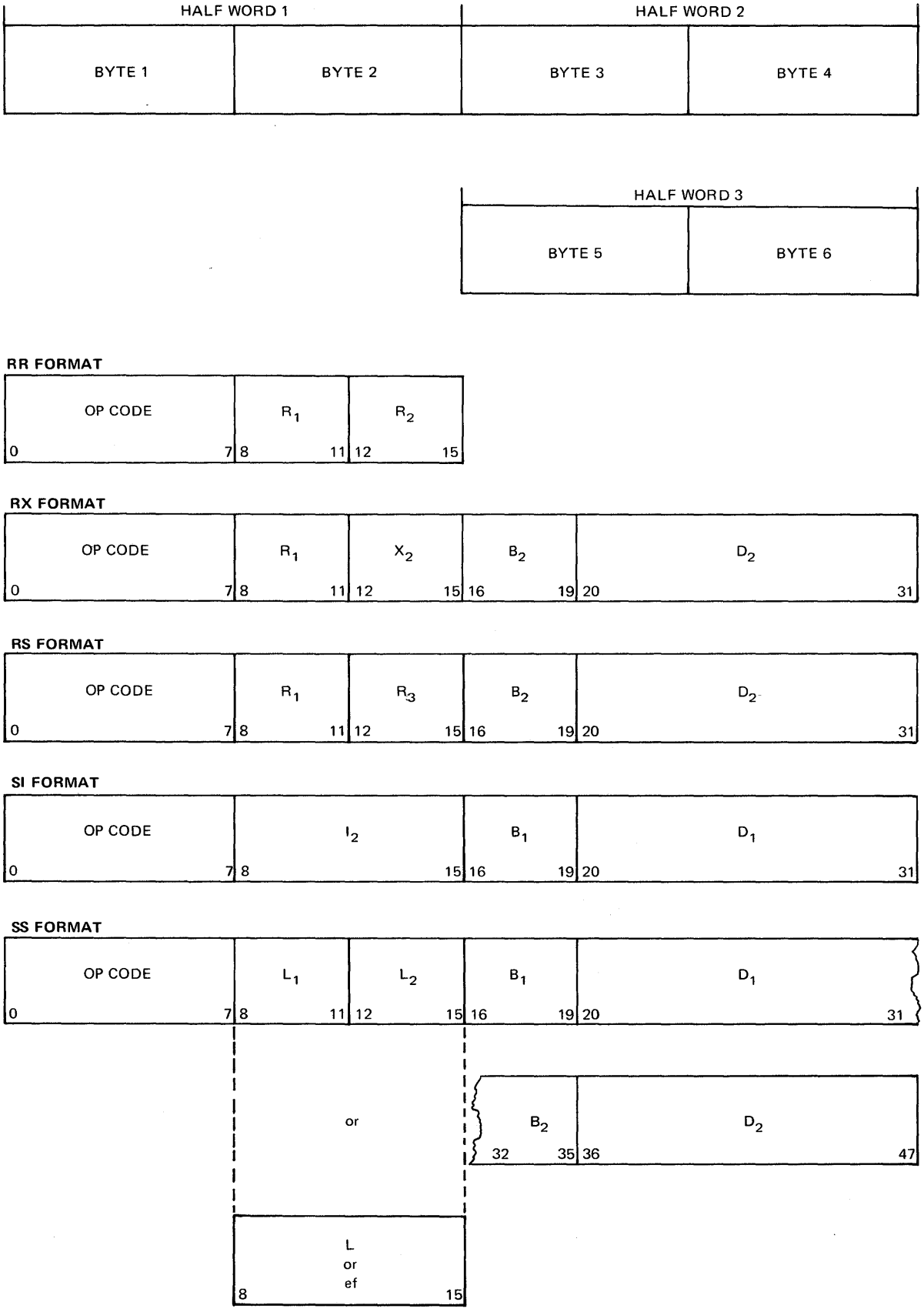
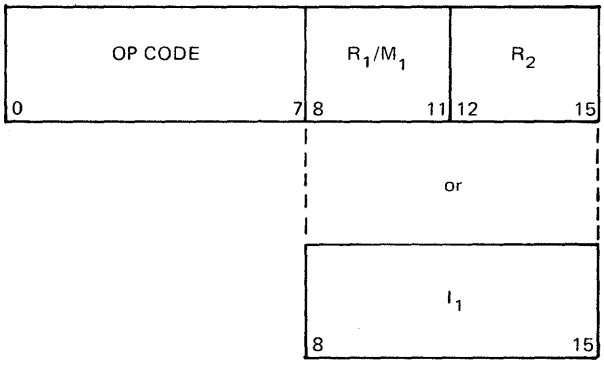


Figure 2–3. Instruction Formats

2.3.2.1. REGISTER TO REGISTER (RR)

The RR instruction is one half word in length and has one of the following formats:

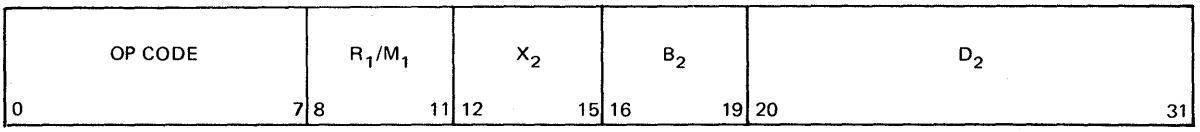


where:

R_1/M_1 (bits 8-11)	R_1 specifies the number of a general register or floating-point register which contains the first operand. M_1 represents a mask for the branch-on-condition (BCR) instruction only.
R_2 (bits 12-15)	Specifies the number of a general register or floating-point register that contains the second operand.
I_1 (bits 8-15)	Represents an 8-bit byte of immediate data.

2.3.2.2. REGISTER TO INDEXED STORAGE (RX)

The RX instruction is one word in length and has the following format:



where:

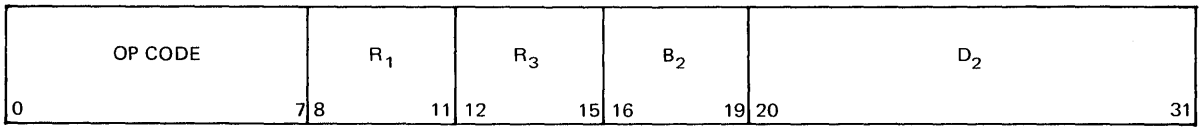
R_1/M_1 (bits 8-11)	R_1 specifies the number of a general register or floating-point register that contains the first operand. R_1 also specifies where the result is to be stored, if required. M_1 specifies the number of a general register that contains an index value in the BCR instruction only.
X_2 (bits 12-15)	Specifies the number of a general register that contains an index value. When set to 0, the index value is 0.

B_2 (bits 16-19) — Specifies the number of a general register that contains an index value representing the base address of the second operand. When set to 0, the base address is 0.

D_2 (bits 20-31) — Contains a 12-bit displacement value that, when added to the contents of B_2 and X_2 and offset (if applicable), represents the address of the second operand. If a load-address instruction is being executed, the final address sum is stored in a general register rather than being used to address main storage.

2.3.2.3. REGISTER TO STORAGE (RS)

The RS instruction is one word in length and has the following format:



where:

R_1 (bits 8-11) — Specifies the number of a general register containing the first operand. These bits also specify the general register where the result is to be stored if required.

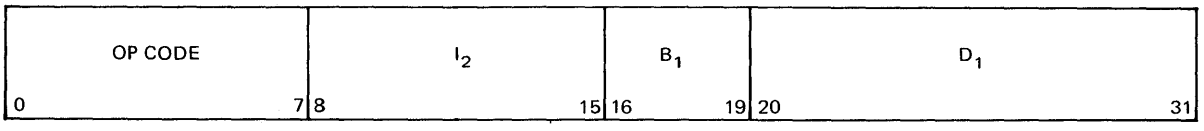
R_3 (bits 12-15) — Specifies the number of a general register containing the third operand. This field is ignored when executing the shift instructions.

R_2 (bits 16-19) — Specifies the number of a general register that contains an index value representing the base address of the second operand. When B_2 is set to 0, the base address is 0.

D_2 (bits 20-31) — Contains a 12-bit displacement value that, when added to the contents of B_2, represents the address of the second operand. For shift instructions, the value of B_2 represents the number of bits to be shifted.

2.3.2.4. STORAGE AND IMMEDIATE OPERAND (SI)

The SI instruction is one word in length and has the following format:



where:

I_2 (bits 8-15) — Contains the second operand (immediate operand). It may be unused, used for a mask, or used as a secondary operation code.

B_1 (bits 16-19) — Specifies the number of a general register that contains an index value representing the base address of the first operand. When B_1 is set to 0, the base address is 0.

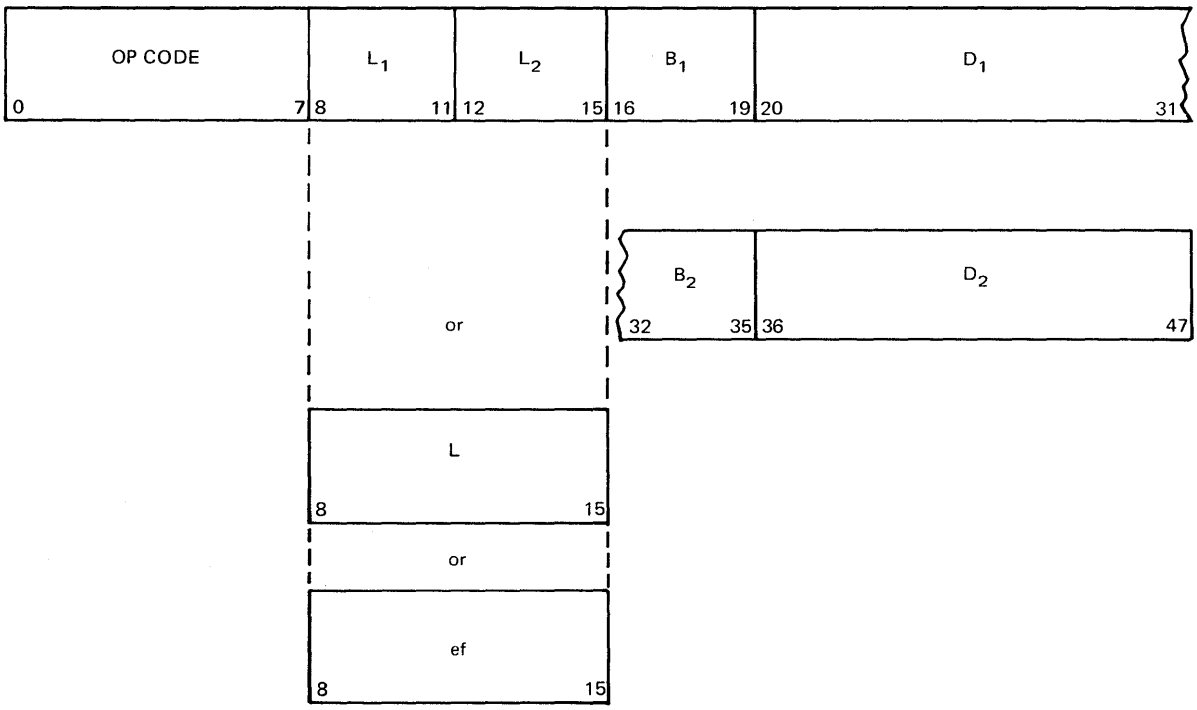
D_1 (bits 20-23)

Contains the 12-bit displacement value that, when added to the contents of B_1 and offset (if required), represents the address of the first operand. For I/O instructions, these bits specify a channel and device number.

Offset is the 12-bit relocation value contained in bits 8-19 of the relocation register located in main storage (4.6.1.3).

2.3.2.5. STORAGE TO STORAGE (SS)

The SS instruction is three half words in length and has one of the following formats:



where:

L_1 (bits 8-11)

Contains a 4-bit number that specifies the number of additional bytes in the first operand field to the right of the left-most byte when the instruction contains Fx op codes, for a decimal instruction where x is any hexadecimal digit 0 through F.

L_2 (bits 12-15)

Contains a 4-bit number that specifies the number of additional bytes in the second operand field to the right of the left-most byte for decimal instructions.

L (bits 8-15)

For instructions with Dx op codes (logical SS instructions), where x is any hexadecimal digit 0 through F; L_1 and L_2 are combined as L to form an 8-bit number that specifies the number of additional bytes to the right of the operand address for both operands, with the exception of translate, translate-and-test, edit, and edit-and-mark instructions, in which case L applies to the first operand only.

ef (bits 8-15)

For the emulation-aid instruction, the second byte of the instruction is used for a secondary operation code.

B_1 (bits 16-19) — Specifies the number of a general register that contains an index value representing the base address of the first operand. When these bits are set to 0, the base address is 0.

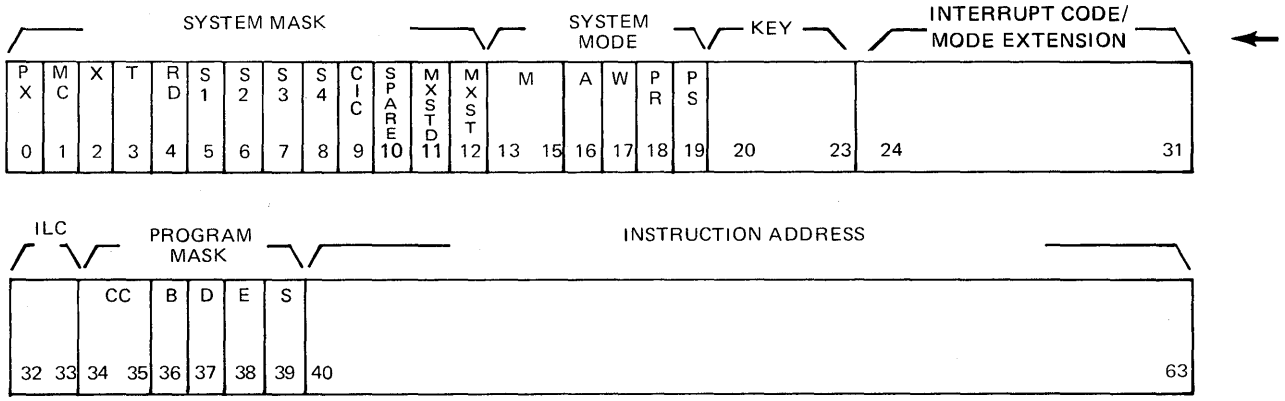
D_1 (bits 20-31) — Contains a 12-bit displacement value that, when added to B_1 and offset (if required), represents the address of the left-most (high order) byte of the first operand.

B_2 (bits 32-35) — Specifies the number of a general register that contains an index value representing the base address of the second operand. When these bits are 0, the base address is 0.

D_2 (bits 36-47) — Contains a 12-bit displacement value that, when added to the contents of B_2, represents the address of the left-most (high order) byte of the second operand.

2.4. PROGRAM STATUS WORD FORMAT

The program status word (PSW) is a double word containing the next instruction address and various control fields which establish operating modes, specify the protection and relocation key, hold program branching conditions, and assist in analyzing interrupt causes. The running program is under control of a PSW which is stored in the processor and is called the current PSW. The format of the PSW is as follows:



All or part of the PSW may be loaded into the current PSW register by use of instructions and interrupts. An entire PSW may be loaded into the current PSW register either by the processor accepting an interrupt or executing a load-PSW instruction. The 13 bits of the system mask may be loaded separately by means of the set-system-mask (SSM) instruction. The six bits of the program mask field may be loaded by means of the set-program mask (SPM) instruction. The ILC, CC, and instruction address fields of the PSW are continually updated, depending on the instructions being executed. The low order 32 bits of the current PSW may be stored by executing the branch-and-link instructions.

2.4.1. System Mask

Thirteen of the fifteen interrupt levels in the processor may be masked by using the system mask bits 0 through 12. If a mask bit is set to 1, the interrupt corresponding to the mask bit occurs.

The individual mask bits are defined as follows:

PX (bit 0)	Program exception level interrupt mask
MC (bit 1)	Machine check level interrupt mask
X (bit 2)	External level interrupt mask
TD (bit 3)	Timer level interrupt mask
RD (bit 4)	Operating system software facility level interrupt mask (I/O channel 6)
S1 (bit 5)	Selector channel 1 level interrupt mask (I/O channel 1)
S2 (bit 6)	Selector channel 2 level interrupt mask (I/O channel 2)
S3 (bit 7)	Selector channel 3 level interrupt mask (I/O channel 3)
S4 (bit 8)	Selector channel 4 level interrupt mask (I/O channel 4)
CIC (bit 9)	Communications intelligence channel level interrupt mask (I/O channel 5)
(bit 10)	Spare (reserved for I/O channel 7)
MUX STD (bit 11)	Multiplexer channel — standard subchannel level interrupt mask (I/O channel 0)
MUX ST (bit 12)	Multiplexer channel — status table subchannel level interrupt mask (I/O channel 0)

2.4.2. System Mode and Mode Extension

The seven bits of the mode field (bits 13–19 of the PSW) and the eight bits of the mode extension field (bits 24–31 of the PSW) are used to establish various operating modes of the processor. The first three bits of the mode field (bits 13 through 15) are designated the M field and are coded to specify one of 5 mutually exclusive special modes of operation. The remaining bits (bits 16 through 19) may occur in any combination along with any mode specified by the first three bits. The codes 011, 110, and 111 are undefined and should not be used. The use of any of the undefined codes produces indeterminate results. The interpretation of the various other bits in the mode field is as follows:

- M Field

Field Code (Bits 13, 14, 15)	Mode	Use
000	UNIVAC 9700 native mode	This is the normal operating mode of the UNIVAC 9700 processor in the absence of any other mode. Interrupt processing and instruction sequencing may be modified per the setting of the mode extension field of the current PSW.
001	UNIVAC 9400 compatibility mode	Used to modify the operation of certain instructions in the repertoire to facilitate the running of programs written for the UNIVAC 9400 System. Interrupt processing and instruction sequencing may be modified per the setting of the mode extension field of the current PSW.
010	Special compatibility	Specification of this mode results in an operation exception interrupt request to be generated if the following instructions are attempted while in this mode: EA, AI, BALE, BCRE, and LBR. Interrupt processing and instruction sequencing can be modified per the setting of the mode extension field of the current PSW.
100	Special emulation mode I	Following the completion of the operation that establishes this mode (LPSW or UNIVAC 9700 IIS), the processor suspends the fetching and execution of instructions in the normal manner. Microprogram control is transferred to the program alterable (PA) portion of the operand manipulation (OM) section of control storage by setting the OM address register to 2048_{10}. Thereafter each time the address calculation (AC) microprogram starts the OM with op code, the most significant OM address register bit (bit 0) is set to 1, thus causing the reference to be made at or above location 2048_{10}. In a similar manner, each time the OM microprogram starts the AC portion of control storage with Start Staticize, the most significant AC address register bit (bit 0), is set to 1 causing the reference to be made at or above AC location 512_{10}. This operation continues so long as the current PSW indicates that the processor is in this mode. This mode would typically be used for emulation of functions (or macros) that are very different from UNIVAC 9700 native mode functions.
101	Special emulation mode II	Following the completion of the operation that establishes this mode (LPSW or UNIVAC 9700 IIS), normal fetching of instructions continues, however, each time the address calculation (AC) microprograms start the operand manipulation (OM) with op code, the most significant OM address bit (bit 0) is set to 1, thus causing the reference to be made at or above OM location 2048_{10}. All other OM references are made according to normal microprogramming conventions. No altering of AC addresses occurs. This operation continues so long as the current PSW indicates that the processor is in this mode. This mode would typically be used for emulation of functions (or macros) that are similar to UNIVAC 9700 native mode functions, particularly if common AC routines (such as staticize, IIS, etc.) can be shared.

- A bit

  The processor operates in either the ASCII or EBCDIC mode. The mode is determined by the A bit (bit 16) of the PSW as follows:

A = 1	ASCII mode
A = 0	EBCDIC mode

  Certain processor instructions interpret or generate code sensitive characters. This mode defines whether the characters are expressed in ASCII or EBCDIC. The unpack, edit, and edit-and-mark instructions generate (binary) code sensitive zones as follows:

ASCII zone	0011
EBCDIC zone	1111

The edit and edit-and-mark instructions detect the following (binary) sensitive control characters:

	ASCII	EBCDIC
Digit Select	1000 0000	0010 0000
Significant start	1000 0001	0010 0001
Field separator	1000 0010	0010 0010

- W bit

  The processor enters the wait mode when the wait (W) bit (bit 17) of the current PSW is set to 1. When the operation that causes this bit to be set to 1 is completed, the processor enters a state where no instructions are fetched or executed. Any previously initiated I/O operations continue as in the normal running mode. Since the processor cannot execute instructions, it cannot change the current PSW by means of the load-PSW instruction. Therefore, the only nonmanual means of exiting from this mode is the acceptance of an interrupt request, which causes a new PSW to be loaded into the current PSW.

- PR bit

  Two sets of general registers, the problem general registers and the supervisor general registers, are contained in the processor. The PR bit (bit 18) of the current PSW determines the selection of the register to be used as follows:

PR = 1	Problem general registers are selected.
PR = 0	Supervisor general registers are selected.

- PS bit

  The processor may operate in one of two modes, the problem mode or supervisor mode. When operating in the problem mode, all privileged instructions are invalid, and their attempted execution results in a program exception interrupt request. When operating in the supervisor mode, all implemented instructions may be executed, provided the processor is not in the special emulation mode. The PS bit (bit 19) of the current PSW specifies the mode selection as follows:

PS = 1	Problem mode
PS = 0	Supervisor mode

2.4.3. Processor Storage Protection and Relocation Key

Bits 20 through 23 of the PSW contain the processor storage protection and relocation key. See 4.5.2. for the description and use of this key.

2.4.4. Interrupt Code/Mode Extension

Bits 24 through 31 of the old PSW (in low order storage) contain the interrupt code, which is used in two ways. When the old PSW is stored during an IIS, the field is defined as in 4.3. When a new PSW is loaded during an IIS or by means of a LPSW instruction, the interrupt code is defined as follows:

Bit 28=1	Machine state simulation mode
Bit 29=1	Storage protect escape mode
Bit 30=1	Monitor
Bit 31=1	Program trace mode

These bits (bits 28, 29, 30, 31) may be specified in any combination. Bits 24 through 27 are set to 0. If they are not set to 0, indeterminate results occur.

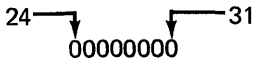
The machine state simulation mode causes the processor to suspend normal serving of interrupts and to transfer control of interrupt servicing to microprogrammed routines located in the program alterable portion of operand manipulation section in the control storage.

Storage protect escape mode results in the protect key (bits 20–23 of current PSW) being treated as though it were 0, the key in the current PSW not being changed to 0, and the specified key remaining intact. The relocation register used is specified in bits 20–23 of the current PSW.

Monitor mode is used to generate a program analysis level interrupt (2.5.7) prior to the execution of every instruction in a running program. Thus a dynamic statistical analysis of programs is performed.

Program trace mode is used to record information pertaining to the execution of branch instructions in a running program. The recorded information is in tabular form and is stored in main storage.

Since monitor mode and program trace mode can be specified together, unique interrupt codes are assigned in the program analysis level of interrupt (2.5.7) as follows:



Monitor Mode

00000001 Program trace mode

2.4.5. Instruction Length Code

Bits 32 and 33 of the PSW contain the instruction length code (ILC) which is used after an interrupt occurs to indicate the length in half words of an instruction that was suppressed or terminated. This field, in conjunction with the instruction address field of the PSW, is used to calculate the beginning address of an instruction that was suppressed or terminated.

2.4.6. Program Mask

Four specific program exceptions have individual program mask bits (bits 34 through 39), as well as being under control of the program exception level interrupt mask. The specific exceptions are as follows:

Bit Position	Identification	Program Exception
36	B	Fixed-point overflow
37	D	Decimal overflow
38	E	Exponent overflow
39	S	Significance

Bits 34 and 35 contain the condition code (CC) field. Many instructions cause the condition code to be set by reflecting the results of an operation. The value of the CC may be tested by various branch instructions. The condition under which the CC is set is determined by the particular instruction being executed.

2.4.7. Instruction Address

Bits 40 through 63 of the PSW contain the instruction address. When a new PSW is loaded into the current PSW register, this address (after being relocated, if required) points to the first instruction in a program to be executed. Thereafter, this field is updated as each instruction is executed. When the current instruction address field is stored, it is pointing to the next instruction that would have been executed. Branch instructions replace the instruction address field in the current PSW in the case of a successful branch.

2.5. INTERRUPTS

The interrupt is the automatic means provided to alert the processor to an exceptional or unexpected condition, the end of an I/O operation, program errors, hardware errors, and various monitoring and tracing operations and to direct the processor to the appropriate program following the detection of such an event. Any task can be interrupted to take on a task of higher priority.

2.5.1. Interrupt Initialization Sequence (IIS)

The PSW specifies the operating characteristics and the current state of the execution of a given program. When an interrupt occurs, this information is saved by storing the current PSW (excluding the mode extension field), held in ← hardware, in the old PSW, which is held in low order storage. When an interrupt occurs, the interrupt code, which carries detailed information pertaining to the interrupt, is transferred to the old PSW location corresponding to the interrupt level. The new PSW, also held in low order storage, establishes the new program environment when it is transferred to the current PSW. The current PSW becomes the old PSW when stored. A new PSW corresponding to the interrupt level is then read and transferred to the current PSW register. This procedure of switching out one program and switching in another is called an interrupt initialization sequence (IIS). After the interrupt has been processed, the system operating characteristics may be restored to what they were prior to the interrupt by transferring the old PSW to the current PSW register by way of an LPSW instruction.

If a particular interrupt level is enabled, an IIS is executed for this level as follows:

1. The hardware generates an interrupt request for this level.

2. Depending on the interrupt type, the processor completes, suppresses, or terminates the instruction currently being executed.

3. The interrupt request for this level is not honored until the hardware priority logic accepts this interrupt. In some cases such as when the program mask is 0, the interrupt request is cancelled if a higher priority IIS is executed first.

4. If the source of the interrupt request is the processor, the appropriate interrupt code is then placed in bit positions 24 through 31 of the current PSW. If the source of the interrupt request is the I/O, the I/O channel itself writes the appropriate interrupt code in bit positions 24 through 31 of the old PSW in low order storage. The instruction address, bits 40 through 63 of the current PSW, is converted from absolute to relative if required. The current PSW (excluding the mode extension field) is then placed in the old PSW location for this level.

5. The new PSW for this level is read and is placed in the current PSW register. Bit positions 32 and 33 (ILC) of the current PSW are set to 0 by the hardware.

6. A new relocation value determined by the new key in the current PSW is read from low order storage and is placed in the current relocation register. Depending on the relative instruction (RI) (4.6.1.3) flag, the instruction address in the current PSW is converted from relative to absolute if required. It should be noted that the current PSW hardware always contains the absolute address.

7. The hardware priority circuit reexamines the state of all interrupt request lines, with two cases resulting:

   a. An interrupt request line is indicating a pending interrupt, and its corresponding mask bit is set to 1. In this case, the preceding procedure is repeated commencing with step 3.

   b. No further interrupt requests are pending and enabled. In this case, the processor resumes the instruction process under control of the current PSW.

2.5.2. Interrupt Request and Handling Priority

Fifteen levels of interrupts are provided; each has associated with it an old PSW and a new PSW located in low order storage. Each level except supervisor call also has an associated mask bit, which controls whether an interrupt request may be accepted. In most cases, this mask bit appears in bit positions 0 through 12 of the system mask of the current PSW. In a few cases, the interrupt is controlled by the mode extension field (bits 24 through 31) of the current PSW. The program exception level has a mask bit, PS (bit 0), pertaining to all conditions that can generate requests in this level, as well as four mask bits, B, D, E, and S (bits 36 through 39) of the current PSW, which control four unique conditions in this level.

The interrupt levels are listed in Table 2–1.

Interrupt Level	Priority
Machine Check	1
Program exception	2
Supervisor call	3
Program analysis	4
External	5
Timer	6
Operating system storage facility	7
Selector channel 1	8
Selector channel 2	9
Selector channel 3	10
Selector channel 4	11
Communication intelligence channel	12
Not used	13
Multiplexer standard subchannel	14
Multiplexer status table subchannel	15

Table 2–1. Interrupt Levels

For each interrupt level, there are one or more interrupt requests and one or more interrupt flip-flops. When an interrupt flip-flop is set by an interrupt request at a time determined by the processor hardware, an interrupt occurs and an IIS is executed. Interrupt requests, except supervisor call, may be blocked from setting the corresponding interrupt flip-flop by means of a mask fit in the current PSW. Interrupt requests thus blocked may be held pending until accepted or they may be rejected and the request dropped.

If multiple interrupt requests are present simultaneously and their corresponding mask bits are set to 1, the appropriate interrupt flip-flops are set simultaneously. A hardware priority circuit then selects which IIS is to be executed. At the conclusion of the IIS, the interrupt flip-flops are cleared and the interrupt request lines are again to be sampled. The mask bits placed in the current PSW by the IIS just completed are effective at this time. The priority circuit establishes the priorities listed in Table 2–2.

Interrupt Request	Priority
Processor machine check	1
Program exception	2
Supervisor call	3
Program analysis trace table	4
Equipment check (machine check level)	5
I/O machine check (machine check level)	6
External	7
Timer	8
Operating system storage facility	9
Selector channel 1	10
Selector channel 2	11
Selector channel 3	12
Selector channel 4	13
Communication intelligence channel	14
Not used	15
Multiplexer standard channel	16
Multiplexer status table subchannel	17
Program analysis monitor	18

Table 2–2. Interrupt Request Handling Priority

Although a hardware priority exists for resolving simultaneously presented interrupt requests, it is stressed that the order of executing continuous IIS is under control of the system mask bits and interrupt requests which may be generated during the execution of IIS.

Within a given level, an interrupt request may be generated by one or more independent conditions. A hardware priority circuit establishes which condition is to be stored in the interrupt code field of the old PSW. In some levels, the specific condition generating the interrupt request changes the interrupt priority of the class itself, as shown in Table 2–2. For example, the program analysis level has interrupt priority 4 or 18, depending on the conditions generating the interrupt request.

An IIS takes place only the the end of an instruction execution and before a new instruction is started. Either the processor waits until the end of the instruction before executing the IIS or it ends the instruction early and then executes the IIS. If the processor waits until the end of the current instruction before honoring the interrupt request, the instruction execution is said to be completed. If the processor ends the instruction early, in order to honor the interrupt request, the instruction execution is said to be either suppressed or terminated. In the case of suppression, the effect is as if no operation were specified. However, the instruction address and ILC of the current PSW have been updated. Results are not stored and the condition code is not changed.

If the execution is terminated, all, part, or none of the result may be stored and, therefore, the result is unpredictable. In general, the results should not be used for further computation. The interrupt code and ILC indicate whether an instruction has been completed, suppressed, or terminated. The machine check and program exception levels are the only two which may cause an instruction to be suppressed or terminated and are the only two levels which carry this information in the interrupt code.

2.5.3. Address Stored in Old PSW

During the handling of certain interrupts, it may be desirable to locate the instruction at the conclusion of the execution of which the interrupt was honored. The next instruction address (bits 40 through 63 of the old PSW) and the ILC (bits 32 and 33) may be used for this purpose.

Machine check, program exception, or supervisor call IIS stores the location of the instruction in the old PSW that would have been executed next had the interrupt not occurred. The ILC reflects the length of the current instruction being interrupted as follows:

ILC	PSW Bits		Instruction Length	Format
	32	33		
0	0	0	Instruction completed	
1	0	1	One half word	RR
2	1	0	Two half words	RX, RS, SI
3	1	1	Three half words	SS

The instruction which generated the interrupt request can be located by subtracting the ILC from the instruction address in the old PSW. This is true whether the instruction was suppressed or terminated. If the instruction was completed and no interrupts occurred, the ILC is set to 0 (except supervisor call, which sets the ILC to 1), and the instruction address in the current PSW is pointing to the next instruction to be executed.

2.5.4. Nonrecoverable Errors

It is possible that certain multiple hardware faults or single faults in critical areas can make program detection and recovery impossible. In these cases, the action taken by the processor is unpredictable. Upon detection of these faults, the hardware takes the following action:

1. Places system in cycle mode.

2. Sends the SYSTEM RESET signal to the I/O channels.

3. Places signal on the machine check-out line, provided that the direct control and external interrupt feature is installed.

4. Illuminates proc check light on the system maintenance panel. The processor then stops at a point as close as possible to where the nonrecoverable error condition was detected.

The following conditions are defined as nonrecoverable errors:

1. Control storage address or data parity errors. (If in the load-control-storage instruction, the specified control storage location is not installed, an address exception is indicated.)

2. The following errors detected during the execution of an IIS (except for I/O writing into the channel status word (CSW) or PSW), an LPSW instruction, or a set-system-mask instruction.

   a. Address bus check

   b. Read bus check

→ c. Storage read check

   d. Write bus check

   e. Protection exception

   f. Address exception

   g. Storage cabinet select exception

3. The generation of a processor machine check interrupt request when the machine check mask bit in the current PSW is set to 0.

4. The presence of an abnormal indication on the storage hold lines.

5. The generation of an I/O machine check interrupt request during initial load.

6. The detection of a dc power fault.

2.5.5. Machine Check Level

The machine check level of interrupt is used to notify a portion of the operating system of the detection of a hardware failure. This detection may result from the use of main storage; certain stall conditions; or overtemperature, blower fault, or ac power fault indications.

The conditions generating the machine check interrupt request may be considered in three classes; I/O machine check, processor machine check, and equipment check. These three classes have different interrupt request priorities, as shown in Table 2–2.

2.5.5.1. INPUT/OUTPUT MACHINE CHECK CLASS

The I/O machine check class interrupt requests are held pending if the machine check mask bit, MC (bit 1), in the current PSW is set to 0. Interrupts of this class may occur only after execution of the current processor instruction is completed. It is therefore not required to note in the interrupt code whether the instruction has been suppressed, terminated, or completed. The ILC stored in the old PSW is 0. The interrupt codes for this class are as follows:

Code	Description
0CCC 0101	An addressing exception or a protection exception occurs in writing a CSW, an ISW, or an interrupt code in an old PSW.
0CCC 0110	A channel has detected the following conditions: The I/O interface stall timer has expired, the channel has sent selective reset to the I/O interface, and all significant interface signals have not reset to 0.
0CCC 1000	An address check occurs in writing a CSW, an ISW, or an interrupt code in an old PSW.
0CCC 1001	A storage parity check occurs in writing a CSW, an ISW, or an interrupt code in an old PSW.
0000 1100	An address check, a storage parity check, an addressing exception, or a protection exception occurs in writing a status table CSW or in writing the error terminate mode in the hard status table control word.
0000 1101	An address check, a storage parity check, an addressing exception, or a protection exception occurs in writing a timer CSW or in writing the error terminate mode or short mode in the hard timer control word.

The CCC in the preceding interrupt codes has the following definition:

000—Multiplexer channel—channel 0

001—Selector Channel 1—channel 1

010—Selector channel 2—channel 2

011—Selector channel 3—channel 3

100—Selector channel 4—channel 4

101—Communication intelligence channel—channel 5

110—Operating system storage facility—channel 6

111—Not assigned (reserved for channel 7)

2.5.5.2. PROCESSOR MACHINE CHECK CLASS

The processor machine check class interrupt requests are inhibited if the machine check mask bit in the current PSW is set to 0. The detection of a processor machine check immediately suppresses or terminates the current instruction being executed and the machine check IIS is entered if the machine check bit is a 1. If this mask bit is set to 0, a nonrecoverable error is generated. The interrupt codes for this class are as follows:

Code	Description
111T 0101	An addressing exception or a protection exception occurs on a processor control hardware-generated address.
111T 0110	COS write data error
111T 0111	A storage parity check (read bus) occurs on a processor read reference.
111T 1000	An address check occurs on any processor reference.
111T 1001	A storage parity check (write bus) occurs on any processor write reference.
111T 1010	A storage parity check (storage read check) occurs on any processor read reference.
111T 1100	A program exception interrupt request occurs and the program exception mask bit, PX (bit 0), of the current PSW is set to 0.
111T 1111	The processor stall timer expires during processer instruction execution, IIS execution, initial load exectuion, or while the processor is in the wait mode or special emulation mode.

Instruction suppression and termination are determined by the value of T in the preceding interrupt codes and are defined as follows:

T = 0 Instruction suppressed

T = 1 Instruction terminated

2.5.5.3. EQUIPMENT CHECK CLASS

The equipment check class interrupt request is held pending if the machine check mask bit, MC (bit 1), of the current PSW is set to 0. However, it should be noted that the primary power in that area of the system containing the fault is turned off 250 microseconds after the detection of the fault. All instructions (except those of the SS type, the emulation-aid instruction, or those instructions executed while in special emulation mode) are completed before the interrupt occurs.

In SS instructions, the T bit is used to determine the disposition of the instruction currently being executed when the equipment check interrupt request occurs. The interrupt code for this class is as follows:

111T 0000 An overtemperature condition, a blower motor fault, or an ac power fault has occurred.

Whether an instruction is completed or terminated is determined by the value of T in the preceding interrupt code and is as follows:

T = 0 Instruction completed (ILC = 0)

T = 1 Instruction terminated

2.5.6. Program Exception Level

A program exception level interrupt request occurs when the hardware detects improper specification or use of instructions and data. Interrupt requests of this level cause the instruction currently being executed to be suppressed or terminated. The program exception IIS is then entered if the program exception mask bit, PX (bit 0), in the current PSW is set to 1. In some cases, as noted in the following list, additional mask bit control exists. For these cases, not only must the PX bit be set to 1, but the appropriate four program mask bits, B, D, E, and S (bits 36 through 39), of the current PSW must be set to 1.

If a program exception interrupt request is generated and the PX bit is set to 0, a machine check interrupt request is generated and the cause of the program exception is lost. For those instructions that examine both the system mask and the program mask, the following list summarizes the result if the instruction generates an interrupt request.

Program Mask	System Mask		Result
	PX	MC	
0	X	X	Instruction processing continues with no pending interrupt request, no nonrecoverable error, and no program exception interrupt.
1	0	0	Nonrecoverable error
1	0	1	Machine check interrupt IIS
1	1	X	Program exception interrupt IIS

NOTE:

X denotes bit is ignored.

The interrupt codes for the program exception level are as follows:

Code	Description
000T 0001	Operation exception. An illegal operation has been attempted, or any of the following instructions have been issued while the current PSW specifies special compatibility mode: EA, AI, BALE, BCRE, and LBR.
000T 0010	Privileged operation exception. A privileged operation has been attempted by a program operating in the program mode (PS, bit 19, of current PSW is set to 1).
000T 0011	Execute exception. The subject instruction of an execute instruction is an execute instruction.
000T 0100	Protection exception. A storage protection violation occurs on a program-generated address.
000T 0101	Addressing exception. A storage location outside the range of the installed storage is referenced by a program-specified address. For load-control-storage instruction only, the referenced control storage location is nonexistant. Also caused by an out of bounds write into COS by a load-control-storage (LCS) instruction.

Code	Description
000T 0110	Specification exception. ■ The unit of information referenced is not on an appropriate boundary. ■ The new flag field specified by the BCRE instruction contains an RI, RO, or RD bit not set to 1, and the flag field in the current relocation register contains an RI bit set to 1. ■ The R_1 field of an instruction which uses an even/odd pair of registers (64-bit operand) does not specify an even register. ■ A floating-point register other than 0, 2, 4, or 6 is specified. ■ A multiplier or divisor in decimal arithmetic exceeds 15 digits and sign. ■ The first operand field is shorter than, or equal in length to, the second operand in decimal multiply and divide instructions. → ■ The four low order address bits specified by the contents of R_2 in a set-storage-key or insert-storage-key instruction are not equal to 0.
000T 0111	Data exception. ■ An invalid sign or digit code is detected in decimal operands. ■ Fields in decimal arithmetic overlap incorrectly. ■ The first operand of the multiply-decimal instruction does not have a sufficient number of high order 0 digits. → ■ No sign code has been detected after scanning 16 bytes in the first operand of the Univac 9400 mode multiply or divide decimal instructions.
000T 1000	Fixed-point overflow exception. A fixed-point add, subtract, shift, or sign control operation exceeds the capacity of the first operand field. This interrupt is masked by B (bit 36) of the current PSW, as well as by PX (bit 1) of the current PSW.
000T 1001	Fixed-point divide exception. The quotient of a fixed-point divide operation exceeds the capacity of the first operand (including division by 0) or the result of a convert-to-binary instruction exceeds 31 bits.
000T 1010	Decimal overflow exception. The result of an add-decimal, subtract-decimal, or zero-and-add instruction exceeds the capacity of the first operand location. This interrupt is masked by D (bit 37) of the current PSW, as well as by PX (bit 1) of the current PSW.
000T 1011	Decimal divide exception. The quotient of a divide-decimal instruction exceeds the capacity of the quotient part of the first operand field.
000T 1100	Exponent overflow exception. The final characteristic resulting from a floating-point arithmetic operation exceeds 127.
000T 1101	Exponent underflow exception. The final characteristic resulting from a floating-point arithmetic operation is less than 0. This interrupt is masked by E (bit 38) of the current PSW, as well as by PX (bit 1) of the current PSW.

Code	Description
000T 1110	Significance exception. The final fraction resulting from a floating-point addition or subtraction is set to 0. This interrupt is masked by S (bit 39) of the current PSW, as well as by PX (bit 1) of the current PSW.
000T 1111	Floating-point divide exception. The divisor fraction in a floating-point divide operation is equal to 0.
001T 0101	Indirect addressing exception. The A bit (bit 5) of an IACW is set to 1 in an IACW that is patched as part of the indirect storage reference.
001T 0110	Indirect address specification exception. The number or indirect address control word (IACW) specified in an indirect storage reference exceeds 8.

Instruction suppression and termination are indicated as follows:

T = 0 Instruction suppressed

T = 1 Instruction terminated

2.5.7. Program Analysis Level

This level of interrupt is utilized by the monitor mode and the program trace mode. The system mask does not apply to this level.

Mode bit setting of 00000001 specifies a program trace interrupt, and 00000000 specifies a monitor interrupt.

The program trace interrupt indicates that the 128-word trace table is full. The monitor interrupt indicates that a processor instruction has been completed and that the next instruction may be monitored.

All instructions are completed in this level of interrupt and the ILC stored in the old PSW is 0.

2.5.8. Supervisor Call Level

A supervisor call level interrupt occurs when a supervisor call instruction is executed. When the supervisor call IIS occurs, the contents of the I field of the instruction are stored in the interrupt code. The hardware does not examine or modify this code. The supervisor call level interrupt cannot be masked. Bits 32 and 33 (ILC) stored in the old PSW are binary 0 and 1, respectively.

2.5.9. External Level

The external level of interrupt alerts a system to the operation of an interrupt key on the operator console or to the presence of certain externally generated signals. These signals require either the use of the direct control and external interrupt feature or the operation of the maintenance trace comparator under control of a switch on the system maintenance panel. This level is under control of the system mask bit X (bit 2) of the current PSW. If bit X is set to 0, all interrupt requests at this level are held pending until the mask bit is set to 1. Interrupt requests occuring at different times while the interrupt is masked are merged, and the interrupt code is the result of ORing together the interrupt codes associated with the separate interrupt requests. Interrupts of this class may occur only after execution of the current processor instruction is completed. The ILC stored in the old PSW is 0. The interrupt codes for the external level are as follows:

Code	Name
1XXX XXXX	Maintenance trace
X1XX XXXX	Interrupt key
XX1X XXXX	External signal 2
XXX1 XXXX	External signal 3
XXXX 1XXX	External signal 4
XXXX X1XX	External signal 5
XXXX XX1X	External signal 6
XXXX XXX1	External signal 7

NOTE:

X bits are other external interrupt conditions.

