9400

SYSTEM

CONTENTS

APPENDICES

FIGURES

TABLES

1. INTRODUCTION

1.1. UNIVAC 9000 SERIES COMPUTER FAMILY

The UNIVAC 9000 Series is a computer family that embodies many bold, new design concepts in a unified and low cost line of data processing equipment. The UNIVAC 9200 computer is a small, card oriented data processing system with a basic internal storage capacity of 8,192 bytes. The 9200 System can be expanded to the higher performance card or tape oriented 9300 System. From this, it is easy to make the transition to the still more powerful tape or disc oriented UNIVAC 9400 System, shown in Figure 1–1, which has a basic storage capacity of 24,576 bytes with a cycle time of 600 nanoseconds. Internal storage for the 9400 System consists of a plated-wire memory as developed by Univac for the 9000 Series computers.

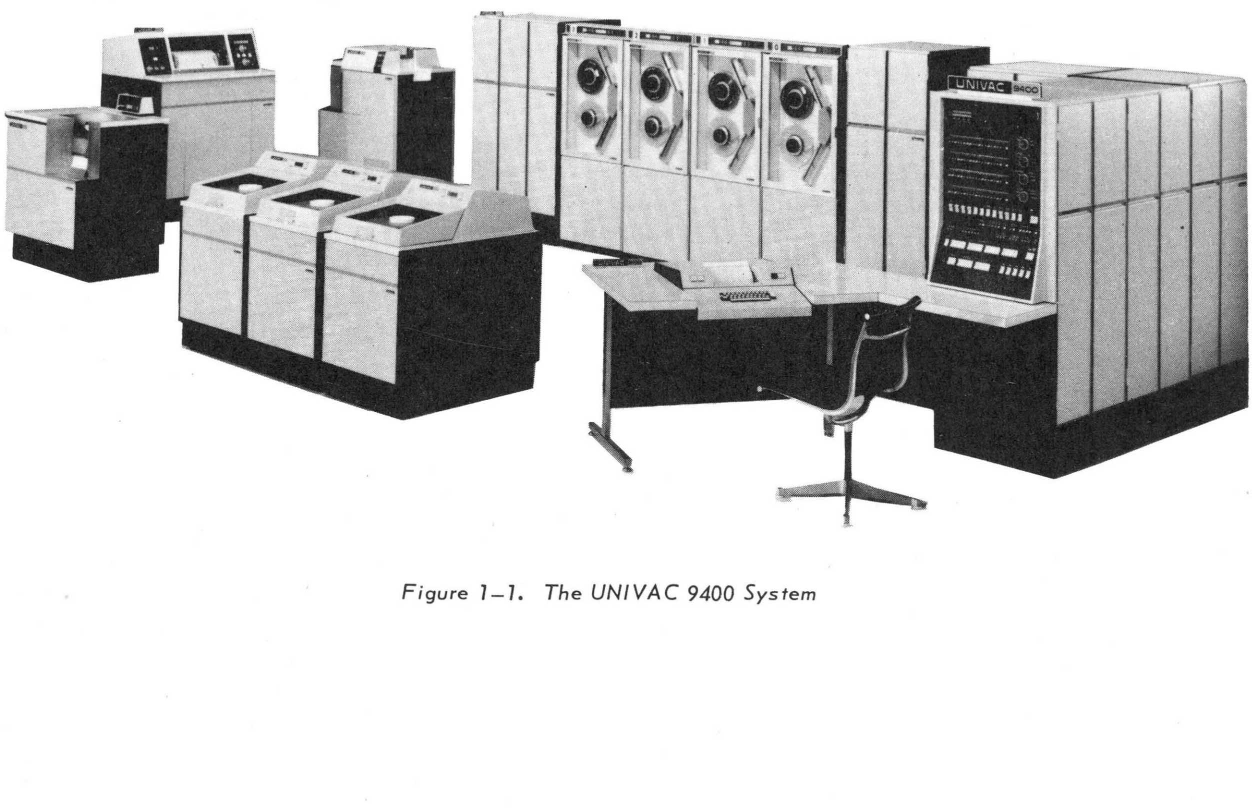


Figure 1–1. The UNIVAC 9400 System

Equipment expansions to larger systems or different configurations of the same system are compatible within the 9000 Series. This highly desirable hardware compatibility also applies to software in the 9000 Series. Programming compatibility within the wide range of data processing capability offered by the 9000 Series allows maximum freedom for growth and expansion into larger equipment configurations.

The equipment and programming compatibility features of the UNIVAC 9000 Series computers allow the entire series to be thought of as essentially one large computer system whose size and configuration are adjustable over a wide range of data processing applications. From the user's point of view, this range of choice is the most economical because a system can always be selected to fit the needs of an installation. Costly time lags where the computing system can be ahead or behind the demands of the user are thereby eliminated.

The UNIVAC 9400 System offers speed, reliability, modularity, compactness, and, most significantly, economy to the user requiring random or sequential batch processing, or communications processing.

The operating system for the UNIVAC 9400 consists of a comprehensive set of programming aids, control programs and utility services. It is modular in design to fulfill a wide range of data processing requirements. The user may write programs in the common higher level programming languages, COBOL and FORTRAN. These higher level programming languages and the symbolic assembly language permit a choice of the language best suited to the application. Control programs provide for both random and sequential batch processing, and communication processing. Data to be processed can be introduced to the system from either central or remote locations.

Conceptually, the approach to the system is the same for all users, regardless of the size of their system. The operating system is designed to provide (as far as possible) the same functions for all configurations. However, system performance is dependent upon the facilities available. In certain instances, the smaller systems may require additional passes over the data to obtain the same output that a more powerful system would produce in a single pass.

Computer applications in the UNIVAC 9400 System can be broadly classified as follows:

- Random and sequential batch processing
- Communications oriented processing

In the past, the selection of data processing systems has been based upon the requirements of the primary application. In many cases, it has been difficult, even impossible, to acquire a system that could handle the broad range of potential applications. Until the present time, only the most expensive and sophisticated systems could process more than one category of application without degrading performance. The UNIVAC 9400 System can perform both of these application categories efficiently. Programs can be performed concurrently while sharing access to all facilities of this system.

CHARACTERISTICS

SYSTEM ORIENTATION	Tape/disc
DATA ORGANIZATION	8-bit byte
BASIC INTERNAL STORAGE	24,576 bytes
MAXIMUM INTERNAL STORAGE	131,072 bytes
MEMORY CYCLE TIME	600 nanoseconds/2 bytes
ADD (BINARY) INSTRUCTION TIME (TWO 32-BIT WORDS)	6 microseconds
DECIMAL MULTIPLY AND DIVIDE INSTRUCTIONS	Standard
EDIT INSTRUCTION	Standard
CARD READER	600 cpm
CARD PUNCH	200 cpm or 250 cpm
READ/PUNCH	Optional
PRINTER	900 to 1100 lpm or 1200 to 1600 lpm
MAGNETIC TAPE RATE	34,160 bytes per second to 192,000 bytes per second
NUMBER OF MULTIPLEXER CHANNELS	1
MULTIPLEXER CHANNEL RATE	85,000 bytes per second
NUMBER OF OPTIONAL SELECTOR CHANNELS	2
SELECTOR CHANNEL RATE	333,000 bytes per second
REGISTERS	16 for problem program functions 16 for Supervisor functions
DISC STORAGE	7.25 million bytes per drive
COMMUNICATIONS	Up to 64 duplex lines

1.2. GROWTH AND COMPATIBILITY

The UNIVAC 9400 System hardware and software features surpass current standards. The features are at the disposal of the user to any extent he wishes. A basic system can be supplemented by many optional features to meet a wide variety of system needs.

The modular design of the UNIVAC 9400 System, coupled with its high speed main storage and I/O architecture, provides a dependable base for future extensions. Changes in business demands, applications, programming techniques, system configuration, or new input/output devices can be readily incorporated in the UNIVAC 9400 System. This architecture extends the usefulness of the initial planning, programming, and operational procedures used with the system.

The UNIVAC 9400 System is complemented by a line of direct access storage, magnetic tape, communications, punched card reading and punching, and high speed printing devices. Peripheral devices are available with different speeds, capacities, and industry data compatibility to permit each user to select the most profitable combination for his application.

The availability of a high speed multiplexer channel enables many input/output devices to exchange data with the central processor at a rate of 85,000 bytes per second. Each device can transmit or accept data without degrading the operation of other devices. Two selector channels are available with the system to provide a combined data exchange capability of 666,000 bytes per second.

The UNIVAC 9400 Central Processor makes storage available on an incremental basis. Thus, each user can select a main storage capacity to best meet his specific requirements. Main storage can be expanded to 32K, 49K, 65K, 98K, and 131K bytes.

The UNIVAC 9400 System and the other computers in the 9000 Series use industry-standard instructions. The instruction repertoire of smaller systems, such as the UNIVAC 9200 System, are subsets of the UNIVAC 9400 repertoire. This family concept permits programs written for smaller systems to be source code compatible with larger systems.

1.3. SOFTWARE FEATURES

The principal objective of the UNIVAC 9400 Operating System is to make the full power of the computing system available to the user to solve his data processing tasks. Implicit in this objective is the need that the software system be consistent with the capabilities of a small- to medium-scale computing system. Overly sophisticated functions, which are of questionable value, have been omitted from the software.

The Supervisor control program is a part of the operating system that operates with problem programs to provide the control necessary for optimum utilization of the UNIVAC 9400 System hardware and software. By use of the Supervisor, the hardware and software systems are effectively coordinated to satisfy the requirements of a growing number of diversified applications. Problems are handled directly and promptly with as little internal bookkeeping as possible without compromising the integrity of the computing system.

1.3.1. Modularity

Functional modularity has been employed in the design of the Supervisor to ensure its adaptation to a wide range of data processing applications. The user can tailor the Supervisor to his particular applications by parameter selection and specification of the various functional modules at systems generation time.

1.3.2. Multiprogramming

Utilization of central processor time is maximized by multiprogramming. A Supervisor can be generated to control from one to five problem programs. In this environment, problem programs are processed concurrently in the computing system. In addition to problem programs, many of the Supervisor functions are designed as autonomous activities capable of being processed as independent programs.

The multiprogramming technique employed by the Supervisor involves the distribution of processing time to independent programs based on program priorities, time allocations, and input/output equipment utilization. Consideration of these factors by the Supervisor assures the user that the distribution of processing time is efficient and equitable.

1.4. COMMUNICATIONS

The UNIVAC 9400 System can be used for communications oriented data processing through the use of a UNIVAC Data Communication subsystem (DCS) and the communications adapter. Each communication subsystem must be attached to the UNIVAC 9400 Central Processor by means of one of the eight subchannels provided in the standard multiplexer channel. The communication configurations provide for any number of simplex line positions up to a maximum of 64 input and 64 output line terminals. In the duplex mode (two directional) a total of 64 line terminals can be accommodated.

The maximum number of remote devices that can be used with the line terminals depends upon the type of device and the volume of data to be transmitted. The use of the dial mode allows the central processor to select remote devices automatically. Thus, the number of possible remote devices is virtually unlimited.

The accuracy of transmission can be controlled through an optional parity check on all communicated data. Either odd or even parity check can be performed for each character in a message to the central processor. Similarly, odd or even parity can be generated for each character transmitted. In addition, a longitudinal redundancy check may be performed for each input message and generated for each output message.

The Data Communication Subsystem is compatible with Data-Phone* Service, TWX Networks, Telex**, and Telpak*** C.

1.5. EQUIPMENT CONFIGURATIONS

The following diagrams (Figures 1–2 to 1–6) illustrate some of the system configurations that are available on the UNIVAC 9400 System. Figure 1–2 shows the UNIVAC 9400 Processor and console configuration.

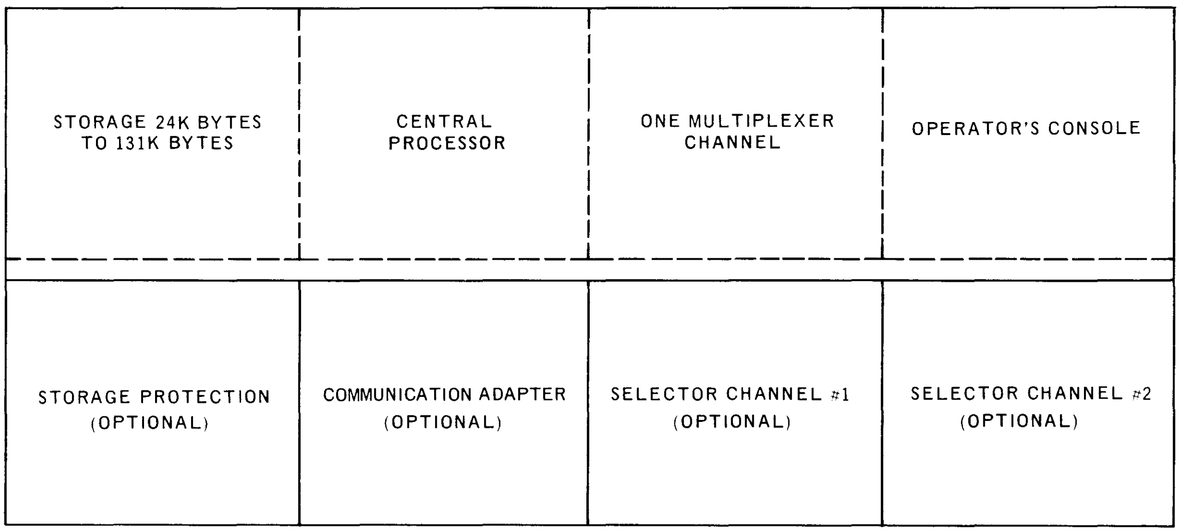


Figure 1–2. UNIVAC 9400 Processor and Console Configuration

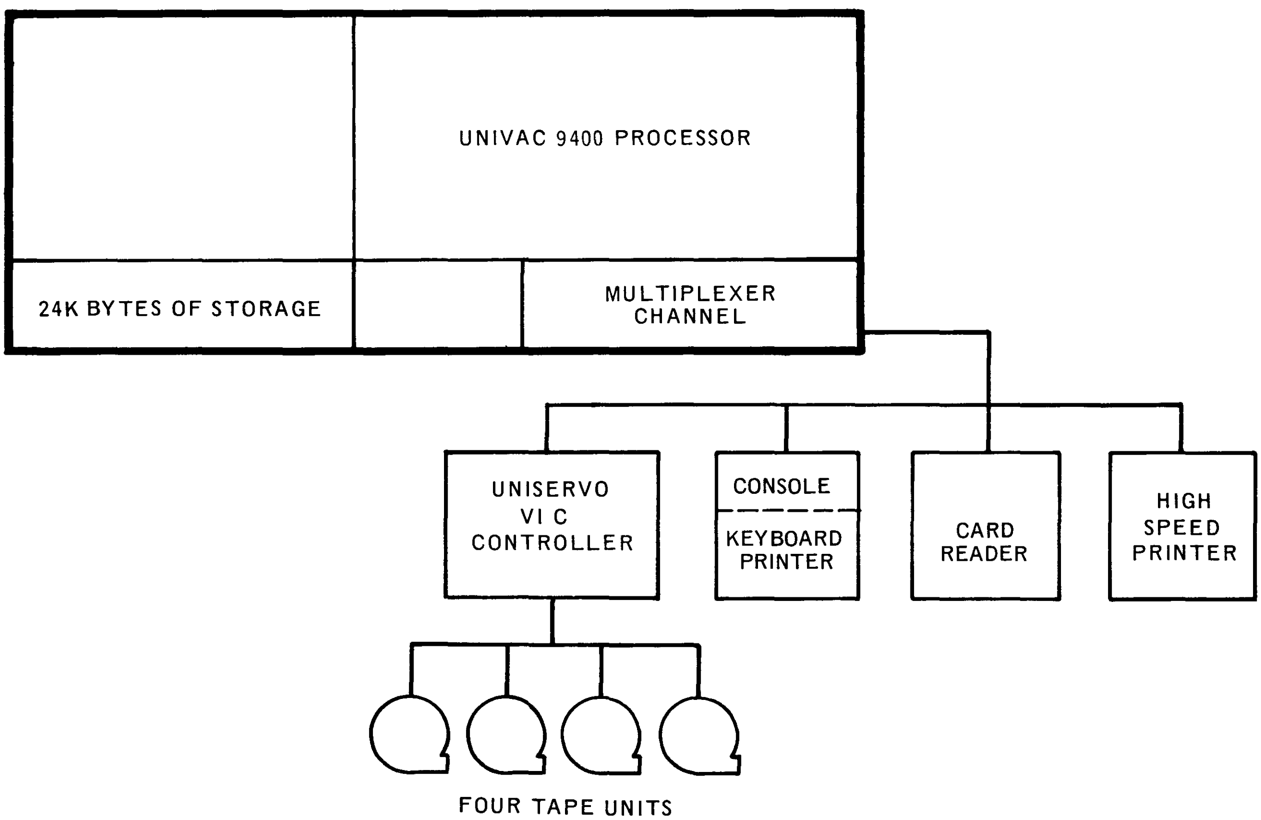
*Trademark and Service Mark of A.T.& T. Co.

**Trademark of Western Union Telegraph Co.

***Trademark of A.T.& T. Co.

The system configuration shown in Figure 1–3 provides a convenient transition from smaller tape systems such as the UNIVAC 9300. This system is suitable for both scientific and business data processing. A few of the many applications possible with this type of system follow:

- Order entry and billing
- Payroll and labor distribution
- Savings and loan
- Sales analysis
- Quality control
- Manpower scheduling
- Personnel analysis
- Engineering design
- Inventory control



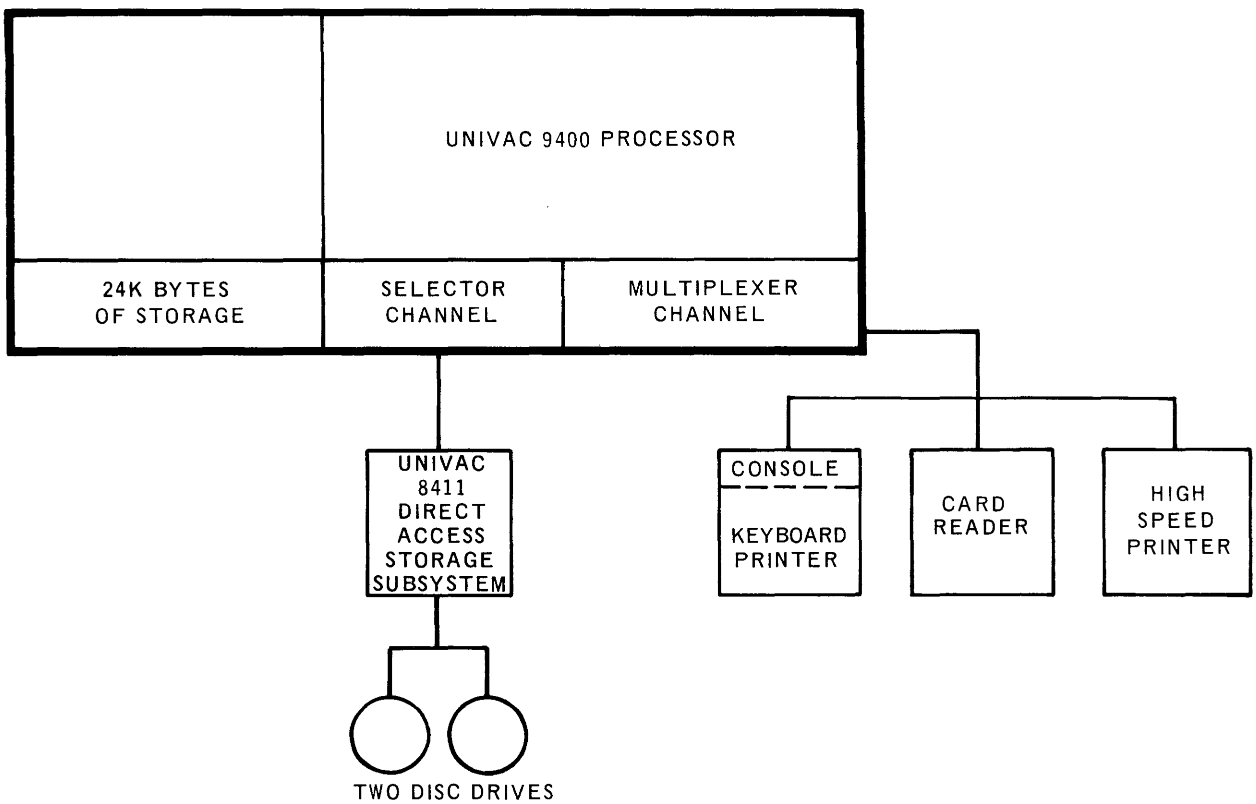
- PROCESSOR WITH MULTIPLEXER CHANNEL AND INTERRUPT TIMER
- CONSOLE WITH KEYBOARD AND PRINTER
- 24,576 BYTES OF STORAGE
- CARD READER
- HIGH SPEED PRINTER
- UNISERVO VI C CONTROLLER WITH FOUR TAPE UNITS

Figure 1–3. UNIVAC 9400 System Basic Configuration (Tape System)

The system configuration shown in Figure 1–4 provides a convenient transition from smaller disc-pack oriented systems. This system, as a result of the auxiliary disc storage, is suited for both scientific and business data processing. A few of the many applications possible with this system follow:

- Billing and accounts receivable
- Savings and loan
- Statistical analysis
- Payroll and labor distribution
- Production control
- Quality control
- Personnel analysis
- Engineering design
- Inventory control

Small files can be more efficiently referenced and maintained using the direct access (disc) storage. Since program libraries, as well as frequently used files, can always be available on disc, this system can be used for short irregularly scheduled runs concurrently with other processing.



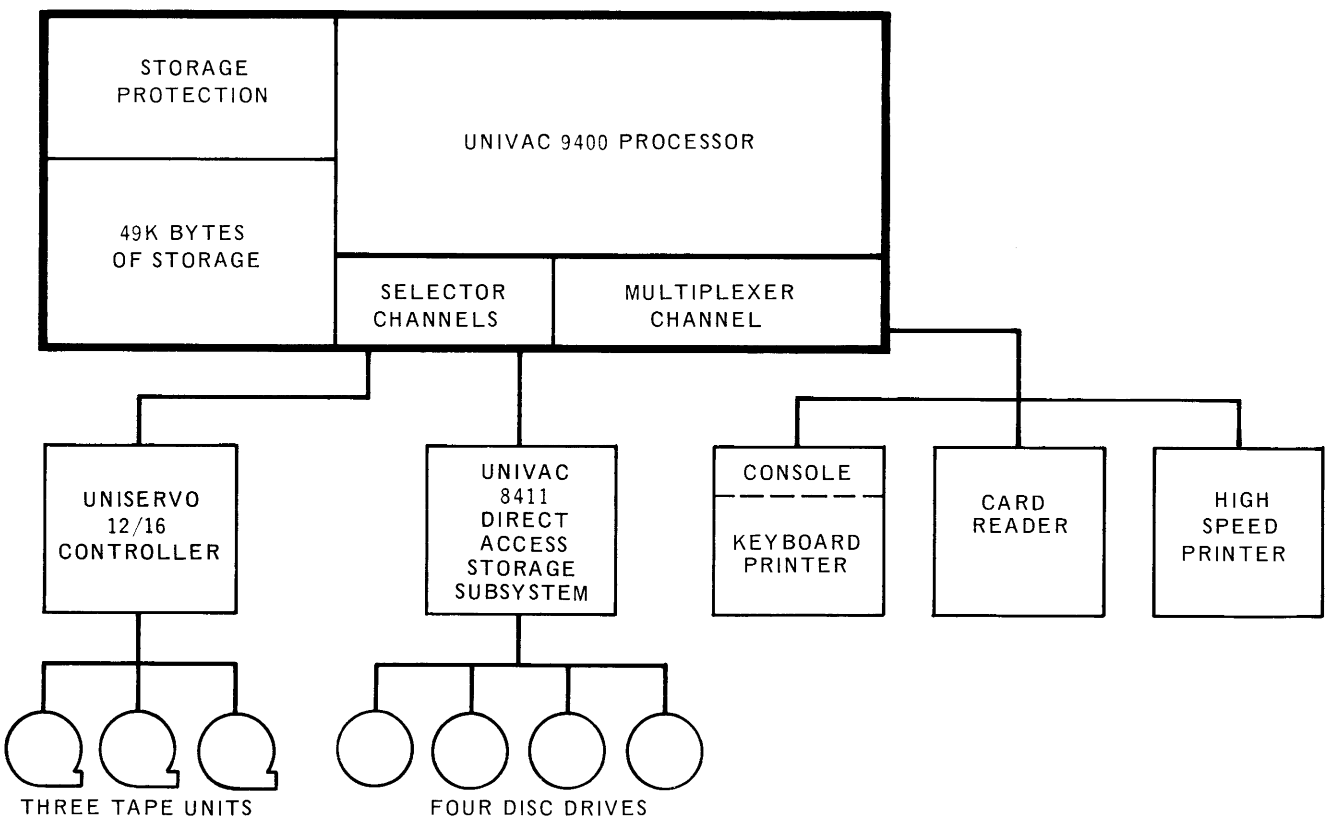
- PROCESSOR WITH MULTIPLEXER CHANNEL AND INTERRUPT TIMER
- CONSOLE WITH KEYBOARD AND PRINTER
- 24,576 BYTES OF STORAGE
- CARD READER
- HIGH SPEED PRINTER
- UNIVAC 8411 DIRECT ACCESS STORAGE SUBSYSTEM WITH TWO DISC DRIVES

Figure 1–4. UNIVAC 9400 System Basic Configuration (Disc System)

The system configuration shown in Figure 1–5 combines the advantages of the tape and the disc systems. A few of the many applications possible with this system follow:

- Billing and accounts receivable
- Insurance
- Pension
- Order entry
- Personnel statistics
- Sales statistics
- Production control
- Engineering design
- Inventory control

The broad flexibility of disc stored program libraries and control stream technique provide software and system organization that has been previously available only in the largest computer systems. The tape and disc combinations are well suited for applications where there are large master files having a high percentage of updates each cycle. Also, the use of tapes saves disc-pack cost and provides a fast method of sequential processing. The discs provide program overlays, table storage, small file storage, and the flexibility of the disc oriented operating system.