2.5.10. Timer Level

The timer level interrupt request occurs when the interval timer interrupt count field is decreased from 1 to 0 or when a timer CSW is written into main storage. This interrupt level is under control of the system mask bit T (bit 3) of the current PSW. If this bit is 0, the timer interrupt request is held pending until the timer mask bit is set to 1. Interrupt requests occurring at different times while the interrupt is masked are merged, and any nonzero timer CSW's are written into low order storage at the time that their interrupt request is generated. The writing of a timer CSW replaces the previous contents of that word in storage. Interrupts of this level may occur only after execution of the current processor instruction is completed. The ILC stored in the old PSW is 0.

2.5.11. Input/Output Channel Levels

There are nine levels of I/O interrupts, including an unused one. The eight levels that are used are:

1. Operating system storage facility (OSSF)
2. Selector channel 1
3. Selector channel 2
4. Selector channel 3
5. Selector channel 4
6. Communication intelligence channel (CIC)
7. Multiplexer standard channels
8. Multiplexer status table subchannel

Each level has its own system mask (bits 4 through 12) in the current PSW. If the mask bit for a given level is a 0, the interrupt request is held pending until the mask bit is set to 1 or the interrupt condition is relieved by an appropriate I/O instruction. A channel status word is written only during the execution of an IIS. Interrupts of this level may occur only after the execution of the current instruction is completed. The ILC stored in the old PSW is 0.

2.5.12. Processor Stall Check

A 16-millisecond timer is triggered at the beginning of the execution of each processor instruction and at the beginning of an IIS. If, due to a hardware fault, a new instruction is not staticized within 16 milliseconds or if the interrupt request lines do not drop properly within the IIS, the processor machine check interrupt request is generated.

The stall timer is also triggered while the processor is waiting for an interrupt when the processor is either in the wait mode or is executing in initial load operation. If, due to a hardware fault, the stall timer expires (not triggered as often as every 16 milliseconds), a processor machine check interrupt request is generated.

The stall timer must also be triggered by the special emulation mode microprogram when the processor is to be operated in the special emulation mode for a continuous period that is longer than the expiration time of the stall timer.

The effect of the expiration of this timer is inhibited under the following conditions:

- The processor is stopped (due to being in a cleared state, maintenance trace stop, halt-and-proceed instruction stop, etc.).
- The processor is operating in the one-instruction mode or the cycle mode.

2.5.13. Power Control Faults

Various switches and indicators exist on the status panel located in the power control cabinet for monitoring and controlling power to the system, as well as indicating certain abnormal conditions, such as overtemperature and blower motor faults. However, only two lines are returned to, and effect the operation of, the processor. One line generates nonrecoverable error and the other generates an equipment check interrupt request. When either line assumes the error indicating state, manual intervention is required to reestablish the normal state.

2.5.14. Program Exceptions

The following program exceptions are generated if the control hardware of the processor attempts execution of any of them:

Program Exception	Type of Exception Generated
Execution of an instruction is not included in the 154 instructions provided.	Operation exception
Execution of a floating-point instruction and a floating-point feature not installed.	Operation exception
Execution of a read-direct or write-direct instruction and direct control and external interrupt feature is not installed.	Operation Exception
Execution of the emulation-aid instruction and the 1401/1440/1460 compatibility feature is not installed.	Operation Exception

Program Exception	Type of Exception Generated
Execution of a privileged instruction when the processor is in the problem mode and the special emulation mode is not specified in the current PSW.	Privileged operation exception
An execute instruction has as its instruction another execute instruction.	Execute exception (instruction is suppressed)
The protection exception is detected during the fetch of an instruction, operand, or IACW.	Protection exception (instruction is suppressed)
The protection exception occurs during the fetch of the subject instruction of an execute instruction.	Protection exception
The protection exception occurs during the writing of an entry into the trace table when operating in program trace mode.	Protection Exception
A program-specified address sent to main storage by the processor lies outside the given storage range.	Addressing exception
The control storage address specified in the load-control-storage instruction is nonexistent.	Addressing Exception
Fixed-point instructions: A double word is not on a 64-bit boundary; a word is not on a 32-bit boundary; a half word is not on a 16-bit boundary; an odd register address is specified where an even/odd pair of registers is required.	Specification exception (instruction is suppressed)
Decimal instructions: A multiplier or a divisor size exceeds 15 digits and sign or is equal to or exceeds the multiplicand or dividend size, respectively.	Specification exception (instruction is suppressed)
Floating-point instructions: A short operand is not on a 32-bit boundary, a long operand is not on a 64-bit boundary, or a floating-point register address is other than 0, 2, 4, or 6.	Specification exception (instruction is suppressed)
Logical instructions: A word operand is not on a 32-bit boundary or an odd register address is specified when an even/odd pair of registers is required.	Specification exception (instruction is suppressed)

Program Exception	Type of Exception Generated
Branching instructions: The branching instruction address is odd and the instruction is an execute instruction.	Specification exception (instruction is suppressed)
Status switching instructions: The PSW address specified in its load instruction is not on a 64-bit boundary. The storage address held in the register specified by R_2 in an insert-storage-key privileged or set-storage-key privileged instruction does not contain four low order 0's.	Specification exception (instruction is suppressed)
Instruction fetching: The instruction address in the current PSW is odd.	Specification exception (instruction is suppressed)
Indirect addressing: An IACW is not on a 32-bit boundary, or the RI flag (in current relocation register) is 1 and new RI, RO, or RD is not equal to 1.	Specification exception (instruction is suppressed)

2.5.15. Instruction Termination and Suppression

An instruction that has been terminated or suppressed may, after the program exception interrupt has occurred, be reconstructed by the software by subtracting the ILC from bit positions 61 and 62 of the instruction address in the program exception in the old PSW. This address then points to the first byte of the instruction that was terminated or suppressed. However, if the instruction terminated or suppressed was result of an execute instruction, the calculated address points to the first byte of the execute instruction.

2.6. STATUS HANDLING

The presentation of status information from I/O operations is handled by two methods:

1. by writing an ISW and setting a condition code at the end of the initiation of an I/O operation by way of an I/O instruction; and

2. by the I/O interrupt and writing CSW's and tabled status words (TSW's).

2.6.1. Intial Status Word

The initial status word (ISW) is a double word located in low order storage at location 180_{16}. This location is used for the multiplexer channel and all selector channels. If subchannel or device status information develops during the initiation of an I/O operation or if pending device or subchannel status information is relieved by the I/O instruction, an ISW is written and the appropriate condition code is set. See 2.6.4 for the format and description of the ISW.

2.6.2 Channel Status Word

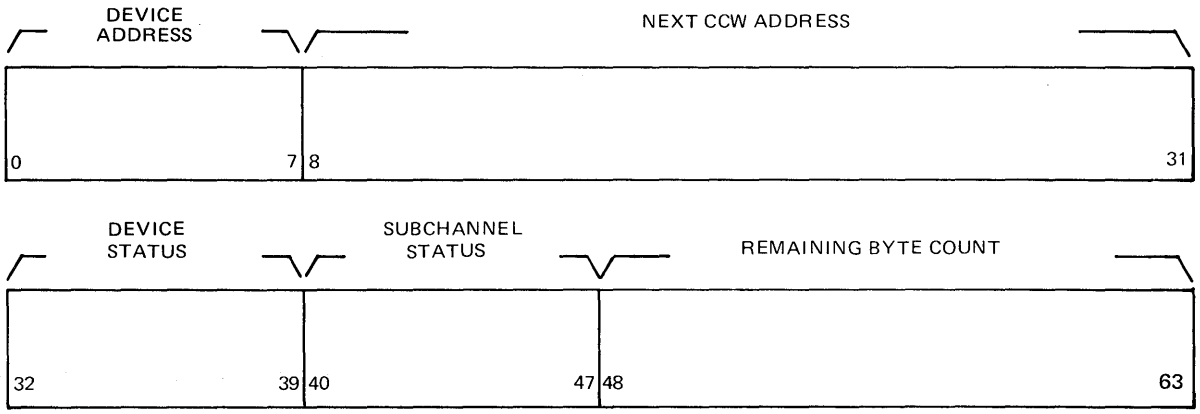
The CSW is located in low order storage at locations $1B0_{16}$, $1C0_{16}$, $1D0_{16}$, and $1E0_{16}$ for selector channel locations 1 through 4, respectively, and at location $1A0_{16}$ for the common use of all the standard subchannels in the multiplexer channel. When either device status is accepted or subchannel status of a type sufficient to cause an interrupt develops, the channel activates an interrupt request signal. Upon receipt of the acknowledgment to the interrupt request, the channel writes the status information into the CSW and clears the condition in the channel or subchannel. See 2.6.4 for the format and description of the CSW.

2.6.3. Tabled Status Word

The data communications subsystem (DCS) subchannel of the multiplexer channel uses the TSW as the vehicle by which device and subchannel status are presented to the software. The format of the TSW is the same as that of the CSW and ISW, but the writing of the TSW is handled by the status table subchannel in the multplexer channel. See 2.6.4 for the format and description of the TSW.

2.6.4. Status Word Format

The ISW, CSW, and TSW have identical formats. The format and description of these words are as follows:



Field	Description
Device address (bits 0-7)	These bits contain the device address associated with the device or subchannel status information. In most ISW's, the device address is the same as the device address in the instruction that caused the ISW write. However, when pending status is relieved by a halt-input/output (HIO) instruction, the device address in the ISW is not necessarily the same as that of the instruction.
Next CCW address (bits 8-31)	These bits contain the value of the next channel command word (CCW) address that was present in the next CCW address field in the hard channel address word (HCAW) at the time the status word was written. Since the next CCW address in the HCAW is an absolute address, the address is absolute in the status word. NOTE: For multiplexer channel only – If the channel control check code (bits 46 and 47) of the subchannel status field indicates a storage error with the HCAW, the contents of the next CCW address field may not be valid. Furthermore, if the channel control check code indicates another type of error, the next CCW address may still not be valid if the error occurred on fetch of the relocation register. This can also occur, if, after the setting of the channel control check code for an error, a second error occurs during the read of the HCAW. Since the channel control check codes cannot be merged, only the first error would be indicated. However, an invalid CCW address would still be written.

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Device status
(bits 32-39)</td><td>These bits contain one byte of status information generated by the subsystem. The codes for the bits in the device status byte are as follows:

Bit Code — Status
32 — Attention
33 — Status modifier
34 — Control unit end
35 — Busy
36 — Channel end
37 — Device end
38 — Unit check
39 — Unit exception</td></tr>
<tr><td>Subchannel status
(bits 40-47)</td><td>These bits contain a byte of status information generated by and pertinent to the subchannel. These bits are not mutually exclusive. The presence of one type of error condition may cause the generation of more than one error indication; that is, a valid channel control check caused by an address check may be accompanied by an erroneous program or protection check. Also, a valid program check caused by an addressing exception may be accompanied by an erroneous protection check. The bits of the subchannel status byte are as follows:
<table>
<tr><th>Bit No.</th><th>Name</th><th>Description</th></tr>
<tr><td>40</td><td>PCI</td><td>The PCI (program controlled interrupt) bit is set in the subchannel status whenever the PCI (bit 36) of the CCW is set to 1. The PCI bit is present in an ISW only if there had been a pending PCI that was relieved by the ISW.</td></tr>
<tr><td>41</td><td>Incorrect length</td><td>The incorrect length bit indicates that the subsystem attempted to transfer a greater or fewer number of bytes than specified in the byte count field of the current CCW. If the SLI flag of the current CCW is set equal to 1 and the chain data (CD) flag is 0, the incorrect length indication is then suppressed. If the CD flag is set to 1, the incorrect length bit may be set irrespective of the SLI flag. If the SLI flag is set to 0, command chaining is suppressed by the incorrect length condition. An incorrect length bit is not set for immediate commands. Therefore, an incorrect length indication in an ISW could only be due to the relieving of a pending status condition or to the execution of halt-input/output (HIO) instruction by the selector channel. The presentation of the incorrect length bit is always withheld until the device presents channel end status. Many of the current subsystems contain no means of counting the number of bytes transferred. As such, these subsystems require the channel to terminate the data transfer by the COMMAND OUT response to the SERVICE IN signal when the byte count in the HCCW is decremented beyond 0. Although this is a normal termination, an incorrect length condition results and is indicated if the SLI flag is not 1.</td></tr>
<tr><td>42</td><td>Program check</td><td>The program check bit indicates that the channel has detected a software-generated error. The following conditions cause the program check bit to be set to 1:

1. The CCW address field in a channel address word (CAW) or in a CCW specifying a transfer in channel (TIC) command does not specify a double-word boundary.

2. Bits 6–7 of the CAW are not set to 0.

3. The channel received an addressing exception indication from main storage during an access for a CCW. The addressing exception indicates that the channel attempted to access an address that was outside of the particular main storage boundary.</td></tr>
</table></td></tr>
</table>

Field	Description		
	Bit No.	Name	Description
			4. The channel received an addressing exception indication during the write of an input data word. 5. The channel received an addressing exception indication during a read of an output data word. The program check bit is set to 1 only if a byte from the word on which the addressing exception indication was generated is actually to be transferred. The transfer is terminated after the last correct byte is transmitted. 6. The command code in the CCW is invalid (for example, bits 4–7 of CCW are set to 0). 7. The byte count in a CCW equals 0. 8. The CCW addressed by a TIC also contains a TIC. 9. The first CCW in a chain specifies a TIC or contains the PCI flag. 10. Bits 37–39 of the CCW are not set to 0. 11. An invalid instruction has been issued.
	43	Protection check	The protection check bit indicates that the channel has received a PROTECTION EXCEPTION signal on an access to main storage for data or CCW. On output data transfers, the protection check bit is set to 1 only if a byte from the output data word on which the protection exception was generated is actually to be transferred. The output transfer is terminated after the last correct byte is transmitted. A PROTECTION EXCEPTION signal is generated by the processor if the key presented by the channel during the storage access does not match the key assigned to the location in storage being accessed and the access is either for a storage write, or for a storage read from a read-protected location. A storage read from a write-only protected location is allowed and generates no exception, regardless of key comparison.
	44	Channel data check	The channel data check bit indicates that the channel detected a data parity error on the I/O input bus or a storage parity check during the transfer of a data word to or from main storage. On an output data transfer, when a byte from the data word on which the storage parity check indication was generated is actually transferred to the subsystem, the error is flagged and the parity bit is intentionally made even on the I/O output bus. The subsystem is then alerted to the parity error. The condition which causes the setting of the channel data check bit suppresses command chaining but does not cause a termination of the transfer operation. The transfer is allowed to proceed until normally terminated either by the subsystem or by the channel upon decrementing the byte count to 0.
	45	Interface contol check	The interface control check bit indicates that the channel detected any one of the following conditions at the I/O interface: ■ The address or status byte received from a device has incorrect (even) parity. If, in the multiplexer channel, the device address byte received during a control unit initiated sequence has bad parity, a selective reset is then issued. However, since the pertinent subchannel cannot be identified, no error indication is provided. ■ The address returned by a device during the initial selection sequence (ISS) did not compare with the address transmitted by the channel.

<table>
<tr><th>Field</th><th colspan="3">Description</th></tr>
<tr><td></td><td>Bit No.</td><td>Name</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td>■ During any ISS (other than the first) associated with command chaining, the device appeared nonoperational or returned a control unit busy (busy and status modifier bits) indication.

■ A signal from a device occurred at the wrong time or was activated for an excessive duration. Interface control check conditions cause both channels to disconnect the device by way of the selective reset. If the condition occurs during the ISS of an I/O instruction, the indication is presented in the ISW. If, in the selector channel, the condition occurs at any other time, the interrupt pending state is entered. If the condition occurs at any other time in the multiplexer channel, the subchannel is placed in the interrupt pending state. However, the interrupt pending condition must be relieved by the TIO or SCHR followed by an LCHR instruction. See Appendix A for mnemonic instruction identification.</td></tr>
<tr><td>Channel control check code (bits 46, 47)</td><td colspan="3">The channel control check code indicates that there has been a hardware malfunction in the channel. The errors included are address checks on storage accesses for data, address checks and storage parity checks on accesses for CCW, and any storage error (such as addressing exception, protection exception, storage parity check, and address check) on accesses for the CAW, relocation value, or (multiplexer only) HCAW. On output data transfers, the channel control check code is set only if a byte from the word in error is actually to be transferred. The transfer is terminated after the last correct byte is transmitted.

The channel control check code bits and their interpretation are as follows:
<table>
<tr><th colspan="2">Bit No.</th><th rowspan="2">Error Occured During</th></tr>
<tr><th>46</th><th>47</th></tr>
<tr><td>0</td><td>0</td><td>No error</td></tr>
<tr><td>0</td><td>1</td><td>Operation with data</td></tr>
<tr><td>1</td><td>0</td><td>Operation with CAW or (multiplexer only) HCAW, or on access of location 180_{16} during LCHR or SCHR.</td></tr>
<tr><td>1</td><td>1</td><td>Operation with CCW or relocation register</td></tr>
</table>
Since channel control check is only a 2-bit code, successive channel control checks cannot be meaningfully merged. Therefore, the code written in the subchannel status field designates the first such error that occurred.</td></tr>
<tr><td>Field (remaining byte count bits 48-63)</td><td colspan="3">These bits contain the value found in the byte count field of the HCCW at the time the status word was written.</td></tr>
</table>

NOTE:

In both selector and multiplexer channels (Section 3), detection of program, protection, and channel control check conditions during data transfers cause the channel to terminate the transfer by the COMMAND OUT response to the SERVICE IN signal. The appropriate subchannel status indication is then presented to the software when the status word (CSW, TSW, ISW) containing the device ending status is written.

Program, protection, and channel control check conditions detected during the first ISS associated with an SIO instruction causes the channel to disconnect the device by a SELECTIVE RESET signal. The subchannel status indication is then presented in the ISW.

Program, protection, and channel control check conditions detected during any ISS (other than the first) associated with command chaining (CC) causes the selector channel to disconnect the device by a SELECTIVE RESET signal and the multiplexer to terminate the operation by a pseudo-TIO command. The selector channel then enters the interrupt pending state to present the appropriate subchannel status. The multiplexer channel presents the subchannel status with the device status transmitted in response to the pseudo-TIO command. This device and subchannel status is handled in a manner that depends on the type of subchannel involved (3.2.3).

2.6.5. Status Table Subchannel

The status table subchannel is the mechanism that permits the multiplexer channel to handle status from data communication subsystems (DCS). A unique characteristic of the DCS that distinguishes it from standard devices is its inability to stack status. The DCS has no means for storing device status within itself. This means the channel must be capable of accepting status from all DCS's at any time. This requirement is satisfied in the multiplexer channel by the sequential storing or tabling of DCS device status in an area of main storage. The channel mechanism that handles this tabling is called the status table subchannel. Control of all DCS operations other than status handling is accomplished by the DCS subchannel. The tabling of DCS device status is handled similarly to handling data by normal data subchannels. This is accomplished by the dedicated working control words in low order storage, software-supplied control words in storage, and the ability to link noncontiguous table areas in storage by way of data chaining.

The control words used by the status table subchannel are:

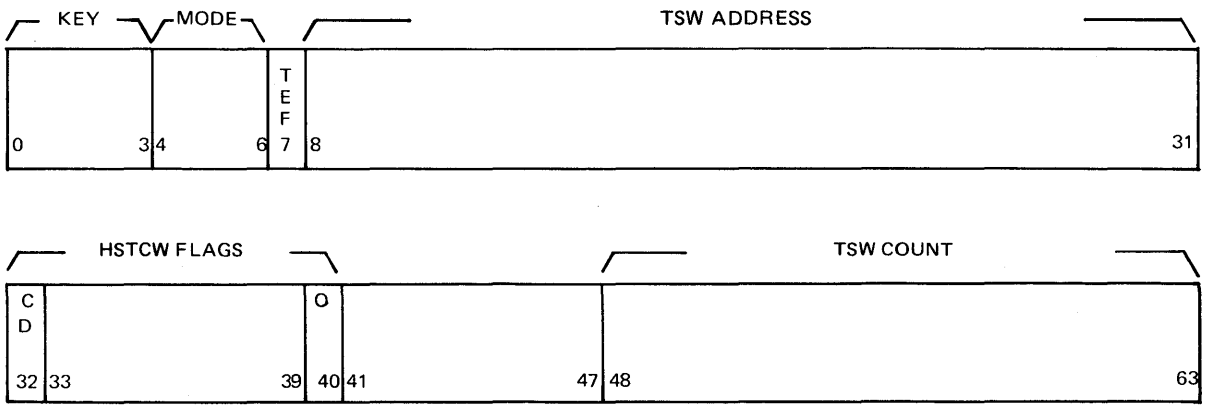
- Tabled status word
- Hard status table control word
- Status table control word
- Status table hard channel address word
- Status table channel status word

2.6.5.1. TABLED STATUS WORD

The tabled status word (TSW) is a double word stored in the status table each time a DCS presents status to the multiplexer channel or DCS subchannel status develops. The location within the status table is designated by the address in the TSW address field of the hard status table control word.

2.6.5.2. HARD STATUS TABLE CONTROL WORD

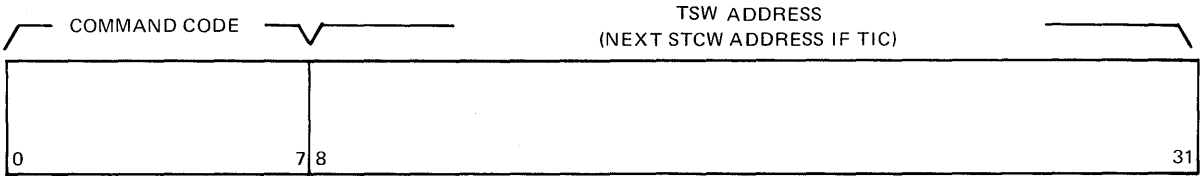
The hard status table control word (HSTCW) is a double word located in low order storage location 130_{16}. This word controls the operation of the status table subchannel by pointing to the location in the status table where the next TSW is to be stored and by allowing the software to link noncontiguous areas in storage for the status table. Software must set up the initial HSTCW at system initialization time. Thereafter, the subchannel provides the replacement of the HSTCW fields by chaining. The format and description of the HSTCW fields are as follows:

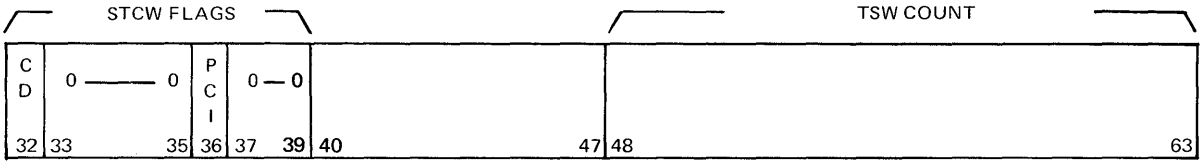


<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Key
(bits 0-3)</td><td>These bits contain a 4-bit storage protection key used by the status table subchannel for all storage accesses for STCW and TSW. Unless the software replaces these bits at some later time, the key supplied in the initial HSTCW is used throughout the operation of the status table subchannel. The key is used for storage protection only. Addresses in the status table subchannel are always in absolute.</td></tr>
<tr><td>Mode
(bits 4-6)</td><td>The contents of the mode field indicates the present state of status table subchannel as follows:
<table>
<tr><th rowspan="2">Mode</th><th colspan="3">Bit No.</th><th rowspan="2">Description</th></tr>
<tr><th>4</th><th>5</th><th>6</th></tr>
<tr><td>Idle</td><td>0</td><td>0</td><td>0</td><td>Causes status table subchannel to cease tabling DCS status. This mode may be set only by software.</td></tr>
<tr><td>Error terminate</td><td>0</td><td>0</td><td>1</td><td>Indicates that the status table subchannel detected an error, halted the subchannel operation, and has or will alert the software by way of a status table interrupt request and the writing of a status table channel status word. The HSTCW remains in this mode until it is replaced by the software in the error recovery procedure.</td></tr>
<tr><td>Active</td><td>1</td><td>0</td><td>0</td><td>Indicates that the status table subchannel is operating in the normal manner.</td></tr>
</table></td></tr>
<tr><td>TEF (bit 7)</td><td>The TEF (table exhausted flag) bit is set to 1 when the TSW count is decremented to 0 and the CD bit is set to 0.</td></tr>
<tr><td>TSW address
(bits 8-31)</td><td>These bits contain the absolute address of the double-word location in the status table area in storage where the next TSW is to be stored. The contents of the TSW address in the initial HSTCW must be set up by software to point to the first double-word location in the status table.</td></tr>
<tr><td>CD (bit 32)</td><td>The chain data (CD) flag indicates that when the TSW count is decremented to 0, a new status table control word is read from storage to obtain new information for the TSW address, HSTCW flags, and TSW count fields of the HSTCW. The location of the STCW is pointed to by the contents of the status table hard channel address word in low order storage location $1F8_{16}$. If, when the TSW count is decremented to 0, the CD flag is not set to 1, the error terminate mode and the TEF are set in the HSTCW. A status table CSW write and a status table interrupt request are not made until DCS status is actually lost. This means that when the HSTCW, with the error terminate mode and TEF set, is read during the next service of DCS status, a status table CSW write and a status table interrupt request are to be made. The CD flag must be set up by software in the initial HSTCW. Subsequently, the CD flag in the HSTCW is set to 1 if the corresponding flag is set to 1 in the STCW read during chaining.</td></tr>
<tr><td>Bits 33-39</td><td>These bits must be set to 0 by the software.</td></tr>
<tr><td>Bit 40</td><td>This bit must be initially set to 0 by the software.</td></tr>
<tr><td>Bits 41-47</td><td>These bits are ignored by the hardware.</td></tr>
<tr><td>TSW count
(bits 48-63)</td><td>These bits specify, in terms of bytes, the number of remaining TSW locations in the area of the status table controlled by the current STCW. The TSW count is decremented by 8 after each TSW is stored. A maximum count of 65,536 bytes or 8192 TSW's may be specified by an all-0 TSW count.</td></tr>
</table>

2.6.5.3. STATUS TABLE CONTROL WORD

The status table control word (STCW) is a double word located on any double-word boundary in main storage. When the TSW count in the HSTCW is decremented to 0 and the CD in the HSTCW is set, an STCW is read from storage to replace certain HSTCW fields. The address of the STCW to be read is found in the status table HCAW. The format and description of the STCW are as follows:

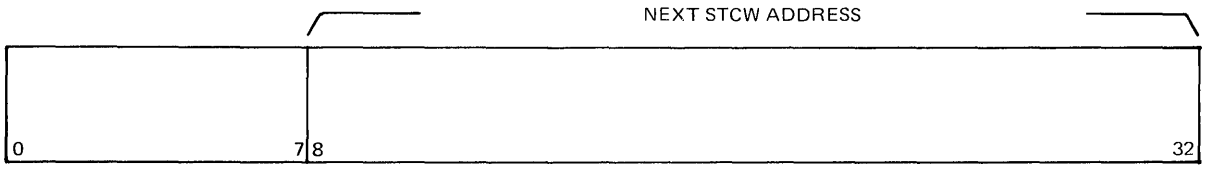




Field	Description
Command Code (bits 0-7)	The only command code recognized is the TIC code, XXXX 1000, where the X's are ignored. The TIC command causes the channel to use the contents of bits 8-31 to access the next STCW. This next STCW may not contain a second TIC command. All other codes are ignored. The second word of the STCW is ignored if a TIC is specified in the first word.
TSW address (bits 8-31)	This field is transferred to the TSW address field of the HSTCW and points to the location in which the next TSW is to be stored. The TSW address must specify a double-word boundary and must be an absolute address. If the CC indicates a TIC, the TSW address field is used to access the next STCW and must specify a double-word boundary.
CD (bit 32)	The chain data (CD) flag is transferred to the corresponding field in the HSTCW during chaining.
Bits 33-35	These bits must be set to 0.
PCI (bit 36)	The program controlled interrupt (PCI) flag causes the status table subchannel to make a status table CSW write and status table interrupt request as soon as possible after it is detected. The PCI bit in the subchannel status field of the status table CSW is set to 1 to indicate that the PCI was detected.
Bits 37-39	These bits must be set to 0.
Bits 40-47	These bits are ignored by the channel.
TSW count (bits 48-63)	The contents of this field are transferred to the corresponding field in the HSTCW during chaining. The TSW count must specify a multiple of eight bytes. A maximum count of 65,536 bytes or 8192 TSW's may be specified by an all-0 TSW count.

2.6.5.4. STATUS TABLE HARD CHANNEL ADDRESS WORD

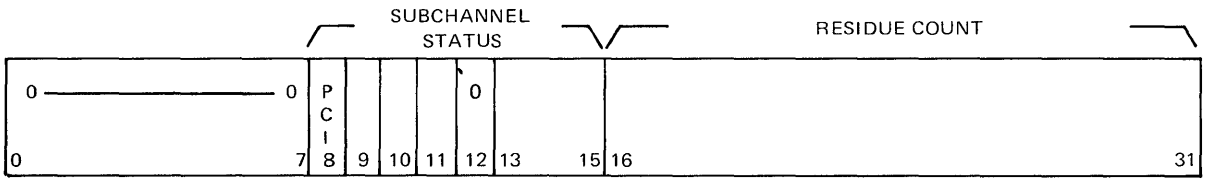
The status table hard channel address word (status table HCAW) is a single word located in low order storage at location $1F8_{16}$. The next STCW address (bits 8 through 31) field contains the absolute address of the STCW that is to be read the next time chaining occurs. The next STCW address must be on a double-word boundary. The initial value of the next STCW address is incremented by 8 each time chaining occurs. Bits 0 through 7 are ignored by the channel. The format of the status table HCAW is as follows:



2.6.5.5. STATUS TABLE CHANNEL STATUS WORD

The status table channel status word (status table CSW) is a full word located in low order storage at location $1A8_{16}$. This word is used to alert the software that the status table subchannel has detected a subchannel error or PCI. The status table CSW is written whenever a subchannel error or PCI is determined. A status table interrupt request is generated after the write of the status table CSW.

The format and description of the status table CSW are as follows:



Field	Description
Bits 0-7	These bits are always set to 0.
PCI (bit 8)	This bit is set to 1 when the subchannel has detected a PCI bit in an STCW read during chaining.
Table exhausted (bit 9)	This bit is set to 1 when, during the servicing of DCS device status, the subchannel detects the error terminate mode and the TEF bit is set to 1 in the HSTCW. The table-exhausted bit indicates to software that DCS device status has been lost because there was no table area available.
Program check (bit 10)	This bit is set to 1 when the subchannel detects any of the following STCW format errors: 1. TSW address not on double-word boundary. 2. If TIC, next STCW address not on double-word boundary. 3. TSW count not a multiple of eight bytes. 4. Bits 33 through 35 and 37 through 39 not equal to 0. 5. STCW addressed by TIC also contains a TIC.

<table>
<tr><th>Field</th><th colspan="2">Description</th></tr>
<tr><td></td><td colspan="2">6. A storage addressing exception on a write of TSW.

7. A storage addressing exception on a read of STCW.</td></tr>
<tr><td>Protection check (bit 11)</td><td colspan="2">This bit is set to 1 when the subchannel detects a storage protection exception on:

1. Write of TSW

2. Read of STCW (if read protect is specified)</td></tr>
<tr><td>Bit 12</td><td colspan="2">This bit is set to 0 by the channel.</td></tr>
<tr><td>Channel control check code (bits 13-15)</td><td colspan="2">This is a 3-bit code that is set when the subchannel detects a hardware malfunction, as follows:</td></tr>
<tr><td></td><td>Code</td><td>Error Occurred During:</td></tr>
<tr><td></td><td>000</td><td>No channel control check</td></tr>
<tr><td></td><td>001</td><td>Read of HSTCW (any storage error)</td></tr>
<tr><td></td><td>010</td><td>Write of HSTCW (any storage error)</td></tr>
<tr><td></td><td>011</td><td>Write of TSW (address check only)</td></tr>
<tr><td></td><td>100</td><td>Not used</td></tr>
<tr><td></td><td>101</td><td>Write of status table HCAW (any storage error)</td></tr>
<tr><td></td><td>110</td><td>Read of status table HCAW (any storage error)</td></tr>
<tr><td></td><td>111</td><td>Read of STCW (address check and storage parity check only)</td></tr>
<tr><td></td><td colspan="2">Since this check code is a 3-bit code, successive channel control checks cannot be meaningfully merged. Thus, the code written in the subchannel status field designates the first such error that occurred.</td></tr>
<tr><td>Residue Count (bits 16-31)</td><td colspan="2">This field contains the value found in the TSW count field of the HSTCW at the time the status table CSW is written.</td></tr>
</table>

Note that a status table CSW containing a PCI may be overlaid by a subsequent status table CSW indicating a subchannel error. The converse is not possible since the status table operations are halted (HSTCW in error terminate mode) after the writing of a status table CSW with an error indication.

Because of the fact that a status table CSW may be written without waiting for the status table interrupt priority signal, the possibility of having an apparent false interrupt arises. This situation could occur as follows:

1. After a TSW is written with no errors, the status table interrupt request is activated.

2. While waiting for the interrupt priority signal, the status table subchannel continues handling status from devices on DCS subchannels.

3. Upon receipt of the PRIORITY signal, the REQUEST is dropped. When the REQUEST is dropped, the IIS is continued and the software enters the interrupt routine.

4. As part of this interrupt routine, the software must check the status table CSW to determine whether a PCI or subchannel error status was written. If, before the software has checked the status table CSW, an error in the subchannel occurs, a new CSW with an error condition is written and an interrupt request is made.

5. If the status table interrupt is not allowed, the processor continues processing the original routine, the new status table CSW is read, and the error is deemed the cause of the interrupt being processed.

6. After reinitializing the status table subchannel, the software exits from the interrupt routine and the status table interrupt is again allowed.

7. At this point, the outstanding status table interrupt request (activated in step 4 above) would be sensed and acknowledged, and the software would again enter the interrupt routine. Since there would now be nothing in the status table CSW, this interrupt would falsely appear to be the first valid interrupt from the reinitialized status table subchannel.

2.6.5.6. STATUS TABLE SUBCHANNEL INITIALIZATION

Unlike the normal data subchannels, the status table subchannel must be initialized by the direct manipulation of the status table control words by the software. This involves the following steps provided by initialization software:

1. If any DCS subchannels are active due to previously issued SIO instructions, the mode in the HSTCW must be set to idle (all 0's) in order to inhibit any further status table control word manipulation by the subchannel hardware. If none of the DCS subchannels are active, the mode can be left unchanged.

2. The MX ST bit (bit 12) in the system mask of the current PSW is set to 1 to allow a possible outstanding interrupt, which occurs only if the status table subchannel has been active previous to this initialization. After this interrupt is allowed, no further status table interrupt requests are possible so long as the HSTCW mode is set to idle.

3. The key field of the HSTCW must be loaded with a value corresponding to the area in storage in which the TSW's are to be written and from which the STCW is to be read.

4. Bit 7 (table-exhausted flag) and bits 33 through 40 of the HSTCW must be set to 0.

5. The TSW address field of the HSTCW must be loaded with the double-word absolute address of the location in storage into which the first TSW is to be stored.

6. The TSW count field of the HSTCW must be loaded with the count in terms of bytes of the total number of TSW's to be stored in the area starting at the intialized TSW address location. If this count is not a multiple of 8, unpredictable results may occur.

7. The CD flag in the HSTCW normally should be set to 1.

8. The next STCW address field of the status table HCAW must be loaded with the double-word absolute address of the STCW which is to be accessed when the TSW count field is decremented to 0 and the CD flag is set to 1.

9. The status table CSW at location $1A8_{16}$ must be cleared to all 0's.

10. The HSTCW mode is set to active (100).

3. INPUT/OUTPUT SECTION

3.1. GENERAL

The input/output (I/O) section initiates, directs, and monitors the transfer of data between main storage and the peripheral subsystem. The I/O section consists of input/output channels that have the standard UNIVAC 9000 Series I/O interface and intput/output channels that are dedicated to special applications such as data communications and operating system storage. These special channels include an integrated disc storage subsystem. The interface between the processor/main storage and the I/O section provides for connection of up to eight channels. All channels are numbered and designated a priority; generally, the higher the data transfer rate of a channel device or subsystem, the higher the priority for that channel. The transfer of information between the processor or main storage and the standard UNIVAC 9000 Series peripheral subsystems is accomplished by the multiplexer and selector channels, one each in the basic processor complex. Three additional selector channels are available by feature addition.

The processor initiates all I/O operations by issuing I/O instructions to a selected channel and a selected subsystem connected to that channel. Once operation is successfully initiated, the channel maintains control of data transfers between main storage and the subsystem, independent of the processor. Upon completion of the I/O operation, the state of the channel and subsystem is presented to the software by way of the appropriate status words and I/O interrupts.

3.1.1. Channel/System Functional Interfaces

Both the multiplexer and selector channels communicate with the other components of the system by way of several relatively independent types of functional interfaces. These interfaces are:

I/O Interface
Instruction Interface
Main Storage Interface
I/O Interrupt Interface

3.1.1.1. I/O INTERFACE

The I/O interface is the 8-bit byte plus parity, compatible interface by which all communication between the channel and peripheral subsystem is accomplished. As many as eight subsystems may be connected to the channel by way of this interface.

3.1.1.2. INSTRUCTION INTERFACE

The instruction interface enables the processor to issue I/O instructions to a selected channel and subsystem; the channel returns a condition code at the completion of the initiation of the operation. The processor begins the execution of an I/O instruction by transferring selected bits of the instruction OP code field, the contents of the instruction device address field, and a request to process the instruction to the selected channel. Depending on the state of the channel, subchannel, and subsystem, the operation may or may not be initiated. The channel then returns the appropriate condition code and the signal acknowledging receipt of the instruction request signal.

3.1.1.3. MAIN STORAGE INTERFACE

This interface enables the channel to request and use main storage, and enables transfers between main storage and the channel. Each channel may independently request the use of main storage by sending a main storage request signal to the processor. Based on the main storage priority scheme, the processor determines which channel is allowed the use of main storage and issues a signal to the selected channel. The selected channel retains use of main storage for one storage cycle. This interface is also used for the transfer of main storage error indications such as addressing exception, protection exception, address check and storage parity check from main storage and the processor to the channel.

3.1.1.4. I/O INTERRUPT INTERFACE

This interface enables the channel to request and to execute I/O interrupts. Each channel may independently request an I/O interrupt by sending a unique interrupt request signal to the processor. Based on the interrupt priority scheme, the processor determines which channel is allowed to make the interrupt and issues a unique interrupt priority signal to the selected channel. The selected channel then takes the appropriate action depending on the interrupt level.

3.1.2. I/O Channel Addressing

I/O channels are addressed 0 through 7 and have a corresponding 3-bit binary code of 000 through 111, which is used to address the channel in I/O instructions and to identify the channel in certain interrupt conditons. The channel address order relates to the physical placement of the channels in the processor complex. Channels 0, 1, and 2 are installed in the I/O A cabinet. The channel address order does not reflect either the interrupt or the main storage priority of the channels.

3.1.3. I/O Channel Address Assignments

The I/O channel addressing assignment and physical location are as follows:

I/O Channel	Assignment	Cabinet Location
0	Multiplexer channel	I/O A
1	Selector channel 1	I/O A
2	Selector channel 2	I/O A
3	Selector channel 3	I/O B
4	Selector channel 4	I/O B
5	Communication intelligence channel (CIC)	I/O B
6	Operating system storage facility (UNIVAC 8405 Disc Subsystem)	I/O C
7	Unassigned	Unused processor/main storage interface in I/O C

3.1.4. I/O Channel Priority

The storage priority for the I/O channels is presented in Table 3–1.

I/O Channel	Channel Number	Priority
Operating system storage facility	6	1
Selector channel 1	1	2
Selector channel 2	2	3
Communications intelligence channel (CIC)	5	4
Selector channel 3	3	5
Selector channel 4	4	6
Unassigned	7	7
Multiplexer channel	0	8

Table 3–1. I/O Channel Storage Priority

3.1.5. I/O Channel Interrupt Priority

There are nine priority levels of I/O interrupts, including one unused. Table 3–2 lists the eight levels that are used.

I/O Channel	Priority Level
Operating system storage facility (OSSF)	1
Selector channel 1	2
Selector channel 2	3
Selector channel 3	4
Selector channel 4	5
Communications intelligence channel (CIC)	6
Multiplexer standard subchannels	7
Multiplexer status table subchannel	8

Table 3–2. I/O Channel Interrupt Priority

3.2. MULTIPLEXER CHANNEL

The multiplexer channel is intended for the attachment of multiple low speed devices through multiple subchannels. It controls and sustains peripheral operations on each of its subchannels concurrently.

The basic multiplexer channel has an 8-subsystem capability, containing fifteen subchannels. These may be increased through the addition of the optional Expanded Interface Feature to sixteen subsystems and by the addition of the optional subchannel expansion to 63 subchannels. Table 3–2A lists the various multiplexer channel configurations.

3.2.1. Subchannels

The multiplexer channel contains two different types of subchannels that are functionally identical except for the manner in which they handle device status. These two types are the standard subchannel and the data communications subsystem (DCS) subchannels.

3.2.1.1. STANDARD SUBCHANNEL

The standard subchannel is used to control the majority of subsystems including both those with shared and nonshared control units. Within the set of standard subchannels, device status is accepted and a request to interrupt with this device status is made for only one subsystem at any one time. Until the interrupt is actually made, no further status is accepted from devices attached to standard subchannels. Rather, when a subsystem attempts to present status, the channel causes the status to be stored (or stacked) within the subsystem. The subsystem then attempts to present the stacked status unless inhibited from doing so by the presence of the SUPPRESS OUT signal from the channel. The channel maintains the SUPPRESS OUT signal until the interrupt condition is cleared. At that point, the SUPPRESS OUT signal drops, the status from the highest priority subsystem is accepted, and the channel reenters the interrupt pending state. The interrupt pending state occurs when the channel has accepted device status and has activated the standard interrupt request and SUPPRESS OUT signals. The multiplexer channel also can enter this state when the program controlled interrupt (PCI) is detected in a channel command word related to a standard subchannel.

Option	Total Allowable Subsystems	Available Subchannels
Standard UNIVAC 9700 Mode of Operation		
Basic Configuration	8	15
Expanded Interface (F1519-00)	15	15
Subchannel Expansion (F1518-00)	8	31
Expanded Interface and Subchannel Expansion (Features: F1519-00 and F1518-00)	16*	31
UNIVAC 9700/Series 70 Mode of Operation		
Subchannel Expansion and Communication Controller – Multichannel (CCM) Subchannel Expansion (Features: F1518-00 and F1518-01)	8	63
Subchannel Expansion, CCM Subchannel Expansion, and Expanded Interface (Features: F1518-00, F1518-01, and F1519-00)	16*	63

*Because of device addressing restrictions, only 15 of the 16 subsystems can be used for devices connected to standard subchannels.

Table 3–2a. Multiplexer Channel Configurations

3.2.1.2. DATA COMMUNICATIONS SUBSYSTEM SUBCHANNELS

The DCS subchannels are used to control subsystems that do not have the capability of stacking status. That is, subsystems controlled by the DCS subchannels must be relieved of their device status whenever they present it to the channel. The multiplexer channel is able to accept status from devices controlled by as many as 29 DCS subchannels (the two remaining subchannels are restricted to standard use) by means of the status table subchannel. The status table subchannel, which is a functional unit within the multiplexer channel, can accept and store device status in an area or table in main storage.

3.2.2. Device Address

Since the channel differentiates between standard and DCS subchannels on the basis of device address, the allowed device addresses are directly related to the configuration of the multiplexer channel in a particular installation. The basic multiplexer channel has 15 subchannels and allows for eight subsystems. This can be expanded to a maximum of 63 subchannels and 16 subsystems by feature additions. Device addresses have one of the following formats:

Standard format: 1ssssddd

DCS format (31 subchannels): 000nnnnn

DCS format (63 subchannels): 0xnnnnnn

where:

ssss	is the binary representation of the subchannel number for standard subchannels.
ddd	is the binary representation of the device number for devices connected to control units on standard subchannels. The d bits allow for the addressing of from one to eight devices connected to a control unit addressed by 1ssss. Shared control units having more than eight devices must use more than one subchannel and must recognize a unique 1ssss field for each subchannel used. (The d bits are not interpreted by the channel.)
nnnnn (31 subchannels) nnnnnn (63 subchannels)	is the binary representation of the subchannel number for DCS subchannels.
x	indicates bit position can be set to 1 or 0.

The 8-bit device address is used by the processor when it issues an I/O instruction to the multiplexer channel and by the channels and subsystem during the prescribed operation.

Table 3–3 lists the 63 subchannel numbers and the associated standard and DCS device addresses. Also listed in Table 3–3 is the communications controller multichannel (CCM) device addresses which are used in the UNIVAC 9700/Series 70 mode of operation.

	Subsystem Number		Hexadecimal Standard Device Address (Format: 1sssddd)	Hexadecimal DC Device Address (Format: 000nnnnn)	Hexadecimal CCM Device Address (Format: 0x0nnnnn, 31 subch; 00nnnnnn 63 subch)
	Device	Binary			
First Subchannel Expansion (F1518-00)	0	00000		00	00 or 40
	1	00001		01	01 or 41
	2	00010		02	02 or 42
	3	00011		03	03 or 43
	4	00100		04	04 or 44
	5	00101		05	05 or 45
	6	00110		06	06 or 46
	7	00111		07	07 or 47
	8	01000		08	08 or 48
	9	01001		09	09 or 49
	10	01010		0A	0A or 4A
	11	01011		0B	0B or 4B
	12	01100		0C	0C or 4C
	13	01101		0D	0D or 4D
	14	01110		0E	0E or 4E
	15	01111		0F	0F or 4F
Basic 15 subchannels	16	10000	80–87	10	10 or 50
	17	10001	88–8F	11	11 or 51
	18	10010	90–97	12	12 or 52
	19	10011	98–9F	13	13 or 53
	20	10100	A0–A7	14	14 or 54
	21	10101	A8–AF	15	15 or 55
	22	10110	B0–B7	16	16 or 56
	23	10111	B8–BF	17	17 or 57
	24	11000	C0–C7	18	18 or 58
	25	11001	C8–CF	19	19 or 59
	26	11010	D0–D7	1A	1A or 5A
	27	11011	D8–DF	1B	1B or 5B
	28	11100	E0–E7	1C	1C or 5C
	29	11101	E8–EF	1D	1D or 5D
	30	11110	F0–F7	1E	1E or 5E
	31	011111	DIAGNOSTIC AID		

Table 3–3. Subchannel/Device Address Correspondence (Part 1 of 2)

Subsystem Number		Hexadecimal Standard Device Address (Format: 1ssssddd)	Hexadecimal DC Device Address (Format: 000nnnnn)	Hexadecimal CCM Device Address (Format: 0x0nnnnn, 31 subch; 00nnnnnn 63 subch)
Device	Binary			
32	100000			20 or 60
33	100001			21 or 61
34	100010			22 or 62
35	100011			23 or 63
36	100100			24 or 64
37	100101			25 or 65
38	100110			26 or 66
39	100111			27 or 67
40	101000			28 or 68
41	101001			29 or 69
42	101010			2A or 6A
43	101011			2B or 6B
44	101100			2C or 6C
45	101101			2D or 6D
46	101110			2E or 6E
47	101111			2F or 6F
48	110000			30 or 70
49	110001			31 or 71
50	110010			32 or 72
51	110011			33 or 73
52	110100			34 or 74
53	110101			35 or 75
54	110110			36 or 76
55	110111			37 or 77
56	111000			38 or 78
57	111001			39 or 79
58	111010			3A or 7A
59	111011			3B or 7B
60	111100			3C or 7C
61	111101			3D or 7D
62	111110			3E or 7E
63	111111			3F or 7F

Second Subchannel Expansion For UNIVAC 9700/Series 70 Mode of Operation (F1518-01, Prerequisites: F1518-00 and Type 0973-00)

Table 3–3. Subchannel/Device Address Correspondence (Part 2 of 2)

3.2.3. Status Handling

Device status or subchannel status that develops or pending device or subchannel status that is relieved during the initiation of an operation by an I/O instruction is presented to the software in the initial status word (ISW) (see 2.6.1). The ISW is written just prior to the indication to sense the condition code. No interrupt is required. This also is true for selector channel status handling.

Asynchronous device status depends on the type of subchannel involved. Device status is always accepted when presented by devices connected to DCS subchannels. As a result, the status table subchannel controls the writing of a tabled status word (TSW) with the device status. Device status from devices connected to standard subchannels is accepted if the channel is not in the interrupt pending state. Upon accepting the status, the channel activates the standard interrupt request and enters the interrupt pending state until the request is acknowledged and the channel status word (CSW) is written. If the channel is in the interrupt pending state when a standard device presents status, the channel stacks the status in the device. Thus, if a standard device is the type that presents separate channel end and device end indications, the device end status is not accepted from the device until after the interrupt for the channel end is made.

Except for the PCI, subchannel status in itself is not sufficient to cause the writing of a CSW or TSW and the generation of the appropriate interrupt request. Incorrect length, channel data check, program check, protection check, and channel control check indications are included in the CSW or TSW written with the ending device status. Most interface control check conditions cause the channel to disconnect the device by way of a selective reset sequence. If the subsystem subsequently attempts to present status, the interface control checks indication is included in the CSW or TSW with the device status. If the subsystem does not subsequently attempt to present status, the interface control check indication may be determined only by way of the test-I/O (TIO) or store-channel-register (SCHR) instruction.

PCI is the only type of subchannel status that can generate an interrupt request. If a PCI is detected in a CCW during the operation of a DCS subchannel, the status table subchannel controls the writing of a TSW and the appropriate interrupt request. If a PCI is detected in a CCW during the operation of a standard subchannel, if not already in the interrupt pending state, the channel activates the standard interrupt request and enters the interrupt pending state. While in the interrupt pending state, other channel operations proceed as usual except that no device status is accepted from standard devices.

When the interrupt is allowed, a standard CSW (located in main storage location $1A0_{16}$) is written and the pending interrupt condition is cleared in the channel. If a PCI develops in a standard subchannel operation while the channel is in the interrupt pending state, the PCI is preserved in the hard channel control word (HCCW). At the next reference to the HCCW, an attempt is again made to present the PCI.

3.2.4. Polling

The multiplexer channel has the ability to poll the I/O interface. That is, when the channel has been inactive for at least 6 microseconds, the SELECT OUT signal is activated by the channel. If none of the subsystems requires service, the SELECT IN signal is returned to the channel. In this manner, the channel can service subsystems that are patched to not generate the REQUEST IN signal for service. This establishes another control on the subsystem priority. Subsystems which have a high data transfer rate but can wait for service may be patched to not generate a REQUEST IN signal and, therefore, will wait for the channel poll. This then prevents subsystems from causing channel overruns on other subsystems.

3.2.5. Channel Register Stack

The channel register stack (CRS) consists of 32 full words (64 full words with F1518-00 installed or 128 full words with F1518-01 installed) of high speed, bipolar registers. The CRS is used for storage of HCCW and for certain diagnostic aids. The CRS may be loaded and unloaded directly by software through use of the load-channel-register (LCHR) and store-channel-register (SCHR) instructions. Figure 3–1 shows the layout and register identification for the CRS.

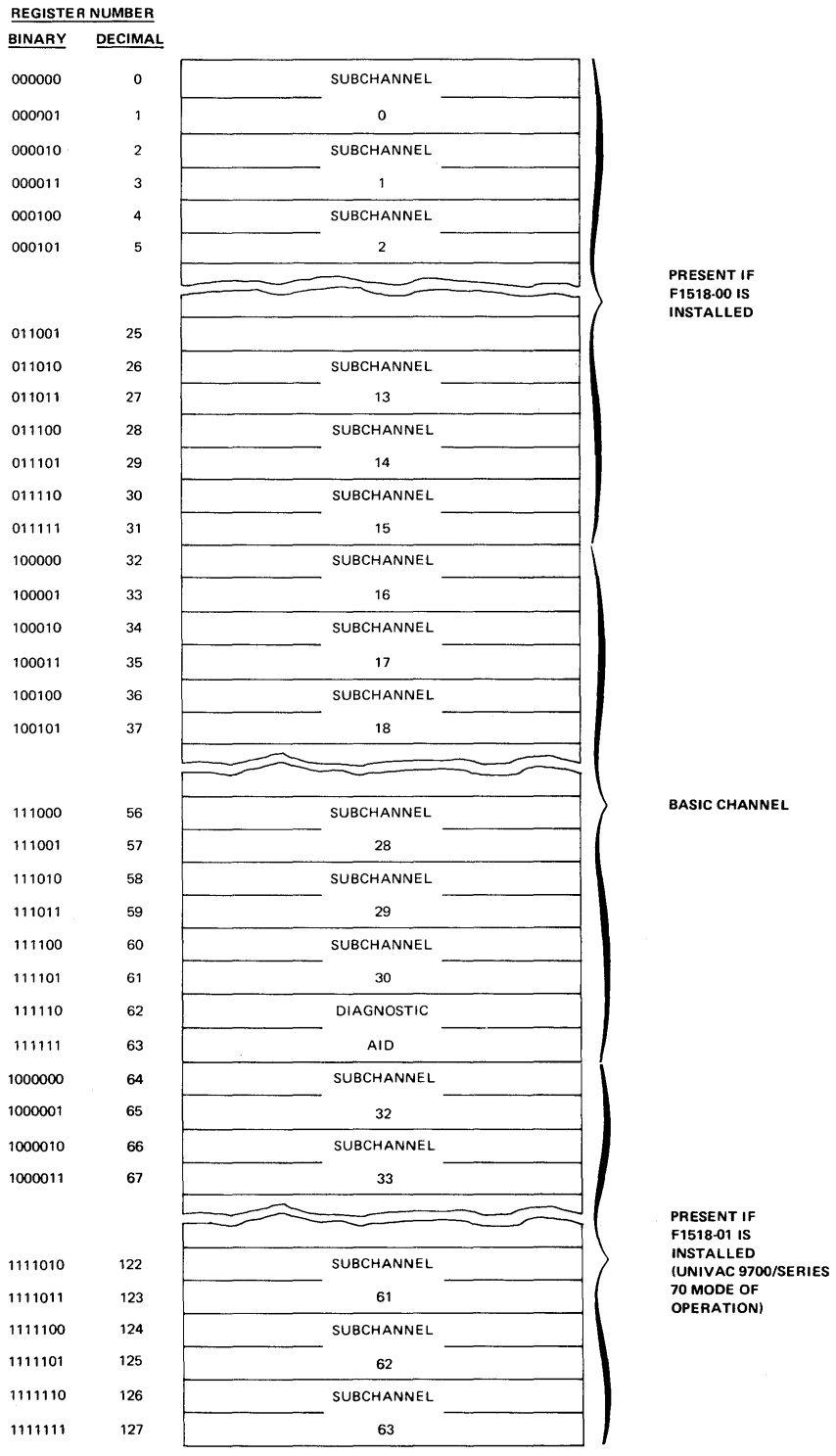


Figure 3–1. Multiplexer Channel – Channel Register Stack (CRS)

When power to the multiplexer channel is deactivated, the resulting state of the bits in the CRS is unpredictable. Furthermore, the SYSTEM RESET signal has no effect on the CRS. Therefore, after power is returned to the channel, the software is required to clear all installed locations (128 maximum) to 0 in the CRS. This may be accomplished by way of a succession of LCHR instructions. Failure to do this leads to unpredicatable results. ←

3.3. SELECTOR CHANNEL

The selector channel controls the exchange of information between any of eight possible subsystems and the processor and main storage. The selector channel operates in burst mode. In burst mode, one of the eight possible subsystems retains control of the interface for the duration of the execution of the I/O operation. Simultaneously, other subsystems may be executing previously initiated operations, such as rewinding tape that does not involve operations with the I/O channel by way of the I/O interface.

Since the selector channel is required to maintain data transfer with only one subsystem at a time, there is only one subchannel in the selector channel. Thus, the channel and subchannel in the selector channel may be considered to be identical entities.

3.3.1. Device Addresses

The format and description of the bits for the device address assigned to each subsystem operating with the selector channel are as follows:

1cccdddd

where:

1ccc — is the binary address of one of eight possible control units connected to the I/O interface.

dddd — is the binary device number of one of 16 devices that may be connected to a shared control unit. A nonshared control unit may use any one of the 16 combinations of d bits.

The 8-bit device address is used by the processor when it issues an I/O instruction to the selector channel and by the channel and subsystem during the prescribed operation.

3.3.2. Status Handling

When device status is accepted or certain subchannel status develops asynchronously during the course of an I/O operation, the selector channel generates a request for an interrupt. Upon receipt of the signal allowing the interrupt, the channel writes a CSW into low order main storage and clears the status in the channel. The subchannel status that causes the channel to request an interrupt includes the following detected condition during the initial selection sequence (other than the first) executed for command chaining:

- PCI
- interface control check
- program check
- protection check
- channel control checks

During command chaining, when a channel control, program, or protection check condition is detected during an initial selection sequence (other than the first) or if an interface control condition arises, the channel disconnects the device by way of a selective reset sequence. An interrupt request is immediately generated and all further operations are halted until the interrupt is made.

Similarly, when asynchronous device status is accepted, an interrupt request is immediately made and all other operations are halted. If a device is the type that presents separate channel end and device end indications, the device end status is not accepted from the device until after the interrupt for the channel end is made.

The PCI is unique to the subchannel status in that, although an interrupt request is made as soon as possible after the detection of the PCI, other channel operations are allowed to proceed normally. When the interrupt is finally allowed, a CSW with the PCI is written.

The remaining subchannel status indications (channel data check, incorrect length program check, protection check, and channel control check during data transfer) are presented in the CSW written with the ending device status.

If an I/O instruction of the type that can relieve pending status is issued before the interrupt is allowed, the pending status is written in the ISW and the interrupt condition is cleared in the channel.

3.4. CHANNEL PROGRAMMING

Both the multiplexer and selector channels have the facility for controlling a series of I/O operations by means of a channel program. A channel program consists of a set or chain of channel command words (CCW'S) that are constructed by the software. The main storage address of the first CCW of the chain is designated in the first CCW address field of the channel address word (CAW). When a start-I/O (SIO) instruction is issued, the channel processes the first CCW and, independently of the processor, any remaining CCW's in the chain. Unless exceptional conditions arise, interrupt requests are made only when ending device status is presented during the operation controlled by the last CCW of the chain or when PCI flags are detected in any CCW of the chain (except the first). Within the chain of CCW's, an existing operation can be extended, a new operation initiated, writing into a buffer in main storage may be skipped, a given CCW may be skipped, and noncontiguous CCW's may be linked by branching within the channel program. The facilities available within channel programming include: command chaining, data chaining, transfer-in-channel commands, CCW skipping, and input buffer skipping.

3.4.1. Command Chaining

Command chaining is specified to the channel by the cc flag (bit 33) in a CCW. The presence of the cc flag in a CCW is recorded in the channel or subchannel registers (bit 33) of the multiplexer HCCW when the operation controlled by the given CCW is initiated. If at the end of this operation the device presents channel end status and no exceptional conditions exist, the status is accepted from the device, but no interrupt request is made.

The channel end indication is not preserved in the channel or subchannel. When the device returns normal device ending status (device end or channel end and device end), the channel accesses the next CCW in the chain. No interrupt request is made and the device ending status is not preserved. The address of the CCW accessed is stored in the next CCW address field of the hard channel address word (HCAW). The channel then proceeds to reconnect the device by way of an initial selection sequence and to initiate the operation indicated in the new CCW.

The operation is terminated immediately if:

- a storage error is detected during the access of the HCAW (multiplexer channel only);
- a storage error is detected during the access of the CCW;
- a storage error is detected in the access of the relocation register; or
- a format error is detected in the contents of the CCW.

In the selector channel, a SELECTIVE RESET signal is issued to the device, the appropriate subchannel status is set, and the channel is then placed in the interrupt pending state.

The multiplexer channel terminates the operation by the use of a pseudo-TIO command. If any storage or format errors occur, the multiplexer channel forces the output bus to all 0's at command out time during initial selection sequence (ISS). The all-0 command byte appears to the device as a TIO operation, and the device responds with a current device status, which should be 0. The multiplexer channel handles this terminate operation as a normal device status sequence and either accepts or stacks the status, depending on the type of subchannel and present state of the channel. In any case, when the device status is accepted and the appropriate status word is written, the subchannel status field indicates the condition that caused the command chaining to be suppressed. Similarly, if during the initial selection sequence the device returns an abnormal device status the operation is halted. Abnormal device status consists of anything other than 0 status (if command chaining is not specified in the new CCW), or anything other than channel end or channel end and device end, with or without status modifier (if command chaining is specified). The abnormal status is accepted or stacked in the device as required, and the interrupt pending state is entered in the selector channel or multiplexer standard subchannel. If the device returns a 0 status during the initial selection sequence, the operation proceeds normally under control of the new CCW.

Unless the SLI flag of the current CCW is set to 1, an incorrect length condition causes command chaining to be suppressed. If both command chaining and data chaining are specified in the current CCW, data chaining is executed. However, if data chaining with or without command chaining is specified in a CCW that also specifies a control-immediate command, the opertion is halted and the interrupt pending state is entered.

If the current CCW specifies command chaining and the device returns device status with the device end or channel end indications accompanied by the status modifier bit, the channel does not access the next CCW in the chain. Rather, the channel address, which is the value in the next CCW address field of the HCAW, is incremented by 8 and accesses the CCW with the resulting address. In this way, the channel program contains a branch which is dependent upon device conditions.

3.4.2. Data Chaining

Data chaining permits the software to transfer data to and from noncontiguous areas in main storage without requiring separate SIO instructions for each area. Data chaining is specified to the channel by the CD flag (bit 32) in a CCW. The presence of the CD flag in a CCW is recorded in the channel or subchannel registers (bit 32 of the multiplexer HCCW) when the operation controlled by a given CCW is initiated. When the byte count in the HCCW is decremented to 0, if data chaining is specified and no exceptional conditions have occurred, the channel accesses the next CCW in the chain. The address of the new CCW is stored in the next CCW address field of the HCAW. The data transfer then continues under the control of the data address, byte count, and flags of the new CCW. Unless a transfer-in-channel (TIC) command is specified, the command code field of the new CCW is disregarded by the channel. The process of data chaining is entirely transparent to and independent of the device.

If any storage errors on the access of the new CCW or any format errors in the contents of the new CCW are detected, the operation is immediately terminated. Both the multiplexer and selector channels terminate the device by generating the COMMAND OUT signal in response to the SERVICE IN signal. The appropriate subchannel status is set and presented to the software when a status word containing the ending device status is written. An incorrect length condition always suppresses data chaining irrespective of the presence of the SLI flag. A channel data check condition does not suppress data chaining.

If both command chaining and data chaining are specified in the current CCW, data chaining is executed. However, if data chaining with or without command chaining is specified in a CCW that also specifies a control immediate operation, the operation is ended at the point where the device presents its status during the initial selection sequence. The status is then accepted or stacked in the device as required by channel conditions.

If the device presents ending status before any data designated by the new CCW is transferred, an incorrect length indication results. However, the remaining byte count and next address fields of the CSW, TSW, or ISW reflect the new CCW.

The channel operations are defined as follows:

Channel Operation	Description
End	The channel considers the operation to be ended. If the operation is an immediate type and has been specified by the first CCW of a chain, a complete ISW is written and the condition code is set to 01_2. In all other cases, the channel handles the device status as required by the type of channel and subchannel.
Terminate	The device is signaled to terminate the transfer via the command-out-to-service-in response. The subchannel remains active until the channel end status is received.
IL	Incorrect length bit (bit 41) is set in the subchannel status field of the status word (CSW, TSW, ISW).
Command chain	The channel fetches the next CCW upon receipt of the device end status.
Data chain	The channel fetches the next CCW immediately upon detecting the exhaused byte count.
Note	This situation cannot occur in normal operations since the channel accesses a new CCW immediately after the byte count of the current operation is exhausted. The channel does not recognize the next request of the device for service (for data or status) until after the new CCW has been accessed and analyzed for validity. A byte count of 0 in the new CCW is a program check condition which would cause the channel to terminate the operation.

Table 3–4 lists the channel programming operation.

HCCW Flags			Channel Operation			Byte Count Nonzero, Ending Status from Device
CD	CC	SLI	Immediate Operation	Byte Count Exhausted, Device Still Active	Byte Count Exhausted, Ending Status from Device	
0	0	0	End, –	Terminate, IL	End, –	End, IL
0	0	1	End, –	Terminate, –	End, –	End, –
0	1	0	Command chain	Terminate, IL	Command chain	End, IL
0	1	1	Command chain	Command chain	Command chain	Command chain
1	0	0	End, –	Data chain	Note	End, IL
→ 1	0	1	End, –	Data chain	Note	End, IL
→ 1	1	0	End, –	Data chain	Note	End, IL
→ 1	1	1	End, –	Data chain	Note	End, IL

Table 3–4. Channel Programming Operation

3.4.3. Transfer in Channel

If the command code field of a CCW access during either data or command chaining specifies a transfer-in-channel (TIC) command, the data address field of the CCW is used to immediately access the next CCW. In this manner, the software can link noncontiguous chains of CCW's.

If any main storage or format errors are detected in the access of the first word of the CCW containing the TIC, the operation is terminated. If the first word of the CCW specifies a TIC command, format and storage errors are disregarded in the second word. If a TIC command is specified in the CCW addressed by a TIC command, the program check bit is set in the subchannel status word and the interrupt pending state is entered. If a TIC command is specified in the first CCW of a chain, the operation is aborted and the program check bit in the ISW is set.

3.4.4. Input Buffer Skipping

Input buffer skipping allows the software to cause a read, read backward, or sense operation at the subsystem without writing the data into main storage. Input buffer skipping is indicated to the channel by the SK flag (bit 35) in a CCW. The input operation specified by the CCW is executed normally in all aspects except that the reference of main storage is inhibited. If an output transfer is specified, the SK flag is ignored. No storage errors or I/O interface parity errors are possible when the SK flag is set in a CCW controlling an input operation.

3.4.5. Relocation in I/O

Relocation in I/O is a means by which the software provides program relative addresses which the channels use in the computation of absolute main storage addresses at execution time. All addresses specified by software in CAW and CCW are relative addresses. At the time that the channel fetches a particular control word, a value is fetched from the appropriate relocation register in low order main storage and added to the relative address. The resulting absolute address is preserved by the channel and subchannel for use in the execution of the operation. The particular relocation register accessed depends on the value of the I/O storage protection and relocation key and the state of certain control flags.

Detailed description of relocation in I/O is as follows:

- Upon accessing a CAW, the channel determines whether the Z bit (bit 4) in the CAW is set to 1. If the Z bit is not set to 1, the channel uses the contents of the key field (bits 0 through 3) in the CAW to access a relocation value from the relocation registers in low order main storage. If the Z bit is set to 1, the channel uses a 0 key value to access the relocation value. In either case, the relocation value is added to the contents of the first CCW address field in the CAW. The resultant absolute address is used to fetch the first CCW. The relocation value is then added to the contents of the data address field of the first CCW, and the resultant absolute address is stored in the HCCW for use in the accessing of the main storage for data transfers. Also, in the selector channel, the relocation value is stored in an internal channel register for subsequent use. In both the selector and multiplexer channels, the value of the Z bit is preserved in the HCCW within the channel.

- Once the particular operation is initiated, the absolute address stored in the HCCW is used to access main storage for data transferred under control of the current CCW. If command chaining or data chaining occurs, the contents of the data address field of the new CCW is again relocated. If the Z bit (bit 40) in the HCCW is set to 0, the relocation value is obtained by the multiplexer channel from the relocation register accessed with the original CAW key value and by the selector channel from its internal register. If the Z bit in the HCCW is set to 1, the relocation value is obtained by the selector channel from its internal register and by the multiplexer channel from the relocation register accessed with a key value of 0.

- If at any time a CCW specifying a TIC command is accessed, the channel determines whether the K bit (bit 3) in the command code field of the CCW (3.6.3) is set to 1. If the K bit is not set to 1 and the Z bit preserved in the HCCW is set to 1, the channel relocates the contents of the TIC command CCW address field with the relocation value which the selector channel obtains from its internal registers and which the multiplexer channel obtains from the relocation register accessed with the 0 key.

  If both the Z and K bits are set to 1, the channel relocates the CCW address with a value obtained by accessing the relocation register with the original CAW key value. The channel then sets the Z bit to 0, and the selector channel stores the new relocation value in its internal register. If the Z bit is set to 0, the channel then ignores the K bit and relocates the CCW address with a relocation value obtained by the selector channel from its internal register and by the multiplexer channel from the relocation registers accessed with the original CAW key value. The relocation value used in the various instances is summarized as follows:

K (Bit 3 in cc field of CCW)	Relocation Value Obtained From:		
	Z (Bit 4 of CAW)	Selector Channel	Multiplexer Channel
0	0	Internal register (previously stored value obtained with CAW key)	Relocation register with CAW key
0	1	Internal register (previously stored value obtained with 0 key)	Relocation register with 0 key
1	0	Internal register (previously stored value obtained with CAW key)	Relocation register with CAW key
1	1	Relocation register with CAW key	Relocation register with CAW key

- Since the next CCW address stored in the HCAW is always an absolute address, the value written in the next CCW address field of a CSW, TSW, or ISW is always an absolute address.

3.4.6. I/O Interface Error Snapshot

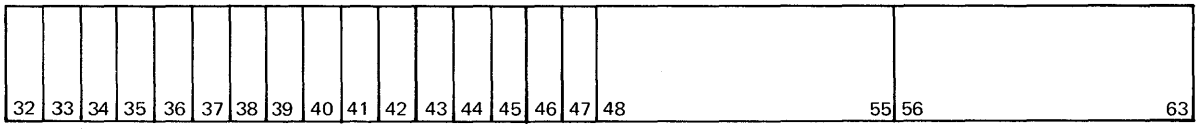
When the channel detects the type of error condition that would cause the setting of the interface control check bit in the status word (CSW, TSW, ISW), the channel executes the following steps to facilitate maintenance.

1. A record or snapshot of the state of all 31 interface lines at the time the interface error condition is detected is preserved within the channel hardware.

2. Following the snapshot, the channel continues handling the error.

3. The selector channel activates the INTERRUPT REQUEST signal and enters the interrupt pending state. When the interrupt priority is received, the selector channel stores a modified form of the CSW. The first word of the double-word CSW is stored as usual. The second word stored contains the snapshot in the following format:

CSW WORD 2 – SNAPSHOT FORMAT



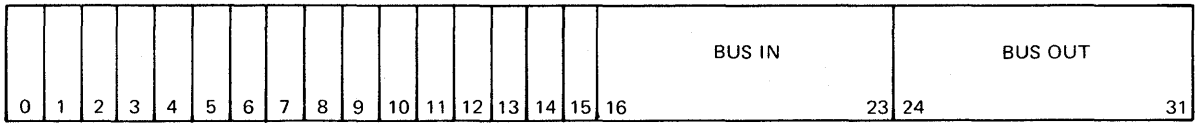
Bit Position	I/O Interface Signal
32	ADDRESS OUT
33	SELECT OUT
34	OPERATIONAL IN
35	ADDRESS IN
36	COMMAND OUT
37	STATUS IN
38	SERVICE IN
39	SERVICE OUT
40	SELECT IN
41	SUPPRESS OUT
42	OPERATIONAL OUT
43	HOLD OUT
44	REQUEST IN
45	INTERFACE CONTROL CHECK BIT
46	BUS OUT PARITY
47	BUS IN PARITY
48–55	BUS IN BITS 0–7
56–63	BUS OUT BITS 0–7

A bit position set to 1 indicates that the corresponding interface line was active (high) at the time the interface error occurred.

4. The multiplexer channel sets the interface control check bit (bit 13 of HCCW terminate mode) and reset mode (bits 4 through 6 of HCCW normal format) and stores the snapshot in register 62_{10} in the channel register stack.

The contents of register 62 is as follows:

REGISTER 62 – SNAPSHOT



Bit Position	I/O Interface Signal
0	ADDRESS OUT
1	SELECT OUT
2	OPERATIONAL IN
3	ADDRESS IN
4	COMMAND OUT
5	STATUS IN
6	SERVICE IN
7	SERVICE OUT
8	SELECT IN
9	SUPPRESS OUT
10	OPERATIONAL OUT
11	HOLD OUT
12	REQUEST IN
13	INTERFACE CONTROL CHECK BIT
14	BUS OUT PARITY
15	BUS IN PARITY
16–23	BUS IN BITS 0–7
24–31	BUS OUT BITS 0–7

To inhibit the channel from overlaying one snapshot with a subsequent one, the channel must examine bit 13 of register 62 before loading a snapshot. If bit 13 is set to 1, the channel bypasses storing the snapshot. If bit 13 is a 0, the channel executes a store operation. Therefore, after the software has examined the snapshot as the result of an SCHR instruction, it must clear the register to all 0's by means of an LCHR instruction.

3.5. CHANNEL TESTER

The channel tester, which is a subsystem resident within the processor, may be connected under software control to the multiplexer or any one of the selector channels. The channel tester is intended to be used with diagnostic and maintenance programs to determine the operating status of the multiplexer and selector channels. The channel tester therefore contains many of the characteristics of the peripheral subsystems normally connected to the multiplexer and selector channels.

3.5.1. Functional Characteristics

The channel tester is to be connected to only one channel at any given time. The connection is made within the channel in such a manner that the I/O interface cable driver and receiver circuits are bypassed. Except for the drivers and receivers, operation of a channel with the channel tester exercises the channel hardware used by a normal peripheral subsystem connected to the channel.

The execution of the diagnose instruction with the secondary operation code of 80_{16} and appropriate channel address causes the selected channel to block the I/O bus interface, enables the interface to the channel tester and generates a system reset sequence in the selected channel. The execution of the diagnose instruction with a secondary operation code of 81_{16} causes the selected channel to disconnect itself from the channel tester and reconnect to the I/O bus. When no channel is connected to the channel tester, the channel tester is in a system reset condition.

The channel tester operates in either multiplexer mode or burst mode. It responds to a channel-initiated burst mode operation and may be set to cause a control unit initiated burst mode operation.

The channel tester simulates a shared control unit with two devices. The tester, therefore, responds to two device addresses and simulates only one device at a time. After the execution of either a system or selective reset, the tester presets to recognize device addresses $C0_{16}$ and $C1_{16}$. Subsequently, the device address may be set to any value, bbbb bbbx, where x is ignored.

The channel tester may be assigned a device address which is associated with a DCS subchannel; thus, a test is furnished for the status table subchannel.

Turnaround data transfers between the channel and channel tester are accomplished by a 32-byte buffer storage within the channel tester. The buffer storage is addressed by a 5-bit address counter, which may be either preset to a specified value by issuing a control command or initialized to address either byte 0 or byte 31_{10} before the execution of any data transfer. If the address counter is neither preset nor initialized and if it has not been cleared to 0 by a selective or system reset sequence, its value remains as it was at the end of the preceding data transfer operation. The address counter is updated after each byte is transferred.

Data transfers may be either fixed or continuous as specified by the command code. A continuous command conditions the channel tester to transfer data until the operation is terminated by the channel. If during the continuous transfer the address counter is updated to either 31_{10} or 0, it wraps around to 0 or 31_{10}, respectively, and the transfer continues. A fixed command conditions the tester to terminate the data transfer when the address counter is incremented beyond 31 or decremented (for the read-backward command) beyond 0.

3.5.2. Command Codes

The channel tester recognizes and responds to the command codes listed in Table 3–5. Any other codes are rejected, the unit check bit is set to 1 in the device status byte presented during the initial selection sequence, and the command reject bit is set in sense byte 1 (3.5.5).

Command	Command Code Bit Position							
	0	1	2	3	4	5	6	7
Write continuous	0	0	0	X	0	0	0	1
Write fixed	0	0	0	X	0	1	0	1
Read continuous	0	0	0	X	0	0	1	0
Read fixed	0	0	0	X	0	1	1	0
Read backward continuous	1	0	0	X	1	1	0	0
Read backward fixed	0	0	0	X	1	1	0	0
Search continuous	0	0	0	X	1	0	0	1
Search fixed	0	0	0	X	1	1	0	1
Sense	0	0	0	0	0	1	0	0
Test I/O	0	0	0	0	0	0	0	0
Set mode	0	0	0	0	0	1	1	1
Set device address register	1	0	0	1	1	1	1	1
Set address counter	0	0	0	0	1	1	1	1
No op	0	0	0	0	0	0	1	1

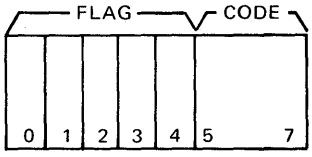
NOTE:

If X (bit 3) is set to 1, the address counter is initialized to 0 before the transfer begins.

Table 3–5. Command Codes for Channel Tester

3.5.3. Mode Byte

The set-mode command causes the channel tester to accept one byte (mode byte) to condition the mode register. The mode register flip-flops corresponding to the bit pattern of the mode byte are set; all other mode register flip-flops are reset. The mode byte format is:



The mode byte is divided into two parts: flag bits and code bits. Each bit in the flag portion represents one of the following modes:

Flag Bit Position	Mode
0	Forced burst
1	Separate ending status
2	Inhibit request in
3	Zero device address
4	Inhibit selective reset

Any combination of the flag bits may be set in one mode byte.

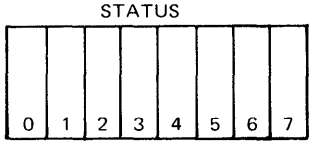
The code bits (bits 5–7) may be set to represent any of the following modes:

Code Bit Position			Mode
5	6	7	
0	0	0	None
0	0	1	Unit check
0	1	0	Even parity on data
0	1	1	Even parity on address
1	0	0	Even parity on status
1	0	1	Interface stall
1	1	0	Two inbound tags
1	1	1	None

The two portions of one mode byte may be independently set to condition the tester to its respectively indicated modes. Care must be taken in the selection of modes to be combined, since the presence of one mode sometimes masks another. For example, if the separate-ending-status mode were combined with the two-inbound-tags mode, the latter mode causes a subsequent operation to be terminated during the ISS. The operation would never progress to the point (ending status) where the former mode was effective.

3.5.4. Status Byte

The status byte furnishes information to the channel on the current state of the channel tester operation. The information is stored in the status register within the channel tester and is presented to the channel during the initial selection sequence at the completion of an operation and in response to a test-I/O command. The status register is cleared when the channel responds to the STATUS IN signal with a SERVICE OUT signal or when a system or selective reset is received.



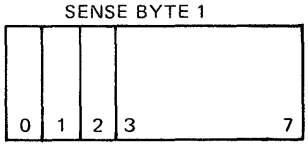
Status Bit Position	Designation
0	Attention
1	Status modifier
2	Control unit end
3	Busy
4	Channel end
5	Device end
6	Unit check
7	Unit exception

3.5.5. Sense Byte

A maximum of three sense bytes may be transferred by the sense command.

- Sense Byte 1

  Sense byte 1 provides detailed information on unusual conditions detected by the channel tester during the previous operation. The significance of the bits in the sense byte is as follows:



Bit Position	Designation
0	Command reject
1	Not used
2	Bus out check
3–7	Not used

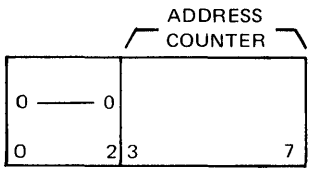
  The register containing sense byte 1 is not cleared by the execution of the sense, test-I/O, or no-op commands. The register containing sense byte 1 is cleared prior to the execution of the write, read, read-backward, search, set-mode, set-address-register, or set-device-address commands.

- Sense Byte 2

  Sense byte 2 contains a copy of the 8-bit mode register (3.5.3). The contents of the mode register are nc affected by the transfer of this byte.

- Sense Byte 3

  Sense byte 3 contains a copy of the 5-bit address counter in the following format:



The contents of the address counter are not affected by the transfer of this byte.

3.5.6. Initial Selection Sequence

The channel tester responds to an initial selection sequence (ISS) by capturing the SELECT OUT signal under the following conditions:

1. The parity of the device address byte transferred on the bus out lines is odd.

2. The device address byte transferred on the bus out lines is equal to the contents of the device address register in the channel tester, bit 7 being ignored for this comparison. If both conditions are not satisfied, the channel tester returns the SELECT IN signal.

3.5.7. Control Unit Busy Sequence

After the channel tester captures the SELECT OUT signal, as specified in 3.5.6, it responds with the STATUS IN signal and a status byte with the busy and status modifier bits transferred on the BUS IN lines under the following conditions:

- The channel tester is executing a previously initiated operation.

- The channel tester contains pending status for other than the addressed device. In order to determine whether the pending status is for the addressed device, a full 8-bit comparison is made between the device address byte and the contents of the device address register in the channel tester.

If the channel tester contains pending status for the addressed device, the ISS continues until the command code is transferred from the channel. If the command code indicates any valid command other than test-I/O, the command is ignored and the pending status accompanied by the busy bit is transferred to the channel at status-in time. If the command code indicates a test-I/O command, the pending status is transferred without the busy bit. If the command code byte is invalid or has even parity, the command is rejected and the status byte contains the pending status and the busy bit. The unit check bit is not set to 1 in this case.

If any of the following modes is set to 1, the listed action causes abnormal conditions in the ISS:

- Two inbound tags
- Zero device address
- Even parity on status
- Even parity on address
- Interface stall

If none of the busy or abnormal conditions exist and the command code is valid and contains odd parity, the status byte transferred at the end of ISS usually contains all 0's to indicate that the command has been accepted and is being executed. However, if the command is a no-op, the status byte transferred during the ISS contains the channel end and device end indications.

3.5.8. Data and Status Transfers

Once the ISS is completed, data and status transfers are executed in a manner depending on whether the channel tester is operating in the multiplexer or burst mode.

- Multiplexer Mode

  After the ISS and after the channel tester disconnects itself from the channel, it then attempts to reconnect to the channel by a control-unit-initiated sequence. If the inhibit-request-in mode is set to 1, the channel tester awaits the SELECT OUT signal transmitted when the channel polls. If the inhibit-request-in mode is not set to 1, the channel tester activates the REQUEST IN signal and then awaits the SELECT OUT response of the channel. Upon receiving the SELECT OUT response, the channel tester continues through the 1-byte transfer sequence and then disconnects itself by dropping the OPERATIONAL IN signal. This connection and disconnection is executed for each byte of data until one of the following conditions occurs:

  – The channel responds to SERVICE IN with COMMAND OUT signals.

  – One byte has been transferred for the set-mode, set-address-counter, or set-device-address-register commands.

  – Three sense bytes have been transferred for the sense command.

  – The address counter has been updated beyond its upper or lower limit for the write, read, read-backward, and search-fixed commands.

  Upon termination of the data transfer, the channel executes a control-unit-initiated sequence to present the ending status byte. If the separate-ending-status mode is set, the channel tester transfers a status byte with channel end and then disconnects itself. Approximately 21 milliseconds later, a control-unit-initiated sequence is executed to transfer the status byte with the device end bit (and possibly other status bits) set to 1.

  If at any time the channel responds to the STATUS IN signal with the COMMAND OUT signal, the status of the channel tester is stacked. If the SUPPRESS OUT signal is not active, the channel tester executes a control-unit-initiated sequence to transfer the status that had been stacked. If the SUPPRESS OUT signal is active, the channel tester is inhibited from executing a control-unit-initiated sequence to present stacked status, (SUPPRESS OUT signal has no effect on control-unit-initiated sequence to transfer either data or unstacked status.)

- Burst Mode

  The channel tester responds properly to a channel-forced burst mode sequence or, if the forced burst mode is set to 1, causes a control-unit-initiated burst mode. In either case, the channel tester remains connected (OPERATIONAL IN signal high) for the duration of an operation from the beginning of the ISS through the presentation of the ending status. If the separate-ending-status mode is set to 1, the status byte containing channel end indication is transferred after the last data byte as part of the burst mode operation. At this time, the channel tester disconnects itself from the channel. Approximately 21 milliseconds later, a control-unit-initiated sequence is executed to transfer the status byte containing the device end indication. The burst mode data transfer continues until one of the following conditions occurs:

- The channel responds to the SERVICE IN signal with the COMMAND OUT signal.
- One byte has been transferred for the set-mode, set-address-counter, or set-device-register commands.
- Three sense bytes have been transferred for the sense command.
- The address counter has been updated beyond its upper or lower limit for the write, read, read-backward, and search-fixed commands.

3.5.9. Selective Reset Sequence

The channel tester responds to the selective reset sequence by disconnecting itself from the channel by deactivating the OPERATIONAL IN signal and all other inbound tag lines, and by clearing the address counter, status, sense, mode, and device address registers. After receipt of the selective reset sequence, the channel tester recognizes device addresses $C0_{16}$ and $C1_{16}$. ←

3.5.10. System Reset Sequence

The channel tester response to the system reset sequence is the same with the response to the selective reset sequence (3.5.9).

3.5.11. Interface Disconnect

The channel tester responds to an interface disconnect sequence by disconnecting itself from the channel as a result of dropping the OPERATIONAL IN signal and all other tag lines. The command that was in progress is terminated, and no more data transfers are attempted. No registers are cleared by the interface disconnect sequence. If the channel tester is performing a data transfer at the time of the disconnect, the appropriate ending status is set and presented afterward by a control-unit-initiated (or initial section) sequence. This status depends on the mode setting and the state of the operation up to the point of disconnect. ←

3.5.12. Command Chaining

The channel tester takes no special action for command chaining.

3.6. INPUT/OUTPUT CONTROL

Control of all I/O operations is accomplished by means of a set of control words located on both main storage (including low order storage) and internal channel hardware. Software conveys information to the channels by means of I/O instructions, channel address words (CAW's), and channel command words (CCW's). The working equivalent of these control words, the hard channel address words (HCAW's) and the hard channel command words (HCCW's), are created and manipulated by the channels during the operations with the I/O subsystems. Information concerning the status of I/O operations is provided to the software in the initial status word (ISW), channel status word (CSW), and tabled status word (TSW) (multiplexer channel only).