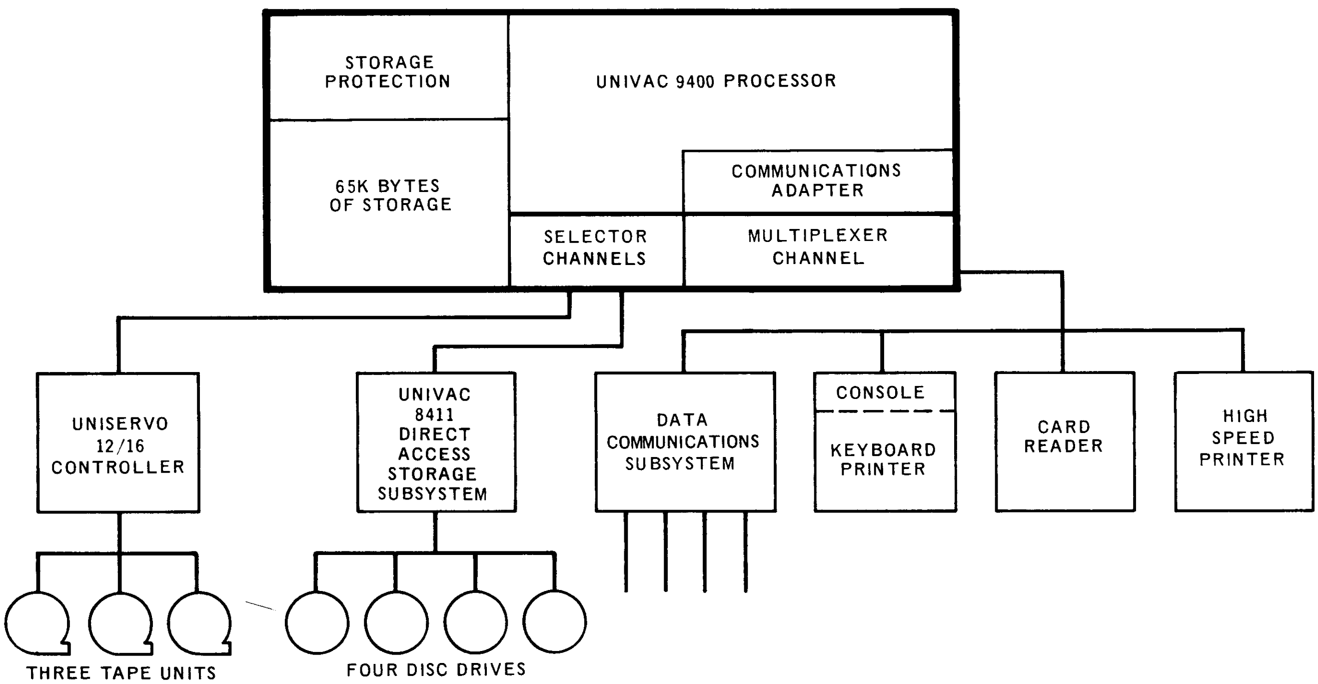
- PROCESSOR WITH MULTIPLEXER CHANNEL AND INTERRUPT TIMER
- CONSOLE WITH KEYBOARD AND PRINTER
- 49,152 BYTES OF STORAGE WITH STORAGE PROTECTION
- CARD READER
- HIGH SPEED PRINTER
- UNISERVO 12/16 CONTROLLER WITH THREE TAPE UNITS
- UNIVAC 8411 DIRECT ACCESS STORAGE SUBSYSTEM WITH FOUR DISC DRIVES

Figure 1–5. Typical UNIVAC 9400 System Configuration (Disc System with Tape)

The system configuration shown in Figure 1–6 provides communication capability in addition to the capabilities of the disc and tape system illustrated in Figure 1–5. A few of the many applications provided by adding the communications facility follow:

- Inquiry systems
- Remote transaction processing
- Data collection from remote terminals

The multiprogramming capabilities of the system permit the communication applications to operate concurrently with batch programs initiated at the system site.



- PROCESSOR WITH MULTIPLEXER CHANNEL AND INTERRUPT TIMER
- CONSOLE WITH KEYBOARD AND PRINTER
- 65,536 BYTES OF STORAGE WITH STORAGE PROTECTION
- CARD READER
- HIGH SPEED PRINTER
- UNISERVO 12/16 CONTROLLER WITH THREE TAPE HANDLERS
- UNIVAC 8411 DIRECT ACCESS STORAGE SUBSYSTEM WITH FOUR DISC DRIVES
- COMMUNICATIONS ADAPTER
- DATA COMMUNICATIONS SUBSYSTEM (DCS-4)

Figure 1–6. Typical UNIVAC 9400 System Configuration (Disc System with Tape and Communications)

2. SYSTEM HARDWARE

2.1. THE UNIVAC 9400 SYSTEM

The UNIVAC 9400 System was designed to be a medium cost, high performance, system with data processing and communication capability that is well within the reach of the majority of data processing users, and in which no compromise is made with operating potential. This design goal has been reached. In the case of the processor storage and circuitry, the resulting high speed permits the UNIVAC 9400 System to compete favorably in the performance of any conventional data processing program.

2.2. CENTRAL PROCESSOR UNIT

Main storage, control, arithmetic, and input/output sections comprise the major parts of the control processor. A description of these parts are described in the following paragraphs.

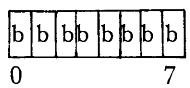
2.2.1. Main Storage Characteristics

The main storage of the UNIVAC 9400 System is contained in free standing units with a 600 nanosecond cycle rate for each halfword (two bytes). The individually addressable units of main storage are called bytes. Each byte contains eight bits plus one bit for parity. Parity is such that the total number of ones in the byte, including the parity bit, is odd. The parity bit is generated when data is written into storage. Because the parity bit cannot be accessed by the program, a byte is considered in terms of the eight accessible bits when further discussed in this manual.

2.2.1.1. Addressing and Data Formats

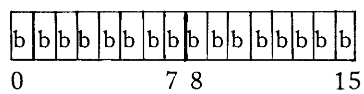
Each eight-bit byte of main storage can be accessed by the problem program. These bytes are addressed consecutively from 0 through a maximum of 131,071. Bytes may be accessed separately or in groups. The address of a group of bytes is addressed by the leftmost byte of the group. The bits in a byte are also numbered from left to right starting with zero.

Byte



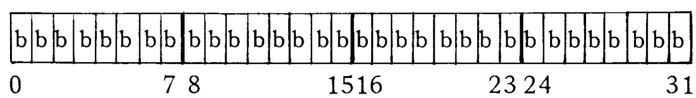
Halfword formats consist of two consecutive bytes.

Halfword



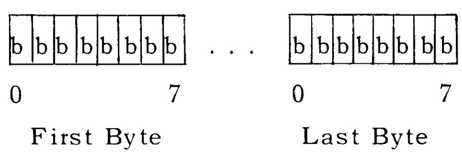
Fullword formats consist of four consecutive bytes.

Fullword



Variable data formats consist of a variable number of consecutive bytes.

Variable Data Format



Fixed length fields, such as halfwords and fullwords, have integral boundaries. Fixed length fields must be loaded into main storage so that the address is evenly divisible by the field length (in bytes). Thus, a halfword must have an address that is a multiple of two and a fullword must have an address that is a multiple of four.

Variable length data fields are not restricted by boundaries. Instruction lengths are two, four, or six bytes, and are restricted only to halfword boundaries.

2.2.1.2. Low Order Storage

The low order 512 bytes of main storage have been reserved to contain specific operating information. The data stored in these locations is accessed by the hardware and the operating system during the execution of the appropriate functions. The operating system provides for loading and protecting the appropriate data in these locations.

2.2.1.3. Storage Protection

Program protection in a multiprogrammed environment is achieved by the write protection feature. Write protection is controlled through the use of the Limits Register, which limits the storage area that any one program can access for storage of data. The Limits Register is under control of the Supervisor which loads the address limits of the particular program in operation. An address exception interrupt is generated whenever a write order attempts to address a location outside the bounds of the Limits Register.

2.2.2. Control Section

The control section controls the sequence in which instructions are executed, and it interprets and controls the execution of each individual instruction. The cycling of main storage is initiated by this section. All of the hardware aspects of interrupt handling, error checking, and protection are performed by the control section. The control section also maintains the program address location counter and provides for the different processor modes of operation.

The central processor can reference two sets of 16 general purpose registers. Both sets are contained in low order storage. One set is used by the Supervisor and the other set is used by problem programs. This design greatly reduces the interrupt processing time overhead required when only a single set of general registers is used. Thus, when the processing mode is changed between problem program and Supervisor modes, the following steps required in single register systems are unnecessary.

(1) Store the contents of problem program registers.

(2) Load the executive routine data into the registers.

(3) Store the executive routine data.

(4) Reload the problem data back into the registers.

These registers can be used for fixed point arithmetic, logical arithmetic, and the indexing of both instruction and operand addresses. The capacity of each register is 32 bits (one word). The registers are identified by the hexadecimal numbers zero through F. The general purpose registers are addressable only through the specific instruction fields provided for their access.

2.2.3. Arithmetic Section

The arithmetic section performs all data manipulations including logical and numerical arithmetic, data comparisons, and shifting. This section also performs single or double indexing of operand addresses. The adder in this section performs arithmetic in twos complement form. There are three classes of arithmetic operations as follows:

2.2.3.1. Fixed Point Arithmetic

A fixed point arithmetic operand can be either a 32-bit word or a 16-bit halfword. When possible, 16-bit operands should be specified to conserve storage. The sign of a fixed point operand is always the leftmost bit of the operand The following figures illustrate the format of halfword and fullword fixed binary values:

Halfword

S	Integer
0	1 — 15

Fullword

S	Integer
0	1 — 31

When a halfword fixed point number is called from storage, it is always expanded to a right justified fullword; the sign is extended to the left.

2.2.3.2. Decimal Arithmetic

Decimal number fields can be variable in length and can exist in two formats; unpacked decimal numbers and packed decimal numbers. Decimal operations including add, subtract, multiply, and divide, can be performed only on packed decimal numbers. Instructions are provided for converting decimal numbers from unpacked to packed and from packed to unpacked format.

In the unpacked decimal format, each byte contains one digit of the number. The byte is divided into two equal fields, a zone field and a digit field. 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 the printer. The format of a three-digit number is shown below:

ZONE	DIGIT	ZONE	DIGIT	SIGN	DIGIT
0	7	8	15	16	23
Byte		Byte		Byte	

Unpacked Decimal Number

In packed format, each byte contains two digits. The least significant four bits of the least significant byte provides the sign of the number. Packed decimal format is used for all decimal arithmetic operations. The format of a five-digit number is shown below:

DIGIT	DIGIT	DIGIT	DIGIT	DIGIT	SIGN
0	7	8	15	16	23
Byte		Byte		Byte	

Packed Decimal Number

Decimal numbers (0 through 9) are represented in the four bit binary coded decimal form (0000 through 1001). The codes 1010 through 1111 are used for sign codes. The binary values 1011 and 1101 represent a minus sign and the binary values 1010, 1100, 1110 and 1111 represent a plus sign. This assignment of sign codes permits the use of either of two conventions: American Standard (ASCII) expanded to eight bits or Extended Binary Coded Decimal (EBCDIC). The codes 1010 (plus) and 1011 (minus) are used in ASCII; the codes 1100 (plus) and 1101 (minus) are used in EBCDIC. A control bit in the Program Status Word determines whether the system is to operate in the ASCII or the EBCDIC mode.

2.2.3.3. Logical Operations

Logical operations such as comparing, translating, editing, bit setting, and bit testing are performed by the arithmetic section. Logical operations can be performed on fixed length operands and variable length operands. Logical operations on fixed length operands are performed in the registers; logical operations on variable length operands are performed in main storage. The instruction format determines whether the logical operation is to be performed in main storage or in a register.

2.2.4. Input/Output Section

The input/output (I/O) section of the UNIVAC 9400 Central Processor initiates, directs, and monitors the transfer of data between storage and the peripheral subsystems. After an I/O instruction has been transferred to the control unit from the control section, the data transfer is performed concurrently with other central processor functions.

The I/O section consists of the input/output channels and the standard UNIVAC 9000 Series I/O Interface which connects channels with the unit controllers. This interface is identical for all the I/O control units; it has been designed for use with all of the available peripheral devices, as well as for future devices.

Two selector channels are available on the UNIVAC 9400 System in addition to the multiplexer channel.

2.2.4.1. Multiplexer Channel

The central processor has one multiplexer channel which consists of eight standard shared subchannels. Through the shared subchannels, 128 nonshared subchannels can be added as a communications adapter option. This option must be used to operate communications subsystems on the channel. A shared subchannel is one which may be addressed by more than one input/output device.

Up to eight I/O control units can be connected to the multiplexer channel by means of the subchannels; these can be any combination of standard control units or Data Communications Subsystems (DCS). Standard control units can handle up to a maximum of 16 devices each depending upon the subsystem selected. Each DCS can (depending on the type) accommodate up to 32 simplex communication lines, with a maximum of four control units permitted allowing 128 simplex lines.

The Subchannel Control Word provides the address of the Buffer Control Word, and the Buffer Control Words provide controlling information needed by the I/O control units to transfer data to or from the central processor. The Subchannel Control Words for all subchannels are located in low order main storage. The Buffer Control Words can be located anywhere in main storage. A separate Subchannel Control Word is used for each of the eight subchannels associated with the multiplexer channel. If communications devices have been included in the configuration, there is a Subchannel Control Word for each communication line. The existence of a separate Subchannel Control Word for each standard control unit and one for each communication line permits all of the low speed devices connected to the multiplexer channel to operate concurrently. Each low speed device on the multiplexer channel sends its eight-bit address over the channel prior to the transfer of each data byte. This device address causes the Subchannel Control Word and the Buffer Control Word to be selected from storage and placed in the Multiplexer Channel Register. This then directs the data byte into or out of the proper main storage location. Thus, data transfers are multiplexed when the Subchannel Control Word and the associated Buffer Control Word related to the appropriate device is specified for each byte transferred.

Examples of low speed devices are printers, card readers, card punches, remote communication devices, I/O subsystems such as the UNIVAC 1004 or 1005 Subsystem, and the UNIVAC 9200 or 9300 Subsystem.

2.2.4.2. Selector Channel

Two selector channels can be added as options to the basic configuration.

High speed devices, such as UNISERVO 12 and UNISERVO 16 Magnetic Tape Units and UNIVAC 8411 Disc Storage Drives may only be connected to the selector channels.

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 buffer control register used for each selector channel is maintained in the channel. Since only one buffer control register is used for each selector channel, the devices attached to a selector channel are serviced on a 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. However, both selector channels, the multiplexer subchannels, and the central processor can operate concurrently.

2.2.5. Interrupt Processing Control

The UNIVAC 9400 System contains a very efficient interrupt system that makes it a highly capable communications system. The UNIVAC 9400 Interrupt System provides the means by which the central processor changes from the problem program state to privileged or supervisor state. The seven types of interrupt employed in the UNIVAC 9400 System are listed below in order of priority:

- Supervisor Call Interrupt – This interrupt results from the execution of a supervisor call instruction. Status information provides the operating system with a link to parameter information in the calling program.

- Program Interrupt – This interrupt occurs when one of the following exceptions are recognized by the central processor.
  - Illegal operation code.
  - Privileged operation instruction encountered while the processor is in the problem state.
  - Storage Protection – attempting to write data outside the bounds of the Limits Register.
  - Addressing Exception – reference to low order storage while in the problem state.
  - Specification Exception – integral boundary reference error.
  - Binary Arithmetic Overflow.
  - Decimal Arithmetic Overflow.

- Timer Interrupt – This interrupt is caused when the millisecond timer storage is decremented to zero.

- Selector Channel 1 Interrupt – This interrupt occurs when an interrupt condition has been generated on Selector Channel 1 by a device.

- Selector Channel 2 Interrupt – This interrupt occurs when an interrupt condition has been generated on Selector Channel 2 by a device.

- Shared Multiplexer Channel Interrupt – This interrupt occurs when an interrupt condition has been generated on a shared multiplexer channel. The status is stored in the appropriate Subchannel Control Word.

- Nonshared Multiplexer Channel Interrupt – This interrupt occurs when an interrupt condition has been generated on a nonshared multiplexer channel. The status is tabled which indicates the source and cause of the interrupt.

2.3. UNIVAC 9400 INSTRUCTION REPERTOIRE

The power and flexibility of the UNIVAC 9400 System is reflected in the instruction repertoire and execution times. The full repertoire includes 67 instructions, many of which offer several optional variations. The instruction set is supplemented by an input/output device instruction which controls each of the possible peripheral devices.

The UNIVAC 9400 instructions fall into two functional categories:

- Supervisor
- Standard

2.3.1. Supervisor Instruction Set

The Supervisor (privileged) instruction set is used primarily by the software operating system when operating in the supervisory state. In this state, all instructions (Supervisor and standard) are valid and can be executed. The privileged general registers are selected and low order storage can be addressed. Instructions in the Supervisor set cannot be executed in a problem program.

2.3.2. Standard Instruction Set

The standard (nonprivileged) instruction set is used to write programs for operation in the problem state (application and data processing programs). In this state, the Supervisor (privileged) instructions are invalid, the problem set of general registers is selected, and low order storage cannot be addressed.

The standard instruction set provides instructions to process fixed length binary numbers, packed and unpacked decimal numbers, and alphabetical characters. Data may be transferred between main storage and the problem set of general registers and from storage to storage. The operations of shifting, branching, and logical functions are also included.

2.3.3. Instruction Types

Instructions can be either two, four, or six bytes in length. In a two byte or halfword instruction, only registers are referenced. A four byte or two halfword instruction references main storage once. A six byte or three halfword instruction references main storage twice. All instructions must be located in storage on halfword boundaries, that is, have an even address. The six instruction types are:

- RR = a register to register operation
- RX = a register and indexed storage operation
- RS = a register and storage operation
- SI = a storage and immediate operand operation
- SS1 = a storage to storage operation (256 byte maximum length)
- SS2 = a storage to storage operation (16 byte maximum length)

The instruction formats for each type are shown and described in the following sections and are also summarized in Figure 2–1 and in 2.3.3.7.

2.3.3.1. Register to Register Instructions (RR)

The RR instructions are two bytes in length and are used primarily to process data between registers. The maximum data operand that can be handled is a fullword of 32 bits; the fullword may or may not be a signed binary number.

In this format, there are 14 instructions:

RR Instructions	Mnemonic OP Codes
Add	AR
AND	NR
Branch and Link	BALR
Branch on Condition	BCR
Branch on Count	BCTR
Compare	CR
Compare Logical	CLR
Exclusive OR	XR
Load	LR
Load and Test	LTR
OR	OR
Set Program Mask	SPM
Subtract	SR
Supervisor Call	SVC

The source code formats of the RR instructions are:

OP R_1

OP R_1, R_2

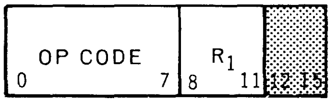
OP I

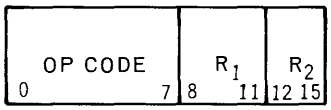
OP M_1, R_2

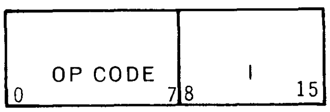
where:

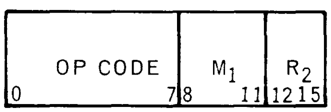
- OP is the mnemonic operation code.
- R_1 is an expression representing a register as operand 1.
- R_2 is an expression representing a register as operand 2.
- I is actual data expressed in bits to be used for control purposes.
- M_1 is a four bit data mask used in testing.

The object codes formed by these codes are:









2.3.3.2. Register to Indexed Storage Instructions (RX)

The RX instructions are four bytes in length and are used primarily to process data between registers and indexed storage. The maximum data operand that can be handled is a fullword of 32 bits; the fullword may or may not be a signed binary number.

In this format, there are 20 instructions:

RX Instructions	Mnemonic OP Codes
Add	A
Add Halfword	AH
AND	N
Branch and Link	BAL
Branch on Condition	BC
Branch on Count	BCT
Compare	C
Compare Halfword	CH
Compare Logical	CL
Exclusive OR	X
Insert Character	IC
Load	L
Load Address	LA
Load Halfword	LH
OR	O
Store	ST
Store Character	STC
Store Halfword	STH
Subtract	S
Subtract Halfword	SH

The source code formats of the RX instructions are:

OP $R_1, D_2(X_2, B_2)$

OP $M_1, D_2(X_2, B_2)$

where:

- OP is the mnemonic operation code.
- R_1 is an expression representing a register as operand 1.
- M_1 is a four bit data mask used in testing.
- D_2 is an expression designating the displacement value of operand 2.
- X_2 is an expression designating a register whose contents is used to index the displacement and base address of operand 2.
- B_2 is an expression designating a register whose contents is added to the displacement value to form the effective address of operand 2.

The object codes formed by these source codes are:

OP CODE	R_1	X_2	B_2	D_2
0 7	8 11	12 15	16 19	20 31

OP CODE	M_1	X_2	B_2	D_2
0 7	8 11	12 15	16 19	20 31

The effective address of operand 2 is formed by adding the contents of the base register B_2 to the displacement value D_2; this may or may not be modified by adding the contents of index register X_2.

2.3.3.3. Register to Storage Instructions (RS)

The RS instructions are four bytes in length and are used to perform multiple register storage and shift operations.

In this format, there are five instructions:

RS Instructions	Mnemonic OP Codes
Load Limits Register (privileged instruction)	LLR
Load Multiple	LM
Shift Left Single Logical	SLL
Shift Right Single Logical	SRL
Store Multiple	STM

The source code formats of the RS instructions are:

OP R_1, $D_2(B_2)$

OP R_1, R_3, $D_2(B_2)$

where:

- OP is the mnemonic operation code.
- R_1 is an expression representing a register as operand 1, or a register which is the first register of a multiregister group.
- R_3 is an expression representing a register which is the last register of a multiregister group.
- D_2 is an expression designating the displacement value of operand 2.
- B_2 is an expression designating a register whose contents is added to the displacement value to form the effective address of operand 2.

The object codes formed by these source codes are:

OP CODE	R_1		B_2	D_2
0 7	8 11	12 15	16 19	20 31

OP CODE	R_1	R_3	B_2	D_2
0 7	8 11	12 15	16 19	20 31

OP CODE			B_2	D_2
0 7	8	15	16 19	20 31

The effective address of operand 2 is formed by adding the contents of the base register B_2 to the displacement value D_2.

2.3.3.4. Storage and Immediate Operand Instructions (SI)

The SI instructions are four bytes in length and are used to provide control data for operation of the processor and peripherals. There are eleven instructions in this format divided into three categories: Logical, Processor Control, and Input/Output Control.

SI Instructions	Mnemonic OP Codes
Logical	
Add Immediate	AI
Compare Logical Immediate	CLI
Move Immediate Data	MVI
AND Immediate Data	NI
OR Immediate Data	OI
TEST Under Mask	TM
Exclusive OR	XI
Processor Control	
Halt and Proceed (privileged instruction)	HPR
LOAD Program State Word (privileged instruction)	LPSW
SET System Mask (privileged instruction)	SSM
Input/Ouput Control	
START I/O (privileged instruction)	SIO

The source code formats of the SI instructions are:

OP $D_1(B_1)$

OP $D_1(B_1),I_2$

where:

- OP is the mnemonic operation code.
- D_1 is an expression designating the displacement value of operand 1.
- B_1 is an expression designating a register whose contents is added to the displacement value to form the effective address of operand 1.
- I_2 is the actual data expressed in bits to be used as operand 2.

The object codes formed by these source codes are:

OP CODE		B_1	D_1
0 7	8 15	16 19	20 31

OP CODE	I_2	B_1	D_1
0 7	8 15	16 19	20 31

The effective address of operand 1 is formed by adding the contents of the base register (B_1) to the displacement value (D_1).

2.3.3.5. Storage to Storage Instructions (SS1)

The SS1 instructions are six bytes in length and are used to process data in storage where the operands are equal in length. The maximum operand may be 256 bytes in length.