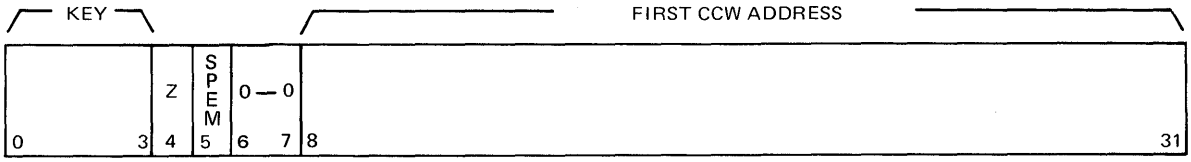
3.6.1. Channel Address Word

The CAW is a full word in low order storage at location 180_{16}. The software is required to load the CAW before each start-I/O (SIO) instruction is issued. The CAW specifies the location of the first CCW and the I/O storage protection and relocation key to be used in the execution of the operation initiated by the SIO instruction. In addition, a bit in the CAW (bit 4) designates whether the value in the key field (bits 0–3) or a 0 is to be used to obtain the relocation value. During the initiation of the SIO operation, the channel adds the appropriate relocation value to the contents of the first CCW address field and writes the results in the appropriate HCAW.

The storage protect escape mode (SPEM, bit 5 of CAW) allows that portion of an I/O operation following the relocation register access to run with a 0 Key instead of that specified in the key field (bit 0–3). The key used to obtain a relocation value is not affected by this mode.

If there is any status information to be provided, the channel writes an ISW in location 180_{16} and thus overlays the CAW just prior to setting the condition code bits in the current PSW.

During execution of the load-channel-register (LCHR) and the store-channel-register (SCHR) instruction, the CAW location 180_{16} is used to transfer information to and from the specified multiplexer channel register. The format and description of the CAW are as follows:

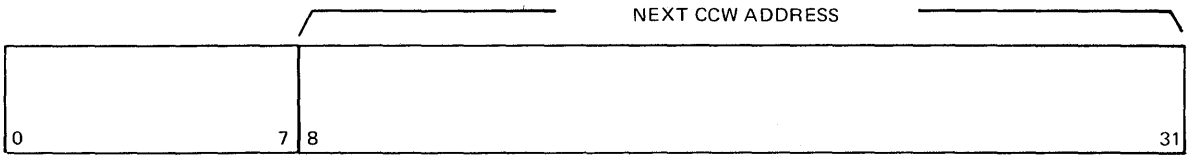


Field	Description
Key (bits 0-3)	These bits contain the 4-bit I/O storage protection and relocation key used by the channel for all storage accesses for data and CCW related to the I/O operation initiated by the SIO instruction. If Z (bit 4) is 0, the key is used to fetch a relocation value to be added to the contents of the first CCW address field. If Z is a 1, a key is used to fetch the relocation value. The contents of the CAW key field are transferred to the key field of the HCCW.
Z (bit 4)	The Z bit designates what key value is to be used for storage protection and relocation purposes. If the Z bit is set to 1, the channel uses a 0 key rather than the contents of the key field to fetch a relocation value that is added to the relative first CCW address to form an absolute first CCW address. The 0 key may also be used to fetch a relocation value to be added to the data address field of the first and subsequent CCW and to the next CCW address field of all transfer-in-channel (TIC) commands in the chain until a TIC with K (bit 3) equal to 1 is encountered. At this point, the original key value is used for all subsequent protection and relocation required in the chain. The value of the Z bit is transferred to the HCCW (bit 40) and is referred to whenever relocation or protection is required.
SPEM (bit 5)	This bit indicates that storage protection is bypassed during the data transfer portion of the I/O operation initiated by a SIO instruction. If SPEM bit is set 1, the channel uses 0 key to fetch or write the data and CCW's associated with the I/O operation in progress. The key used to fetch a relocation value is that specified by the interaction of the key field and the Z bit (bits 0–4) in the CAW and a bit (bit 3) of a transfer-in-channel command (3.4.3).
Bits 6–7	These bits are set to 0.
First CCW address (bits 8-31)	These bits specify the relative address of the double-word location in main storage which contains the first CCW that is to control the I/O operation being initiated by the SIO instruction. After the relocation value has been added to the first CCW address, the result is used to fetch the first CCW. The result is then incremented by 8 and copied into the HCAW.

3.6.2. Hard Channel Address Word

The hard channel address word (HCAW) contains the working equivalent of the CAW. It specifies the address of the next CCW to be accessed when command or data chaining occurs. After the relocation value has been added to the value of the first CCW address in the CAW, the result is incremented by 8 and copied into the HCAW during the initiation of the I/O operation. Thereafter, the value is incremented by 8 each time chaining occurs.

For each subchannel in the multiplexer channel, there is one 32-bit HCAW located in low order storage between locations 200_{16} and 278_{16} inclusive (first 31 channels) and between locations 300_{16} and $37C_{16}$ inclusive (32 subchannel expansion, F1518-01). The format and description of the multiplexer channel HCAW are as follows:

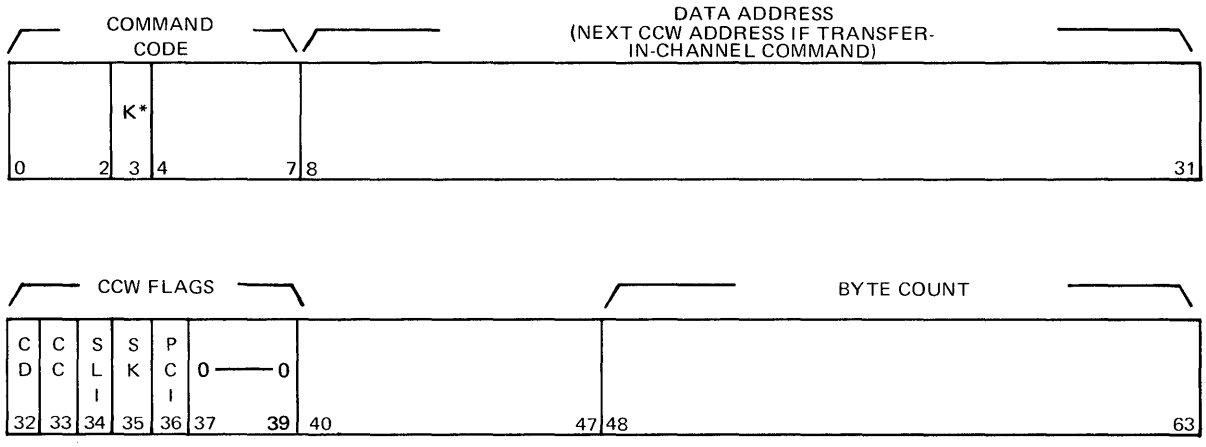


Field	Description
Bits 0-7	These bits are disregarded by the hardware.
Net CCW address (bits 8-31)	These bits contain the absolute address of the double-word location in storage immediately following the current CCW.

3.6.3. Channel Command Word

The CCW is a 64-bit double word located on any double-word boundary in main storage. This word specifies the operation to be performed by the channel or device. The actual control of the operation is handled by the HCCW. The first CCW is accessed during the initiation of an I/O operation by an SIO instruction. The contents of a CCW are transferred to the appropriate HCCW fields and are used to control the I/O operation until either another CCW is fetched during chaining or the operation is completed. Fetching of the CCW by the channel does not affect the contents of the location in main storage.

The format and description of the CCW are as follows:



*TIC only

Field	Description		
Command code (bits 0-7)	These bits specify the operation to be performed by the device and channel. If valid, the command code is transferred to the device during the initial selection sequence of an SIO instruction or a command chain operation. The following is a list of the command codes and general descriptions; the X and M bits are not interpreted by the channel.		
	Command Code Bit Position	**Command**	**Description**
	0123 4567		
	XXXX 0000	Invalid	Upon detection of an invalid command code, the channel aborts the operation and sets the program check bit in the subchannel status field of the status word.

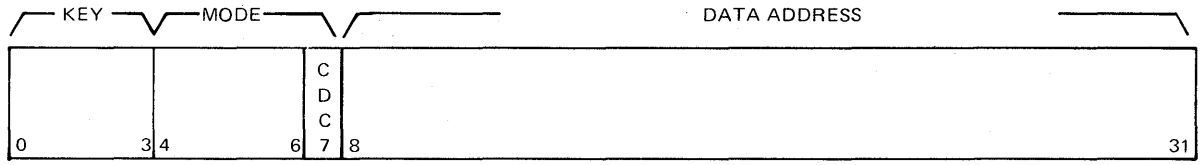
Field	Description			
Command code (bits 0-7) (cont.)	Command Code Bit Position 0123	4567	Command	Description
	XXXK	1000	Transfer in channel (TIC)	This command causes the channel to consider bits 8–31 as the relative address of the next CCW to be fetched. If the Z bit (bit 40) of the HCCW is set to 1 and the K bit (bit 3) of the TIC is not set to 1, the multiplexer channel uses a 0 key to fetch the relocation value to be added to the next CCW address. The selector channel obtains the relocation value from its internal register. If the Z bit is 0, the K bit is ignored. If both the Z and K bits are set to 1, the channel uses the value found in the key field of the HCCW to fetch the relocation value and resets the Z bit to 0. The second word of the CCW is ignored if a TIC is detected in the first word. A program check bit is set in the subchannel status field if a TIC command is detected in the first CCW or the CCW addressed by a TIC also contains a TIC.
	MMMM	0100	Sense	The channel interprets the sense code as an input data transfer (a write into main storage with data address incremented). The program must set up the appropriate byte count and data address fields.
	MMMM	MM01	Write	The write command causes the channel to initiate an output data transfer (a read from main storage with the data address incremented).
	MMMM	MM10	Read	The read command causes the channel to initiate an input data transfer (a write into main storage with the data address incremented).
	MMMM	1100	Read backward	The read-backward command causes the channel to initiate an input data transfer (a write into main storage with the data address decremented).
	MMMM	MM11	Control	The control command code is interpreted by the channel as an output data transfer. Therefore, even if no data bytes are to be transferred, the program must set up a valid byte count (nonzero) field. Control commands that cause the device to present either device end or channel end and device end status during the initial selection sequence are referred to as immediate commands.
Data address (bits 8-31)	These bits contain the relative address of the location in storage into or from which the first byte of data is to be transferred. The channel fetches a relocation value which is added to the data address. The result is stored in the data address field of the HCCW. The key used to fetch the relocation value depends on the value of the Z bit (bit 40) in the HCCW. If the command code field specifies a TIC, bits 8–31 are then considered the relative address of the next CCW. In this case, bits 8–31 must specify a double-word boundary. After relocation is executed and the new CCW is fetched, the result is incremented by 8 bytes and stored in the next CCW address field of the HCAW.			

Field	Description
CD (bit 32)	The CD (chain data) flag is transferred to the corresponding position in the HCCW. The CD flag specifies that, upon completion of the portion of the data transfer controlled by the current CCW, a new CCW is read from storage and the operation is continued under control of the new CCW. The address of the new CCW fetched is found in the next CCW address field of the appropriate HCAW.
CC (bit 33)	The CC (chain command) flag is transferred to the corresponding position in the HCCW. The CC flag specifies that, upon receipt of valid ending device status, a new CCW is fetched and the operation specified by the new command code is initiated. The address of the new CCW is found in the next CCW address of the appropriate HCAW. If both the CC and CD flags are set to 1, data chaining takes place.
SLI (bit 34)	The SLI (suppress length indication) flag is transferred to the corresponding position in the HCCW. If the SLI flag is set to 1, an incorrect length condition is not indicated to the program; and if the CC is 1, command chaining is not suppressed. If data chaining is specified and the device causes an early termination, the incorrect length is indicated whether or not the SLI flag is present. An incorrect length condition arises when the subsystem attempts to transfer a fewer or greater number of bytes than specified in the byte count. NOTE: Many subsystems cannot count the number of bytes transferred. These subsystems require the channel to terminate the data transfer by the COMMAND OUT response to SERVICE IN when the byte count in the HCCW is decremented beyond 0. Although this is a normal termination, an incorrect length condition results and is indicated if the SLI flag is not set.
SK (bit 35)	The SK (skip) flag is transferred to the appropriate position in the HCCW. The SK flag indicates that, for the duration of the data transfer controlled by the CCW containing the SK flag, input data is not written into storage. With respect to the subsystem and subchannel, the input operation proceeds normally. The presence of the SK flag during an input data transfer causes the suppression of all subchannel status except program controlled interrupt, incorrect length, interface control check, and program checks due to CCW format errors. The SK flag is disregarded during output data transfers.
PCI (bit 36)	The PCI (program controlled interrupt) flag causes the channel to generate an interrupt request and write a status word (CSW, TSW) as soon as possible after detection of the flag. The PCI flag is not permitted in the first CCW of a chain. If a PCI is detected in the first CCW of an operation, an ISW with the program check bit is written. In the multiplexer channel, the interrupt for the PCI is made as soon as detected, if possible. For DCS subchannels, a TSW is written immediately. For standard channels, if an interrupt request is not possible upon detection of the PCI because the channel is already in the interrupt pending state, the PCI is then stored in the HCCW. Thereafter, on every reference to the HCCW, the interrupt request for the PCI is similarly attempted. In the selector channel, an interrupt request is activated as soon as the PCI is detected. When the interrupt is allowed, a CSW with the PCI bit set is written. In both the selector and multiplexer channels, if some other condition arises which is sufficient to cause an interrupt (such as ending device status), the PCI is included with the other appropriate status bits when the status word is written. If the PCI is still pending when a new CCW containing a PCI is accessed, the two PCI conditions merge. If the PCI is still pending, when a new CCW not containing a PCI is accessed, the pending PCI condition is retained. Also, if an instruction which is executed causes an ISW write, the PCI is included in the ISW. In all cases, the PCI is presented by way of the CSW, TSW, or ISW only once. Thereafter, the condition is cleared in the channel or subchannel.
Bits 37-39	These bits must be set to 0.
Bits 40-47	These bits are disregarded by the channel.
Byte count (bits 48-63)	The value in the byte count field specifies the number of bytes to be transferred during the operation controlled by the current CCW. The byte count is transferred to the byte count field of the HCCW and is decremented by 1 after each byte is transferred. A byte count of 0 is invalid and results in a program check when detected. The maximum byte count that may be specified is $2^{16}-1$ or 65,535 bytes.

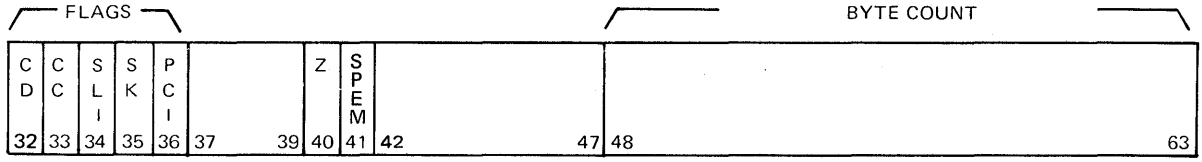
3.6.4. Hard Channel Command Word

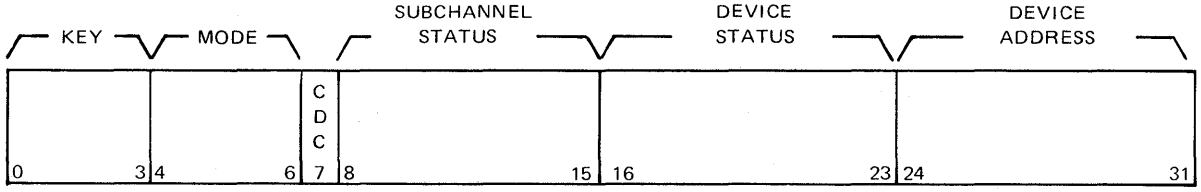
The hard channel command word (HCCW) is a double word. One double word is contained in the internal hardware of each selector channel and 15 double words (31 double words if subchannel feature F1518-00 is installed or 63 double words if CCM subchannel expansion F1518-01 is installed) are contained in the high speed channel register stack within the multiplexer channel. The HCCW contains the working copy of the current CCW. The byte count and data address fields are updated after each byte is transferred. The key and flag fields are used as required. In the multiplexer channel, the HCCW in a second format is also used for holding pending device and subchannel status information.

The two formats with their descriptions for the HCCW are as follows:

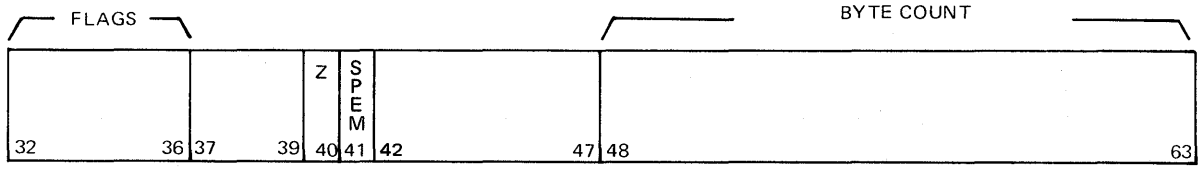


HCCW (NORMAL FORMAT)





HCCW (TERMINATE FORMAT)



<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Key
(bits 0-3)</td><td>Contains the 4-bit I/O storage protection and relocation key that the channel uses for all storage accesses for data and CCW's related to the I/O operation initiation. (Same as key bits of the CAW, 3.6.1).</td></tr>
<tr><td>Mode
(bits 4–7)</td><td>These bits specify the current state of subchannel operation.

Normal Format

<table>
<tr><th rowspan="2">Mode</th><th colspan="4">Bit Code</th><th rowspan="2">Description</th></tr>
<tr><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>Idle</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Indicates subchannel is available for the initiation of an operation by way of an SIO instruction. The mode is set to idle from:

1. terminate mode when any device status excluding BUSY and STATUS MODIFIER has been presented by way of the write of an ISW, CSW, or TSW;

2. reset mode when any device or subchannel status has been presented by way of an ISW, CSW, or TSW.</td></tr>
<tr><td>Subchannel idle alert mode (SIAM)</td><td>0</td><td>0</td><td>0</td><td>1</td><td>Indicates subchannel is available for initiation of an operation via an SIO instruction. The mode is set by a LCHR instruction.

Subchannel responds to a control-unit-initiated sequence for data with terminate response. No data is processed by the subchannel/channel.

The subchannel responds to a control-unit-initiated sequence to present status with a positive acknowledge response. However, the subchannel/channel does not process the status; that is, the status is not stored, no status words are written, no interrupts are generated, and the mode is not changed.

Bit 7 of HCCW is used as temporary storage for channel data when HCCW is not SIAM set.</td></tr>
<tr><td>Active (input, forward)</td><td>1</td><td>0</td><td>0</td><td>X*</td><td>Indicates the subchannel is in the midst of an input data transfer where the data address is being incremented.</td></tr>
<tr><td>Active (input, backward)</td><td>1</td><td>0</td><td>1</td><td>X*</td><td>Indicates the subchannel is in the midst of an input data transfer where the data address is being decremented.</td></tr>
<tr><td>Active (output)</td><td>1</td><td>1</td><td>0</td><td>X*</td><td>Indicates the subchannel is in the midst of an output data transfer where the data address is being incremented.

The mode is set to active from:

1. idle mode when a valid command code is detected in the first CCW associated with an SIO instruction;

2. chain mode when a valid command code (other than a TIC) is detected in any CCW (other than first) associated with command chaining.</td></tr>
</table>
</td></tr>
</table>

*X denotes bit is disregarded in this mode.

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Mode
(bits 4-7)
(cont.)</td><td>
Normal Format
<table>
<tr><th rowspan="2">Mode</th><th colspan="4">Bit Code</th><th rowspan="2">Description</th></tr>
<tr><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>Chain</td><td>1</td><td>1</td><td>1</td><td>X*</td><td>Indicates that the data transfer portion of the previous operation has ended and command chaining is specified. The mode is set to chain from active mode when:

1. the byte count is decremented to 0 and command chaining is specified;

2. the byte count is not exhausted, the device terminates, and command chaining and the SLI flag are specified.

The mode is set to chain from idle mode when the device returns channel end status only during the first ISS associated with an SIO instruction and command chaining is specified.</td></tr>
<tr><td colspan="6">Terminate Format</td></tr>
<tr><td>Terminate</td><td>0</td><td>1</td><td>0</td><td>X*</td><td>Indicates the data transfer portion of the operation has been completed (normally or abnormally) and that chaining is not specified.

The mode is set to terminate from:

1. active mode when the byte count is decremented to 0 and data chaining is not specified; when the byte count is not exhausted, the device terminates and chaining is not specified; or when program, protection or channel-control-check subchannel status develops during the data transfer;

2. idle mode when unsolicited device status is presented by way of a control-unit-initiated sequence. This mode is not set to terminate if SIAM is set.</td></tr>
<tr><td>Reset</td><td>0</td><td>1</td><td>1</td><td>X*</td><td>Indicates subchannel operation has been terminated due to an I/O interface error or a subchannel error during an interrupt initialization sequence (IIS) (other than first) associated with command chaining.

The mode is set to reset from:

1. active mode when an interface control check condition occurs;

2. chain mode when subchannel status develops which causes command chaining to be terminated by way of a pseudo-TIO command.</td></tr>
</table>
</td></tr>
<tr><td>CDC
(bit 7)</td><td>This bit serves as a temporary storage for the channel data check error indication. The channel data check bit is set to 1 when a parity error is detected during the transfer of data to or from main storage or on the input data bus. The data transfer is allowed to continue to its normal completion point, whereupon the channel data check indication is presented with the device ending status when the status word is written.</td></tr>
<tr><td>Data address
(bits 8-31, normal format only)</td><td>These bits contain the absolute address of the location in storage to or from which the next byte of data is transferred. The data address is updated (incremented or decremented depending on mode) by 1 after each byte is transferred.</td></tr>
<tr><td>Flags (bits 32-36, normal format only)</td><td>Same as for the CCW flags (3.6.3).</td></tr>
<tr><td>Bits 37-39</td><td>These bits are disregarded by the channel.</td></tr>
</table>

*X denotes bit is disregarded in this mode.

Field	Description
Z (bit 40)	When set to 1, this bit indicates that, if required, the relocation register in low order storage is to be accessed with a 0 key. When set to 0, this bit indicates that the relocation register is to be accessed with the key found in the HCCW key field.
SPEM (Bit 41)	When set to 1, SPEM bit indicates that all main storage accesses after the fetch of the appropriate relocation register will use a 0 key. When set to 0, SPEM bit indicates that the key specified in CAW (bit 0–4) is used to access storage during the operation in progress.
Bits 42–47	These bits are set to 0 and disregarded by the channel.
Byte count (bits 48-63)	Same as CCW byte count field.
Subchannel status (bits 8-15, terminate format only)	When the HCCW is in terminate or reset mode, the subchannel status field is used to store several subchannel status indications until they can be relieved by a CSW, TSW, or ISW write. The bits are defined as follows:

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Subchannel status (bits 8-15, terminate format only) (cont.)</td><td>
<table>
<tr><th>Bit</th><th>Subchannel Status</th></tr>
<tr><td>8</td><td>Set to 0 by channel</td></tr>
<tr><td>9</td><td>Incorrect length</td></tr>
<tr><td>10</td><td>Incorrect length</td></tr>
<tr><td>11</td><td>Protection check</td></tr>
<tr><td>12</td><td>Set to 0 by channel</td></tr>
<tr><td>13</td><td>Interface control check</td></tr>
<tr><td>14–15</td><td>Channel control check code</td></tr>
</table>
</td></tr>
<tr><td>Device status (bits 16-23)</td><td>These bits are used to store standard device status from the time the status is accepted until it is relieved by way of a CSW or ISW write. The device status bits are as follows:
<table>
<tr><th>Bit</th><th>Device Status</th></tr>
<tr><td>16</td><td>Attention</td></tr>
<tr><td>17</td><td>Status modifier</td></tr>
<tr><td>18</td><td>Control unit end</td></tr>
<tr><td>19</td><td>Busy</td></tr>
<tr><td>20</td><td>Channel end</td></tr>
<tr><td>21</td><td>Device end</td></tr>
<tr><td>22</td><td>Unit check</td></tr>
<tr><td>23</td><td>Unit exception</td></tr>
</table>
</td></tr>
<tr><td>Device addresses (bits 24-31)</td><td>These bits contain the address of the device which was active with the subchannel at the time the mode was set to terminate or reset. Thus, the address is the one associated with the device or subchannel status, if any.</td></tr>
</table>

3.6.4.1. IDLE MODE VARIATION

A variation of the idle mode is provided for handling channel/subsystem operations for subsystems/devices that can accept certain commands when busy. This new idle mode is called subchannel idle alert mode (SIAM).

- SIAM mode is used with standard or DCS subchannels and is intended for very particular applications.
- SIAM mode is set by way of the load-channel-register (LCHR) instruction.
- SIAM mode is specified by bits 4 through 7 of the HCCW set to 0001, which is effectively the idle mode (bits 4 through 6 set to 000) plus bit 7 (CDC) set to 1; however, bit 7 continues to be used as temporary storage for CDC when the HCCW is not in SIAM mode (either normal or terminate formats).

When the SIAM mode is set in a subchannel:

- The subchannel responds to any multiplexer channel I/O instruction in the same manner as when in the normal idle mode.
- The subchannel responds to a control-unit-initiated sequence for data (SERVICE IN) with the terminate response (COMMAND OUT). No data will be processed by the channel/subchannel.
- The subchannel responds to a control-unit-initiated sequence to present status (STATUS IN) with the positive acknowledge response (SERVICE OUT). However, the channel/subchannel does not process the status; that is, the status is not stored, no CSW, TSW, etc., is written, and no interrupt request is generated.

3.7. TIMERS

There are two timers available with the processor: the stall timer and the interrupt timer.

3.7.1. Stall Timer

An I/O interface stall timer is located in the multiplexer channel and each selector channel. The function of the stall timer is to detect any stall condition in the I/O interface operation due to hardware malfunctions in either the channel or the subsystem. Each stall timer consists of timers to detect two distinctive types of stall conditions. The first is a 32-microsecond timer, which detects stalls between the channel and the subsystem. The second is a 15-second timer, which detects inactive periods of excessive duration. Upon detection of either type of stall, the channel issues a selective reset sequence and sets the interface control check bit in the subchannel status word.

When operating in cycle mode, the stall timer is inhibited from indicating a stall condition.

3.7.2. Interval Timer

An interval timer is part of the multiplexer channel. It provides the software with a relative running time count (RTC) and an incremental interrupt count. The counting rate of both is 1 kHz, or once every millisecond. The RTC allows for 4.66 hours of continuous operation before wrapping around through 0 and continuing without an interruption. The interrupt count allows the interval between interrupts to range from 1 millisecond to 65,536 seconds. The RTC may be either incremented or decremented but the interrupt count is only decremented.

The following control words, described in 2.4.2, are used by the timer throughout its operation:

- Hard timer control word (HTCW)
- Timer control word (TCW)
- Timer hard channel address word (timer HCAW)
- Timer channel status word (timer CSW)

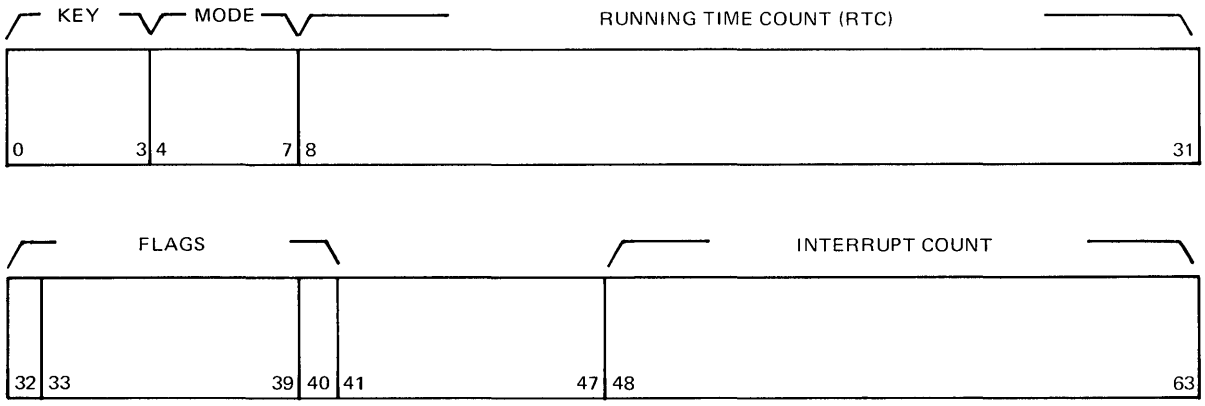
3.8. TIMER CONTROL WORDS

The formats and descriptions of each timer control word (TCW) are presented in the following paragraphs.

3.8.1. Hard Timer Control Word

The HTCW is a double word located in fixed, low order storage location 138_{16}. This is the working control word for the timer operation. The contents of this word are updated once every millisecond and represent the actual counts at any given time. Thus, an instantaneous relative time reference and incremental time count is furnished for the system.

The format and description of the HTCW bits are as follows:



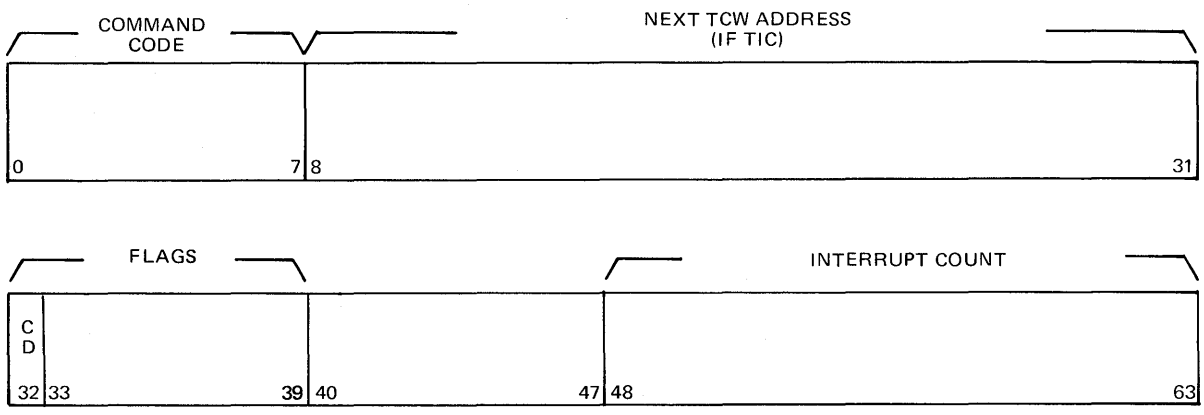
Field	Description
Key (bits 0-3)	Contains a 4-bit storage protection key that is used by the timer subchannel for all storage accesses for TCW's. Since all storage addresses used in the timer subchannel are absolute addresses, the key value is used for protection purposes only (no relocation is accomplished).
Mode (bits 4-7)	Specifies the current state of the timer subchannel operation. The mode can be changed both by channel hardware and the software. The valid modes and associated codes and definitions are as follows:

Mode	Bit Location 4	5	6	7	Definition
Idle	0	0	0	0	When set, count fields (RTC and interrupt count) of HTCW are not updated if this mode is set. However, this mode may only be set by the software via instructions because both the RTC and interrupt count wrap around through 0 without causing termination.
Active	1	0	0	0	Increment RTC; decrement interrupt count.
Active	1	0	1	0	Decrement RTC; decrement interrupt count.
Error terminate	1	0	X	1	Indicates that the subchannel has detected an error other than one on accessing or writing back the first word of the HTCW, has terminated the interrupt count operation only, and has indicated the error condition through the subchannel status information in the timer CSW. In other words, the RTC will continue to be updated with this mode set, but not the interrupt count. The X bit (bit 6) remains unchanged from its previous active mode setting.
Timer abort	0	0	1	1	Indicates that the subchannel has detected an error on accessing or writing back the first word of the HTCW, has terminated the timer operation completely, and has indicated the error condition by way of the subchannel status information in the timer CSW.

Field	Description
Running time count (RTC) (bits 8-31)	Contains a continuous count of the number of millisecond time intervals which have occurred since the field was loaded. The RTC counts to a maximum of 4.66 hours (counting in one direction only) and then wraps around through 0 and continues counting. No interrupt is made when this wraparound through 0 occurs.
CD (bit 32)	Chain data (CD) flag. When set to 1, specifies that, when the interrupt count field of the current HTCW is decremented to 0, a new TCW is accessed from storage and the operation is continued under control of the new interrupt count placed in the HTCW. The location of the TCW is pointed to by the contents of the timer HCAW in low order storage location IFC_{16}
Bits 33–39	These bits are set to 0 by the software and remain 0 under control of the TCW bits (bits 33–40) which are transferred to the HTCW whenever a new TCW is accessed for data chaining.
Bit 40	This bit is set to 0 by the software.
Bits 41–47	These bits are ignored by the hardware.
Interrupt count (bits 48–63)	These bits contains the current interval count that is decremented once every millisecond. Whenever this count is decremented to 0, no timer CSW is written but an interrupt request is generated. If, when this count is decremented to 0, chaining is not specified (CD=0), the count continues to be decremented through 0 on succeeding timer updating operations. If chaining is specified, this count is replaced by a new interrupt count from the TCW. The maximum count from the TCW or the software that can be loaded into the HTCW is all 0's because the interrupt occurs only when the count is decremented from 1 to 0. The range between interrupts then becomes from 1 millisecond to 65.536 seconds.

3.8.2. Timer Control Word

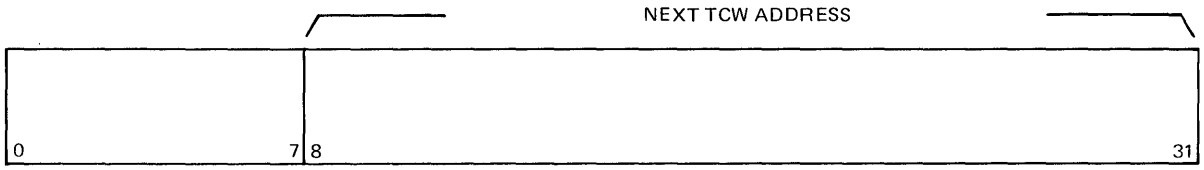
The TCW, which is a double word located anywhere in storage, must be on a double-word boundary. This word provides a new incremental interrupt count during chaining operations, to indicate by way of the chain data (CD) flag that data chaining is to be accomplished the next time the HTCW interrupt count is decremented to 0, and to provide the next TCW address if the operation is to be a transfer-in-channel (TIC). This word is not a working word but only contains information to be transferred to the HTCW. The channel can never change the contents of this word. The TCW are created by the software and can only be changed by the software. The format and description of the TCW are as follows:



Field	Description
Command code (bits 0-7)	The only command code recognized is the TIC command code, XXXX 1000, where the X's are ignored. The TIC command causes the timer subchannel to use the contents of bits 8–31 of the TCW to access the next TCW. This next TCW may not specify a second TIC command. All other codes are ignored. If a TIC is specified, the second word of the TCW is ignored.
Next TCW address (bits 8-31)	These bits contain the address of the new TCW to be addressed if the CC specifies a TIC command; otherwise this field is ignored.
CD (bit 32)	This flag, when set to 1, specifies that, when the interrupt count field of the current HTCW is decremented to 0, a new TCW is read from storage and the operation is continued under control of the new interrupt count in this new TCW. The CD flag is transferred to the HTCW.
Bits 33-39	These bits must be set to 0.
Bits 40-47	These bits are ignored by the hardware.
Interrupt count (bits 48-63)	These bits contain the new interrupt count to replace the interrupt count in the HTCW. This interrupt count specifies a maximum length interval when it is all 0's.

3.8.3. Timer Hard Channel Address Word

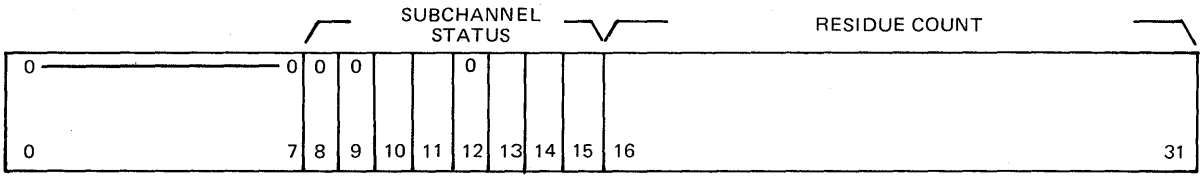
The timer HCAW is a full word located in fixed, low order storage location $1FC_{16}$. This word contains the working copy of the next TCW address, which is an absolute address used for chaining. Each time chaining occurs, the next TCW address of the timer HCAW is incremented by 8 to specify the current value of the next TCW address. The timer HCAW must be set up by the software at system initialization time. The format and description of the timer HCAW are as follows:



Field	Description
Bits 0-7	These bits are disregarded by the hardware.
Next TCW address (bits 8-31)	These bits contain the absolute address of the TCW that is to be read the next time chaining occurs.

3.8.4. Timer Channel Status Word

The timer CSW is a full word located in fixed, low order storage at location $1AC_{16}$. This word is used to alert the software that the timer subchannel has detected an error in its operation. A timer interrupt request is generated after the write of the timer CSW. It should be noted that this same interrupt request is made to indicate that the interrupt count in the HTCW has been decremented to 0. Then, whenever a timer interrupt occurs, the software should check the timer CSW to determine whether a subchannel error has occurred. The format and description of the timer CSW are as follows:



Field	Description
Bits 0-7, 8, 9, and 12	These bits are set to 0 by the hardware.
Program check (bit 10)	This bit is set when the subchannel detects any of the following: ■ TCW format errors: – If TIC, next TCW address does not specify a double-word boundary. – Bits 33–39 not equal to 0. – TCW addressed by a TIC also contains a TIC. ■ Addressing exception on read of TCW.
Protection check (bit 11)	This bit is set when the subchannel detects a storage protection exception on reading a TCW.
Channel control check code, (bits 13-15)	This is a 3-bit code that is set when the channel detects a hardware malfunction. These errors include all storage errors associated with the timer subchannel operation except for addressing and protection exceptions on the reading of a TCW. The channel control check code bits and their interpretation are as follows.

Bit Position 13	Bit Position 14	Bit Position 15	Error Interpretation
0	0	0	No channel control error occurred.
0	0	1	Error occurred during read of HTCW.
0	1	0	Error occurred during write of HTCW.
0	1	1	Not used
1	0	0	Not used
1	0	1	Error occurred during write of timer HCAW.
1	1	0	Error occured during read of timer HCAW.
1	1	1	Error occurred during read of TCW.

Since the channel control check is a 3-bit code, successive channel control checks cannot be meaningfully merged. Therefore, the code written in the subchannel status field designates the first error that occurred.

Residue count (bits 16-31)	These bits contain the value of the interrupt count field of the HTCW at the time the timer CSW is stored.

3.8.5. Timer Subchannel Initialization

Unlike the normal data subchannels, the timer subchannel must be initialized by direct manipulation of the TCW's by the software. This involves the following steps:

1. The mode field of the HTCW is set to idle (all 0's) to inhibit any further TCW manipulation by the subchannel hardware.

2. The T bit (bit 3) in the system mask of the current PSW is set to 1 to allow possible outstanding timer interrupt. After this interrupt is allowed, no further timer interrupt requests are possible as long as the HTCW mode is set to idle.

3. The next TCW address field of the timer HCAW must be loaded with the absolute address of the TCW which is to be accessed when the initial interrupt count is decremented to 0.

4. The interrupt count and RTC fields of the HTCW must be loaded with the desired initial values.

5. The key field of the HTCW must be loaded with the value that corresponds to the area in storage in which the TCW chain is stored.

6. The CD flag of the HTCW must be set to 0 or 1 as desired.

7. Bits 33–40 of the HTCW must be set to 0.

8. The timer CSW location ($1AC_{16}$) should be cleared to all 0's.

9. The mode field of the HTCW should be set to active, incrementing or decrementing the RTC as desired.

4. MAIN STORAGE

4.1. GENERAL

Main storage of the UNIVAC 9700 System is nondestruct readout plated-wire storage with a cycle time of 600 nanoseconds for read or write of one word (four bytes). This rate remains the same for one, two, three, or four bytes.

The smallest addressable unit of storage is one byte which is eight binary bits. With each byte is associated a parity bit. Bytes are numbered consecutively from 0 to a maximum of 1,048,575. These numbers are the byte addresses. Other addressable units of storage and their address requirements are:

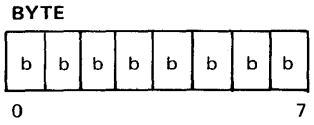
Unit	Number of Bytes	Beginning Address
Half word	2	Multiple of 2
Word	4	Multiple of 4
Double word	8	Multiple of 8

Storage functions are an integral part of the central processor even though housed in one or more separate cabinets. Minimum capacity is 131K bytes and may be increased to a maximum of 1048K bytes. ←

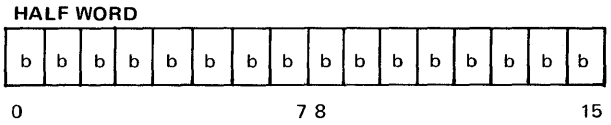
Each main storage cabinet can contain up to 131,072 bytes of storage. Switches on each cabinet are used to assign the beginning storage address, a multiple of 131,072 for each cabinet. If it becomes necessary to place a cabinet offline, another cabinet can be assigned the beginning address of the former cabinet if it is necessary to preserve the consecutive address series of the remaining storage.

4.2. INFORMATION POSITIONING

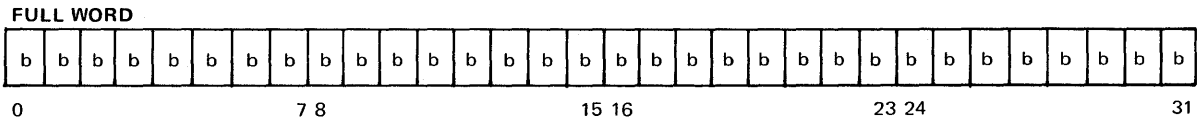
Since main storage locations are addressed consecutively, bytes may be accessed separately or in groups. A group of bytes is addressed by the leftmost byte of the group. The bits in a byte are numbered from left to right starting with 0.



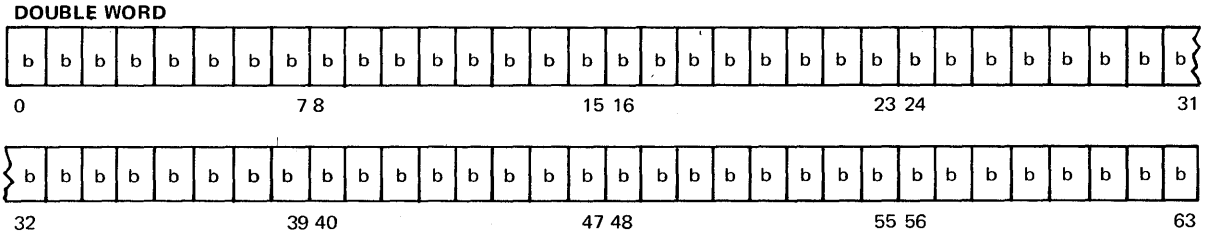
Half-word formats consist of two consecutive bytes:



Full-word formats consist of four consecutive bytes:



Double-word formats consist of eight consecutive bytes:



Variable formats consist of a variable number of consecutive bytes:

Fixed-length fields such as half words and full words have integral boundaries and must be loaded into main storage so that the address is evenly divisible by the field length in bytes. Thus, a half word must have an address that is a multiple of 2, a full word must have an address that is a multiple of 4, and a double word must have an address that is a multiple of 8. The binary address of these fields must contain 0's in the low order bit positions as indicated:

Field	Binary Address
Byte	X XXXX
Half Word	X XXX0
Word	X XX00
Double word	X X000

Variable-length data fields are not restricted by their boundaries. Instructions must begin on half-word boundaries and must have lengths of two, four, or six bytes.

4.3. FIXED STORAGE ASSIGNMENTS

The first 640 bytes of main storage are reserved for special hardware uses, such as the initiation and control of input/output (I/O) operations, program interrupt execution, interval timer, and program analysis, and may be directly addressed by software functioning in the supervisor state (Figure 4–1). The problem mode can also access low order storage if allowed by the software and storage protect assignments.

Byte Address (Hex) X =	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00X	(Reserved)															
01X	Mux Chnl 0 - Status Tbl				Old PSW				Mux Chnl 0 - Status Tbl				New PSW			
02X	Mux Chnl 0 - Standard								Mux Chnl 0 - Standard							
03X	Selector Chnl 1								Selector Chnl 1							
04X	Selector Chnl 2								Selector Chnl 2							
05X	Selector Chnl 3								Selector Chnl 3							
06X	Selector Chnl 4								Selector Chnl 4							
07X	Chnl 5 - CIC								Chnl 5 - CIC							
08X	Chnl 6 - OSSF								Chnl 6 - OSSF							
09X									(Reserved)							
0AX	Interval Timer								Interval Timer							
0BX	External								External							
0CX	Supervisor Call								Supervisor Call							
0DX	Machine Check								Machine Check							
0EX	Program Exception								Program Exception							
0FX	Program Analysis								Program Analysis							
10X	(Reserved for CIC)								(Reserved for CIC)							
11X	(Reserved for CIC)								(Reserved for CIC)							
12X	(Reserved for CIC)								(Reserved for CIC)							
13X	HSTCW								HTCW							
14X	Rel Reg 0				Rel Reg 1				Rel Reg 2				Rel Reg 3			
15X	Rel Reg 4				Rel Reg 5				Rel Reg 6				Rel Reg 7			
16X	Rel Reg 8				Rel Reg 9				Rel Reg A				Rel Reg B			
17X	Rel Reg C				Rel Reg D				Rel Reg E				Rel Reg F			
18X	CAW/ISW_1				ISW_2				Trace Tbl Addr				Absent IACW			
19X	(Reserved)															
1AX	Mux Standard CSW								Mux Sta Tbl CSW				Timer CSW			
1BX	Selector Chnl 1 CSW								CIC CSW							
1CX	Selector Chnl 2 CSW								OSSF CSW							
1DX	Selector Chnl 3 CSW								(Reserved)							
1EX	Selector Chnl 4 CSW								(Reserved)							
1FX	(Reserved)								Mux Sta Tbl HCAW				Timer HCAW			
20X	Mux Subch 0 HCAW				Mux Subch 1 HCAW				Mux Subch 2 HCAW				Mux Subch 3 HCAW			
21X	Mux Subch 4 HCAW				Mux Subch 5 HCAW				Mux Subch 6 HCAW				Mux Subch 7 HCAW			
22X	Mux Subch 8 HCAW				Mux Subch 9 HCAW				Mux Subch 10 HCAW				Mux Subch 11 HCAW			
23X	Mux Subch 12 HCAW				Mux Subch 13 HCAW				Mux Subch 14 HCAW				Mux Subch 15 HCAW			
24X	Mux Subch 16 HCAW				Mux Subch 17 HCAW				Mux Subch 18 HCAW				Mux Subch 19 HCAW			
25X	Mux Subch 20 HCAW				Mux Subch 21 HCAW				Mux Subch 22 HCAW				Mux Subch 23 HCAW			
26X	Mux Subch 24 HCAW				Mux Subch 25 HCAW				Mux Subch 26 HCAW				Mux Subch 27 HCAW			
27X	Mux Subch 28 HCAW				Mux Subch 29 HCAW				Mux Subch 30 HCAW				(Reserved)			

Figure 4–1. Fixed Main Storage Assignments (Part 1 of 2)

Byte Address (Hex) X=	0 1 2 3	4 5 6 7	8 9 A B	C D E F
28X	(Reserved)		(Reserved)	
29X	(Reserved)		(Reserved)	
2AX	(Reserved)		(Reserved)	
2BX	(Reserved)		(Reserved)	
2CX	(Reserved)		(Reserved)	
2DX	(Reserved)		(Reserved)	
2EX	(Reserved)		(Reserved)	
2FX	(Reserved)		(Reserved)	
30X	Mux Subch 32 HCAW	Mux Subch 33 HCAW	Mux Subch 34 HCAW	Mux Subch 35 MCAW
31X	Mux Subch 36 HCAW	Mux Subch 37 HCAW	Mux Subch 38 HCAW	Mux Subch 39 HCAW
32X	Mux Subch 40 HCAW	Mux Subch 41 HCAW	Mux Subch 43 HCAW	Mux Subch 43 HCAW
33X	Mux Subch 44 HCAW	Mux Subch 45 HCAW	Mux Subch 46 HCAW	Mux Subch 47 HCAW
34X	Mux Subch 48 HCAW	Mux Subch 49 HCAW	Mux Subch 50 HCAW	Mux Subch 51 HCAW
35X	Mux Subch 52 HCAW	Mux Subch 53 HCAW	Mux Subch 54 HCAW	Mux Subch 55 HCAW
36X	Mux Subch 56 HCAW	Mux Subch 57 HCAW	Mux Subch 58 HCAW	Mux Subch 59 HCAW
37X	Mux Subch 60 HCAW	Mux Subch 61 HCAW	Mux Subch 62 HCAW	Mux Subch 63 HCAW
38X	(Reserved)		(Reserved)	
39X	(Reserved)		(Reserved)	
3AX	(Reserved)		(Reserved)	
3BX	(Reserved)		(Reserved)	
3CX	(Reserved)		(Reserved)	
3DX	(Reserved)		(Reserved)	
3EX	(Reserved)		(Reserved)	
3FX	(Reserved)		(Reserved)	

Figure 4–1. Fixed Main Storage Assignments (Part 2 of 2)

4.4. PARTIAL WRITE

The partial write capability of main storage enables the processor to write into main storage on a byte basis. Four control lines to main storage are used to identify the bytes of the word written into on a write cycle. The processor activates the proper control line or lines at the beginning of a storage write cycle. The unselected byte or bytes are unchanged in cases where less than a full word of four bytes is written.

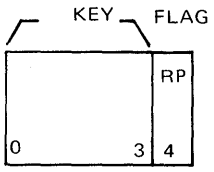
4.5. STORAGE PROTECTION

Storage protection is provided to ensure the integrity of the supervisor and the individual worker programs located in main storage. Up to 16 programs, including the supervisor, may be protected against adverse interaction. The storage protection capability protects the contents of main storage from unwanted destruction or misuse in both single-program and multiprogramming environments.

Two types of protection are provided:write only or read/write in blocks of 2048 bytes. Access by both the processor and the I/O channels are subject to protection checking.

4.5.1. Storage Key

Storage protection is accomplished by dividing main storage into blocks of 2048 bytes and by associating a 5-bit storage key with each block. Each storage key is stored in a location in a set of hardware registers in the processor. The capacity of main storage determines the number of locations required in key storage in a given configuration. Sixty-four 5-bit storage keys are required for a minimum main storage capacity of 131,072 bytes; 512 five-bit storage keys are required for a maximum main storage capacity of 1,048,576 bytes. The storage key has the following format:



Field	Definition
Key (bits 0-3)	Four-bit code assigning the associated 2048-byte block of main storage to one of 16 programs
RP (bit 4)	Read protect flag. RP = 0 Specifies write protection. RP = 1 Specifies read and write protection.

The supervisor software establishes the storage key for each 2048-byte block. This is accomplished by executing a set-storage-key (SSK) instruction which loads a storage key into a key memory loction. The insert-storage-key (ISK) instruction is used for reading out the contents (key) or a location in the key memory. These two instructions are privileged and, therefore, must be executed when the processor is operating in the supervisor mode. There may be more than one location in the key memory which contains the same storage key corresponding to several blocks of main storage being assigned to the same program.

4.5.2. Storage Protection and Relocation Key

Storage protection also utilizes a 4-bit protection key which identifies the program or operation that is requesting access to main storage. Storage references initiated by the processor use the key fields (bits 20 through 23) in the current program status word (PSW); accesses by I/O channels are controlled by the protection key assigned to the associated I/O operation.

In addition to its function in storage protection, this key is also used for accessing the appropriate relocation register in low order main storage as part of the address relocation capability. This protection key, therefore, is called the current PSW (or I/O) storage protection and relocation key. The source and implementation of the I/O storage protection and relocation key is specified in the individual description for each selector and multiplexer channel.

4.5.3. Key Comparison

When access to main storage is initiated by the processor or I/O, the 13 most significant bits of the 24-bit storage byte address selects a storage key which corresponds to the addressed block of main storage. The key portion (four high order bits) of the storage key location is compared for equality with the 4-bit current PSW or I/O storage protection and relocation key. If the result is either that the two 4-bit keys are equal or that the PSW or I/O key is binary 0, a match condition exists and access to main storage is granted. In the case of processor storage accesses only a match condition also exists irrespective of the current PSW storage and relocation key if the storage escape mode is specified in the current PSW. If a match condition does not exist, but the RP flag (bit 4) in the storage key is 0, the access is granted if the reference is a main storage read.

An exception to the rules for a match condition exists in indirect addressing. The value F_{16} in the key portion (bits 0-3) of a storage key has a special meaning when the main storage reference is to read or write an indirect address control word (IACW). This special facility updates an IACW located in a main storage block which is not assigned to the program accessing it. Where the storage reference is initiated by the processor to access an IACW, read or write is permitted unconditionally if the key portion (bits 0-3) of the storage key equals F_{16} or if a normal match condition exists. The value of the RP flag (bit 4) in the storage key and the value of the current PSW key are ignored and the main storage access is granted.

Whenever the main storage access is not granted because a match condition does not exist or the special IACW condition does not exist, a protection exception occurs. If the reference is for a write operation, the processor signals main storage to abort the write cycle and the original data is preserved. If the reference is for a read operation, main storage executes the read cycle; however, the processor blocks the read data from being used either in the processor or in the I/O channels. When a protection exception occurs on a program-specified reference, a program exception interrupt request is generated. The action taken by an I/O channel when a protection exception occurs on an access initiated by I/O is described in the individual selector and multiplexer channel descriptions.

4.5.4. Summary of Storage Protection Rules

Writing is permitted if any of the following is true:

- The storage protection and relocation key equals the key portion of the storage key.
- The storage protection and relocation key equals 0.
- The key portion of the storage key equals F_{16}, and the reference is for an IACW during the execution of an indirect addressing sequence.
- The access is by the processor and the storage protect escape mode is set to 1.
- The processor is operating in program trace mode; the access is by the processor for a write into the trace table and the machine state simulation mode (MSSM) is set.

Reading is permitted if any of the following is true:

- The storage protection and relocation key equals the key portion of the storage key.
- The storage protection and relocation key equals 0.
- The RP flag in the storage key equals 0.
- The key portion of the storage key equals F_{16}, and the reference is for an IACW during the execution of an indirect addressing sequence.
- The access is by the processor and the store protect escape mode is set.

4.6. ADDRESSING

Even though the maximum main storage capacity is 1,048,576 bytes, the addressing hardware of the processor complex accomodates a 24-bit binary number, which provides an addressing capability of 16,777,216 bytes. The low order 20 bits are used to allow access to 1,048,576 bytes. Byte locations are numbered consecutively, beginning with 0; each number is considered to be the address of the byte stored at that location. The address field in the processor addresses an individual byte; however, the two least significant address bits are not sent from the processor to main storage. Therefore, a 4-byte group (one word) in main storage is referenced by the address of the most significant byte location. The full address capability is provided in the processor/main storage interface. The full address capability is 24 bits less the two least significant bits. (See partial write, 4.4.)

The 22-bit word address is sent to storage along with four control lines. A signal on a specific control line indicates that the corresponding byte within a specified word location in main storage is to be written. Thus, a signal on all four control lines specifies that a full 32-bit word (width of main storage) is written. The absence of a signal on all four control lines signifies a read operation is to occur from the specified storage word location. Three parity bits accompany the 22 address bits and four control lines to storage. One parity bit is associated with the low order nine address bits; the second parity bit is associated with the next nine address bits; and the third parity bit is associated with the high order four address bits and the four control lines.

On a read cycle, the main storage presents four bytes to the processor. If the particular reference requires byte addressability, the processor or I/O channel chooses the appropriate byte based upon the two least significant bits of the address field. Similarly, on a write cycle, four bytes are written unless the partial write capability of main storage is employed.

The processor hardware provides for main storage addressing to wrap around from the maximum byte address of 16,777,215 to address 0. Note that the maximum byte address which can access main storage is 1,048,575. An addressing exception error is detected and indicated by storage whenever the address exceeds the capacity of main storage in the configuration.

Storage addresses in instructions are recorded in base-displacement format symbolized as B1-D1 or B2-D2. The base (B1 or B2) portion indicates the number of a general register which contains a binary-base beginning address; and the displacement (D1 or D2) portion contains a 12-bit additional binary address value which can vary from 0 to 4095. Instructions of the RX type can also specify another general register which contains another binary address value: the index value. All of these values are determined by the language processor used to write the program and adjusted by linking procedures to achieve proper program relative addresses ranging from 0 to the maximum address required by the program. The relative address is determined by adding:

Displacement value + Base (register) value	RS, SI, SS, and RX types
+ Index value	RX type only
Relative address value	

4.6.1. Address Relocation

Address relocation provides a means for repositioning a program in main storage without program aid or knowledge. The supervisor can place the rolled-in program or data in any portion of main storage on 4096-byte boundaries by adjusting an associated relocation value. Address relocation is supported by three processor instructions:

- Branch and link external (BALE)
- Branch on condition to return external (BCRE)
- Load base and relativize (LBR)

Modification of the program-specified address is by an indexing scheme and address relocation.

The indexing scheme modifies the program-specified operand and branch addresses in RX, RS, SI, and SS type instructions. The address is formed by the addition of the operand address displacement value, the contents of the D field in the instruction, and the base (B), which is the content of the register specified by the B field in the instruction. Addresses specified in RX type instructions can be modified further by the addition of the contents of the index register (X_2).

The address relocation also can be effective on these program-specified operand and branch addresses, and, in addition, it can modify the addresses of instructions and data including data and channel command words specified for I/O.

4.6.1.1. ABSOLUTE AND RELATIVE ADDRESSES

Main storage addresses are specified by the program and the hardware. The value of the address when it references main storage is the number of the absolute location in main storage being accessed. Hardware-generated addresses are always in this form. The value of program-generated addresses, as specified in the program, may be different from the value of the corresponding address used to reference main storage. For example, program-specified addresses of intructions and data in object code may appear relative to a starting address of the program. This starting address and all addresses relative to it may or may not specify the same values as the values of the absolute locations where the program or data is placed in main storage when it is loaded; that is, the address may be relocated. There are also special cases where a program address must always specify the absolute location in main storage; for example, addresses specified for use by the timer subchannel in the multiplexer channel.

- Relative Address

  An address capable of being relocated as it appears prior to being modified by address relocation.

- Absolute Address

  – An address not capable of being relocated. Such addresses are hardware-generated addresses, address of a program trace table, and addresses used by the status table and timer subchannels in the multiplexer channel.

- A relative address that has been relocated. This includes cases when a relocation value of 0 is added to a relative address resulting in the absolute address having the same value as the relative address.

- A relative address that has not been relocated because the appropriate R_n bit is 0 in the flag field of the relocation register or the indirect address control word (IACW). This only applies to relative addresses which are sensitive to control by an R_n flag bit. In this case, the absolute address has the same value as the relative address.

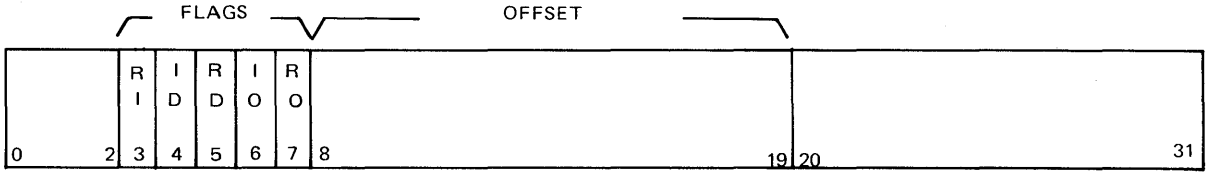
4.6.1.2. CHARACTERISTICS OF ADDRESS RELOCATION

Address relocation is accomplished by adding a 12-bit relocation value (OFFSET) to bits 8 through 19 (most significant bits) of the main storage address which would normally have been used to access main storage if relocation had not been used. The 12-bit offset value provides a relocation base which is on a 4096-byte boundary.

Relocation is controlled by a set of 16 relocation registers that are in low order main storage locations 140_{16} to $17F_{16}$. The registers are numbered 0_{16} through F_{16}, each associated with its corresponding storage protection and relocation key. These registers are used by both the processor and the I/O channels. The storage protection and relocation key (processor or I/O) is used to select the appropriate register. There are no special instructions to store the relocation registers in low order main storage. The processor also has a hard copy, called the current relocation register, of the relocation register that is actively in use by the processor. The current relocation register is loaded in the execution of a load-program-status-word (LPSW) instruction and in an interrupt initialization sequence. The current relocation register can also be modified by executing a BALE, BCRE, or LBR instruction. When this occurs, the corresponding relocation register in low order main storage is also updated with the new values. When the processor is in the clear state, the contents of the current relocation register is 0.

4.6.1.3. RELOCATION REGISTER FORMAT

Each relocation register in main storage and the current relocation register in the processor have the following format:



Field	Definition
Bits 0-2	These bits are set to 0 by the hardware when the flags field is written into main storage after being modified by either the BALE, BCRE, or LBR instruction. These bits are not provided in the current relocation register.
Flags (bits 3-7)	The flags field is ignored by I/O channels.
RI (bit 3)	Relative instruction fetch including all branch addresses. RI = 0 Specifies that instruction addresses and branch addresses are absolute. RI = 1 Specifies that instruction addresses and branch addresses are relative.

Field	Definition
ID (bit 4)	Indirect destination operand control. ID = 0 Specifies that destination operand addresses in instructions are direct addresses. ID = 1 Specifies that destination operand addresses in instructions are IACW addresses.
RD (bit 5)	Relative destination operand control. RD is evaluated before ID (bit 4). RD = 0 Specifies that destination operand addresses in instructions are absolute. RD = 1 Specifies that destination operand addresses in instructions are relative.
IO (bit 6)	Indirect origin operand control. IO = 0 Specifies that origin operand addresses in instructions are direct addresses. IO = 1 Specifies that origin operand addresses in instructions are IACW addresses.
RO (bit 7)	Relative origin operand control. RO is evaluated before IO (bit 6). RO = 0 Specifies that origin operand addresses in instructions are absolute. RO = 1 Specifies that origin operand addresses in instructions are relative.
OFFSET (bits 8-19)	A 12-bit relocation value. This value is added to the 12 high order bits of all addresses that are to be relocated to form the absolute main storage address, subject to the control of the RI, RD, and RO flag bits. If the software sets the value of the offset to 0, the relative address equals the absolute address. NOTE: The address in an IACW is subject to the control of the R (bit 6) of the IACW, and is not controlled by any of the R_n flag bits in the current relocation register or corresponding relocation register in main storage. In input/output main storage references, address relocation is not subject to the control of the flags field.
Bits 20-31	These bits in the relocation registers are ignored, not altered by the hardware, and not provided in the current relocation register.

4.6.1.4. Loading Current Relocation Register

The current relocation register is loaded from main storage with the contents of a relocation register whenever there is a new processor storage protection and relocation key placed in the current PSW. This occurs when the current PSW is changed in the execution of the load-program-status-word (LPSW) instruction and in an interrupt initialization sequence (IIS). The new key is used as a pointer to access the appropriate relocation register in low order main storage. The contents of this relocation register is loaded in the current relocation register in the processor.

The flags field in the current relocation register can be changed by the execution of a BALE, BCRE, or LBR instruction. The updated flags field is written into the appropriate relocation register in main storage during the execution of these instruction. The BCRE instruction can modify the entire flags field (bits 3 through 7), while the BALE instruction can modify only the RI flag (bit 3), and the LBR instruction places a fixed pattern in the flags field (RI, RD, and RO = 1).

4.6.1.5. INSTRUCTION ADDRESS RELOCATION

Prior to loading the current PSW and current relocation register, the instruction address in the current PSW is converted from relative to absolute. The relocation offset is added, if required, depending upon the value of the RI flag (bit 3) in the current relocation register. (If RI is set to 1, the addition is performed). Processing of instructions is initiated and continues with the instruction address in the current PSW maintained in absolute.

The instruction address in the current PSW can be changed by the following conditions:

- Execution of a branch instruction when branching occurs. The branch address becomes the new instruction address in the current PSW. When loading the branch address into the instruction address field in the current PSW, it is converted from a relative to an absolute address. The relocation offset is added if required, depending upon the value of the RI flag (bit 3) in the current relocation register. In the case of the BCRE instruction, the value of RI is determined after the new flags field is loaded in the current relocation register.

  Although the branch address is either in or specified by the operand address field in the instruction, it is subject to control by the RI flag and not by either the RO or RD flag normally used for relocation of operand addresses. Except for the BALE instruction, branch addresses are always direct. The address in the BALE instruction (relocated subject to control by the RI flag as is the case for all branch instructions) is not the branch address, but is always the address of the IACW. At the last level of indirection, the actual branch is determined. This address is relocated subject to control by R (bit 6) in the last IACW. The BALE also specifies that the value of the RI flag in the current relocation register is replaced by the value of R (bit 6) in the last IACW.

  The instruction address that is being replaced in the current PSW in the execution of a branch and link instruction (BAL, BALR, BALE) is stored as the link information in relative address form; that is, before storing the instruction address, the offset in the current relocation register is subtracted from the absolute instruction address, if required, depending upon the value of the RI flag in the current relocation register (if RI is set to 1, the subtraction is performed). In the case of BALE, the value of RI is determined prior to RI being set to the value of R (bit 6) in the last IACW.

- Execution of an LPSW instruction or execution of an IIS. The instruction address in the current PSW that is being replaced is stored in the old PSW as a relative address; that is, the offset in the current relocation register is subtracted from the absolute instruction address, if required, depending upon the value of the RI flag in the current relocation register. The values in the current relocation register mentioned here are determined prior to the loading of a new relocation register, which also occurs during the execution of an IIS.

- Execution of a BCRE instruction when branching does not occur and the value of the RI flag is changed from 0 (absolute address) to 1 (relative address). The new RI flag equal to 1 causes the updated instruction address in the current PSW to be converted by adding the relocation offset value of it.

  No special action occurs when the RI flag is not changed. If in the execution of a BCRE instruction a new flags field containing RI, RD, RO set to 0 is detected when the current RI flag is set to 1, a specification exception occurs and a program exception interrupt request is generated. Detection of the specification exception inhibits loading the current relocation register with the new flag field. The corresponding relocation register in main storage is not updated.

- Cleared state of the current relocation register. When the processor is in the cleared state, the contents of the current relocation register are 0. The contents of the current PSW and, therefore, the processor storage protection and relocation key are also set to 0 when the processor is in the cleared state. When the load switch on either the system maintenance panel or the system console is pressed, 0's are written into relocation register 0 (main storage location 140_{16}) prior to the initiation of the load operation. The I/O operation accesses the relocation register 0 during the load operation since the I/O storage protection and relocation key specified in the channel address word (CAW) in main storage location 180_{16} is set to 0 when the load operation is initialized by the hardware.

Note that if processing is initiated from the cleared state without initial loading, the contents of relocation register 0 may differ from the contents of the current relocation register which are 0.

4.6.1.6. OPERAND ADDRESS RELOCATION

Operand addresses in instructions are converted from relative to absolute, if required, depending on the value of RD or RO (bits 5 and 7) in the current relocation register and the particular instruction being executed. Address relocation does not apply to all instruction operand address fields; for example, in the load-address, shift, and I/O instructions, the operand address field is never converted from relative to absolute.

The operand address in an instruction may be the address of an IACW and not the address of the operand itself.

In such cases, the address is converted from relative to absolute, if required, depending upon the value of RD or RO in the current relocation register.

The actual operand address is determined by the address in the last IACW (last level of indirection), which is relocated subject to control by R (bit 6) in the last IACW. Intermediate IACW's, if accessed, specify an address of another IACW in multilevel indirect addressing. The address of each of these IACW's is relocated subject to control by R (bit 6) in that IACW.

The relocation offset is added to an instruction operand address at the same time the D+(B) modification takes place. In Rx type instructions, the contents of the indexing register (X_2) is added, if required, following the D+(B) + offset addition unless indirect addressing is specified. If indirect addressing is specified, the contents of the indexing register (X_2) is added, if required, to the address in the last IACW(also subject to modification by the addition of the offset value determined by the value of R (bit 6) in the IACW). The result forms the actual operand address.

4.6.1.7. INPUT/OUTPUT ADDRESS RELOCATION

The implementation of address relocation in I/O channels is channel dependent. Addresses are always specified as relative addresses for the multiplexer and selector channels with the following exceptions:

- All addresses to be used by the timer subchannel in the multiplexer channel must be specified as absolute.
- All addresses to be used by the status table subchannel in the multiplexer channel must be specified as absolute.
- All hardware-generated addresses are absolute; for example, the addresses used to access the CAW and the channel status word (CSW).

The relocation offset, which is added to the relative address to convert it to absolute, is read from a relocation register in low order main storage. The I/O storage protection and relocation key specified in the CAW is used as the pointer to access the appropriate relocation register. The flags field in the relocation register is ignored. Under special conditions, the multiplexer or selector channel can access data and channel command words under control of a 0 value key instead of the key specified in the CAW. In these cases, relocation register 0 (corresponding to the 0 key) is accessed to obtain the relocation offset that is added to the relative addresses.

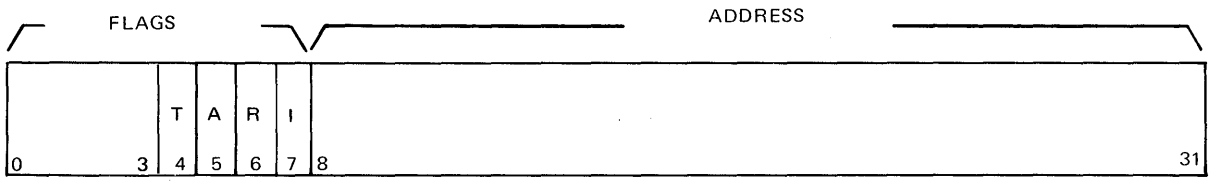
4.6.2. Indirect Addressing

Indirect addressing assists in providing uncomplicated entry to common or reentrant subroutines that are under control of the supervisor. It also allows different programs to reference shared data or a single program to reference data which is not always stored in the same relative address.

Indirect addressing is specified to the processor in either of two ways: explicitly, by the setting of the ID and IO flags in the current relocation register; and implicitly, as the operand address of the BALE instruction. Indirect addressing has no meaning to I/O channels.

4.6.2.1. INDIRECT ADDRESS CONTROL WORD

The indirect address control word (IACW) is used to implement indirect addressing. An IACW can reside anywhere in main storage on a word boundary subject to the rules of storage protection. The format and description of the IACW are as follows:



Field	Definition
Flags (bits 0-7)	This field is examined by the processor after the IACW is fetched from main storage.
Bits 0-3	These bits are ignored by the hardware.
T (bit 4)	Trail control bit. If A (bit 5) or I (bit 7) in the IACW is equal to 1, T is ignored. If A (bit 5) and I (bit 7) in the IACW are both equal to 0, T is examined and the following occurs: ■ If T is 0, T remains unchanged. ■ If T is 1, T is reset to 0 and the resultant updated flags field is written back into the IACW in main storage. The write of the IACW is subject to control by the rules of storage protection.
A (bit 5)	Absence control bit. A = 0 Specifies that the T, R, and I flag bits and the address field in the IACW are to be interpreted by the processor. A = 1 Specifies that the T, R, and I flag bits and the address field in the IACW are not to be interpreted by the processor; instead, an indirect addressing exception occurs and a program exception interrupt request is generated. In addition, the absolute main storage address of the IACW is stored in bit positions 8–31 of the word at location $18C_{16}$ in low order main storage. Bit positions 0–7 (byte locations $18C_{16}$ of this word in main storage are not used by the hardware.
R (bit 6)	Relative address control bit. Ignored if A (bit 5) is equal to 1. R = 0 Specifies that the address in the address field of the IACW is absolute. R = 1 Specifies that the address in the address field of the IACW is relative and must be converted to absolute.
I (bit 7)	Indirect control bit. Ignored if A (bit 5) is equal to 1. I = 0 Specifies that the address in the address field, subject to control by R (bit 6), is the object of reference (operand address or in the case of the BALE instruction, the branch address; both subject to modification under control of R, bit 6, in the IACW). In RX type instructions, this operand address (branch address in the case of a BALE instruction) is subject to further modification by the addition of the contents of the index register (X_2). I = 1 Specifies that the address in the address field, subject to control by R, is the address of another IACW.

Field	Definition
Address (bits 8-31)	Address of operand (branch address in the case of a BALE instruction) or another IACW. The address field is subject to control by the R and I flags (bits 6 and 7). It is ignored if A (bit 5) is equal to 1. When the address field is the operand address (branch address in the case of a BALE instruction) of an RX type instruction, the address is subject to further modification by the addition of the contents of the index register (X_2).

4.6.2.2. STORAGE PROTECTION – IACW REFERENCES

If the main storage reference is for a read of an IACW, the access is allowed if any of the following is true:

- The key portion of the storage key equals F_{16}.
- Any of the following normal storage protection rules pertain:
  - The relocation key in PSW equals 0.
  - The storage protection and relocation key equals the key portion of the storage key.
  - The RP flag in the storage key equals 0.
  - The storage protect escape mode is set in the current PSW.

If the main storage reference is for a write of an IACW (update of the flags when T (bit 0) is changed from a 1 to 0), the access is allowed if any of the follwoing is true:

- The key portion of the storage key equals F_{16}.
- Any of the following normal storage protection rules pertains:
  - The storage protection and relocation key equals 0.
  - The storage protection and relocation key equals the key portion of the storage key.
  - The storage protect escape mode is set in the current PSW.

4.6.2.3. INDIRECT ADDRESSING OPERATION

The operand address in an instruction may be the address of an IACW and not the address of the operand itself. The first level of indirect addressing is determined by the value of ID or IO (bits 4 and 6) in the current relocation register and the particular instruction being executed. Indirect addressing does not apply to all instruction operand address fields; for example, in the load address, shift and I/O instruction, the operand address field is never the address of an IACW. In the BALE instruction, the operand address is always the address of an IACW. The branching address in a branch instruction (other than BALE) is never the address of an IACW. In cases where indirect addressing does apply, the description specifies the flag bit, IO (origin) or ID (destination) that controls whether the operand address is the address of an IACW or the address of the operand itself.

When indirect addressing is specified, the operand address in the instruction, formed by D+(B)+ relocation offset (if required), is used to reference an IACW. The IACW is examined by the processor and the appropriate action is taken according to the setting of the flag bits. If the flags field of the IACW (A, bit 5, equals 0; and I, bit 7, equals 1) indicates that the address field in the IACW is the address of another IACW, the new IACW is accessed and examined by the processor. This condition is called multilevel indirect addressing. Multilevel indirect addressing is permitted up to eight levels. If the eighth IACW accessed in multilevel indirect addressing specifies that another IACW is to be accessed, an indirect address specification exception occurs and a program exception interrupt request is generated.

The RD and RO flags (bits 5 and 7) in the current relocation register control the determination of relative or absolute-specified operand addresses in the first level of indirect addressing, except for the BALE instruction, which uses the RI flag (bit 3) for this purpose. Levels 2 through 8 in multilevel indirect addressing use R (bit 6) in the respective IACW to determine if the address in the address field of the IACW is specified as relative or absolute.

When the object of the reference is reached in the last level of indirection (A, bit 5, and I, bit 7, in the IACW both set to 0), indirect addressing ends the address in the address field in the operand address; or in the case of a BALE instruction, the value of the RI flag (bit 3) in the current relocation register is set to the value of R (bit 6) in the IACW at the last level of indirection. The special meaning for F_{16} in the key portion of a storage key does not apply to the operand (nor BALE branching address) once indirect addressing ends. In RX type instructions, when indirect addressing ends, the contents of the index register (X_2) is added, if required, to the operand address; or in the case of a BALE instruction, it is added to the branching address. In the BALE instruction, the operand address is always the address of an IACW. The branching address in a branch instruction (other than BALE) is never the address of an IACW. In cases where indirect addressing does apply, the description specifies the flag bit, IO (origin) or ID (destination) that controls whether the operand address is the address of an IACW or the address of the operand itself.

4.7. PRIORITY ASSIGNMENTS

Main storage and the main storage interface is shared by the processor and I/O. The priority for the access to main storage is specified in 3.1.4.

4.8. MAIN STORAGE ERRORS

The following error conditions are detected and indicated to the processor and I/O channels whenever main storage is used:

- Address check
- Addressing exception
- Storage parity check
- Storage hold check
- Protection exception

4.8.1. Address Check

An address check is indicated if either an address bus check or a storage cabinet select exception is indicated.

- Address Bus Check

  If a parity error is detected in the address sent to it, the main storage cabinet so indicates on the address parity check line. The error indicated by a signal on this line is an address check. The detection of this error by main storage suppresses any specified writing of information.

- Storage Cabinet Select Exception

  If two or more storage cabinets signal simultaneously on the address-accepted line, a storage-cabinet-select-exception error exists. This error also suppresses any specified writing of information.

4.8.2. Addressing Exceptions

Each main storage cabinet examines all addresses sent to it. If the cabinet recognizes an address as being contained and installed in its storage and the cabinet is online, the cabinet so indicates by signaling on the address-accepted line. If no storage cabinet indicates the acceptance of an address within a certain time following presentation of that address, an addressing exception error condition exists.

4.8.3. Storage Parity Checks

A storage parity check is indicated if any of the following conditions is detected:

- Storage Read Check

  If a parity error is detected by main storage during a read operation, a storage read check signal is placed on the storage parity check line.

- Read Bus Check

  The word transmitted from main storage is accompanied by four parity bits, one for each of the four bytes read. The unit (processor, I/O cabinet) receiving this information performs a parity check. If an error is detected, it is a read bus check error.

- Write Bus Check

  Parity bits, one per byte, are attached to information sent to main storage to be written. The storage cabinet accepting the address of this information performs a parity check on the data to be written. If a parity error is detected, the storage cabinet so indicates by signaling on the storage parity check line. The error indicated is a write bus check error. The data, including the bad parity, is written into main storage.

4.8.4. Storage Hold Check

A maintenance switch located on the storage cabinet sets the storage hold mode. If a parity error is detected by the storage cabinet while in this mode, operation of the cabinet is halted to preserve the contents of all of its internal registers. The cabinet is thus rendered useless to the system. The condition generates a nonrecoverable error in the processor. This condition is cleared manually by operating the STORAGE CLEAR switch.

4.8.5. Protection Exception

A storage protection capability is included to prevent unauthorized writing or reading in various sections of main storage. If such an operation is detected, the error is a protection exception error.

5. SYSTEM CONSOLE

5.1. GENERAL

The UNIVAC 9700 System Console (system console) is a freestanding modular operator console which provides central operator control of the UNIVAC 9700 System, necessary monitoring functions, and control through the use of the operational panel and a hard copy for permanent records of console activity.

Implementation of these functions consists of a keyboard/display and control panel as part of the system console. Add-in modular cabinets provide expansion features such as the incremental printer for hard copy printout, and the multi-channel switch (MCS) control panel that provides operator controls and indicators to perform subsystem switching functions.

The operator console portion of the system console, shown in Figure 5–1, does not include the printer, which is discussed in Section 6. The UNISCOPE 100 Display Terminal is to the left of the drawer assembly. The display unit is mounted so that it can be rotated 180° on a turntable-type base, and the drawer assembly can be physically removed and turned around, enabling the display unit to appear to the right of the drawer assembly. Therefore, the front of the console is defined by the equipment located in the bottom portion of the console cabinet. The side showing the circuit breaker switches, the plug-in boards, and test panel is the rear or back of the console, and the side showing the terminal boards (backboard wiring side) is the front of the console.

Figure 5–1. UNIVAC 9700 Operator Console

5.2. SUBSYSTEM COMPONENTS

The basic components comprising the operator console are discussed in the following paragraphs.

5.2.1. Keyboard/Display

The keyboard/display (type 3536-04/05) is a modified UNISCOPE 100 Display Terminal located on a mechanical enclosure compatible to the system. The UNISCOPE 100 is configured with the upper and lowercase alphabetic typewriter keyboard, a dual-case alphanumeric character set, and a 16-line by 64-character-per-line protected screen format.

Protected format protects selected data from operator alteration. The processor may prevent the operator from modifying the format or information on any portion of the screen and leave certain unprotected areas for operator input. If the operator attempts to position the cursor in a protected area, it automatically moves until an unprotected area is reached. If all data positions are protected, the cursor stops at the home position. Protected areas are defined as all data sent to the UNISCOPE 100 between shift out (SO) and shift in (SI) codes (Figure 5–2).

5.2.1.1. DISPLAY AND CONTROL INDICATOR PANEL

Located directly below the display screen is a series of controls and indicators indicating the status of both the message and display along with controls required by the operator.

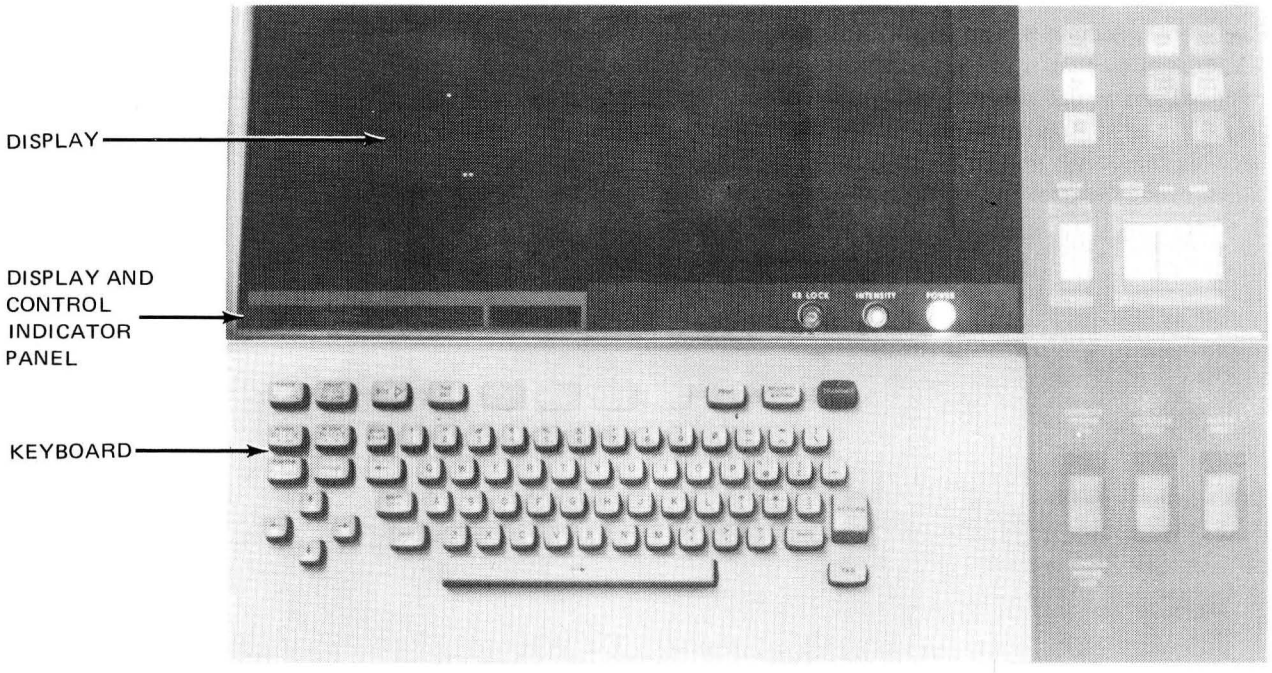


Figure 5–2. UNIVAC 9700 Operator Console Keyboard and Display

5.2.2. System Operator Panel

The system operator panel consists of switches, indicators, and switch indicators that provide direct connection to the maintenance panel of the UNIVAC 9700 System Processor (processor). See Figure 5–3.

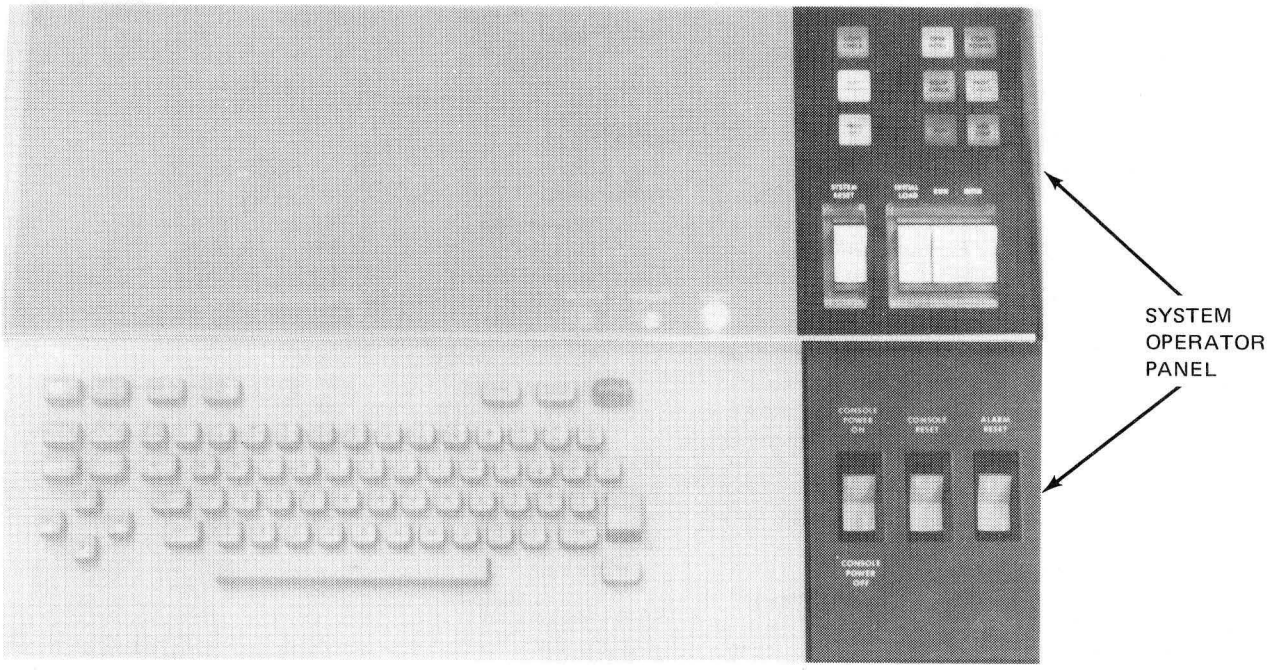


Figure 5–3. UNIVAC 9700 Operator Console System Operator Panel

5.2.3. System Console Control Unit

The system console control unit provides the logical interface between the processor and the keyboard/display (and/or printer) by way of one subchannel of the processor multiplexer channel. The control unit provides the control logic, data path, and code translation and interface logic required by the incremental printer. Code translation between the EBCDIC and ASCII codes is provided in the control unit so that data to and from the processor is transmitted only in EBCDIC code.

5.2.3.1. CONTROL UNIT TRANSLATION CODES

Keyboard translation code from EBCDIC to ASCII is given in Table 5–1; translation from ASCII to EBCDIC is given in Table 5–2.