In this format, there are nine instructions:

SS1 Instructions	Mnemonic OP Codes
Compare Logical Character	CLC
Edit	ED
Move Characters	MVC
Move Numerics	MVN
Move Zones	MVZ
AND Characters	NC
OR Characters	OC
Translate	TR
Exclusive OR	XC

The source code format for the SS1 instructions is:

OP $D_1(L, B_1), D_2(B_2)$

where:

- OP is the mnemonic operation code.
- D_1 is an expression designating the displacement value of operand 1.
- L is a value which states the length of both operand 1 and operand 2.
- B_1 is an expression designating a register whose contents is added to the displacement D_1 to form the effective address of operand 1.
- D_2 is an expression designating the displacement value of operand 2.
- B_2 is an expression designating a register whose contents is added to the displacement D_2 to form the effective address of operand 2.

The object code formed by this source code is:

OP CODE	L	B_1	D_1	B_2	D_2
0 7	8 15	16 19	20 31	32 35	36 47

The effective address ot operand 1 is formed by adding the contents of base register B_1 to the displacement value D_1 and the effective address of operand 2 is formed by adding the contents of base register B_2 to the displacement value D_2.

2.3.3.6. Storage to Storage Instructions (SS2)

The SS2 instructions are six bytes in length and are used to process data in storage where the operands are unequal in length. The maximum operand can be 16 bytes in length.

In this format, there are nine instructions:

SS2 Instructions	Mnemonic OP Codes
Add Packed Decimal	AP
Compare Packed Decimal	CP
Divide Packed Decimal	DP
Multiply Packed Decimal	MP
Move With Offset	MVO
Pack	PACK
Subtract Packed Decimal	SP
Unpack	UNPK
Zero Add Packed Decimal	ZAP

The source code format for these instructions is:

OP $D_1(L_1, B_1), D_2(L_2, B_2)$

where:

- OP — is the mnemonic operation code.
- D_1 — is an expression designating the displacement value of operand 1.
- L_1 — is a value designating the length of operand 1.
- B_1 — is an expression designating a register whose contents is added to the displacement D_1 to form the effective address of operand 1.
- D_2 — is an expression designating the displacement value of operand 2.
- L_2 — is a value designating the length of operand 2.
- B_2 — is an expression designating a register whose contents is added to the displacement D_2 to form the effective address of operand 2.

The object code formed by this source code is:

OP CODE	L_1	L_2	B_1	D_1	B_2	D_2
0 7	8 11	12 15	16 19	20 31	32 35	36 47

The effective address of operand 1 is formed by adding the contents of base register B_1 to the displacement value D_1 and the effective address of operand 2 is formed by adding the contents of base register B_2 to the displacement value D_2.

2.3.3.7. Instruction Formats

The basic instruction formats are illustrated in Figure 2–1 and an explanation of the symbols used is provided in Table 2–1. The subscripts 1, 2, and 3 indicate that the given field refers to operand 1, operand 2, or operand 3 respectively. In general, data are processed from operand 2 to operand 1, with the result often replacing the original contents of operand 1. Operand 3 is a special notation explained in the RS type instruction in 2.3.3.3.

Most instructions are two halfwords in length, however, the RR instructions require only one halfword and therefore are executed faster. The SS1 and SS2 instructions use three halfwords and require more execution time, which is dependent upon the length of the data processed by the instruction.

In the instruction formats in Figure 2–1, the first halfword contains the operation code, register numbers, immediate data, or length as specified by the individual instruction. The second halfword specifies either operand 1 or operand 2. The third halfword, when used, always specifies operand 2.

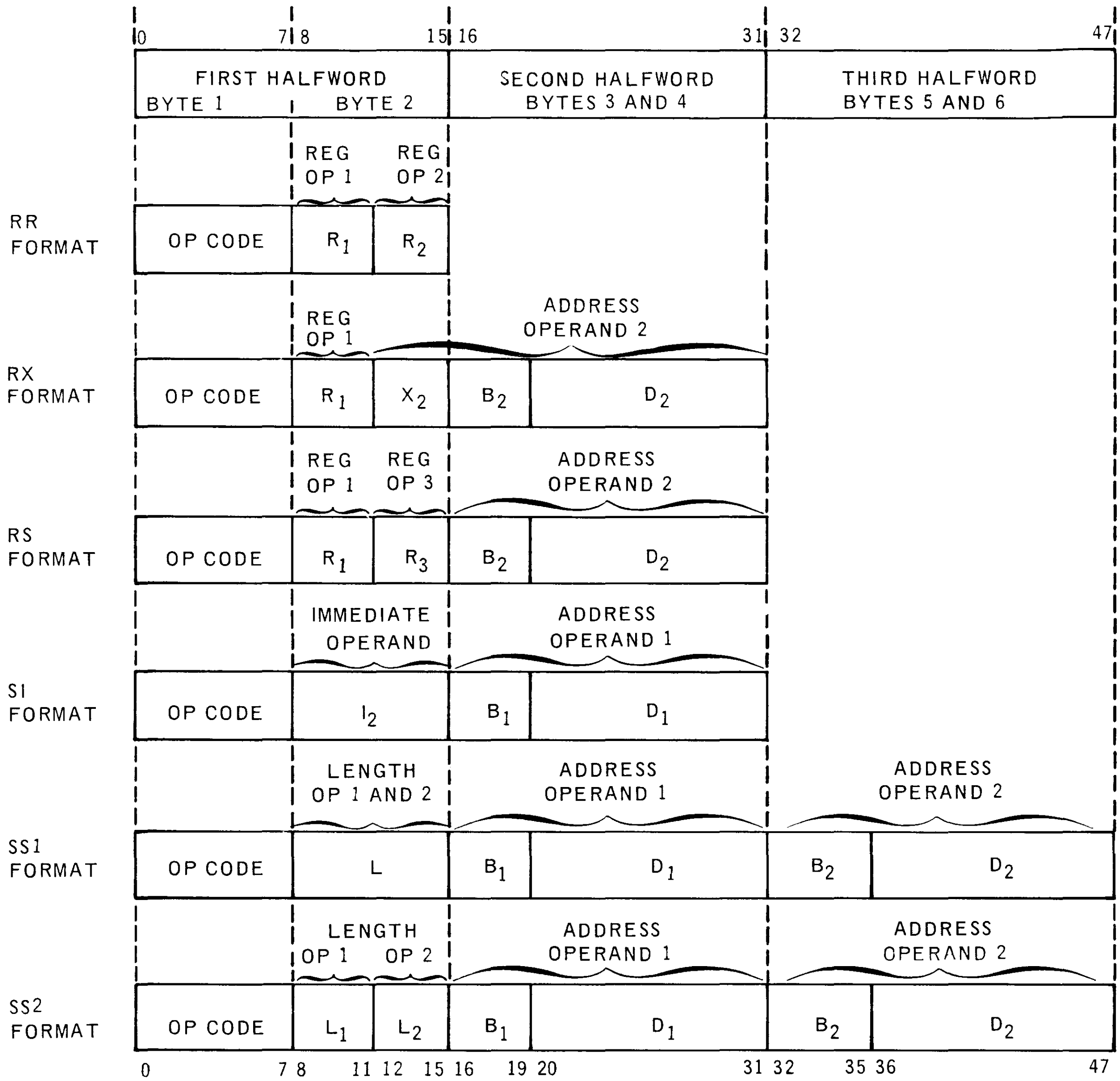


Figure 2–1. Basic Instruction Formats

The entries in the following table explain the meaning of the symbols used in describing the instruction formats in Figure 2–1.

SYMBOL	MEANING
OP CODE	Instruction operation code.
R_1	The number of the register addressed as operand 1, or a register which is the first register of a multi-register group.
R_2	The number of the register addressed as operand 2.
R_3	An expression representing a register which is the last register in a multiregister group.
X_2	The number of the register to be used as an index for operand 2 of an RX instruction.
I_2	The immediate data or device address used as operand 2 of a SI instruction.
L	The length of operands 1 and 2.
L_1	The length of operand 1.
L_2	The length of operand 2.
B_1	The base register for operand 1.
B_2	The base register for operand 2.
D_1	The displacement for operand 1.
D_2	The displacement for operand 2.
OP1	Operand 1
OP2	Operand 2

Table 2–1. Symbols Used to Describe Operand Formats

2.4. CONSOLE

The UNIVAC 9400 System console is a means for communication between the operator and operating system. The console consists of a keyboard, printer, switches, and indicators, which are housed in a platform attached to the processor as shown in Figure 2–2. The associated control and interface I/O logic is housed in the processor.

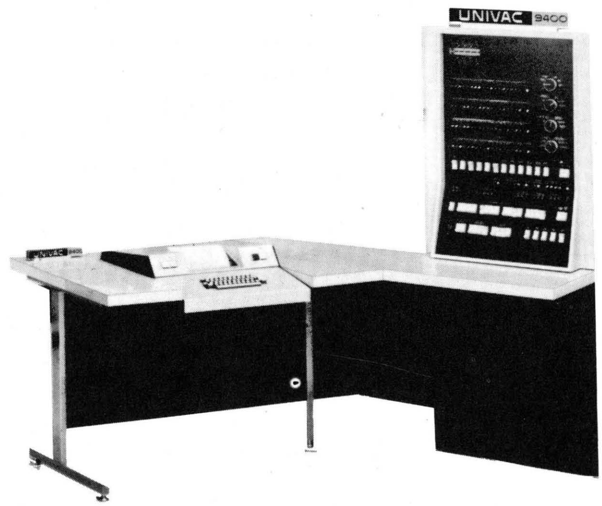


Figure 2–2. UNIVAC 9400 System Console

2.4.1. Keyboard Assembly and Control

The keyboard resembles a standard typewriter keyboard. This assembly consists of 46 keyin switches and a core logic encoding network. Pressing one of the keyin switches causes an EBCDIC coded character and a sprocket pulse to be transmitted in bit parallel from the encoding network to the keyboard control logic interface.

The keyboard assembly and control is activated by a read command from the UNIVAC 9400 Processor. This command is requested by pressing the ATTENTION switch located next to the printer, and it remains active throughout the input operation. During the input operation the printing of each character signifies that the particular character has been stored. The input operation can be terminated at any point by pressing the End-of-Message key (EOM STOP). If an incorrect character is typed, pressing the ERROR switch located next to the printer also terminates the input operation.

2.4.2. Printer Assembly and Control

The UNIVAC 9400 System printer is capable of printing 10 characters per second on 8–1/2 inch friction feed roll paper. Information in EBCDIC code from the processor is converted to ASCII code by the printer control logic and transmitted to the printer assembly in a bit serial sequence.

The printer executes three nonprint functions: carriage return, line feed, and space.

An override is provided to permit an output operation to terminate an input operation.

2.4.3. Switches and Indicators

There are four switches and indicators located on the console (two on each side of the printer). These controls and indicators permit the operator to initiate or terminate a message to the operating system, and to control printer paper movement.

The function of each of these is summarized as follows:

Switch/Indicator	Function
PAPER FEED (Sw.) (white)	This momentary contact switch activates the paper feed mechanism when pressed. Paper is fed continuously as long as the switch is pressed.
ATTENTION (Sw. and Ind.) (white)	This momentary contact switch is used to solicit a read command. The indicator is is extinguished when the attention status is received by the processor.
ERROR (Sw.) (white)	This momentary contact switch is used to terminate the input operation, and signals the processor that there was an incorrect character typed in from the keyboard.
READ (Ind.) (green)	This indicator is lit to indicate that the console is in the input mode.

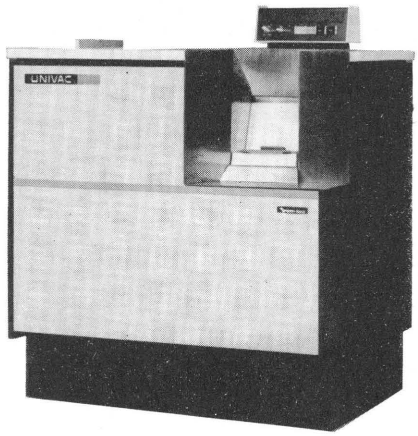
2.5. INPUT/OUTPUT DEVICES

A full line of peripheral devices and subsystems are available for use with the UNIVAC 9400 System. These devices and subsystems are:

- Card Reader
- Card Punch
- High Speed Printer
- Data Communications Subsystem
- UNIVAC 1004 or 1005 Subsystem
- UNIVAC 9200 or 9300 Subsystem
- UNISERVO Magnetic Tape Subsystems
- UNIVAC 8411 Direct Access Storage Subsystem

Descriptions of these devices and subsystems follow.

2.5.1. Card Reader

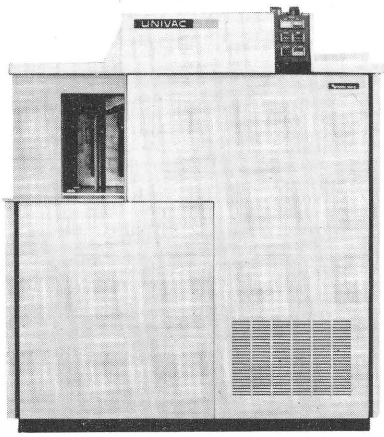


CHARACTERISTICS	
CARD READING SPEED	600 cpm
INPUT HOPPER CAPACITY	1200 cards
OUTPUT STACKER CAPACITY	1500 cards
READ MODES	Image mode – 160 6-bit characters per card EBCDIC – 80 characters per card
I/O CHANNELS	1 subchannel
OPTIONAL FEATURES	51- or 66-column short card feeds

The card reader operates at a rate of 600 cards per minute on a column by column basis. The read check feature is standard to ensure correct input. Information read from the card is transferred to the processor in either image mode or EBCDIC mode.

The card reader includes a self-contained control unit and synchronizer that regulates flow of data and control signals to and from the reader mechanism. This control unit is attached to the UNIVAC 9400 Central Processor by means of one of the eight subchannels provided in the standard multiplexer channel. A separate multiplexer subchannel is required for each card reader. The reader can have either a 51- or a 66-column stub card optional feature.

2.5.2. Card Punch



CHARACTERISTICS	
CARD PUNCHING SPEED	200 or 250 cpm
INPUT HOPPER CAPACITY	1000 cards
OUTPUT STACKER CAPACITY	2 stackers - 1000 cards per stacker
PUNCH MODES	Image mode - 160 6-bit characters per card Translate mode - 80 characters per card
I/O CHANNELS	1 subchannel
OPTIONAL FEATURE	Read before punching

The card punch operates at a rate of 200 or 250 cards per minute on a row by row basis (12 punching positions per card). Standard features include processing 80-column cards in either punched card code or main storage image code modes. Output cards can be directed to either stacker under control of the program.

The card punch includes a selfcontained control unit and synchronizer which regulates the flow of data and control signals to and from the punch mechanism. This control unit is attached to the UNIVAC 9400 Central Processor by means of one of the eight subchannels provided in the standard multiplexer channel. A separate multiplexer subchannel is required for each card punch.

The speed of the card punch may be increased to 250 cpm as an option. A second option is the inclusion of the read/punch feature. The read/punch option allows prepunched cards to be sensed and read into the punch buffer from a prepunch station.

Data to be punched may be transferred to the output buffer in either of two formats. In the translate mode each byte of data is translated by program into the corresponding card code. In the image mode data transferred has the two most significant bits stripped from the byte. The remaining six bits are punched in image code in the upper or lower half of a card column. The image mode provides the means for punching up to 160 characters of information into a single 80-column card. A post-punch read station checks card punching and directs error cards to a selected output stacker.

2.5.3. High Speed Printer

CHARACTERISTICS	
PRINTING SPEED	900 through 1100 lines per minute average or 1200 through 1600 lines per minute average
NUMBER OF CHARACTERS	132 character print positions
PRINTABLE CHARACTERS	63 characters plus space
HORIZONTAL SPACING	10 characters per inch
VERTICAL SPACING	6 or 8 lines per inch
FORM ADVANCE RATE	33 ips at 6 lines per inch spacing 22 ips at 8 lines per inch spacing
FORM WIDTH	4 to 22 inches
FORM LENGTH	1 to 22 inches
NUMBER OF FORM COPIES	Up to 6 part continuously sprocketed forms
FORM ADVANCE	Loop control
LINE ADVANCE	Single and double spacing under program control
FORMS ADVANCE	Up to 132 lines per command
SPEED OF FORM ADVANCE	12.5 – 5.2 (N – 1) if set at 6 lpi 12.5 – 5.7 (N – 1) if set at 8 lpi N = number of lines advanced

The high speed printer operates at an average speed of 900 through 1100 printed lines per minute. The high rate of 1100 lpm is maintained if the characters within a print line are included within any 49 contiguous characters of the 63 character set. As an option the print speed can be increased to 1200 through 1600 lines per minute average. The high rate of 1600 lpm is maintained if the characters within a print line are included within any 43 contiguous characters of the 63 character set.

The controlling and synchronizing circuitry, including the 132 character print buffer and the print mechanisms, are housed within a single free-standing cabinet. This complete printer subsystem is connected to the UNIVAC 9400 Central Processor by means of one of the eight subchannels of the standard multiplexer channel.

A forms container at the base of the unit houses the supply of forms being fed into the printer. Controls are provided to allow manual adjustment of paper tension, form thickness, paper alignment, vertical print positioning, horizontal print positioning, and advancement of forms. The forms handling mechanism is designed to eliminate buildup of static electricity.

2.5.4. Data Communication Subsystem

Communications oriented data processing is obtained through the use of the UNIVAC 9400 with from one to four Data Communication Subsystems (DCS) and a communications adapter. Each Data Communication Subsystem (DCS-1, DCS-4, or DCS-16) can accommodate one, four, or sixteen duplex lines, depending on the type. Each subsystem must be connected to one of the eight multiplexer subchannels. Circuitry is included in the DCS to control data transmission between the UNIVAC 9400 Central Processor and the line terminals. The DCS establishes the priority for individual lines when service is requested simultaneously. Also, the DCS signals the operating system when a data transfer between the DCS and the central processor has been lost.

The subsystem can take a variety of forms, depending upon the particular installation. The modular elements comprising the subsystem (that is, line terminal controller, line terminal, communication interface unit, automatic dialing adapter, and timing assembly) are described in the following paragraphs. A block diagram, Figure 2–3, follows showing interconnections between UNIVAC 9400 Central Processor, Data Communication Subsystem, and remote devices.

2.5.4.1. Line Terminal Controller

The Line Terminal Controller provides control for the various line terminals and the automatic dialing adapter.

2.5.4.2. Line Terminal

One or more line terminals can be used in simplex mode as data handlers for either sending or receiving information to or from the central processor. Pairs of line terminals can be interconnected to create a half duplex (send or receive) or full duplex (send and receive) communications environment. Several types of line terminals are available to provide low and medium speed asynchronous operation or to meet synchronous high speed requirements. Data characters may range from four to eight bits in size (level) depending on the model and mode of the line terminal used. Line terminal characteristics are summarized in Table 2–2.

FEATURE	DESCRIPTION
NUMBER of TERMINALS	Simplex mode – 2, 8, or 32 positions Duplex mode – 1, 4, or 16 positions
TYPE of OPERATION	Simplex, Half duplex, Full duplex
LINE TERMINAL DATA TRANSFER RATE	Low speed – 75 to 300 bits per second Medium speed – 300 to 1800 bits per second Synchronous (voice grade) – 2000 to 4800 bits per second Synchronous (broadband data terminal) – 40,800 or 50,000 bits per second Synchronous (Telpak† C) 230,400 or 250,000 bits per second
LINE TERMINAL CHARACTER SIZE	Low Speed – 5 or 8 bit Medium Speed – 4 to 8 bit Synchronous – 5 to 10 bit
SPECIAL APPLICATION	■ One dialing adapter can be attached to a DCS-4 (or four dialing adapters in the case of a DCS-16) to permit the processor to control an A.T. & T. 801 A or 801 C Automatic Calling Unit. ■ A parallel line terminal (input only) enables the UNIVAC 9400 System to interface with a Touch-Tone* telephone. It operates at 10 characters per second by means of an A.T. & T. 403 Data Set (2 out of 8 code). ■ A Dual Channel Access feature allows two UNIVAC 9400 multiplexer channels to access the same DCS-16 through operator intervention.

Table 2–2. Line Terminal Characteristics

† *Trademark of A.T. & T. Co.*

**Registered Service Mark of A.T. & T. Co.*

2.5.4.3. Communication Interface Unit

The communication interface unit is used to connect the line terminals with the common carrier lines. The available Communication Interfaces meet both the EIA RS 232B (Industry Standard Interface) and the MIL–STD–188B (Electrical Circuit Compatibility – Government) specifications. Each input/output line pair requires one Communication Interface Unit.

2.5.4.4. Asynchronous Timing Assembly

The asynchronous timing assembly provides a clock source for asynchronous line terminals. A single unit provides one baud rate for an entire Data Communication Subsystem. Asynchronous timing assemblies are available in baud rates up to 1800 baud. Each different speed asynchronous line terminal requires an asynchronous timing assembly. The maximum number of ATA's are four and six in the case of the DCS-4 and DCS-16.

2.5.4.5. Synchronous Timing Assembly

The synchronous timing assembly provides a clock source for synchronous communication. Clock signals are supplied for transmission, and provide the baud rates of 1200 to 40,800. These assemblies are needed only in a synchronous mode of operation where an asynchronous modem is used or where there is no external synchronized clock.

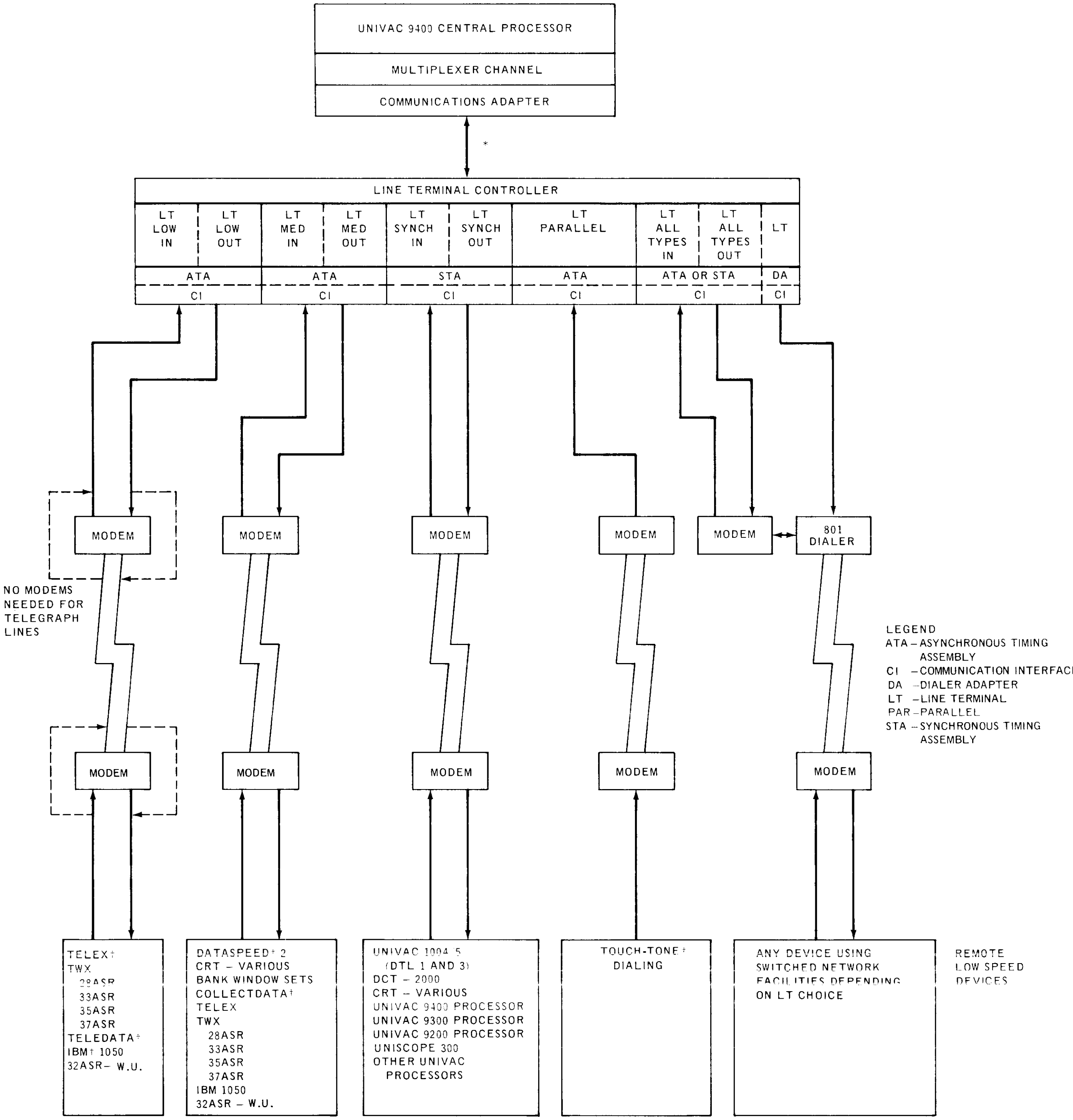
2.5.4.6. Configurations

The Data Communication Subsystem configurations have been classified into the following three types:

- DCS-1 provides either two simplex or one duplex position.
- DCS-4 provides either eight simplex or four duplex positions.
- DCS-16 provides either 32 simplex or 16 duplex positions.