EBCDIC					ASCII				
Hex Code	8-Bit Code 0123	8-Bit Code 4567	Graphic or Control	Name of Graphic or Control	Name of Graphic or Control	Graphic or Control	8-Bit Code 7654	8-Bit Code 3210	Hex Code
00	0000	0000	NUL	Null	Null	NUL	0000	0000	00
01	0000	0001	SOH	Start of Heading	Start of Heading	SOH	0000	0001	01
02	0000	0010	STX	Start of Text	Start of Text	STX	0000	0010	02
03	0000	0011	ETX	End of Text	End of Text	ETX	0000	0011	03
04	0000	0100		NOT	USED				
05	0000	0110	HT	Horizontal Tab	Horizontal Tab	HT	0000	1001	09
06	0000	0110		NOT	USED				
07	0000	0111	DEL	Delete	Delete	DEL	0111	1111	7F
08	0000	1000		NOT	USED				
09	0000	1001		NOT	USED				
0A	0000	1010		NOT	USED				
0B	0000	1011	VT	Vertical Tab	Vertical Tab	VT	0000	1011	0B
0C	0000	1100	FF	Form Feed	Form Feed	FF	0000	1100	0C
0D	0000	1101	CR	Carriage Return	Carriage Return	CR	0000	1101	0D
0E	0000	1110	SO	Shift Out	Shift Out	SO	0000	1110	0E
0F	0000	1111	SI	Shift In	Shift In	SI	0000	1111	0F
10	0001	0000	DLE	Data Link Escape	Data Link Escape	DLE	0001	0000	10
11	0001	0001	DC1	Device Control 1	Device Control 1	DC1	0001	0001	11
12	0001	0010	DC2	Device Control 2	Device Control 2	DC2	0001	0010	12
13	0001	0011	DC3	Device Control 3	Device Control 3	DC3	0001	0011	13
14	0001	0100		NOT	USED				
15	0001	0101		NOT	USED				
16	0001	0110	BS	Backspacer	Backspacer	BS	0000	1000	08
17	0001	0111		NOT	USED				
18	0001	1000	CAN	Cancel	Cancel	CAN	0001	1000	18
19	0001	1001	EM	End of Medium	End of Medium	EM	0001	1001	19
1A	0001	1010		NOT	USED				
1B	0001	1011		NOT	USED				
1C	0001	1100	FS	Field Separator	Field Separator	FS	0001	1100	1C
1D	0001	1101	GS	Group Separator	Group Separator	GS	0001	1101	1D
1E	0001	1110	RS (SOE)	Reader Stop (Start of Entry)	Reader Stop (Start of Entry)	RS (SOE)	0001	1110	1E
1F	0001	1111	US	Unit Separator	Unit Separator	US	0001	1111	1F
20	0010	0000		NOT	USED				
21	0010	0001		NOT	USED				
22	0010	0010		NOT	USED				
23	0010	0011		NOT	USED				
24	0010	0100		NOT	USED				
25	0010	0101	LF	Line Feed	Line Feed	LF	0000	1010	0A
26	0010	0110	ETB	End of Trans. Block	End of Trans. Block	ETB	0001	0111	17
27	0010	0111	ESC	Escape	Escape	ESC	0001	1011	1B
28	0010	1000		NOT	USED				
29	0010	1001		NOT	USED				
2A	0010	1010		NOT	USED				
2B	0010	1011		NOT	USED				
2C	0010	1100		NOT	USED				
2D	0010	1101	ENQ	Enquiry	Enquiry	ENQ	0000	0101	05
2E	0010	1110	ACK	Acknowledge	Acknowledge	ACK	0000	0110	06
2F	0010	1111	BEL	Bell	Bell	BEL	0000	0111	07
30	0011	0000		NOT	USED				
31	0011	0001		NOT	USED				
32	0011	0010	SYN	Synchronous Idle	Snychronous Idle	SYN	0001	0110	16
33	0011	0011		NOT	USED				
34	0011	0100		NOT	USED				
35	0011	0101		NOT	USED				
36	0011	0110		NOT	USED				
37	0011	0111	EOT	End of Transmission	End of Transmission	EOT	0000	0100	04
38	0011	1000		NOT	USED				
39	0011	1001		NOT	USED				

Table 5–1. Control Unit Translation Codes, EBCDIC to ASCII (Part 1 of 5)

EBCDIC					ASCII				
Hex Code	8-Bit Code 0123	8-Bit Code 4567	Graphic or Control	Name of Graphic or Control	Name of Graphic or Control	Graphic or Control	8-Bit Code 7654	8-Bit Code 3210	Hex Code
3A	0011	1010		NOT	USED				
3B	0011	1011		NOT	USED				
3C	0011	1100	DC4	Device Control 4	Device Control 4	DC4	0001	0100	14
3D	0011	1101	NAK	Negative Acknowledge	Negative Acknowledge	NAK	0001	0101	15
3E	0011	1110		NOT	USED				
3F	0011	1111	SUB	Substitute	Substitute	SUB	0001	1010	1A
40	0100	0000	SP	Space	Space	SP	0010	0000	20
41	0100	0001		NOT	USED				
42	0100	0010		NOT	USED				
43	0100	0011		NOT	USED				
44	0100	0100		NOT	USED				
45	0100	0101		NOT	USED				
46	0100	0110		NOT	USED				
47	0100	0111		NOT	USED				
48	0100	1000		NOT	USED				
49	0100	1001		NOT	USED				
4A	0100	1010	[	Left Bracket	Left Bracket	[	0101	1011	5B
4B	0100	1011	.	Period (Decimal Point)	Period (Decimal Point)	.	0010	1110	2E
4C	0100	1100	<	Less Than	Less Than	<	0011	1100	3C
4D	0100	1101	(	Left Parenthesis	Left Parenthesis	(	0010	1000	28
4E	0100	1110	+	Plus Sign	Plus Sign	+	0010	1011	2B
4F	0100	1111	|	Logical OR	Exclaimation Point	!	0010	0001	21
50	0101	0001	&	Ampersand	Ampersand	&	0010	0110	26
51	0101	0000		NOT	USED				
52	0101	0010		NOT	USED				
53	0101	0011		NOT	USED				
54	0101	0100		NOT	USED				
55	0101	0101		NOT	USED				
56	0101	0110		NOT	USED				
57	0101	0111		NOT	USED				
58	0101	1000		NOT	USED				
59	0101	1001		NOT	USED				
5A	0101	1010	]	Right Bracket	Right Bracket	]	0101	1101	5D
5B	0101	0111	$	Dollar Sign	Dollar Sign	$	0010	0100	21
5C	0101	1100	*	Asterisk	Asterisk	*	0010	1010	2A
5D	0101	1101	)	Right Parenthesis	Right Parenthesis	)	0010	1001	29
5E	0101	1110	;	Semicolon	Semicolon	;	0011	1011	3B
5F	0101	1111	¬	Logical NOT	Circumflex	^	0101	1110	5E
60	0110	0000	–	Minus Sign, Hyphen	Minus Sign, Hyphen	–	0010	1101	2D
61	0110	0001	/	Slash	Slash	/	0010	1111	2F
62	0110	0010		NOT	USED				
63	0110	0011		NOT	USED				
64	0110	0100		NOT	USED				
65	0110	0101		NOT	USED				
66	0110	0110		NOT	USED				
67	0110	0111		NOT	USED				
68	0110	1000		NOT	USED				
69	0110	1001		NOT	USED				
6A	0110	1010	|	Vertical LIne	Vertical Line	|	0111	1100	7C
6B	0110	1011	,	Comma	Comma	,	0010	1100	2C
6C	0110	1100	%	Percent	Percent	%	0011	0101	25
6D	0110	1101	—	Underline	Underline	—	0101	1111	5P
6E	0110	1110	>	Greater Than	Greater Than	>	0011	1110	3E
6F	0110	1111	?	Question Mark	Question Mark	?	0011	1111	3F
70	0111	0000		NOT	USED				
71	0111	0001		NOT	USED				
72	0111	0010		NOT	USED				
73	0111	0011		NOT	USED				
74	0111	0100		NOT	USED				

Table 5–1. Control Unit Translation Codes, EBCDIC to ASCII (Part 2 of 5)

EBCDIC					ASCII				
Hex Code	8-Bit Code 0123	8-Bit Code 4567	Graphic or Control	Name of Graphic or Control	Name of Graphic or Control	Graphic or Control	8-Bit Code 7654	8-Bit Code 3210	Hex Code
75	0111	0101		NOT	USED				
76	0111	0110		NOT	USED				
77	0111	0111		NOT	USED				
78	0111	1000		NOT	USED				
79	0111	1001	`	Grave Accent	Grave Accent	`	0110	0000	60
7A	0111	1010	:	Colon	Colon	:	0011	1010	3A
7B	0111	1011	#	Number Sign	Number Sign	#	0010	0011	23
7C	0111	1100	@	Commercial At Symbol	Commercial At Symbol	@	0110	0000	40
7D	0111	1101	'	Prime, Apostrophe	Prime, Apostrophe	'	0010	0111	27
7E	0111	1110	=	Equal Sign	Equal Sign	=	0011	1101	3D
7F	0111	1111	"	Quotation Mark	Quotation Mark	"	0010	0010	22
80	1000	0000		NOT	USED				
81	1000	0001	a			a	0110	0001	61
82	1000	0010	b			b	0110	0010	62
83	1000	0011	c			c	0110	0011	63
84	1000	0100	d			d	0110	0100	64
85	1000	0101	e			e	0110	0101	65
86	1000	0110	f			f	0110	0110	66
87	1000	0111	g			g	0110	0111	67
88	1000	1000	h			h	0110	1000	68
89	1000	1001	i			i	0110	1001	69
8A	1000	1010		NOT	USED				
8B	1000	1011		NOT	USED				
8C	1000	1100		NOT	USED				
8D	1000	1101		NOT	USED				
8E	1000	1110		NOT	USED				
8F	1000	1111		NOT	USED				
90	1001	0000		NOT	USED				
91	1001	0001	j			j	0110	1010	6A
92	1001	0010	k			k	0110	1011	6B
93	1001	0011	l			l	0110	1100	6C
94	1001	0100	m			m	0110	1101	6D
95	1001	0101	n			n	0110	1110	6E
96	1001	0110	o			o	0110	1111	6F
97	1001	0111	p			p	0111	0000	70
98	1001	1000	q			q	0111	0001	71
99	1001	1001	r			r	0111	0010	72
9A	1001	1010		NOT	USED				
9B	1001	1011		NOT	USED				
9C	1001	1100		NOT	USED				
9D	1001	1101		NOT	USED				
9E	1001	1110		NOT	USED				
9F	1001	1111		NOT	USED				
A0	1010	0000		NOT	USED				
A1	1010	0001	~	Tilde	Tilde	~	0111	1110	7E
A2	1010	0010	s			s	0111	0011	73
A3	1010	0011	t			t	0111	0100	74
A4	1010	0100	u			u	0111	0101	75
A5	1010	0101	v			v	0111	0110	76
A6	1010	0110	w			w	0111	0111	77
A7	1010	0111	x			x	0111	1000	78
A8	1010	1000	y			y	0111	1001	79
A9	1010	1001	z			z	0111	1010	7A
AA	1010	1010		NOT	USED				
AB	1010	1011		NOT	USED				
AC	1010	1100		NOT	USED				
AD	1010	1101		NOT	USED				
AE	1010	1110		NOT	USED				

Table 5–1. Control Unit Translation Codes, EBCDIC to ASCII (Part 3 of 5)

EBCDIC					ASCII				
Hex Code	8-Bit Code 0123	8-Bit Code 4567	Graphic or Control	Name of Graphic or Control	Name of Graphic or Control	Graphic or Control	8-Bit Code 7654	8-Bit Code 3210	Hex Code
AF	1010	1111		NOT	USED				
B0	1011	0000		NOT	USED				
B1	1011	0001		NOT	USED				
B2	1011	0010		NOT	USED				
B3	1011	0011		NOT	USED				
B4	1011	0100		NOT	USED				
B5	1011	0101		NOT	USED				
B6	1011	0110		NOT	USED				
B7	1011	0111		NOT	USED				
B8	1011	1000		NOT	USED				
B9	1011	1001		NOT	USED				
BA	1011	1010		NOT	USED				
BB	1011	1011		NOT	USED				
BC	1011	1100		NOT	USED				
BD	1011	1101		NOT	USED				
BE	1011	1110		NOT	USED				
BF	1011	1111		NOT	USED				
C0	1100	0000	{	Left Brace	Left Brace	{	0111	1011	7B
C1	1100	0001	A			A	0100	0001	41
C2	1100	0010	B			B	0100	0010	42
C3	1100	0011	C			C	0100	0011	43
C4	1100	0100	D			D	0100	0100	44
C5	1100	0101	E			E	0100	0101	45
C6	1100	0110	F			F	0100	0110	46
C7	1100	0111	G			G	0100	0111	47
C8	1100	1000	H			H	0100	1000	48
C9	1100	1001	I			I	0100	1001	49
CA	1100	1010		NOT	USED				
CB	1100	1011		NOT	USED				
CC	1100	1100		NOT	USED				
CD	1100	1101		NOT	USED				
CE	1100	1110		NOT	USED				
CF	1100	1111		NOT	USED				
D0	1101	0000	}	Right Brace	Right Brace	}	0111	1101	7D
D1	1101	0001	J			J	0100	1010	4A
D2	1101	0010	K			K	0100	1011	4B
D3	1101	0011	L			L	0100	1100	4C
D4	1101	0100	M			M	0100	1101	4D
D5	1101	0101	N			N	0100	1110	4E
D6	1101	0110	O			O	0100	1111	4F
D7	1101	0111	P			P	0101	0000	50
D8	1101	1000	Q			Q	0101	0001	51
D9	1101	1001	R			R	0101	0010	52
DA	1101	1010		NOT	USED				
DB	1101	1011		NOT	USED				
DC	1101	1100		NOT	USED				
DD	1101	1101		NOT	USED				
DE	1101	1110		NOT	USED				
DF	1101	1111		NOT	USED				
E0	1110	0000	\	Reverse Slash	Reverse Slash	\	0101	1100	5C
E1	1110	0001		NOT	USED				
E2	1110	0010	S			S	0101	0011	53
E3	1110	0011	T			T	0101	0100	54
E4	1110	0100	U			U	0101	0101	55
E5	1110	0101	V			V	0101	0110	56
E6	1110	0110	W			W	0101	0111	57
E7	1110	0111	X			X	0101	1000	58
E8	1110	1000	Y			Y	0101	0101	59

Table 5–1. Control Unit Translation Codes, EBCDIC to ASCII (Part 4 of 5)

EBCDIC					ASCII				
Hex Code	8-Bit Code		Graphic or Control	Name of Graphic or Control	Name of Graphic or Control	Graphic or Control	8-Bit Code		Hex Code
	0123	4567					7654	3210	
E9	1110	1001	Z			Z	0101	1010	5A
EA	1110	1010		NOT	USED				
EB	1110	1011		NOT	USED				
EC	1110	1100		NOT	USED				
ED	1110	1101		NOT	USED				
EE	1110	1110		NOT	USED				
EF	1110	1111		NOT	USED				
F0	1111	0000	0			0	0011	0000	30
F1	1111	0001	1			1	0011	0001	31
F2	1111	0010	2			2	0011	0010	32
F3	1111	0011	3			3	0011	0011	33
F4	1111	0100	4			4	0011	0100	34
F5	1111	0101	5			5	0011	0101	35
F6	1111	0110	6			6	0011	0110	36
F7	1111	0111	7			7	0011	0111	37
F8	1111	1000	8			8	0011	1000	38
F9	1111	1001	9			9	0011	1001	39
FA	1111	1010		NOT	USED				
FB	1111	1011		NOT	USED				
FC	1111	1100		NOT	USED				
FD	1111	1101		NOT	USED				
FE	1111	1110		NOT	USED				
FF	1111	1111		NOT	USED				

Table 5–1. Control Unit Translation Codes, EBCDIC to ASCII (Part 5 of 5)

ASCII					EBCDIC				
Hex Code	8-Bit Code		Graphic or Control	Name of Graphic or Control	Name of Graphic or Control	Graphic or Control	8-Bit Code		Hex Code
	7654	3210					0123	4567	
00	0000	0000	NUL	Null	Null	NUL	0000	0000	00
01	0000	0001	SOH	Start of Heading	Start of Heading	SOH	0000	0001	01
02	0000	0010	STX	Start of Text	Start of Text	STX	0000	0010	02
03	0000	0011	ETX	End of Text	End of Text	ETX	0000	0011	03
04	0000	0100	EOT	End of Transmission	End of Transmission	EOT	0011	0111	37
05	0000	0101	ENQ	Enquiry	Enquiry	ENQ	0010	1110	2D
06	0000	0110	ACK	Acknowledge	Acknowledge	ACK	0010	1101	2E
07	0000	0111	BEL	Bell	Bell	BEL	0010	1111	2F
08	0000	1000	BS	Backspacer	Backspacer	BS	0001	0110	16
09	0000	1001	HT	Horizontal Tab	Horizontal Tab	HT	0000	0101	05
0A	0000	1010	LF	Line Feed	Line Feed	LF	0010	0101	25
0B	0000	1011	VT	Vertical Tab	Vertical Tab	VT	0000	1011	0B
0C	0000	1100	FF	Form Feed	Form Feed	FF	0000	1100	0C
0D	0000	1101	CR	Carriage Return	Carriage Return	CR	0000	1101	0D
0E	0000	1110	SO	Shift Out	Shift Out	SO	0000	1110	0E
0F	0000	1111	SI	Shift In	Shift In	SI	0000	1111	0F
10	0001	0000	DLE	Data Link Escape	Data Link Escape	DLE	0001	0000	10
11	0001	0001	DC1	Device Control 1	Device Control 1	DC1	0001	0001	11
12	0001	0010	DC2	Device Control 2	Device Control 2	DC2	0001	0010	12
13	0001	0011	DC3	Device Control 3	Device Control 3	DC3	0001	0011	13
14	0001	0100	DC4	Device Control 4	Device Control 4	DC4	0011	1100	3C
15	0001	0101	NAK	Negative Acknowledge	Negative Acknowledge	NAK	0011	1101	3D
16	0001	0110	SYN	Synchronous Idle	Synchronous Idle	SYN	0011	0010	32
17	0001	0111	ETB	End of Transmission Block	End of Transmission Block	ETB	0010	0110	26
18	0001	1000	CAN	Cancel	Cancel	CAN	0001	1000	18
19	0001	1001	EM	End of Medium	End of Medium	EM	0001	1001	19
1A	0001	1010	SUB	Substitute	Substitute	SUB	0011	1111	3F
1B	0001	1011	ESC	Escape	Escape	ESC	0010	0111	27
1C	0001	1100	FS	Field Separator	Field Separator	FS	0001	1100	1C
1D	0001	1101	GS	Group Separator	Group Separator	GS	0001	1101	1D
1E	0001	1110	RS	Reader Stop (SOE) - Start of Entry)	Reader Stop (SOE) - Start of Entry)	RS	0001	1110	1E
1F	0001	1111	US	Unit Separator	Unit Separator	US	0001	1111	1F
20	0010	0000	SP	Space	Space	SP	0100	0000	40
21	0010	0001	!	Exclamation Point	Exclamation Point	!	0100	1111	4F
22	0010	0010	"	Quotation Mark	Quotation Mark	"	0111	1111	7F
23	0010	0011	#	Number Sign	Number Sign	#	0111	1011	7B
24	0010	0100	$	Dollar Sign	Dollar Sign	$	0101	1011	5B
25	0010	0101	%	Percent Sign	Percent Sign	%	0110	1000	6C
26	0010	0110	&	Ampersand	Ampersand	&	0101	0000	50
27	0010	0111	'	Apostrophe	Apostrophe	'	0111	1101	7D
28	0010	1000	(	Left Parenthesis	Left Parenthesis	(	0100	1101	4D
29	0010	1001	)	Right Parenthesis	Right Parenthesis	)	0101	1101	5D
2A	0010	1010	*	Asterisk	Asterisk	*	0100	1110	5C
2B	0010	1011	+	Plus Sign	Plus Sign	+	0100	1110	4E
2C	0010	1100	,	Comma	Comma	,	0110	1011	6B
2D	0010	1101	–	Minus Sign	Minus Sign	–	0110	0000	60
2E	0010	1110	.	Period (Decimal Point)	Period (Decimal Point)	.	0100	1011	4B
2F	0010	1111	/	Slash	Slash	/	0110	0001	61
30	0011	0000	0	Zero	Zero	0	1111	0000	F0
31	0011	0001	1			1	1111	0001	F1
32	0011	0010	2			2	1111	0010	F2
33	0011	0011	3			3	1111	0011	F3
34	0011	0100	4			4	1111	0100	F4
35	0011	0101	5			5	1111	0101	F5
36	0011	0110	6			6	1111	0110	F6
37	0011	0111	7			7	1111	0111	F7
38	0011	1000	8			8	1111	1000	F8
39	0011	1001	9			9	1111	1001	F9
3A	0011	1010	:	Colon	Colon	:	0111	1010	7A
3B	0011	1011	;	Semicolon	Semicolon	;	0101	1110	5E
3C	0011	1100	<	Less Than	Less Than	<	0100	1100	4C
3D	0011	1101	=	Equal Sign	Equal Sign	=	0111	1110	7E
3E	0011	1110	>	Greater Than	Greater Than	>	0110	1110	6E

Table 5–2. Control Unit Transition Codes, ASCII to EBCDIC (Part 1 of 2)

ASCII					EBCDIC				
Hex Code	8-Bit Code		Graphic or Control	Name of Graphic or Control	Name of Graphic or Control	Graphic or Control	8-Bit Code		Hex Code
	7654	3210					0123	4567	
3F	0011	1111	?	Question Mark	Question Mark	?	0110	1111	6F
40	0100	0000	@	Commercial At Symbol	Commercial At Symbol	@	0111	1100	7C
41	0100	0001	A			A	1100	0001	C1
42	0100	0010	B			B	1100	0010	C2
43	0100	0011	C			C	1100	0011	C3
44	0100	0100	D			D	1100	0100	C4
45	0100	0101	E			E	1100	0101	C5
46	0100	0110	F			F	1100	0110	C6
47	0100	0111	G			G	1100	0111	C7
48	0100	1000	H			H	1100	1000	C8
49	0100	1001	I			I	1100	1001	C9
4A	0100	1010	J			J	1101	0001	D1
4B	0100	1011	K			K	1101	0010	D2
4C	0100	1100	L			L	1101	0011	D3
4D	0100	1101	M			M	1101	0100	D4
4E	0100	1110	N			N	1101	0101	D5
4F	0100	1111	O			O	1101	0110	D6
50	0101	0000	P			P	1101	0111	D7
51	0101	0001	Q			Q	1101	1000	D8
52	0101	0010	R			R	1101	1001	D9
53	0101	0011	S			S	1110	0010	E2
54	0101	0100	T			T	1110	0011	E3
55	0101	0101	U			U	1110	0100	E4
56	0101	0110	V			V	1110	0101	E5
57	0101	0111	W			W	1110	0110	E6
58	0101	1000	X			X	1110	0111	E7
59	0101	1001	Y			Y	1110	1000	E8
5A	0101	1010	Z			Z	1110	1001	E9
5B	0101	1011	[	Left Bracket	Left Bracket	[	0100	1010	4A
5C	0101	1100	\	Reverse Slash	Reverse Slash	\	1110	0000	E0
5D	0101	1101	]	Right Bracket	Right Bracket	]	0101	1010	5A
5E	0101	1110	^	Circumflex	Circumflex	^	0101	1111	5F
5F	0101	1111	—	Underline	Underline	—	0110	1101	6D
60	0110	0000	`	Grave accent	Grave accent	`	0111	1001	79
61	0110	0001	a			a	1000	0001	81
62	0110	0010	b			b	1000	0010	82
63	0110	0011	c			c	1000	0011	83
64	0110	0100	d			d	1000	0100	84
65	0110	0101	e			e	1000	0101	85
66	0110	0110	f			f	1000	0110	86
67	0110	0111	g			g	1000	0111	87
68	0110	1000	h			h	1000	1000	88
69	0110	1001	i			i	1000	1001	89
6A	0110	1010	j			j	1001	0001	91
6B	0110	1011	k			k	1001	0010	92
6C	0110	1100	l			l	1001	0011	93
6D	0110	1101	m			m	1001	0100	94
6E	0110	1110	n			n	1001	0101	95
6F	0110	1111	o			o	1001	0110	96
70	0111	0000	p			p	1001	0111	97
71	0111	0001	q			q	1001	1000	98
72	0111	0010	r			r	1001	1001	99
73	0111	0011	s			s	1010	0010	A2
74	0111	0100	t			t	1010	0011	A3
75	0111	0101	u			u	1010	0100	A4
76	0111	0110	v			v	1010	0101	A5
77	0111	0111	w			w	1010	0110	A6
78	0111	1000	x			x	1010	0111	A7
79	0111	1001	y			y	1010	1000	A8
7A	0111	1010	z			z	1010	1001	A9
7B	0111	1011	{	Left Brace	Left Brace	{	1100	0000	C0
7C	0111	1100	|	Vertical Line	Vertical Line	|	0110	1010	6A
7D	0111	1101	}	Right Brace	Right Brace	}	1101	0000	D0
7E	0111	1110	~	Tilde	Tilde	~	1010	0001	A1
7F	0111	1111	DEL	Delete	Delete	DEL	0000	0111	07

Table 5–2. Control Unit Transition Codes, ASCII to EBCDIC (Part 2 of 2)

5.3. CONTROLS AND INDICATORS

5.3.1. Keyboard and System Control Switches and Indicators

The operator console is the main area of control for the UNIVAC 9700 System operator. The operator console consists of a UNISCOPE 100 Display Terminal mounted as part of the cabinet. The operator console includes a display screen with three controls/indicators, a keyboard for entering data and information, and 16 controls and indicators for further system control. Figures 5–4 and 5–5 show the operator console, and Table 5–3 describes the keyboard controls and indicators.

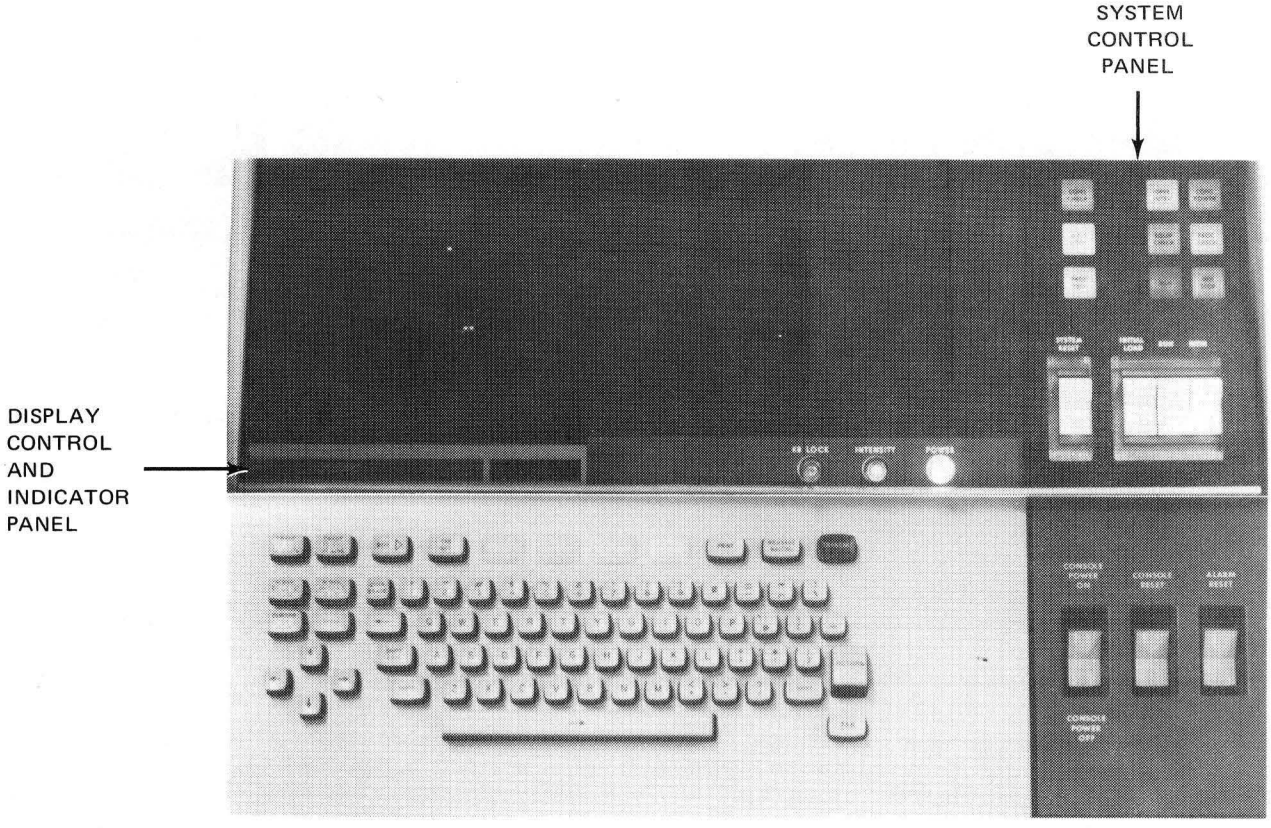
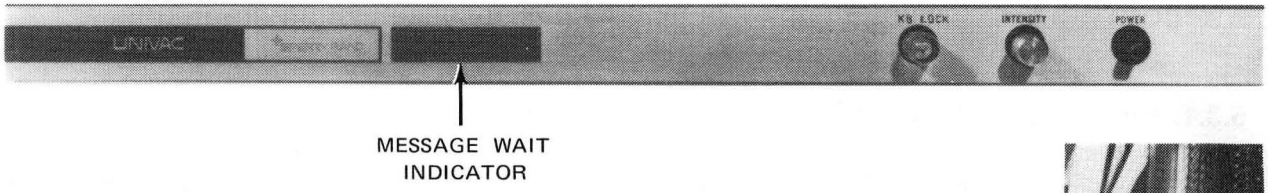


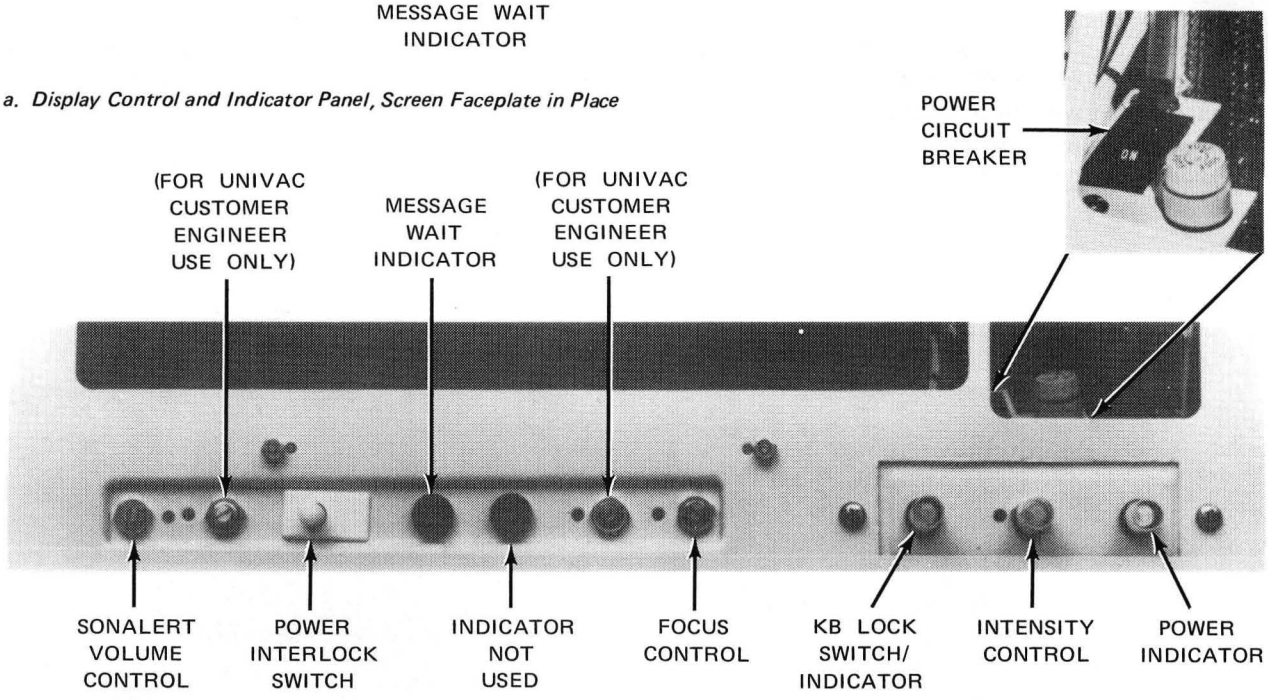
Figure 5–4. UNIVAC 9700 Operator Console, Controls and Indicators

5.3.1.1. DISPLAY CONTROL AND INDICATOR PANEL

The display controls and indicators shown in Figure 5–5 are located above the keyboard. They control and indicate the status of the equipment and messages. The functions of these controls and indicators are also given in Table 5–3.



a. Display Control and Indicator Panel, Screen Faceplate in Place



b. Display Control and Indicator Panel, Screen Faceplate Removed

Figure 5–5. Display Control and Indicator Panel

Name	Function
	Screen
Display Screen	The display screen has a total display capacity of 1024 characters. Data is displayed on 16 lines numbered vertically 0 through 15 from the top of the screen, each line having a 64-character capacity. Depending on what is selected at SYSGEN time, either one or two lines are reserved at the bottom of the screen for operator input. Data from the system is displayed starting at line 13 or 14 depending on how many lines are reserved for the operator. As system messages appear on the screen, and as operator requests are answered, they are displayed on either line 13 or 14. As new messages appear, the first lines are rolled up to line 12, then 11, and so forth, in a scrolling effect until, when they reach the top of the screen, they are rolled off and lost.
	Display Control and Indicator Panel
MESSAGE WAIT indicator	The indicator lights when a message displayed at the operator console requires a response by the operator.
KB LOCK switch/indicator	This switch/indicator lights when the keyboard is locked. The keyboard, except for the MESSAGE WAITING key, is locked whenever the TRANSMIT key is pressed or an input data transfer is occurring on the screen (in which case the control unit unlocks the keyboard at the end of the transfer). The operator may manually unlock the keyboard by pressing this switch/indicator.
INTENSITY control	This control adjusts the brightness of the screen display and should be adjusted for a clear image and not for maximum brightness, which shortens screen life.
POWER indicator	This indicator lights when power is applied to the operator console and the UNISCOPE 100.

Table 5–3. Operator Console Keyboard Controls and Indicators (Part 1 of 4)

Name	Function
	Keyboard
Typewriter Keys	These keys are arranged in standard typewriter fashion and include alphanumeric and symbolic functions. These are the keys for message composition to the processor. As a key is pressed, the character (uppercase or lowercase depending on the SHIFT key) is displayed on the screen at the cursor position. Pressing the SHIFT LOCK key locks the keyboard in the uppercase mode, and pressing the SHIFT key returns the keyboard to the lowercase mode.
	Message Control Keys
SOE ▷	This is the start-of-entry key which places the start-of-entry symbol on the screen at a position corresponding with the cursor. The location of this symbol on the screen designates the starting point of the message to be transmitted to the processor. The cursor determines the end of message. Automatic carriage return is employed following the last character of each line (nonsignificant space codes are suppressed). If more than one start-of-entry symbol happens to be entered upon the screen, transmission will start at the start-of-entry symbol closest to the cursor and continue until the cursor is reached. If the start-of-entry symbol is not inserted, all data on the screen from the beginning of the display to the cursor is transmitted.
TRANSMIT	Pressing this key alerts the processor that the input message (located between the start-of-entry symbol and cursor, or from beginning of display if no start-of-entry symbol is present) is to be transmitted. Pressing this key locks the keyboard, except for the MESSAGE WAITING key and special function keys, until a read command is received from the processor. The processor can lock the console keyboard by transmitting a lock character in the return message.
MESSAGE WAITING	Pressing this key alerts the processor that the operator wishes to enter a message on the screen. The processor must unlock the console keyboard to permit operator data entry.
	Cursor Control Keys
Cursor Symbol ⌐	The cursor symbol, small right angle (⌐), is always present in the screen display and indicates the location where the next keyboard-generated character will be displayed. When in a blank space, the cursor is displayed; however, when in a position already occupied, the character and cursor blink alternately. When in the eighth character position from the end of a line, an audible alarm is momentarily sounded.
→ scan forward	If this key is pressed momentarily, it causes the cursor to move forward (to the right) one position. If the key is pressed and held, the cursor moves forward approximately 10 character positions per second.
← scan backward	If this key is pressed momentarily, it causes the cursor to move backward (to the left) one position. If the key is pressed and held, the cursor moves backward approximately 10 character positions per second.
↑ scan upward	If this key is pressed momentarily, it causes the cursor to move up one line. If the key is pressed and held, the cursor moves upward approximately 10 lines per second.
↓ scan downward	If this key is pressed momentarily, it causes the cursor to move down one line, If the key is pressed and held, the cursor moves downward approximately 10 lines per second. The space bar provides the same result as the scan forward key. The backspace key provides the same result as the scan backward key.
CYCLE	Pressing this key causes the next alphanumeric or symbolic character (chosen by the operator) to be repeated at a rate of about 10 cycles per second. This key operates with all keys except the ERASE, DELETE, INSERT, SHIFT, SHIFTLOCK, PRINT, MESSAGE WAITING, TRANSMIT, and CURSOR TO HOME keys. Repetition continues as long as the CYCLE key and typewriter keys are held pressed.
CURSOR TO HOME	Pressing this key causes the cursor to move to the first character position on the screen (upper left-hand corner).

Table 5–3. Operator Console Keyboard Controls and Indicators
(Part 2 of 4)

Name	Function
RETURN	Pressing this key moves the cursor to the first character position of the next line.
TAB SET	Pressing this key sets the tab code by inserting a tab character in storage. The position corresponds to the position of the cursor.
TAB	Pressing this key moves the cursor ahead to a position on the screen one space beyond that which was set by the TAB SET key. If no tab was set, the cursor will stop at the home position.
Editing Control Keys	
CHAR ERASE	Pressing this key replaces a character at the cursor position with a space.
ERASE TO END OF LINE	Pressing this key replaces all characters from and including the cursor to the end of the line with spaces.
ERASE TO END OF DISPL	Pressing this key replaces all characters from and including the cursor to the end of the display with spaces.
IN DISPL INSERT IN LINE	Lowercase operation of this key causes all the characters in the line from and including the cursor position to move right one position. A space is inserted in the cursor position, and if a character is moved out of the last position of the line, it is discarded. Uppercase operation of this key causes all the characters in the display from and including the cursor position to move right one position. A space is inserted in the cursor position, and if a character is moved out of the last position of the display, it is discarded.
IN DISPL DELETE IN LINE	Lowercase operation of this key causes the characters in the line to the right of the cursor to shift left one position. The original character at the cursor position is deleted, and a space is inserted into the last character position in the cursor line. This key is pressed once for each character to be deleted. Uppercase operation of this key causes all characters in the display to the right of the cursor to move left one position. The original character at the cursor position is deleted, and a space is inserted into the last character position of the display. This key is pressed once for each character deleted.
System Control Panel	
CONS CHECK indicator	This indicator lights when any fault indicator or test mode setting is activated at the printer or test panels.
WAIT MODE indicator	This indicator lights when the processor is operating in the wait mode.
PROC TEST indicator	This indicator lights when a switch on the processor or maintenance panel is set to oerate the processor in the test mode.
OPR INTRV indicator	When this indicator lights, the operator must press the KB LOCK switch to clear the UNISCOPE 100.
EQUIP CHECK indicator	This indicator lights when a power fault is detected by the system power control.

Table 5–3. Operator Console Keyboard Controls and Indicators (Part 3 of 4)

Name	Function
RUN indicator	This indicator lights when the processor is in the run mode of operation.
CONS POWER CHECK indicator	This indicator lights when a power fault is detected by the console power control facility.
PROC CHECK indicator	This indicator lights when a nonrecoverable error is detected by the processor.
HPR STOP indicator	This indicator lights when the processor is stopped at a display point requested by a halt-and-proceed instruction.
SYSTEM RESET switch	This momentary-contact rocker switch, when pressed, clears the system and prepares for initial loading. The processor, channels, and all nonshared control units and their associated devices are cleared.
INITIAL LOAD switch	This momentary-contact rocker switch, when pressed, starts the initial load sequence on the channel and device specified by the CHANNEL ADDRESS and DEVICE ADDRESS switches on the operator panel of the system maintenance panel. The error-free completion of this sequence activates the RUN indicator.
RUN switch	This momentary-contact rocker switch, when pressed, initiates processor operation.
INTER switch	This momentary-contact rocker swtich, when pressed, initiates a request for an external processor interrupt.
CONSOLE POWER ON/ CONSOLE POWER OFF switch	This two-position rocker switch, when placed in the ON (top half pressed) position, initiates the console power on sequence; when placed in the OFF (bottom half pressed), initiates console power-down sequence.
CONSOLE RESET switch	This two-position rocker switch, when top half is pressed, clears console, console control, and printer (if attached); also, lights all console indicators for approximately two seconds to serve as a lamp check.
Sonalert Alarm	
ALARM RESET switch	This monentary-contact rocker switch, when pressed, resets the alarm and MESSAGE WAIT indicator. The audible alarm sounds for any of the following conditions: ■ When cursor enters last 8-character position of any line of the display, alarm sounds once. ■ When the cursor enters the first character position of the last line of the display, alarm sounds once. ■ When MESSAGE WAIT indicator is lit, alarm sounds continuously. (Alarm may be silenced under control of processor.)
Volume Control	This screwdriver-adjust control is located behind the display screen faceplate (behind UNIVAC bezel) to the left of the MESSAGE WAIT indicator and is used to vary the sound level of the sonalert.
Focus	
Focus Control	This screwdriver-adjust control is located behind the screen display faceplate to the far right of the MESSAGE WAIT indicator.

Table 5–3. Operator Console Keyboard Controls and Indicators
(Part 4 of 4)

5.3.1.2. POWER CIRCUIT BREAKER

The power circuit breaker (Figure 5–5) switch is located inside the unit, below and toward the right of the display screen and behind the screen faceplate. When the switch is set toward the back, ac power is on. If a power surge or overload condition occurs, the circuit breaker automatically disconnects power from the UNISCOPE 100.

5.3.1.3. SECURITY SWITCH

A security switch is located on the right front corner on the underside of the display unit, and is used as a security measure against unauthorized operation. When the switch is set to the OFF (front) position, the keyboard is locked and the high voltage to the CRT display screen is off. The switch is accessed by removing two screws supporting a long narrow cover plate located under and toward the back of the keyboard. The switch is a toggle type and next to the front right rubber cushion leg. The switch cannot be set forward without the use of a hook-type tool at least six inches long, even though it can be pushed to the back (set to ON) without the aid of a tool.

5.3.2. Power Distribution Panel

The power distribution panel (Figure 5–6) is located in the lower rear right portion of the operator console, and it is accessed by opening the lower rear panel door.

The function of each of the circuit breakers, switches, and indicators is described in Table 5–4.

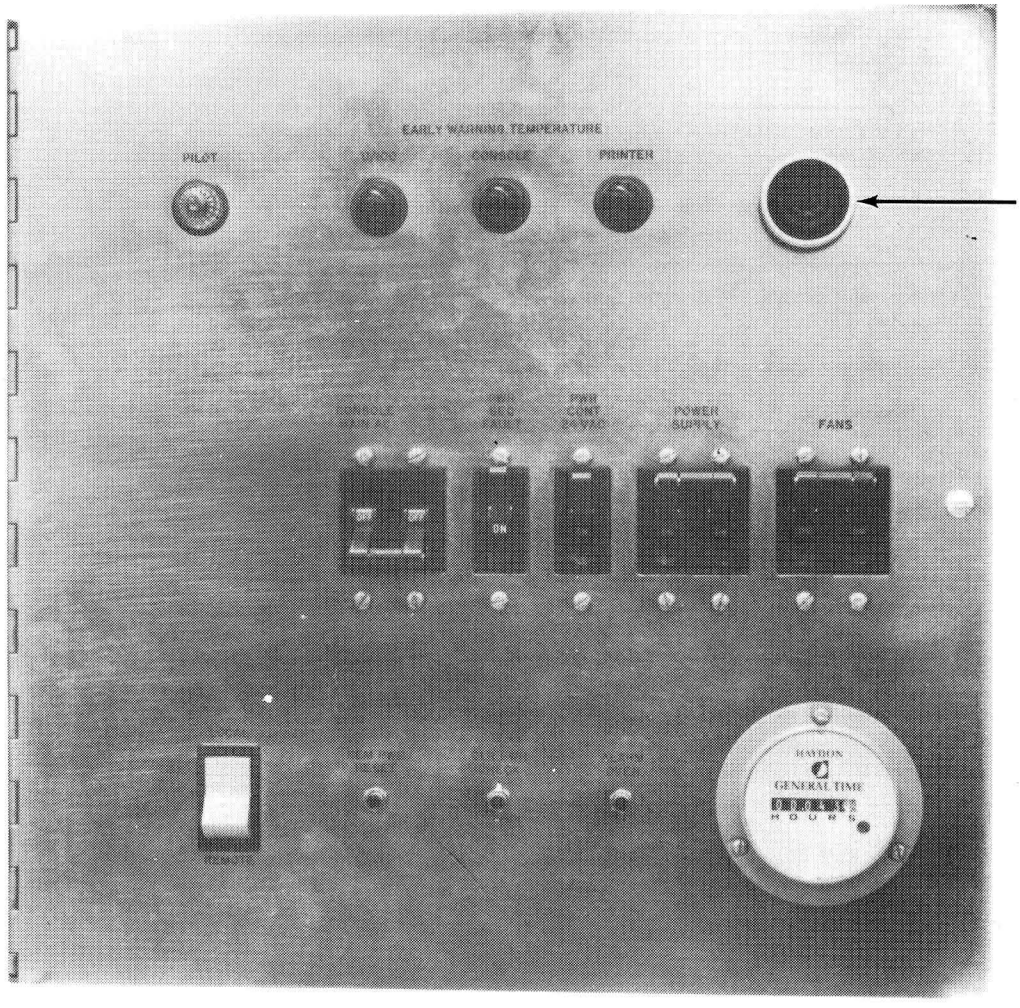


Figure 5–6. Operator Console Power Distribution Panel

Control/Indicator	Function
CONSOLE MAIN AC circuit breaker	Two-gang circuit breaker, when ON (up position) applies ac power to the system console; when set down, removes ac power from system console.
PWR SEQ FAULT circuit breaker	Single circuit breaker, when ON (up position) protects power sequencing circuits. Any power sequencing error detected causes this breaker to trip off.
PWR CONT 24 VAC circuit breaker	Single circuit breaker, when ON (up position) applies 24 VAC to the power sequence control circuits.
POWER SUPPLY circuit breaker	Two-gang circuit breaker, when ON (up position) applies ac power to two +6V and one −12V power supplies.
FANS circuit breaker	Two-gang circuit breaker, when ON (up position) applies ac power to the cooling fans.
LOCAL/REMOTE switch	For this two-position rocker-switch to be effective, CONSOLE POWER ON/CONSOLE POWER OFF rocker switch at system control panel must be set to CONSOLE POWER ON (top half pressed). When set to LOCAL position (top half pressed), permits power turnon from operator console control panel. When set to REMOTE (bottom half pressed), permits power turnon to be initiated by the processor.
REM PWR RESET pushbutton	When system console is used with any processing system other than UNIVAC 9700 System Processor and LOCAL/REMOTE switch is set to REMOTE, system console power will cycle down when power at the processor cycles down. Pressing this pushbutton permits system console power to be cycled up again, independent of the processor. This pushbutton has no effect when system console is used with UNIVAC 9700 System Processor.
CLR PWR CHECK pushbutton	When pressed, clears CONS POWER lamp circuit to extinguish lamp.
ALARM OVER pushbutton	When pressed, silences sonalert alarm.
PILOT indicator	When lit, indicates primary power is available at the system console power distribution panel if main ac power is on.
EARLY WARNING TEMPERATURE indicators	
U/100 (red)	When lit, indicates that the internal temperature of the UNISCOPE 100 has reached or exceeded 130°F.
CONSOLE (red)	When lit, indicates that the internal temperature of the console logic has reached or exceeded 130°F, or FANS circuit breaker is OFF.
PRINTER (red)	When lit, indicates that the internal temperature of the printer, when attached, has reached or exceeded 130°F.
Elapsed time indicator	Indicates in hours and tenths of an hour, the total accumulated operating time for the system console.

Table 5–4. Operator Console Power Distribution Panel, Controls and Indicators

5.3.3. Power Control Panel

The power control panel (Figure 5–7) is located below the power distribution panel in the lower rear, right portion of the operator console. It is accessed by opening the lower right rear panel door that is held shut by magnetic latches. The function of each of the circuit breakers and controls is described in Table 5–5.

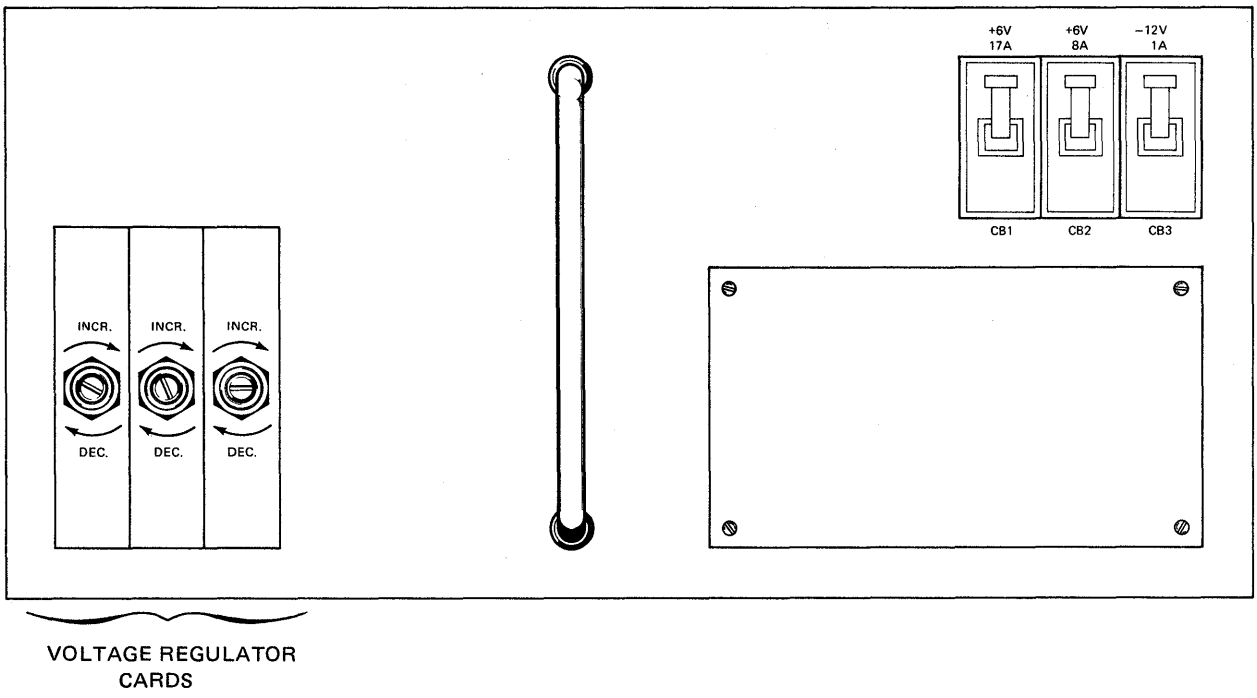


Figure 5–7. Operator Console Power Control Panel

Control	Function
CB1 +6VDC 17A	When set to ON (up position), applies +6VDC to the operator console circuits.
CB2 +6VDC 8A	When set to ON (up position), applies +6VDC to the UNIVAC 9000 Channel Adapter.
CB3 –12VDC 1A	When set to ON (up position), applies –12VDC to the operator console circuits.
INCREASE DECREASE voltage adjustment control on regulator cards	Adjust regulated dc voltage of the +6VDC 17A, +6VDC 8A, and –12VDC 1A power supplies. These controls are not to be adjusted by the operator.

Table 5–5. Operator Console Power Control Panel Controls

5.3.4. Logic Test Panel

A small test panel (Figure 5–8) for the logic section of the operator console is located in the left half of the lower rear portion of the cabinet. This test panel, except for one switch, is for maintenance purposes only and is not generally used by the operator. The one exception is the LINE off/on toggle switch. This switch is set to on (online) for regular operation, and set to off (offline) when maintenance is being performed on the operator console logic circuits.

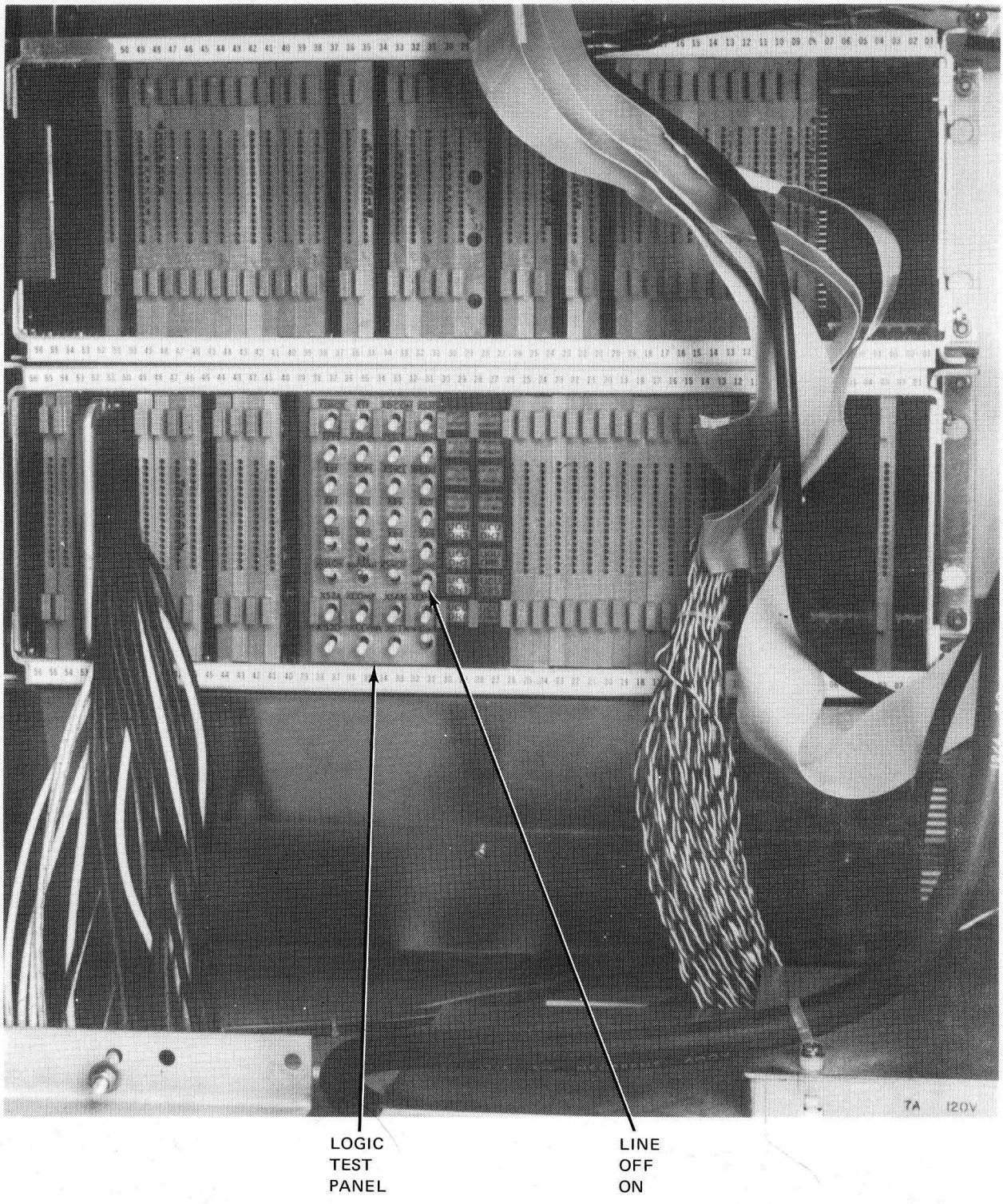


Figure 5–8. Operator Console Logic Test Panel

6. INCREMENTAL PRINTER

6.1. GENERAL

The UNIVAC 0722-00, -01 Receive-Only Incremental Printer (printer) is capable of printing the full 94-character American Standard Code for Information Interchange (ASCII) character set, plus spaces. Printing is at the rate of 30 characters per second on 132 columns. A synchronizer, data path circuitry, control logic, and power supplies are included in the printer cabinet (Figure 6–1).

6.2. CHARACTERISTICS

The printer may be located up to 30 feet from the operator console, and is directly addressable from the processor via the I/O channel. No buffer is required for the printer; therefore, all data inputs are supplied one character at a time. During transfer of data, the printer control unit remains in a busy state.

Figure 6–1. UNIVAC 0722 Receive-Only Incremental Printer

6.2.1. Physical Characteristics

The printer is mounted in a cabinet housing the printer mechanism, power supply, logic deck, control panel, and paper supply (hopper). Table 6–1 lists the printer's physical characteristics.

The printer cabinet is specifically designed to provide ready access to the unit. Ease of maintenance capabilities on the cabinet are:

- A hinged cover on the cabinet top permits access to the printer mechanism.
- Two hinged front doors permit access to the paper.
- Two hinged front doors permit access to the paper input hopper, power supply, and electronic plug-in cards.
- Two hinged rear doors permit access to electronic plug-in assemblies and to the 48-volt power supply.

Item	Characteristics
Size (nominal)	34" long 20" deep 47" high
Weight (nominal)	300 pounds
Operating voltage	Type 0722-00: 120/208, 120/240 volts, 60 Hz ± 0.5 Hz Type 0722-01: 220,230,240 volts, 50 Hz ± 1.0 Hz
Power consumption (nominal)	1300 watts
Heat dissipation (nominal)	2160 BTU/hr.
Temperature and relative humidity: Recommended temperature range Operating temperature limits Recommended humidity range Operating humidity limits Input Cable Length	 70 to 75° F 60 to 94° F 40 to 60% 20 to 75% 30 feet

Table 6–1. Printer Physical Characteristics

6.2.2. Printing Characteristics

Table 6–2 lists the printing characteristics.

Item	Characteristics
Print speed	Up to 30 characters per second
Print positions	Adjustable from 26 to 132 positions
Print spacing	Six lines per inch; ten characters per inch
Printed character size (typical)	Height: 0.090 to 0.110 inch Width: 0.060 to 0.085 inch (on single-part paper, 0.002-inch thickness)
Paper feed rate	One line per line feed signal; 30 lines per second for continuous line feed signals
Home paper rate	72 lines (12 inches) per second
Carriage return rate	385 milliseconds from 132nd print position. See Figure 6–4 for return rate of other print positions.
Printable characteristics	94 ASCII character set

Table 6–2. Printing Characteristics

6.3. FUNCTIONAL DESCRIPTION

The printer uses an advanced method of ribbonless printing with an ink-impregnated roller, rotating helical print wheel, and a single print hammer-actuator which operates with incremental carriage motion.

6.3.1. Ink Rollers

The helical print wheel is inked with an ink-impregnated porelon roller mounted in a hinged-covered housing in front of the print wheel (Figure 6–2). A free-running shaft is inserted into a bushing on the roller, allowing the roller to rotate freely. The roller and shaft are retained, but not fastened, between two mounting slots inside the housing.

As the print wheel turns, constant contact maintained with the ink roller causes ink to be transferred to the print wheel, continuously supplying it with ink.

An ink roller is easily replaced by holding the shaft and lifting the roller out of the housing. The shaft is inserted into the new roller and both are mounted in the housing in the slots provided for the shaft. The standard replacement ink roller uses black ink; however, rollers with red, green, or violet ink are also available.

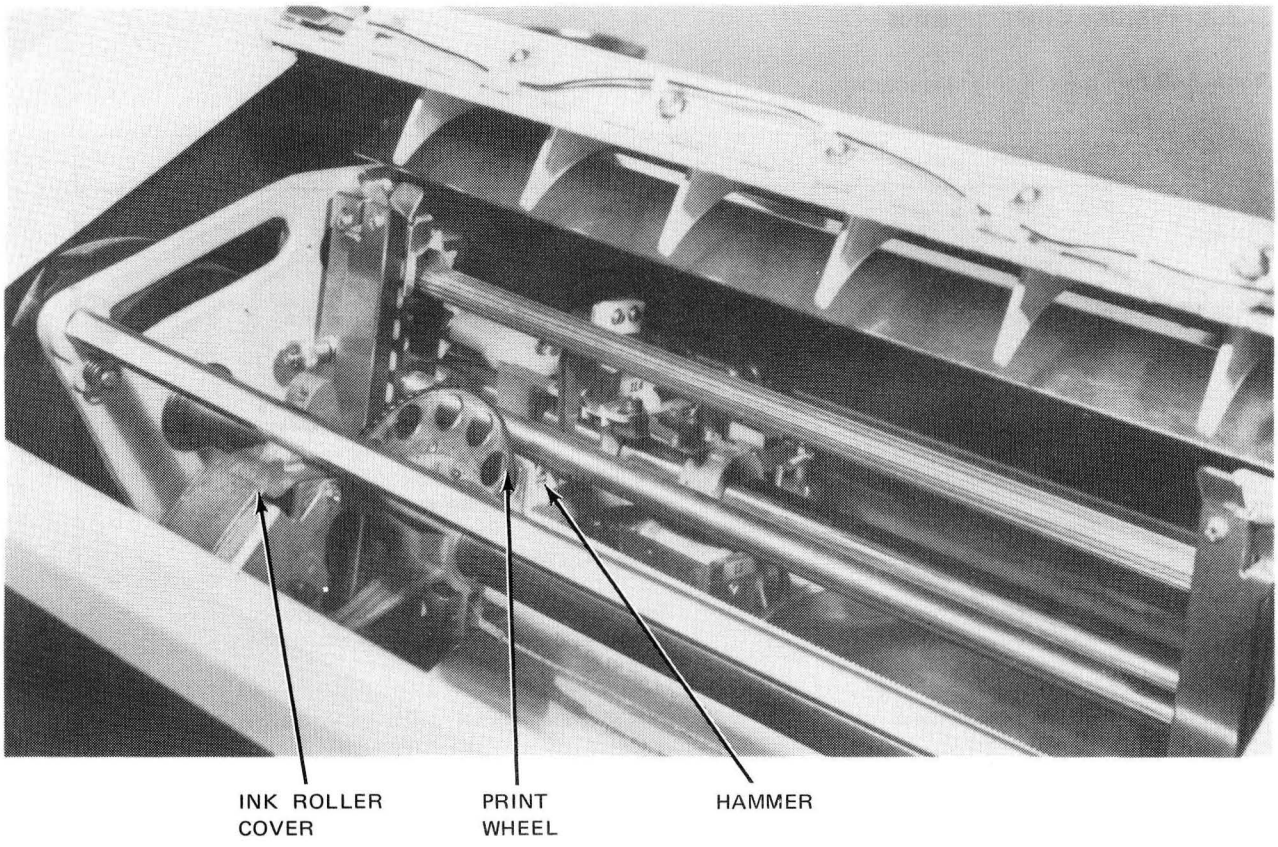


Figure 6–2. Printing Mechanism

6.3.2. Print Wheel and Print Hammer

The print wheel (Figure 6–2) contains a left and right font to provide 94 characters plus vertical line and diamond characters. Table 6–3 lists the character sequence for each font. As noted in the table, the left font on the print wheel (Univac part number 3533793-00) contains 63 characters plus nine diamonds, and the right font contains 31 vertical lines, 31 characters, and 10 diamonds.

Printing is accomplished with the rotating print wheel and hammer (Figure 6–2). When the selected character is in the print position, the hammer located behind the printing form is released, impressing the print wheel character to the paper's printing surface.

6.3.3. Incremental Carriage Movement

A pawl and rack mechanism provides the escapement required for incremental carriage motion. Carriage traverse control is obtained with a left margin microswitch and a right margin microswitch. A bar, mounted in front of the print wheel, is marked with graduated column positions (Figure 6–3). A pointer mounted on the print wheel housing moves along the marked bar so that the operator can determine the carriage column position when the carriage is at rest.

Four nonprintable control characters are interpreted by the printer: carriage return (CR), line feed (LF), space (SP), and form feed (FF). The CR and LF characters may be supplied in any order, but are activated automatically when the carriage return reaches the right margin plus two columns on the full 132-character print line. Automatic carriage return and line feed do not present an error indication to the processor.

The carriage is moved forward by the move-forward signal generated in the printer. The CR signal supplied from the processor causes the carriage to return to the left margin position. Return from the 132nd column position requires 385 milliseconds. Carriage return time for each print position up to the 132nd column is shown in Figure 6–4.

Sequence	Left Font	Right Font	Sequence	Left Font	Right Font	Sequence	Left Font	Right Font
1	!		25	9		49	Q	q
2	"		26	:		50	R	r
3	#		27	;		51	S	s
4	$		28	<		52	T	t
5	%		29	=		53	U	u
6	&		30	>		54	V	v
7	'		31	?		55	W	w
8	(		32	@		56	X	x
9	)		33	A	a	57	Y	y
10	*		34	B	b	58	Z	z
11	+		35	C	c	59	[	{
12	,		36	D	d	60	\	|
13	–		37	E	e	61	]	}
14	.		38	F	f	62	^	~
15	/		39	G	g	63	—	◇
16	0		40	H	h	64	◇	
17	1		41	I	i	65		
18	2		42	J	j	66		
19	3		43	K	k	67		
20	4		44	L	l	68		
21	5		45	M	m	69		
22	6		46	N	n	70		
23	7		47	O	o	71		
24	8		48	P	p	72	◇	◇

Table 6–3. Print Wheel Character Set

6.3.4. Paper Feed

The paper feed system uses two paper feed tractors (Figure 6–3). The left hand paper feed tractor is stationary but the right tractor may be adjusted for forms to a minimum width of 3-5/8 inches and a maximum width of 14-7/8 inches. When printing 132 characters per line on a maximum width form, the printed line is centered between the perforations on the left and right edges of the paper. Paper pressure plates (Figure 6–3) hold the paper firmly in place on the left and right tractors. Vertical adjustments of the form may be made with the platen knob on the right of the carriage.

Printing in columns 1 and 132 is 0.337 inch away from the left and right perforations, respectively. Additional space from the left perforations can be obtained by adding spaces to the first column positions.

Besides manual control of vertical movement of paper with the platen knob, the paper is indexed one line upon receipt of each LF function. Continuous LF signals move the paper at a maximum rate of 30 lines per second (lps).

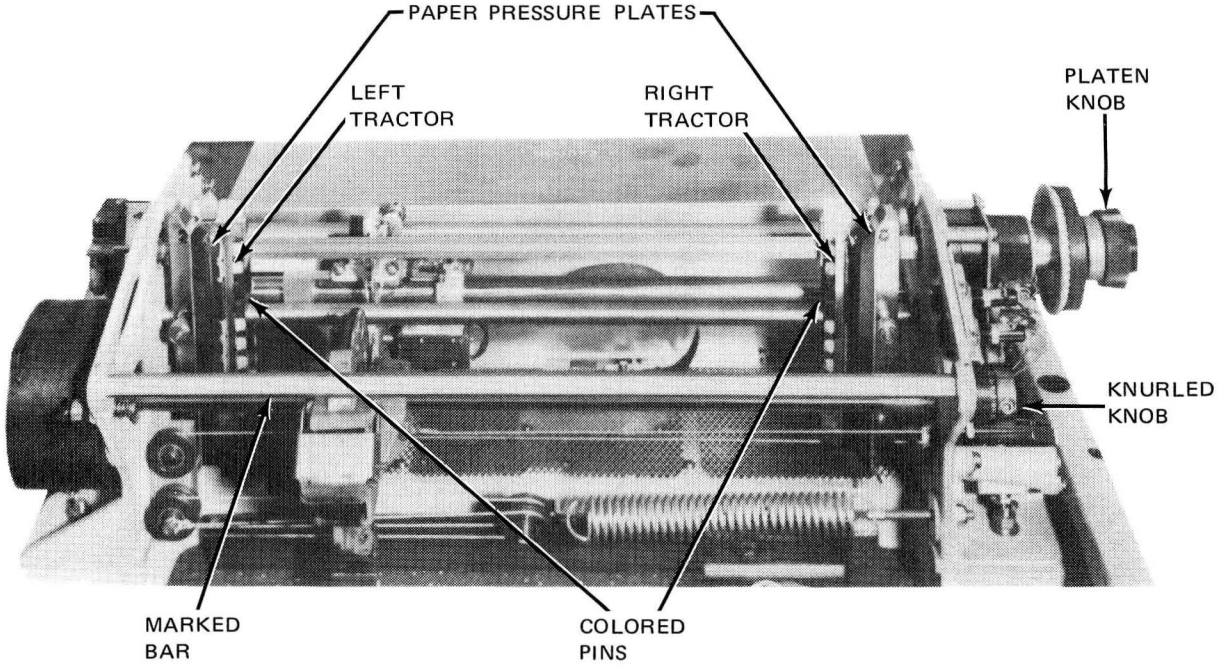
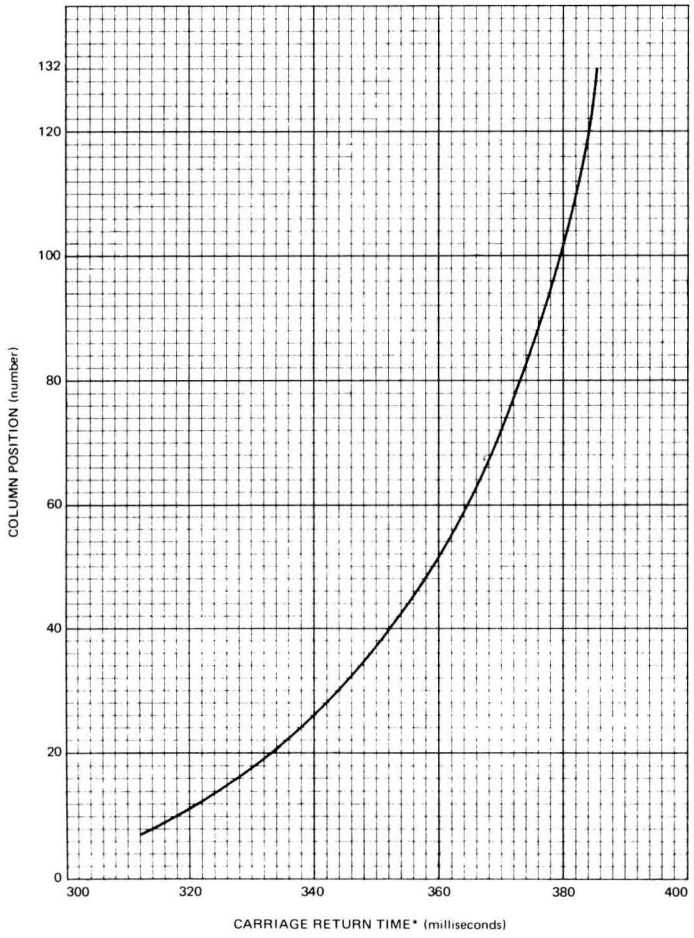


Figure 6–3. Printer Carriage



*Maximum carriage return time is for any mechanism adjusted to specification and operating within specified environmental limits.

Figure 6–4. Carriage Return Time for Print Positions

The print wheel carriage should be near center position when loading paper. Paper is supplied from the hopper located below the printing mechanism (Figure 6–5). After printing, the paper moves toward the rear of the printing mechanism and out through a slot at the rear of the cabinet (Figure 6–6). When approximately two inches of paper remain available for printing, FORMS END indicator lights to inform the operator that more paper is required and the printer operation stops.

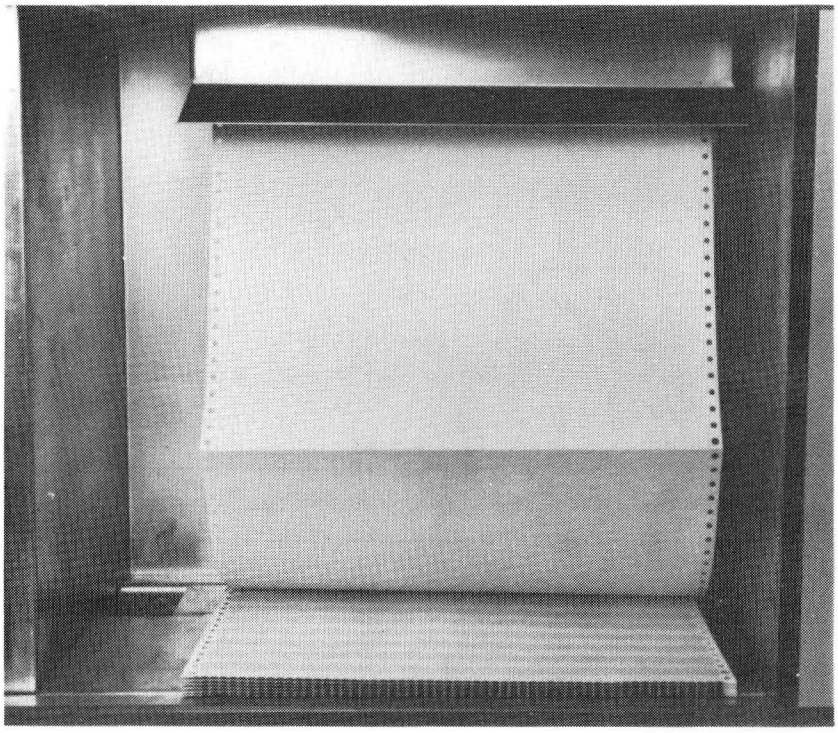


Figure 6–5. Paper Supplied From Input Hopper

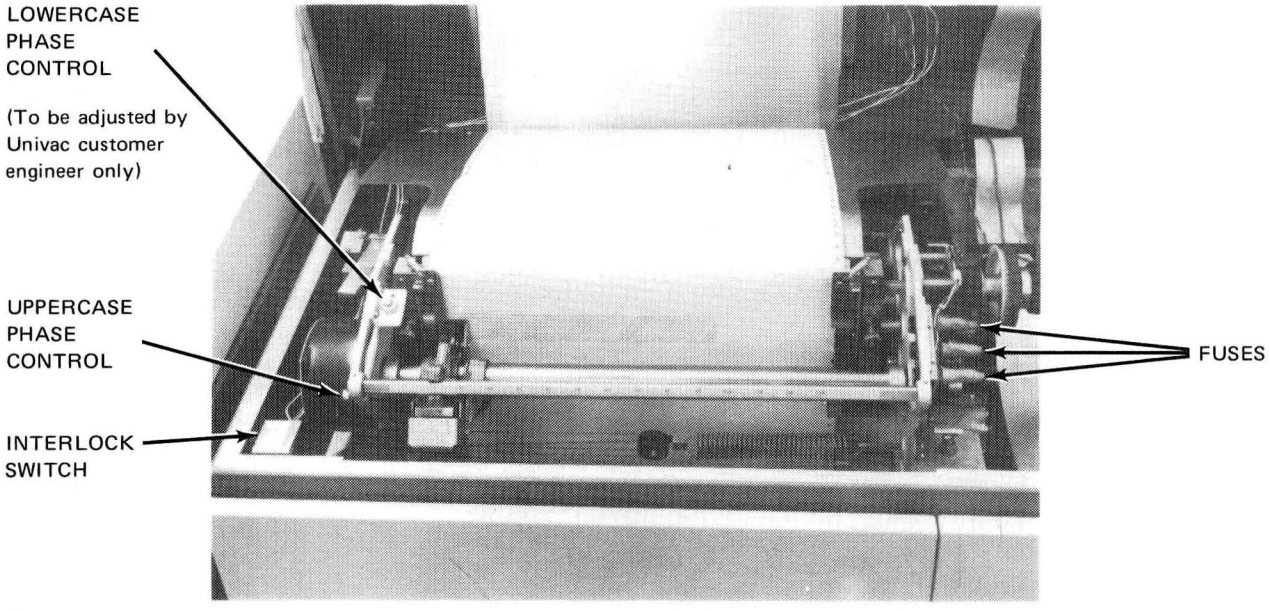


Figure 6–6. Paper Loaded in Printer

6.3.5. Requirements for Printed Forms

Using continuous printed forms that comply with standard requirements assures optimum print quality. Continuous printing forms may be pre-printed or blank, and may be in single part (one original) or multipart (original with carbon copies). The forms must contain sprocket holes in both margins (Figure 6–7), and meet the requirements specified in Tables 6–4, 6–5, and 6–6.

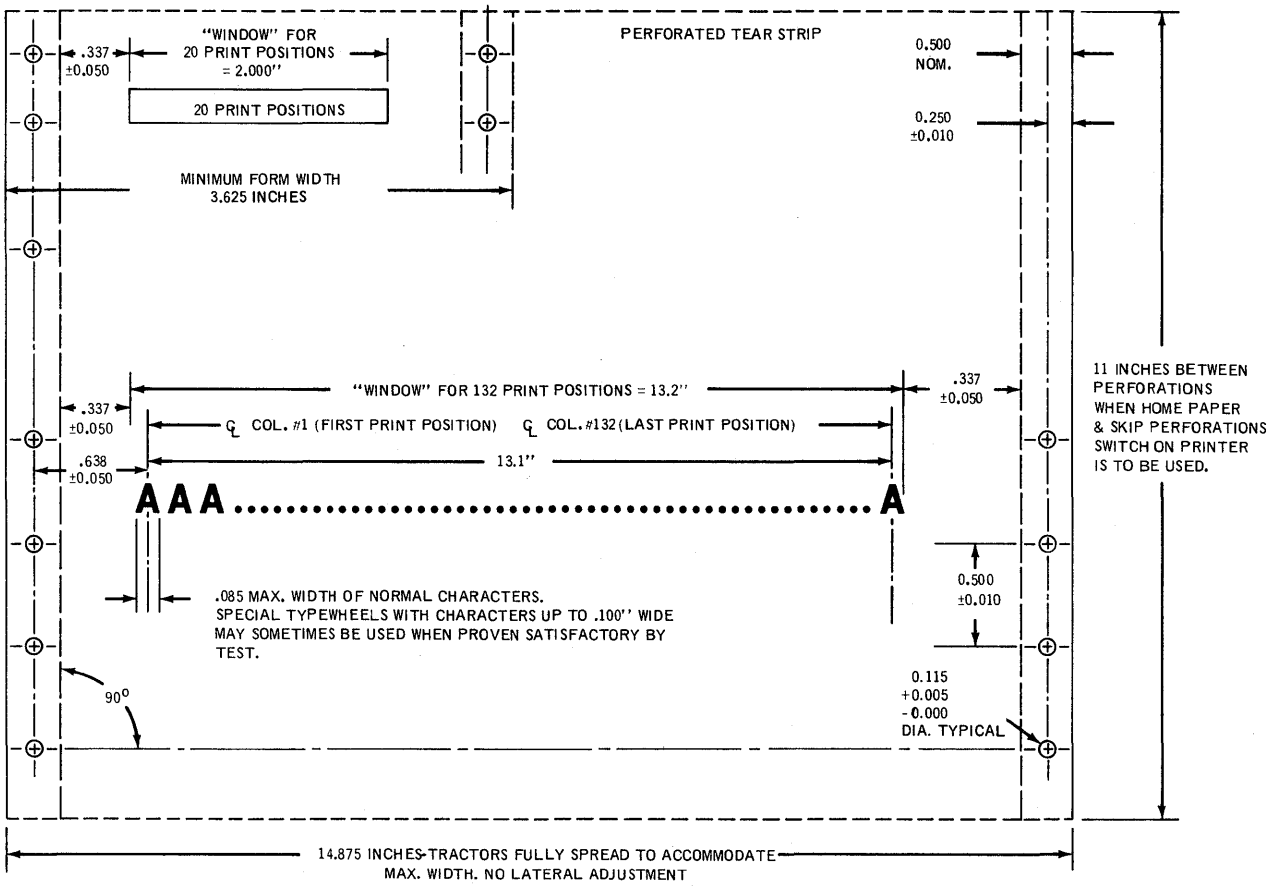


Figure 6–7. Continuous Form Design

Condition	Requirements
Form Design	Certain requirements are essential in design of continuous forms: ■ Forms with sprocket holes on both ends may be from 3.625 inches minimum to 14.875 inches maximum width. Printing dimensions within these limitations are illustrated in Figure 6–7. ■ Continuous forms may be up to 11 inches long between perforations. ■ Sprocket hole diameter is 0.155 inch (+0.005,–0.000). Holes are located 0.250 inch (±0.010) from each vertical edge of the paper, and are spaced vertically 0.500 inch (±0.010). The horizontal centerline of opposite sprocket holes is at a 90 degree angle to the vertical centerline of sprocket holes (Figure 6–7). ■ Vertical lines may be placed between adjacent characters, and characters may touch the lines because of normal shrinkage or expansion of paper. ■ Horizontal lines should be separated by multiples of 0.167 inch because print spacing is at six lines per inch.
Paper Stock and Weight	Paper may be in single or multipart forms: ■ Single-part forms require 20-pound bond mimimum weight for best print quality; however, ledger and other papers are suitable. Slightly reduced print quality is obtained with 16-pound paper; but any standard paper must not exceed 0.0155-inch thickness. ■ Multipart forms may be no thicker than 0.0155 inch to assure clear print. Pack thickness of up to 0.020 inch can be handled, but with reduced print quality. ■ The recommended maximum number of parts is one original and five carbons. The heaviest form should always be the last part of the pack. Combinations of bond paper, ledger, index bristol, tabulating cards, or postcard stock may be used in packs, provided the maximum pack thickness does not exceed 0.0155 inch and the heaviest form is the last part of the pack. Table 6–5 lists the thickness and weight of various paper types as a guide in selection of form paper.
Duplicating Forms	Continuous forms may be used as masters for various duplicating processes such as heat transfer, spirit master, and stencil. Tests should first be made to determine print and reproduction quality before extensive use. Additional considerations are: ■ Heat transfer duplication masters and carbon papers should be kept apart, then collated as they are fed into the printer. Soft carbon paper is recommended, but carbon which is too soft produces indistinct characters. ■ Packing, handling, and softness considerations of spirit masters should be handled the same as heat transfer duplicators. ■ Soft stencil stock is recommended.
NCR Paper	When using National Cash Register (NCR) paper, no more than four parts of NCR multipart form should be used. Each part should be of 15-pound stock weight.
Multipart Form Fasteners	The printer can handle most types of fasteners; including glued margin, bump, sewn, crimped, and partially punched sprocket holes (such as in Standard Register STANLOCK*). Staples on forms must be restricted to the margin and must be tested to assure satisfactory performance. The tractor area clearance of 0.500 inch (nominal) cannot be varied (Figure 6–7).

Table 6–4. Continuous Form Requirements (Part 1 of 2)

Condition	Requirements
Side and Interform Perforations	Considerations that should be made if perforations are present on forms are: ■ Side and interform perforations having extremely long slit perforations or cut areas, as compared to uncut areas, should not be used. These perforations can cause snags in the printer carriage when a form feed or advance from command is presented to the printer. ■ Single or two-part forms should have a closely spaced or needle design having minute round punched holes or dashed line stitches approximately 1/16 inch cut, and approximately 1/32 inch uncut. ■ The ratio of cut and uncut areas increases on multipart forms as the number of parts increase. ■ Refer to Table 6–6 for recommended side and form-to-form perforations in proportion to the number of parts required in multipart forms.
Stock Forms	Single or multicopy stock forms with horizontal lines, strips, or bands of tinted colors can be used when supplied by venders of continuous forms. Horizontal markings facilitate eye movement across a line.
Care of Continuous Forms	Recommendations on care and handling of continuous printing forms: ■ Forms should be packaged in one continuous string within a carton to reduce setup time. If a break occurs in this continuity, insert a marker at that point and note on the container that a break is present. ■ Container size should be: — easy to handle; — within the width and height dimensions of the hopper input area in front of the printer; — less than 20 inches high. ■ Strong shipping containers should be used to eliminate damage during shipment. Forms can be fed directly from the shipping containers if the top of the carton is cut away cleanly. ■ Purchase orders for continuous forms to nationally known venders should include the statement: "To be used on UNIVAC Incremental Printer". These venders are familiar with form construction requirements for this equipment. ■ Storage of continuous forms should be in containers and in an environment like that in which they are used. If storage must be in cold or damp areas, the forms should be conditioned to the operating environment in open containers for at least 12 hours prior to use. ■ Carbons should not be placed in areas having a temperature above 100 degrees F for an extended period of time, unless noted otherwise by the vender.

*Registered trademark of Standard Register.

Table 6–4. Continuous Form Requirements (Part 2 of 2)

Type	Paper Weight Basis (inches)	Density (lbs/cu. in)	Parameter*						
Bond	17 by 22	0.0267	Weight (lbs.)	9	11	13	16	20	24	28
			Thickness (mils)	1.8	2.2	2.6	3.2	4.0	4.8	5.6
Ledger	17 by 22	0.0242	Weight (lbs.)	24	28	32	36	40	44	–
			Thickness (mils)	5.6	6.1	7.0	7.9	8.9	9.6	–
Index Bristol	25½ by 30½	0.0333	Weight (lbs.)	90	110	140	170	220	–	–
			Thickness (mils)	7.0	8.5	11.0	13.0	17.0	–	–
Tabulating Card	24 by 36	0.0335	Weight (lbs.)	101± 5%	–	–	–	–	–	–
			Thickness (mils)	7.0± 0.4	–	–	–	–	–	–
Post Card	24 by 36	0.0322	Weight (lbs.)	125± 5%	–	–	–	–	–	–
			Thickness (mils)	9.0± 0.5	–	–	–	–	–	–
Carbon Paper	20 by 30		Weight (lbs.)	5.5	6.5	8	9	10	–	–
			Thickness (mils)	0.6	0.7	1.0	1.1	1.2	–	–

*The standard weights and thicknesses listed are based on a ream (500 sheets).

Table 6–5. Paper Thickness and Weight

Form Type	Side Perforations			Form-To-Form Perforations		
	Maximum Cut (inches)	Minimum Uncut (inches)	Ratio	Maximum Cut (inches)	Minimum Uncut (inches)	Ratio
Single-Part	1/8	1/32	4:1	1/16	1/32	2:1
Two-Part	1/8	1/32	4:1	1/16	1/32	2:1
Three-Part	3/16	1/32	6:1	1/8	1/32	4:1
Four-Part	1/4	1/32	8:1	1/8	1/32	4:1
Five- or Six-Part	5/16	1/32	10:1	3/16	1/32	6:1

NOTE:

If the printing format requires printing 1/2 inch above or below the perforation, closely spaced or needle type performation should be specified for all parts of the continuous form. Printing close to the perforation can cause partial breaking which may result in jamming or separation during feeding.