The configuration shown in Figure 2–3 shows a UNIVAC Data Communication Subsystem connected to some of the more commonly used remote devices. It should be noted that each remote device requires a specific combination of the following units:

- Line Terminal Controller
- Line Terminals
- Timing Assemblies – Asynchronous Timing Assemblies and/or Synchronous Timing Assemblies
- Automatic Dialing Adapter
- Communication Interface Unit
- Modems

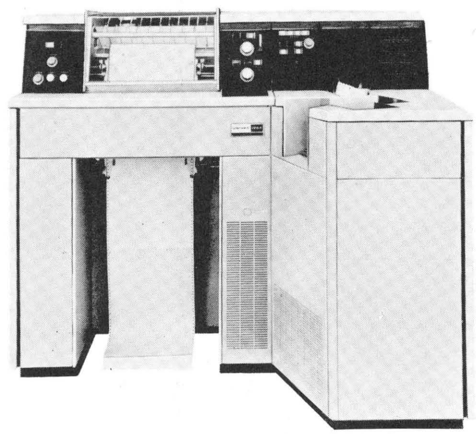


* A SECOND UNIVAC 9400 CENTRAL PROCESSOR CAN BE CONNECTED TO THE DATA COMMUNICATION SUBSYSTEM (DCS-16) THROUGH THE USE OF THE DUAL CHANNEL ADAPTER. THIS CONNECTION IS MANUALLY SWITCHED AND ALLOWS ONLY ONE PROCESSOR AT A TIME TO INTERFACE WITH THE DATA COMMUNICATION SUBSYSTEM.

† TELEX – TRADEMARK OF WESTERN UNION TELEGRAPH CO.
TELEDATA AND COLLECTDATA – TRADEMARKS OF FRIDEN, INC.
IBM – REGISTERED TRADEMARK OF INTERNATIONAL BUSINESS MACHINES CORP.
DATASPEED – TRADEMARK AND SERVICE MARK OF A.T. & T. CO.
TOUCH-TONE – REGISTERED SERVICE MARK OF A.T. & T. CO.

Figure 2–3. UNIVAC 9400 Central Processor with Data Communications Subsystems

2.5.5. UNIVAC 1004 or 1005 Subsystem



The UNIVAC 1004 or 1005 Subsystem is a powerful processing unit in its own right, with arithmetic, logical, and editing capabilities allied to a modular 961-character core storage. Standard peripheral units are a 400 cpm or 615 cpm card reader, a high speed printer operating at 400 lpm or 600 lpm with a 63 character set and 132 character print line width. A card punch operating at 200 cards per minute may also be included. External interrupt, punch stacker select, and alternate print code features are required when the UNIVAC 1004 or 1005 is used as a subsystem to the UNIVAC 9400 System.

Optional units for the offline configuration are a second bank of 961 characters of core storage, a second card reader (400 cpm), a card punch or card read/punch (200 cpm), UNIVAC 1001 Card Controller, paper tape reader (400 cps), a paper tape punch (110 cps), and one or two UNISERVO VI C Magnetic Tape Units.

A UNIVAC 1004 or 1005 Subsystem can be connected online, by means of one of the eight subchannels of the basic multiplexer channel, to the UNIVAC 9400 System To provide card reading, card punching, and printing capability.

A special plugboard is required when the UNIVAC 1004 or 1005 Subsystem is to be used online with the UNIVAC 9400 System. The 1004 or 1005 Subsystem retains its freestanding processing power when used in this configuration. At any time, the 1004 or 1005 Subsystem can be switched to offline mode, and it then operates as a standard freestanding model.

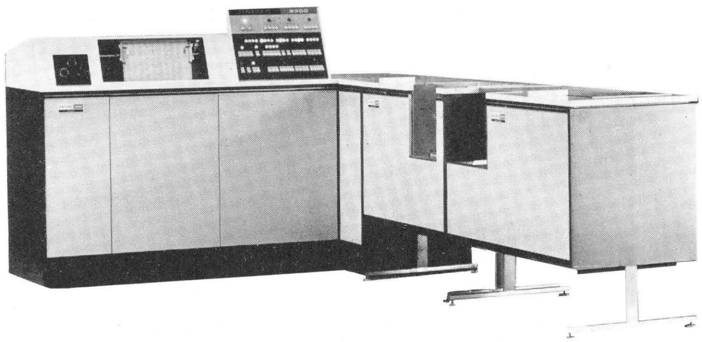
By attaching one of several types of line terminals, this subsystem can function as a remote data processor connected through a communications line to the UNIVAC 9400 System. Transmission at speeds of 2000, 2400 or 40,800 bauds is possible depending upon the types of line terminals and communications facility employed.

For detailed description of this subsystem, see UNIVAC 1004 and 1005 Description Manuals UP-4052, UT-2541, UT-2543, and UT-3927 (current versions).

UNIVAC 1004 OR 1005 SUBSYSTEM CHARACTERISTICS

ONLINE AND OFFLINE	
CARD READING SPEED	400 or 615 cpm
CARD PUNCHING SPEED	200 cpm
PRINTING SPEED	400 or 600 lpm
PRINTABLE CHARACTERS PER LINE	63 plus space
NUMBER OF CHARACTERS PER LINE	132
NUMBER OF LINES PER INCH	6 or 8
MAIN STORAGE	961 character positions
NUMBER OF INPUT/OUTPUT CHANNELS USED	1
OFFLINE ONLY	
CARD READING SPEED	400 cpm auxiliary reader 1000 or 2000 cpm with UNIVAC 1001 Card Controller
MAIN STORAGE	1922 character positions
MAGNETIC TAPE	One or two UNISERVO VI C Tape Units
MAGNETIC TAPE CODE CONVERSION	Available in UNIVAC 1004 Subsystem using the automatic translate feature
TAPE TRANSFER RATE	8,500; 23,700; and 34,200 characters per second
RECORDING DENSITY	200, 556 or 800 ppi
PAPER TAPE READ	400 characters per second
PAPER TAPE PUNCH	110 characters per second
REQUIRED FEATURES	
EXTERNAL INTERRUPT	
PUNCH STACKER SELECT	
ALTERNATE PRINT CODE	

2.5.6. UNIVAC 9200 or 9300 Subsystem



UNIVAC 9300 SUBSYSTEM CHARACTERISTICS

ONLINE AND OFFLINE	
MAIN STORAGE	8092 8-bit bytes of main storage
NUMBER OF CHANNELS	One general purpose channel
SPECIAL DEVICES	One channel to channel adapter
PRINTING	600 lpm printer – 120 or 132 print positions – (63 character font) or 1200 lpm printer – 120 or 132 print positions – (16 character font)
CARD READING SPEED	600 cpm card reader
STUB CARDS	Stub card read (51 and 66 column)
UNIVAC 1001 SUBSYSTEM	1000/2000 cpm (online)
CARD PUNCHING SPEED	75 to 200 cpm card punch (column) or 200 cpm card punch (row)
INSTRUCTION EXECUTION	52 microsecond add decimal instruction time (two 5 digit fields)
OFFLINE ONLY	
PREREAD	Preread in punch feedpath
STACKER SELECT	Punch stacker select control
UNIVAC 1001 SUBSYSTEM	1000/2000 cpm (online)
MAGNETIC TAPE	UNISERVO VI C Magnetic Tape Subsystem
ROW PUNCH	200 cpm card punch (row)
MAIN STORAGE CAPACITY	Maximum main storage of 32,768 8-bit bytes

The UNIVAC 9200 or 9300 Subsystem is a standard version of the UNIVAC 9200 or 9300 System. It is connected to the UNIVAC 9400 System by means of a channel to channel adapter attached to one of the eight subchannels provided in the standard multiplexer channel. The UNIVAC 9200 or 9300 Subsystem operating in an online mode to the UNIVAC 9400 System provides an economical combination of card reading, card punching, and printing for the UNIVAC 9400 System.

The UNIVAC 9200 or 9300 Subsystem is a freestanding offline system and may include all peripheral units and hardware features available in the UNIVAC 9200 or 9300 System configurations. The UNIVAC 9200 or 9300 Subsystem can be operated offline or online with the UNIVAC 9400 System. However, as a software convention, when the UNIVAC 9200 or 9300 Subsystem is used online with the UNIVAC 9400 System, operations must be performed according to the specific requirements and configurations of the UNIVAC 9400 System and the UNIVAC 9200 or 9300 Subsystem.

When operating in the online mode, the main storage of the UNIVAC 9200 or 9300 Subsystem contains a control program to perform the functions of card reading, card punching, and printing. I/O operations are initiated from the UNIVAC 9400 Central Processor by sending functional commands to the UNIVAC 9200 or 9300 Subsystem by means of the channel interface. The UNIVAC 9200 or 9300 Subsystem accepts these commands and the data transfers in the sequence they were requested. The UNIVAC 9200 or 9300 Subsystem instruction set is identical to that of the UNIVAC 9200 or 9300 System.

The operations of card reading, card punching, and printing, can be performed simultaneously by the UNIVAC 9200 or 9300 Subsystem in either the offline or online mode. The channel adapter is housed in the UNIVAC 9200 or 9300 Subsystem and precludes the inclusion of more than one other internal control unit in the UNIVAC 9200 or 9300 System.

For description of this subsystem, see UNIVAC 9200 System Description Manual UP-4086 (current version) and UNIVAC 9300 System Description Manual UP-4119 (current version).

2.5.7. UNISERVO Magnetic Tape Subsystems

Two types of magnetic tape subsystems are available for use with the UNIVAC 9400 Data Processing System. These are the UNISERVO 12/16 Subsystem and the UNISERVO VI C Subsystem. The UNISERVO 12/16 Subsystem can be attached to the central processor through one or two optional selector channels, while the UNISERVO VI C Subsystem must be attached through the multiplexer channel. The recording modes employed on all NRZI tape units are industry standard compatible, thereby ensuring interchangeability of data tapes with tape units of other processors.

2.5.7.1. UNISERVO 12/16 Magnetic Tape Subsystem

The UNISERVO 12/16 Subsystem consists of a synchronizer and any combination of UNISERVO 12 and 16 Magnetic Tape Units up to a maximum of 16 magnetic tape units. The magnetic tape units and special devices available in the UNISERVO 12/16 Subsystem provide a high degree of modularity and versatility. Included among these features are: two recording modes, two levels of recording, several pulse densities, two tape read/write speeds, and certain operating features to enhance the overall performance of the subsystem.

Both UNISERVO 12 and UNISERVO 16 Magnetic Tape Units are available in nine-track and seven-track models. The nine-track tape units produce the higher rate of throughput. In this record level, data is phase encoded in nine-bit frames across the width of the tape. Each frame contains eight data bits plus one parity bit (one byte). The coding scheme for nine-track tapes is EBCDIC, which is the internal code of the central processor.

An optional dual density feature can be added to the nine-track units. This feature enables each UNISERVO 16 Magnetic Tape Unit, or UNISERVO 12 Master Magnetic Tape Unit and the nine-track slave units it controls, to read and record data in the Non-Return to Zero (NRZI) mode at a density of 800 bytes per inch. When this feature is included, the recording mode and density are controlled through the program instructions.

In the seven-track recording level, data is recorded in seven-bit frames (NRZI only) across the width of the tape. In this case, each frame contains six data bits plus one parity bit. The synchronizer automatically converts the internal code of the central processor (EBCDIC) to or from six-bit (BCD) when seven-track tape units are used. A special data convert feature is available to record and read three eight-bit bytes of information in four six-bit frames, if EBCDIC to BCD translation is not desired.

Reading can take place with the tape moving either forward or backward. The ability to read magnetic tape backward in operations such as sorting and merging, gives to the system increased processing time that is normally used in rewinding the tape reel.

Writing can take place when the tape is moving in a forward direction only. The number of records to be included in each record block is established through directives to the operating system; the block size can be up to 4096 bytes in length. A minimum block size of 18 bytes is imposed when recording in seven or nine-track NRZI (compatible) mode to allow automatic detection of noise blocks.

An optional feature is available to produce concurrent execution of multiple tape functions in addition to rewind. This simultaneous operation feature must be added to the synchronizer, to the UNISERVO 12 Master Magnetic Tape Units, and to each UNISERVO 16 Magnetic Tape Unit which is to operate in parallel.

When this feature is included, the subsystem must be connected to the central processor through two channels. With this configuration, two UNISERVO 12 Magnetic Tape Units can be read at the same time, or one can be read while another unit is written. The UNISERVO 16 Magnetic Tape Units have the same capability; in addition, two tapes can be written at the same time.