Table 6–6. Cut Versus Uncut Dimensions on Forms

6.3.6. Character Codes

The printer recognized the UNIVAC modified ASCII character set. Codes are derived from a seven-bit (plus parity bit) defined in Figure 6–8. The left font on the print wheel contains 63 characters, including all uppercase characters. The right font on the print wheel contains 31 characters, including all lowercase characters. The remaining character spaces on the left font (positions 64 to 72) are all diamond characters. On the right font, the first 31 positions are vertical-line characters, and positions 63 to 72 are diamond characters (Table 6–3).

The LF, FF, CR, and SP characters are used by the operator console to control printer operations, and they are not printable.

		Control Characters		Graphic Characters					
	Col.	0	1	2	3	4	5	6	7
Row	Bits	000	001	010	011	100	101	110	111
0	0000			SP	0	@	P	`	p
1	0001			!	1	A	Q	a	q
2	0010			"	2	B	R	b	r
3	0011			#	3	C	S	c	s
4	0100			$	4	D	T	d	t
5	0101			%	5	E	U	e	u
6	0110			&	6	F	V	f	v
7	0111			'	7	G	W	g	w
8	1000			(	8	H	X	h	x
9	1001			)	9	I	Y	i	y
10	1010	LF		*	:	J	Z	j	z
11	1011			+	;	K	[	k	{
12	1100	FF		,	<	L	\	l	|
13	1101	CR		–	=	M	]	m	}
14	1110			.	>	N	^	n	~
15	1111			/	?	O	—	o	DEL

63 Characters (Uppercase—Left Font) | 31 Characters (Lowercase—Right Font)

Example:

100	0001

= A

b_7.....b_1

NOTE: Shaded characters are nonprintable.

Figure 6–8. ASCII-Based Code Used on Printer

6.3.7. Character Sequence

Information supplied by the operator console causes the print wheel carriage to move to the left margin and start printing towards the right, across the paper. A CR character in text causes the next character to be printed at the beginning of the same line. An LF character moves the form up one line. If the end of the line is reached during printing, and a CR or LF character is not supplied, the next character will print at the beginning of the next line.

An FF character in text causes the next character to print at the beginning of the first line of the next form. The CR character performs only a carriage return function, just as the LF character performs only the line feed function; therefore, to move the form one line and begin printing at the left margin, both the CR and the LF characters must be supplied to the printer.

The bit sequence for an ASCII character (Figure 6–8) is from the least significant bit (bit 1) to the most significant bit (bit 7). Characters are supplied serially after the printer is addressed by the operator console. During reception of characters, the printer control circuits are in a busy condition because input characters cannot be stored in the printer.

6.4. CONTROLS AND INDICATORS

The printer contains an operator panel located at the top right side of the printer, a power control panel located inside the printer on the front right side, and a 48-volt power supply located inside the rear of the cabinet.

6.4.1. Operator Control Panel

The operator control panel (Figure 6–9) is located at the top right of the printer and contains the switches and indicators used during the operation. The switches are of a rocker type which cause their associated functions to be activated when the upper portion of the switch is pressed. The indicators are located immediately above the switches, and are visible only when lit. Table 6–7 lists the switches and indicators on the operator's control panel, and a description of each.

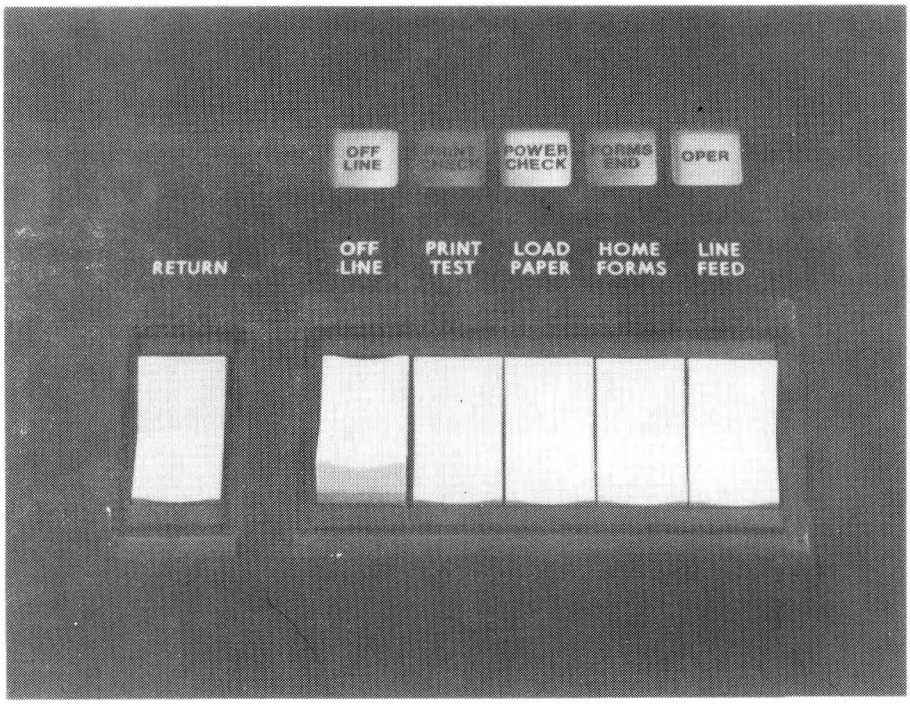


Figure 6–9. Operator Control Panel

Control/Indicator	Function
RETURN switch	This switch can be activated only after OFF LINE switch is set for offline operation. When pressed, causes carriage to return to left margin.
OFF LINE switch	When upper portion of switch is pressed, printer is disabled, placed in an offline condition for maintenance, and OFF LINE indicator lights. When lower portion of switch is pressed, printer is enabled to accept data from processor.
PRINT TEST switch	This switch can be activated only after OFF LINE switch is set for offline operation. When pressed, causes one line of character E to be printed permitting operator to accurately adjust phasing controls.
LOAD switch	This switch can be activated only after OFF LINE switch is set for offline operation. When pressed, causes printer carriage to move away from left tractor, permitting operator to open tractor when loading paper.
HOME FORMS switch	This switch can be activated only after OFF LINE switch is set for offline operation. When pressed, causes printer to feed form to top of next form or home position.
LINE FEED switch	This switch can be activated only after OFF LINE switch is set for offline operation. When pressed causes form to move up one line.
OFF LINE indicator	When lit, indicates that OFF LINE switch has been pressed and printer is disabled for maintenance or adjustments.
PRINT CHECK indicator	When lit, indicates that a fuse is open to: form feed actuator (IF1) left front print actuator (IF2), or right front print actuator (IF3).
POWER CHECK indicator	When lit, indicates that a power failure in the 48-volt power supply has been detected.
FORMS END indicator	When lit, indicates that supply of forms in printer is exhausted.
OPER Indicator	When lit, indicates that top cover of printer is open. Print wheel motor also stops whenever cover is opened.

Table 6–7. Operator Control Panel, Controls and Indicators

6.4.2. Power Control Panel

The power control panel (Figure 6–10) is located behind the right front printer cabinet door. This panel contains circuit breakers and indicators used by the operator to control operating power for the printer. Switches and indicators on the power control panel are listed and described in Table 6–8.

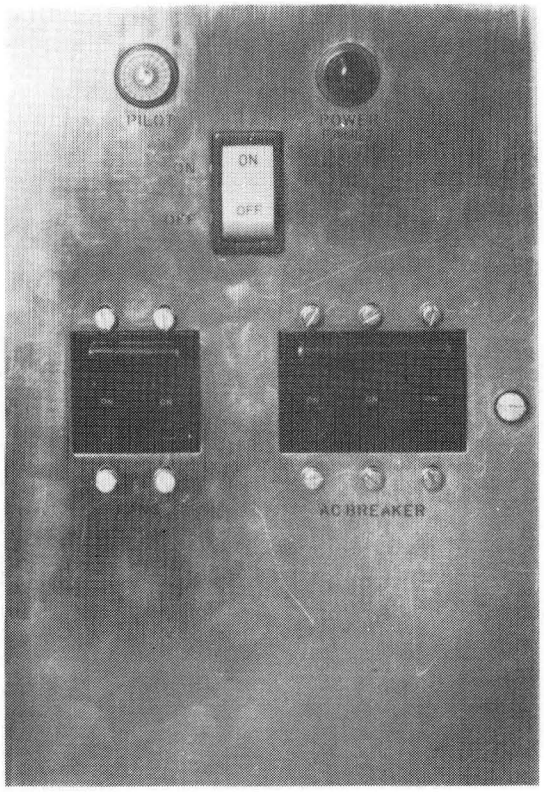


Figure 6–10. Power Control Panel

Control/Indicator	Function
PILOT indicator	When lit, indicates that primary power is turned on at the UNIVAC 9700 System Console (system console) and is available to printer ac circuit breakers.
POWER FAULT indicator	When lit, indicates a power failure in the 48-volt power supply has been detected.
ON/OFF switch	ON – Activates control circuits to console permitting ac primary power to be supplied to printer. If power is turned on at console system, PILOT indicators lights. OFF – Disconnects primary power to printer at source in system console.
FANS circuit breaker	Provides protection and power to cooling fans within printer cabinet. When breaker is set to OFF or is tripped off due to a current overload, PRINTER EARLY WARNING indicator lights on the power control panel at the system console and sonalert sounds.

Table 6–8. Power Control Panel, Controls and Indicators

6.4.3. Power Supply

A 48-volt power supply is located behind the rear left printer cabinet door. This supply contains a single circuit breaker to control 48 volts to the printer switching and control circuits. When a power fault is detected, the 48-volt power supply circuit breaker should be checked to determine whether it has tripped due to the fault.

APPENDIX A. INSTRUCTIONS

The UNIVAC 9700 Processor is equipped with 154 instructions, each having a unique operational code (op code). All 154 instructions are used in the normal mode while 149 instructions are used in the special compatibility mode. These instructions are divided into five instruction-type formats.

RR, RX, RS, SI, and SS

Figure A–1 shows the basic UNIVAC 9700 System instruction formats (object code form).

Table A–1 lists the symbols used to describe op code formats.

Table A–2 lists the instructions alphabetically according to instruction name.

Table A–3 lists the instructions according to the hexadecimal op code.

Table A–4 lists the instruction execution times.

Table A–5 is a legend for Table A–4.

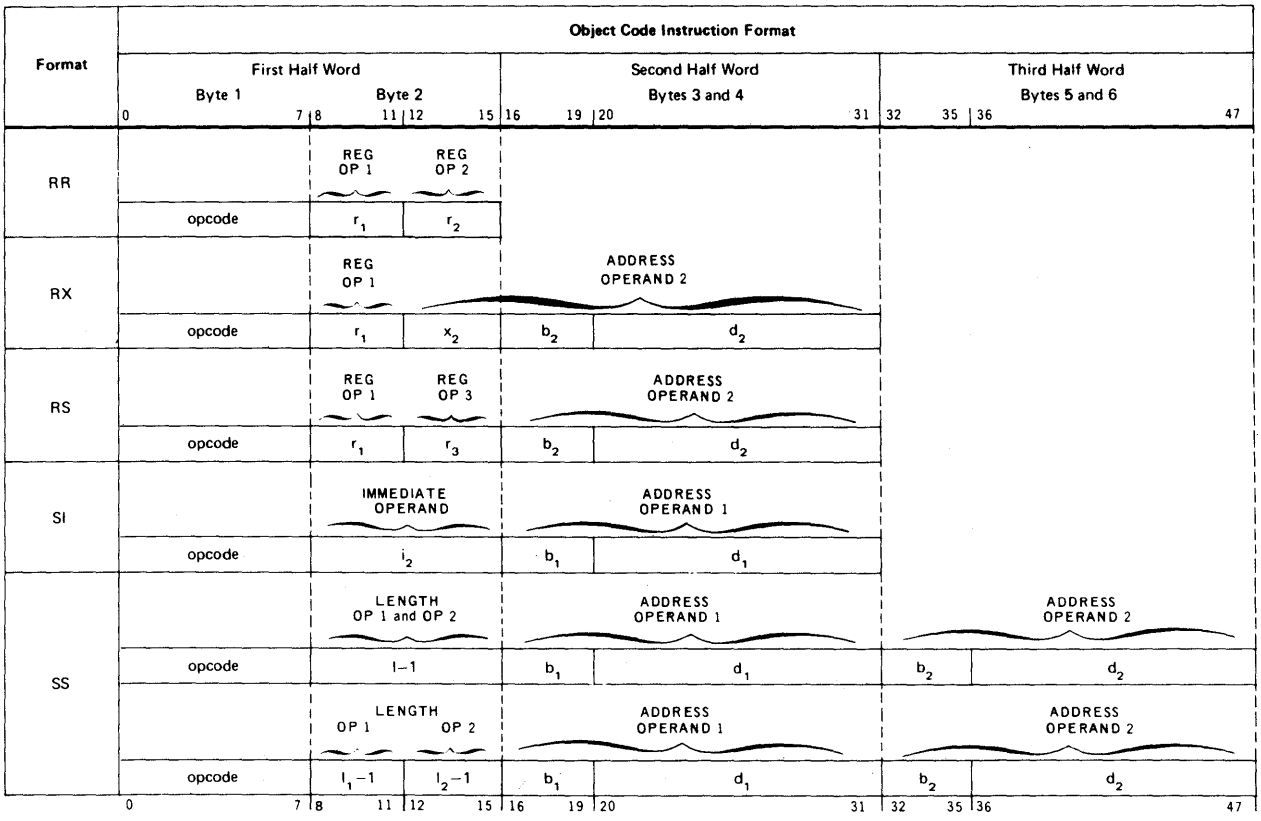


Figure A–1. Basic Instruction Formats (Object Code Form)

Symbol	Meaning
opcode	Instruction operation code
r_1	Number of the register addressed as operand 1, a mask, or a register which is the first register of a multiregister group
r_2	Number of the register addressed as operand 2
r_3	An expression representing a register which is the last register in a multiregister group, an increment, an operand address, or a control storage address
x_2	Number of the register to be used as an index for operand 2 of an RX instruction
i_2	Immediate data used as operand 2 of an SI instruction
l	Length of operands 1 and 2 as stated in source code
l_1	Length of operand 1 as stated in source code
l_2	Length of operand 2 as stated in source code
b_1	Base register for operand 1
b_2	Base register for operand 2
d_1	Displacement for operand 1
d_2	Displacement for operand 2
OP 1	Operand 1
OP 2	Operand 2
OP 3	Operand 3

Table A–1. Symbols Used to Describe Op Code Formats

Instruction	Op Code	Mnemonic
Add	1A	AR
Add	5A	A
Add Decimal	FA	AP
Add Half Word	4A	AH
Add Immediate	9A	AI
Add Logical	1E	ALR
Add Logical	5E	AL
Add Normalized (Long)	2A	(F)ADR
Add Normalized (Long)	6A	(F)AD
Add Normalized (Short)	3A	(F)AER

Instruction	Op Code	Mnemonic
Add Normalized (Short)	7A	(F)AE
Add Unnormalized (Long)	2E	(F)AWR
Add Unnormalized (Short)	3E	(F)AUR
Add Unnormalized (Long)	6E	(F)AW
Add Unnormalized (Short)	7E	(F)AU
And	14	NR
And	54	N
And	94	NI
And	D4	NC

NOTE:

Tag symbol (F) before mnemonic indicates instructions that are added as features.

Table A–2. Alphabetical List of Instructions (Part 1 of 4)

Instruction	Op Code	Mnemonic
Branch and Link	05	BALR
Branch and Link	45	BAL
Branch and Link External	4D	BALE
Branch on Condition	07	BCR
Branch on Condition Return External	0C	BCRE
Branch on Condition	47	BC
Branch on Count	06	BCTR
Branch on Count	46	BCT
Branch on Index High	86	BXH
Branch on Index Low or Equal	87	BXLE
Compare	19	CR
Compare	59	C
Compare Decimal	F9	CP
Compare Half Word	49	CH
Compare Logical	15	CLR
Compare Logical	55	CL
Compare Logical	95	CLI
Compare Logical	D5	CLC
Compare (Long)	29	(F)CDR
Compare (Long)	69	(F)CD
Compare (Short)	39	(F)CER
Compare (Short)	79	(F)CE
Convert to Binary	4F	CVB
Convert to Decimal	4E	CVD
Diagnose - Privileged	83	DIAG
Divide	1D	DR
Divide	5D	D
Divide Decimal	FD	DP
Divide (Long)	2D	(F)DDR
Divide (Long)	6D	(F)DD
Divide (Short)	3D	(F)DER
Divide (Short)	7D	(F)DE
Edit	DE	ED
Edit and Mark	DF	EDMK
Emulation Aid	EA	EA
Exclusive OR	17	XR
Exclusive OR	57	X
Exclusive OR	97	XI
Exclusive OR	D7	XC
Execute	44	EX
Halt and Proceed - Privileged	99	HPR
Halt I/O - Privileged	9E	HIO
Halve (Long)	24	(F)HDR
Halve (Short)	34	(F)HER
Insert Character	43	IC
Insert Storage Key - Privileged	09	ISK
Load	18	LR
Load	58	L
Load Address	41	LA
Load and Test	12	LTR
Load and Test (Long)	22	(F)LTDR
Load and Test (Short)	32	(F)LTER
Load Base and Relativize	0B	LBR
Load Channel Register - Privileged	AD	LCHR
Load Complement	13	LCR
Load Complement (Long)	23	(F)LCDR
Load Complement (Short)	33	(F)LCER
Load Control Storage - Privileged	B1	LCS
Load Half Word	48	LH
Load (Long)	28	(F)LDR
Load (Long)	68	(F)LD

NOTE:

Tage symbol (F) before mnemonic indicates instructions that are added as features.

Table A–2. Alphabetical List of Instructions (Part 2 of 4)

Instruction	Op Code	Mnemonic
Load Multiple	98	LM
Load Negative	11	LNR
Load Negative (Long)	21	(F)LNDR
Load Negative (Short)	31	(F)LNER
Load Positive	10	LPR
Load Positive (Long)	20	(F)LPDR
Load Positive (Short)	30	(F)LPER
Load PSW - Privileged	82	LPSW
Load (Short)	38	(F)LER
Load (Short)	78	(F)LE
Move	D2	MVC
Move	92	MVI
Move Numerics	D1	MVN
Move With Offset	F1	MVO
Move Zones	D3	MVZ
Multiply	1C	MR
Multiply	5C	M
Multiply Decimal	FC	MP
Multiply Half Word	4C	MH
Multiply (Long)	2C	(F)MDR
Multiply (Long)	6C	(F)MD
Multiply (Short)	3C	(F)MER
Multiply (Long)	6C	(F)MD
Multiply (Short)	3C	(F)MER
Multiply (Short)	7C	(F)ME
OR	16	OR
OR	56	O
OR	96	OI
OR	D6	OC
Pack	F2	PACK
Read Direct - Privileged	85	(F)RDP
Set Program Mask	04	SPM

Instruction	Op Code	Mnemonic
Set Storage Key - Privileged	08	SSK
Set System Mask - Privileged	80	SSM
Shift Left Double Logical	8D	SLDL
Shift Left Double	8F	SLDA
Shift Left Single Logical	89	SLL
Shift Left Single	8B	SLA
Shift Right Double Logical	8C	SRDL
Shift Right Double	8E	SRDA
Shift Right Single Logical	88	SRL
Shift Right Single	8A	SRA
Start I/O - Privileged	9C	SIO
Store	50	ST
Store Channel Register - Privileged	AC	SCHR
Store Character	42	STC
Store Half Word	40	STH
Store (Long)	60	(F)STD
Store (Short)	70	(F)STE
Store Multiple	90	STM
Subtract	1B	SR
Subtract	5B	S
Subtract Decimal	FB	SP
Subtract Half Word	4B	SH
Subtract Logical	1F	SLR
Subtract Logical	5F	SL
Subtract Normalized (Long)	2B	(F)SDR
Subtract Normalized (Short)	3B	(F)SER
Subtract Normalized (Long)	6B	(F)SD
Subtract Normalized (Short)	7B	(F)SE
Subtract Unnormalized (Long)	2F	(F)SWR
Subtract Unnormalized (Short)	3F	(F)SUR
Subtract Unnormalized (Long)	6F	(F)SW

NOTE:

Tag symbol (F) before mnemonic indicates instructions that are added as features.

Table A–2. Alphabetical List of Instructions (Part 3 of 4)

Instruction	Op Code	Mnemonic
Subtract Unnormalized (Short)	7F	(F)SU
Supervisor Call	0A	SVC
Supervisor Load Multiple - Privileged	B8	SLM
Supervisor Store Multiple - Privileged	B0	SSTM
Test and Set	93	TS
Test Channel - Privileged	9F	TCH
Test I/O - Privileged	9D	TIO
Test Under Mask	91	TM
Translate	DC	TR
Translate and Test	DD	TRT
Unpack	F3	UNPK
Write Direct - Privileged	84	(F)WRD
Zero and Add	F8	ZAP

NOTE:

Tag symbol (F) before mnemonic indicates instructions that are added as features.

Table A–2. Alphabetical List of Instructions (Part 4 of 4)

Op Code	Mnemonic	Instruction
04	SPM	Set Program Mask
05	BALR	Branch and Link
06	BCTR	Branch on Count
07	BCR	Branch on Condition
08	SSK	Set Storage Key - Privileged
09	ISK	Insert Storage Key - Privileged
0A	SVC	Supervisor Call
0B	LBR	Load Base and Relativize
0C	BCRE	Branch on Condition to Return External
10	LPR	Load Positive
11	LNR	Load Negative
12	LTR	Load and Test
13	LCR	Load Complement
14	NR	AND
15	CLR	Compare Logical
16	OR	OR
17	XR	Exclusive OR
18	LR	Load
19	CR	Compare
1A	AR	Add
1B	SR	Subtract
1C	MR	Multiply
1D	DR	Divide
1E	ALR	Add Logical
1F	SLR	Subtract Logical
20	(F)LPDR	Load Positive (Long)
21	(F)LNDR	Load Negative (Long)
22	(F)LTDR	Load and Test (Long)
23	(F)LCDR	Load Complement (Long)
24	(F)HDR	Halve (Long)

NOTE:

Tag symbol (F) before mnemonic indicates instructions that are added as features.

Table A–3. List of Instructions by Op Code (Part 1 of 3)

Op Code	Mnemonic	Instruction
28	(F)LDR	Load (Long)
29	(F)CDR	Compare (Long)
2A	(F)ADR	Add Normalized (Long)
2B	(F)SDR	Subtract Normalized (Long)
2C	(F)MDR	Multiply (Long)
2D	(F)DDR	Divide (Long)
2E	(F)AWR	Add Unnormalized (Long)
2F	(F)SWR	Subtract Unnormalized (Long)
30	(F)LPER	Load Positive (Short)
31	(F)LNER	Load Negative (Short)
32	(F)LTER	Load and Test (Short)
33	(F)LCER	Load Complement (Short)
34	(F)HER	Halve (Short)
38	(F)LER	Load (Short)
39	(F)CER	Compare (Short)
3A	(F)AER	Add Normalized (Short)
3B	(F)SER	Subtract Normalized (Short)
3C	(F)MER	Multiply (Short)
3D	(F)DER	Divide (Short)
3E	(F)AUR	Add Unnormalized (Short)
3F	(F)SUR	Subtract Unnormalized (Short)
40	STH	Store Half Word
41	LA	Load Address
42	STC	Store Character
43	IC	Insert Character
44	EX	Execute
45	BAL	Branch and Link
46	BCT	Branch on Count
47	BC	Branch on Condition
48	LH	Load Half Word
49	CH	Compare Half Word

Op Code	Mnemonic	Instruction
4A	AH	Add Half Word
4B	SH	Subtract Half Word
4C	MH	Multiply Half Word
4D	BALE	Branch and Link External
4E	CVD	Convert to Decimal
4F	CVB	Convert to Binary
50	ST	Store
54	N	AND
55	CL	Compare Logical
56	O	OR
57	X	Exclusive OR
58	L	Load
59	C	Compare
5A	A	Add
5B	S	Subtract
5C	M	Multiply
5D	D	Divide
5E	AL	Add Logical
5F	SL	Subtract Logical
60	(F)STD	Store (Long)
68	(F)LD	Load (Long)
69	(F)CD	Compare (Long)
6A	(F)AD	Add Normalized (Long)
6B	(F)SD	Subtract Normalized (Long)
6C	(F)MD	Multiply (Long)
6D	(F)DD	Divide (Long)
6E	(F)AW	Add Unnormalized (Long)
6F	(F)SW	Subtract Unnormalized (Long)
70	(F)STE	Store (Short)
78	(F)LE	Load (Short)
79	(F)CE	Compare (Short)

NOTE:

Tag symbol (F) before mnemonic indicates instructions that are added as features.

Table A–3. List of Instructions by Op Code (Part 2 of 3)

Op Code	Mnemonic	Instruction
7A	(F)AE	Add Normalized (Short)
7B	(F)SE	Subtract Normalized (Short)
7C	(F)ME	Multiply (Short)
7D	(F)DE	Divide (Short)
7E	(F)AU	Add Unnormalized (Short)
7F	(F)SU	Subtract Unnormalized (Short)
80	SSM	Set System Mask - Privileged
82	LPSW	Load PSW - Privileged
83	DIAG	Diagnose - Privileged
84	(F)WRD	Write Direct - Privileged
85	(F)RDD	Read Direct - Privileged
86	BXH	Branch on Index High
87	BXLE	Branch on Index Low or Equal
88	SRL	Shift Right Single Logical
89	SLL	Shift Left Single Logical
8A	SRA	Shift Right Single
8B	SLA	Shift Left Single
8C	SRDL	Shift Right Double Logical
8D	SLDL	Shift Left Double Logical
8E	SRDA	Shift Right Double
8F	SLDA	Shift Left Double
90	STM	Store Multiple
91	TM	Test Under Mask
92	MVI	Move Immediate
93	TS	Test and Set
94	NI	AND
95	CLI	Compare Logical
96	OI	OR
97	XI	Exclusive OR
98	LM	Load Multiple
99	HPR	Halt and Proceed - Privileged
9A	AI	Add Immediate

Op Code	Mnemonic	Instruction
9C	SIO	Start I/O - Privileged
9D	TIO	Test I/O - Privileged
9E	HIO	Halt I/O - Privileged
9F	TCH	Test Channel - Privileged
AC	SCHR	Store Channel Register - Privileged
AD	LCHR	Load Channel Register - Privileged
B0	SSTM	Supervisor Store Multiple - Privileged
B1	LCS	Load Control Storage - Privileged
B8	SLM	Supervisor Load Multiple - Privileged
D1	MVN	Move Numerics
D2	MVC	Move
D3	MVZ	Move Zones
D4	NC	AND
D5	CLC	Compare Logical
D6	OC	OR
D7	XC	Exclusive OR
DC	TR	Translate
DD	TRT	Translate and Test
DE	ED	Edit
DF	EDMK	Edit and Mark
EA	EA	Emulation Aid
F1	MVO	Move with Offset
F2	PACK	Pack
F3	UNPK	Unpack
F8	ZAP	Zero and Add
F9	CP	Compare Decimal
FA	AP	Add Decimal
FB	SP	Subtract Decimal
FC	MP	Multiply Decimal
FD	DP	Divide Decimal

NOTE:

Tag symbol (F) before mnemonic indicates instructions that are added as features.

Table A–3. List of Instructions by Op Code (Part 3 of 3)

Mnemonic	Time in Microseconds*
A	1.5
AD	5.64+.3[ce+pn(m+2n+r)+2a+2rp]
ADR	5.64+.3[ce+pn(m+2n+r)+2a+2rp]
AE	4.3+.3[ce+pn(m+2n+r)+2a+2rp]
AER	4.3+.3[ce+pn(m+2n+r)+2a+2rp]
AH	1.8+.3ms
AI	2.7
AL	1.5
ALR	.9
AP	7.5+1.5(2w1+w2)+1.2r(w1+1)
AR	.9
AU	4.3+.3(ce+r+2a)
AUR	4.3+.3(ce+r+2a)
AW	5.64+.3(ce+r+2a)
AWR	5.64+.3(ce+r+2a)
BAL	1.2+.3hw
BALE	2.1+b(2.1+x)+3hw
BALR	1.2+.3hw
BC	1.2+.3(hw+nb)
BCR	1.2+.3hw
BCRE	2.4+.3(hw—nb)
BCT	1.2+.3(hw+3nb)
BCTR	1.2+.3(hw+3nb)
BXH	1.8+.3(hw+4nb)
BXLE	1.8+3(hw+4nb)
C	.9
CD	3.3+fz1.95+.ce+1.5sc
CDR	3.0+fz1.95+.3ce+.9sc
CE	2.7fz1.95+.3ce+.9sc
CER	2.4+fz1.95+.3ce+.9sc
CH	1.8+.3ns
CL	1.5
CLC	2.7+2.7wz+.9bn
CLI	1.8
CLR	.9
CP	6.9+2.4w1+1.5w2
CVB	.9+1.8d1+.6d2+.3z
CVD	6.45+3.6h1+4.2h2+.3z
CR	1.5
D	27.0
DD	72.9+.3(p1+p2)
DDR	73.2+.3(p1+p2)
DE	27.1+.3(p1+p2)
DER	27.1+.3(p1+p2)
DIAG	1.8
DP	3.6+2.1w1+1.2w2+12.0(n1—n2)
DR	27.0
ED	6+2.1w1+1.5w2+1.8no+2.1fs+2.4(ds+ss)
EDMK	.6+2.1w1+1.5w2+1.8no+2.1fs+2.4(ds+ss)+1.2mk
EX	2.25+e

Mnemonic	Time in Microseconds*
HDR	3.0+.3n+.3np
HER	3.0+.3n+.3np
HPR	1.5+hd
HIO	2.7+cu
IC	1.5
ISK	1.8
L	1.5
LA	1.8
LBR	2.1
LCDR	1.5
LCER	1.5
LCHR	2.7+cu
LCR	.9
LCS	2.1+2.4w(1+om)
LD	2.1
LDR	1.5
LE	1.5
LER	.9
LH	1.8
LM	2.4+.6gr
LNDR	1.5
LNER	.9
LNR	1.5
LPDR	1.5
LPER	.9
LPR	1.5
LPSW	3.9+.3hw
LR	.9
LTDR	1.5
LTER	.9
LTR	.9
M	12.9
MD	37.6+.3(p1+p2+pn)
MDR	38.6+.3(p1+p2+pn)
ME	12.7+.3(p1+p2+pn)
MER	12.4+3(p1+p2+pn)
MH	9.15
MP	15.3+2.7w1+.6w2+21.6(n2—1)
MR	12.9
MVC	2.7+1.8w1+.9bn
MVI	1.8
MVN	2.7+2.7w1+.9bn
MVO	3.3+1.5w1+.9w2
MVZ	2.7+2.7w1+.9bn
N	1.5
NC	2.7+2.7w1+.9bn
NI	2.4
NR	.9
O	1.5
OC	2.7+2.7w1+.9bn
OI	2.4
OR	.9

*See Table A–5 for legend.

Table A–4. Instruction Execution Times (Part 1 of 2)

Mnemonic	Time in Microseconds*
PACK	1.5+1.2n1+.9w1+2.4w2
RDD	3.0+v
S	1.5
SCHR	2.7+cu
SD	5.64+.3[ce+pn(m+2n+r)+2a+2rp]
SDR	5.64+.3[ce+pn(m+2n+r)+2a+2rp]
SE	4.3+.3[ce+pn(m+2n+r)+2a+2rp]
SER	4.3+.3[ce+pn(m+2n+r)+2a+2rp]
SH	1.8+
SIO	2.7+cu
SL	1.5
SLA	2.1+.6(1+k1)k3
SLDA	2.7+.6k1+.3k2+.3k2
SLDL	2.7+.6k1+.3k2
SLL	2.1+.6(1+k1)k3
SLM	2.7+.6gr
SLR	.9
SP	7.5+1.5(2w1+w2)+1.2r (w1+1)
SPM	1.5
SR	.9
SRA	2.1+.6 (1+k1) k3
SRDA	2.7+6k1+.6k2
SRDL	2.7+6k1+.6k2
SRL	2.1+.6(1+k1)k3
SSK	1.8
SSM	2.1+.3hb

Mnemonic	Time in Microseconds*
SSTM	2.4+.6gr
ST	1.8
STC	2.1
STD	2.4
STE	1.8
STH	2.1
STM	2.1+.6gr
SU	4.3+.3(ce+r+2a)
SUR	4.3+.3(ce+r+2a)
SVC	.3
SW	5.64+.3(ce+r+2a)
SWR	5.64+.3(ce+r+2a)
TCH	2.7+cu
TIO	2.7+cu
TM	1.8
TR	3.0+2.1w1+1.8n1
TRT	3.0+2.1wz+2.4nzi+1.8nz
TS	2.1
UNPK	1.5+.9n1+.9w1+2.4w2
WRD	2.4
X	1.5
XC	2.7+2.7w1+.9bn
XI	2.4
XR	.9
ZAP	6.0+1.5(2w1+w2)

*See Table A–5 for legend.

Table A–4. Instruction Execution Times (Part 2 of 2)

Symbol	Definition	Symbol	Definition
a	1 If fraction overflow adjustment is necessary; otherwise 0.	m	1 If the most significant hexadecimal digit of the fraction is 0.
b	1 If b2 designator is not 0; otherwise 0.	mk	Number of times the mark function is performed.
bn	1 If the two low order address bits of operands 1 and 2 are unequal; otherwise 0.	n	1 If 2 or more most significant hexadecimal digits or the fraction is 0.
ce	Number of digit shifts (4 bits each) required to equalize characteristics.	n1	Number of operand 1 bytes.
		n2	Number of operand 2 bytes.
cu	Channel and control unit delay.	nb	1 If branch is unsuccessful; otherwise 0.
ds	Number of digit select characters in the pattern.	no	Number of pattern bytes other than fs, ds, or ss.
d1	Number of significant decimal digits in the operand (not including high order 0's and sign).	np	1 If postnormalization is unnecessary; otherwise 0.
		ns	1 If sign of operand 2 is negative; otherwise 0.
d2	1 If d1≤7; otherwise 0.	nz	Number of operand 1 bytes processed.
e	Execution time of subject instruction.	nzi	1 If result byte is nonzero; otherwise 0.
fs	Number of field separator characters in the pattern.	om	1 If OM control storage; 0 if AC control storage.
fz	1 If operand 1 and 2 not equal to 0.	p1	Number of digit shifts (4 bits each) required to prenormalize the first operand.
gr	Number of general registers loaded or stored.		
h1	Number of significant hexadecimal digits in the operand (not including high order 0's).	p2	Number of digit shifts (4 bits each) required to prenormalize the second operand.
h2	1 If H1≤6; otherwise 0.	pn	1 If result requires postnormalization; otherwise 0.
hd	Delay until RUN switch is depressed.	r	1 If recomplementing is required; otherwise 0.
hb	1 If the instruction is on a half-word boundary; otherwise 0.	rp	1 If recomplementing without postnormalization; otherwise 0.
hw	3 If the object instruction of the branch instruction is on an odd half-word boundary and the object instruction is an RX, RS, or SI instruction; 1 if the object instruction is an SS instruction; otherwise 0.	sc	Signs and characters are equal.
		ss	Number of significance starter characters in the pattern.
		v	The time until Direct In is valid.
		w	Number of control storage words to be written.
k1	1 If the shift count is 1, 4 or 5.	w1	Number of storage words which contain operand 1.
	2 If the shift count is 2 or 6.	w2	Number of storage words which contain operand 2.
	3 If the shift count is 3, 7, or 11; otherwise the shift count is divided by 4 (remainder is ignored).	wz	Number of leading 0 words plus 1.
		x	1.2 times the number of IACW's in succession (up to 8 IACW's).
k2	1 If the shift count is greater than or equal to 32; otherwise 0.		
k3	1 If the shift count is less than 32; otherwise 0.	z	Number of leading 0 digits in operand to be converted.

Table A–5. Legend for Table A–4

APPENDIX B. GLOSSARY

A

AC
Address Calculation Section

ASM
Assembler

B

BALE
Branch and Link External (Instruction)

BCRE
Branch on Condition to Return External (Instruction)

BDW
Branch Discipline Word

C

CAW
Channel Address Word

CC
Condition Code

CCW
Channel Command Word

CDC
Channel Data Check

CIC
Communications Intelligence Channel

COS
Control Storage

CSW
Channel Status Word

D

DCS
Data Communications Subsystem

H

HCAW
Hard Channel Address Word

HCCW
Hard Channel Command Word

HIO
Halt Input-Output

HSTCW
Hard Status Table Control Word

HTCW
Hard Timer Control Word

I

IACW
Indirect Address Control Word

IGR
Integrated General Register

ILC
Instruction Length Code

IIS
Interrupt Initialization Sequence

ISS
Initial Selection Sequence

ISW
Initial Status Word

L

LCHR
Load Channel Register

LD
Load

LSB
Least Significant Bit

M

MAM
Manipulation Alter Matrix

MSB
Most Significant Bit

O

OM
Operand Manipulation Section

OSSF
Operating System Storage Facility

P

PCI
Program Controlled Interrupt

PSW
Program Status Word

R

RP
Read Protect

RPA
Reverse Positive Load A

RPG
Report Program Generation

RWM
Read/Write Memory

RWS
Read/Write Storage

S

SCHR
Store Channel Register

SRF
Set Relocation Flags

SIAM
Subchannel Idle Alert Mode

SIO
Start Input/Output

SSM
Set System Mask

Status Table CSW
Status Table Channel Status Word

Status Table HCAW
Status Table Hard Channel Address Word

STCW
Status Table Control Word

T

TCH
Test Channel

TCW
Timer Control Word

TIC
Transfer in Channel

Timer CSW
Timer Channel Status Word

Timer HCAW
Timer Hard Channel Address Word

TIO
Test Input/Output

TSW
Tabled Status Word

INDEX

UNIVAC 9700 System
Processor Programmer Reference
UP-7936

March 1973

UPDATING PACKAGE B

File pages as specified below

SECTION	DESTROY FORMER PAGES NUMBERED	FILE NEW PAGES NUMBERED
Front Cover & Disclaimer	†	†
Page Status Summary	PSS-1	PSS-1B
Contents	5 and 6 7	5B and 6 7B
Appendix A	1 and 2 7 N. A.	1B and 2 7 and 8** 9** and 10**
Index	5 and 6 9 and 10	5B and 6 9 and 10B

†Destroy old cover and file new cover
**These are new pages

All the technical changes in an update are denoted by an arrow (→) in the margin. A downward pointing arrow (↓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (↑) is found. A horizontal arrow (→) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.

PAGE STATUS SUMMARY

ISSUE: Update B to UP-7936

Section	Page Number	Update Level
Cover/Disclaimer		B
PSS	1	B
Contents	1 thru 4 5 6 7	Orig. B Orig. B
1	1 thru 11	Orig.
2	1 thru 41	Orig.
3	1 thru 37	Orig.
4	1 thru 15	Orig.
5	1 thru 19	Orig.
6	1 thru 15	Orig.
Appendix A	1 2 thru 7 8 thru 10*	B Orig. B
Appendix B	1 thru 3	Orig.
Index	1 thru 4 5 6 thru 9 10 11	Orig. B Orig. B Orig.
User Comment Sheet		

Section	Page Number	Update Level

Section	Page Number	Update Level

*New pages

SPERRY UNIVAC COMPUTER SYSTEMS

PUBLICATIONS UPDATE

UNIVAC 9700 SYSTEM

Processor

Programmer Reference

UP-7936-C

This UNIVAC 9700 System Library Memo announces the release and availability of Updating Package C to "UNIVAC 9700 System Processor Programmer Reference," UP-7936.

Updating Package C should be utilized as specified on the Updating Summary Sheet. Upon completion of manual updating, file the Updating Summary Sheet after the Contents section so that a record of updating is maintained.

This update contains additions and corrections to the *UNIVAC 9700 System Processor Programmer Reference, UP-7936* to incorporate UNIVAC Series 70 compatibility. Major changes are in the areas of:

- Processor working registers
- Control storage and control storage map
- System configuration and expansion features
- System mode and mode extension
- Program status word bits, (interrupt code/mode extension).
- System mode and mode extension (M-field)
- Multiplexer channel configurations
- Subchannel/device address correspondence
- Channel address word (SPEM)
- Hard channel command word (SPEM and SIAM)

Various other corrections have been made throughout the manual to complete this update.

Copies of Updating Package C are now available for requisitioning. Either the updating package alone, or the complete manual with the updating package may be requisitioned by your local Sperry Univac Representative. To receive the updating package alone, order UP-7936-C. To receive the complete manual, order UP-7936.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS:
Mailing Lists 217, 630 and 692	Mailing Lists 65, 66, 67 and 68 (Package C to UP-7936, Cover and 81 pages plus Memo and 2 Updating Summary Sheets)	Library Memo for UP-7936-C
		RELEASE DATE: December 1973

UNIVAC 9700 System
Processor Programmer
Reference
UP-7936

December 1973

UPDATING PACKAGE C
File pages as specified below

SECTION	DESTROY FORMER PAGES NUMBERED	FILE NEW PAGES NUMBERED
Front Cover & Disclaimer	†	†
Page Status Summary	PSS-1-B	PSS-1-C
Contents	1 thru 4	1C thru 4C
	5B and 6	5C and 6C
	7B	7C
Section 1	1 and 2	1 and 2C
	3 thru 6	3C thru 6C
	N.A.	6aC**
	7 and 8	7C thru 8C
	9 and 10	9C and 10
	11	11C
Section 2	1 and 2	1C and 2C
	3 and 4	3C and 4
	13 and 14	13C and 14C
	N.A.	14aC**
	15 and 16	15C and 16C
	N.A.	16aC**
	17 and 18	17C and 18C
	21 and 22	21 and 22C
	25 and 26	25C and 26C
	31 and 32	31C and 32
	33 and 34	33C and 34
Section 3	3 and 4	3 and 4C
	N.A.	4aC**
	5 and 6	5C and 6C
	N.A.	6aC**
	7 and 8	7C and 8C
	N.A.	8aC**
	11 and 12	11 and 12C
	23 and 24	23C and 24C
	N.A.	24aC**
	27 and 28	27 and 28C
	29 and 30	29C and 30C
	N.A.	30aC**
Section 4	1 and 2	1C and 2
	3 and 4	3C and 4C
	N.A.	4aC**
	5 and 6	5C and 6

UNIVAC 9700 System
Processor Programmer
Reference
UP-7936

December 1973

UPDATING PACKAGE C
File pages as specified below

SECTION	DESTROY FORMER PAGES NUMBERED	FILE NEW PAGES NUMBERED
Appendix A	1B and 2	1C and 2
Index	1 thru 11	1C thru 11C
User Comment Sheet		

† Destroy old cover and file new cover

** These are new pages

All the technical changes in an update are denoted by an arrow (→) in the margin. A downward pointing arrow (↓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (↑) is found. A horizontal arrow (→) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.

Comments concerning this manual may be made in the space provided below. Please fill in the requested information.

System: _____

Manual Title: _____

UP No: _____ Revision No: _____ Update: _____

Name of User: _____

Address of User: _____

Comments:

CUT

FOLD

FIRST CLASS
PERMIT NO. 21
BLUE BELL, PA.

BUSINESS REPLY MAIL NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

UNIVAC

P.O. BOX 500

BLUE BELL, PA. 19422

ATTN: SYSTEMS PUBLICATIONS DEPT.

FOLD

CUT