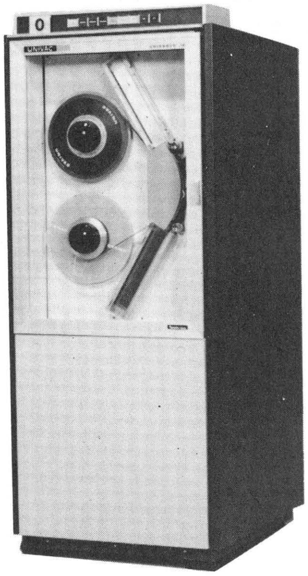
Each UNISERVO 12/16 Magnetic Tape Control Unit equipped with the simultaneous operation feature may be accessed by two selector channels. Therefore, a second central processor can be interfaced with the same bank of magnetic tape units. In this configuration, auxiliary storage is provided for two independent UNIVAC 9400 Systems. However, the magnetic tape subsystem can be accessed by only one processor at a time. Subsystem characteristics are summarized in Table 2–3.

A parity check is performed for each character and block (horizontal and vertical) read or written. A read-after-write head allows immediate verification of all data written.

MODEL	COMBINATION OF COMPONENTS	DATA TRANSFER RATE	PULSE DENSITY	RECORDING MODE	SIMULTANEOUS OPERATION
UNISERVO 16 Magnetic Tape Units (9-Track)	Standard	192,000 bytes per second	1600 ppi	Phase Encoded	Optional Control Device 1 per Control Unit and 1 per Magnetic Tape Unit
	Standard plus Dual Density Control Device 1 per Synchronizer and 1 per Tape Unit	96,000 bytes per second	800 ppi	NRZI	
UNISERVO 16 Magnetic Tape Units (7-Track)	Standard	96,000 characters per second	800 ppi	NRZI	Optional Control Device 1 per Control Unit and 1 per Magnetic Tape Unit
UNISERVO 12 Magnetic Tape Units (9-Track)	Standard	68,320 bytes per second	1600 ppi	Phase Encoded	Optional Control Device 1 per Control Unit and 1 per Master Magnetic Tape Unit
	Standard Plus Dual Density Control Device 1 per Synchronizer and 1 per Master Unit	34,160 bytes per second	800 ppi	NRZI	
UNISERVO 12 Magnetic Tape Units (7-Track)	Standard (density under control of the read/write instructions)	34,160 characters per second	800 ppi	NRZI	Optional Control Device 1 per Control Unit and 1 per Master Magnetic Tape Unit
		23,740 characters per second	556 ppi		
		8,540 characters per second	200 ppi		

Table 2–3. UNISERVO 12/16 Magnetic Tape Subsystem Characteristics

2.5.7.1.1. UNISERVO 12 Magnetic Tape Unit



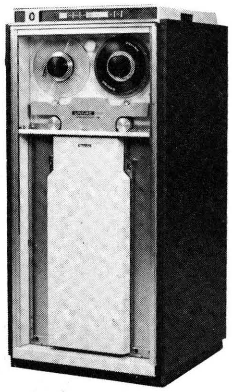
CHARACTERISTICS	
TAPE SPEED	42.7 inches/second
TAPE DIRECTION READING WRITING	 Forward or backward Forward
TAPE WIDTH	0.5 inch
TAPE LENGTH	2,400 feet (plastic)
THICKNESS	1.5 mils
BLOCK LENGTH	Variable
INTERBLOCK GAP	0.75 inch (7-track) 0.6 inch (9-track)
INTERBLOCK GAP TIME (7-TRACK)	17.6 milliseconds (non-stop) 23.6 milliseconds (start-stop)
INTERBLOCK GAP TIME (9-TRACK)	14.1 milliseconds (non-stop) 20.1 milliseconds (start-stop)
REVERSAL TIME	25 milliseconds
REWIND TIME	3 minutes (2,400 feet)
DUAL DENSITY	Optionally available
SIMULTANEOUS OPERATION	Optionally available

The UNISERVO 12 Magnetic Tape Unit is a lower cost tape handler with moderate speed. The master/slave concept is employed in the logic of these units. The master unit, having the power supply and control circuitry, governs up to three additional slave units. There are no differences in programming for master or slave units. UNISERVO 12 Magnetic Tape Units can be added to the subsystem one at a time providing the first unit is a master unit. A combination of three slave units and a master unit is called a quad.

UNISERVO 12 Magnetic Tape Units are available in both nine-track and seven-track models. The nine-track tape units read and record data in the phase encoded mode at a density of 1600 bytes per inch. If the optional dual density feature is added, the nine-track units can also record at 800 bytes per inch density. The seven-track UNISERVO 12 Magnetic Tape Units read and record in NRZI mode only. These units can be programmed to read and record data at densities of 200, 556, or 800 frames per inch.

The physical tape passing speed is 42.7 inches per second, giving nine-track tape a transfer rate of 68,320 bytes per second in the phase encoded mode, or 34,160 bytes per second in NRZI mode. The seven-track tape units transfer data in NRZI mode at 34,160, 23,740 or 8,540 characters per second depending upon the density selected.

2.5.7.1.2. UNISERVO 16 Magnetic Tape Unit



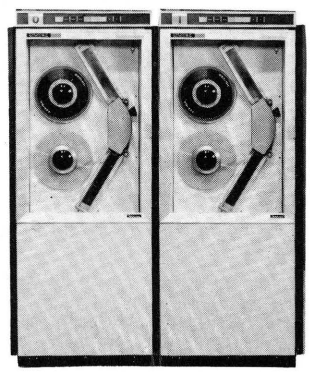
CHARACTERISTICS	
TAPE SPEED	120 inches/second
TAPE DIRECTION READING WRITING	 Forward or backward Forward
TAPE WIDTH	0.5 inch
TAPE LENGTH	2,400 feet (plastic)
THICKNESS	1.5 mils
BLOCK LENGTH	Variable
INTERBLOCK GAP	0.75 inch (7-track) 0.6 inch (9-track)
INTERBLOCK GAP TIME (7-TRACK)	6.25 milliseconds (non-stop) 9.25 milliseconds (start-stop)
INTERBLOCK GAP TIME (9-TRACK)	5.0 milliseconds (non-stop) 8.0 milliseconds (start-stop)
REVERSAL TIME	10 milliseconds
REWIND TIME	2 minutes (2,400 feet)
DUAL DENSITY	Optionally available
SIMULTANEOUS OPERATION	Optionally available

The UNISERVO 16 Magnetic Tape Unit is a high performance tape unit. These units can be added to the subsystem individually until the maximum of 16 is reached. UNISERVO 16 Magnetic Tape Units are available in nine-track and seven-track models.

The UNISERVO 16 Magnetic Tape Unit reads and records data in phase encoded mode at 1600 bytes per inch. If the optional dual density feature is added, the nine-track units can also record data in NRZI mode at 800 bytes per inch. The seven-track model reads and records data in NRZI mode at a density of 800, 556, or 200 characters per inch, depending on the density selected.

Physical tape speed is 120 inches per second, giving the nine-track tape unit a maximum transfer rate of 192,000 bytes per second in the phase encoded mode, or 96,000 bytes per second when the NRZI mode is used. Seven-track tape is recorded only in NRZI mode with a data transfer rate of 96,000, 66,720, or 24,000 characters per second, depending on the density selected.

2.5.7.2. UNISERVO VI C Magnetic Tape Subsystem



UNISERVO VI C MAGNETIC TAPE UNIT CHARACTERISTICS	
TAPE SPEED	42.7 inches/second
TAPE DIRECTION READING WRITING	 Forward or backward Forward
TAPE WIDTH	0.5 inch
TAPE LENGTH	2,400 feet (plastic)
THICKNESS	1.5 mils
BLOCK LENGTH	Variable
INTERBLOCK GAP	0.75 inch (7-track) 0.6 inch (9-track)
INTERBLOCK GAP TIME	14.1 milliseconds (non-stop) 20.1 milliseconds (start-stop)
REVERSAL TIME	25 milliseconds
REWIND TIME	3 minutes (2,400 feet)
DUAL DENSITY	Optionally available
SIMULTANEOUS OPERATION	Optionally available

The UNISERVO VI C Subsystem is a low cost magnetic tape subsystem suitable for use with the UNIVAC 9400 System. This subsystem consists of a synchronizer control unit and from two to eight magnetic tape units. The master/slave concept is employed in the logic of the UNISERVO VI C Subsystem. The master unit, having the power supply and control circuitry governs the functions of up to three slave magnetic tape units. However, all units are treated alike from a programming or operating system viewpoint. Subsystem characteristics are summarized in Table 2–4.

The synchronizer control, one master tape unit, and one slave unit are housed in a single cabinet. Additional tape units must be added as needed. A second master unit is needed when more than four tape units are used.

UNISERVO VI C Magnetic Tape Units are available in either nine-track or seven-track models. The nine-track model reads and records data in eight bit EBCDIC code. Each frame recorded across the width of the tape contains eight data bits plus a parity bit. Data is recorded in NRZI mode at a density of 800 bytes per inch.

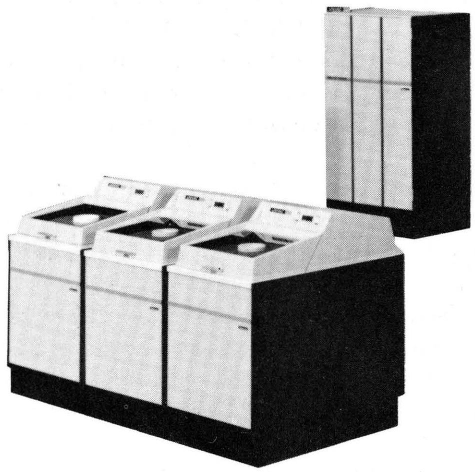
The seven-track models read and record data in six bit BCD code. Each frame recorded across the width of the tape contains six data bits plus a parity bit. The data is automatically converted from EBCDIC to BCD during the recording process and conversely during reading. If this automatic translation is not desired, an optional data conversion feature is available to record three eight bit-bytes in four six-bit frames leaving the data in EBCDIC code. This feature also causes four six-bit frames to be read into the central processor main storage as three eight-bit bytes.

Data can be read or recorded by seven-track UNISERVO VI C Magnetic Tape Units in NRZI mode at densities of 800, 556, or 200 characters per inch. The tape instructions, together with a manual switch on each tape unit, establish the applicable density.

MODEL	COMBINATION OF COMPONENTS	DATA TRANSFER RATE	PULSE DENSITY	RECORDING MODE
UNISERVO VI C (9-track)	Standard	34,160 bytes per second	800 ppi	NRZI
	Standard plus 7-track feature	34,160 characters per second	800 ppi	NRZI
UNISERVO VI C (7-track)	Standard	34,160 characters per second	800 ppi	NRZI
	Standard	23,740 characters per second	556 ppi	NRZI
	Standard	8,540 characters per second	200 ppi	NRZI

Table 2–4. UNISERVO VI C Magnetic Tape Subsystem Characteristics

2.5.8. UNIVAC 8411 Direct Access Storage Subsystem



CHARACTERISTICS	
NUMBER OF DRIVES	1–8
NUMBER OF DISC DRIVES PER CABINET	1
NUMBER OF R/W HEAD ACCESSOR MECHANISMS	1
NUMBER OF R/W HEADS PER DISC DRIVE	10
NUMBER OF TRACKS PER DISC SURFACE	203
NUMBER OF RECORDING SURFACES PER DISC DRIVE	10
NUMBER OF ADDRESSABLE TRACKS PER SURFACE	200
NUMBER OF ADDRESSABLE TRACKS PER DISC DRIVE	2,000
MAXIMUM NUMBER OF BYTES PER TRACK	3,625
CAPACITY (8-BIT BYTES PER DISC PACK)	7,250,000
MINIMUM ARM POSITIONING TIME	25 milliseconds
AVERAGE ARM POSITIONING TIME	75 milliseconds
MAXIMUM ARM POSITIONING TIME	135 milliseconds
MINIMUM SEARCH LATENCY TIME	0 milliseconds
AVERAGE SEARCH LATENCY TIME	87.5 milliseconds
MAXIMUM SEARCH LATENCY TIME	160 milliseconds
DISC DRIVE SPEED	2,400 rpm
STORAGE TRANSFER RATE	156,000 bytes per second

The UNIVAC 8411 Direct Access Storage Subsystem is available for use with the UNIVAC 9400 Data Processing System. This Subsystem can be attached to the central processor through an optional selector channel. The UNIVAC 8411 Direct Access Storage Subsystem offers many advantages in standard data processing as well as in communication operation, especially in applications where rapid file processing are more prevalent. Large capacity is combined with rapid accessibility to provide convenient intermediate storage.

This subsystem allows an installation to make use of an extensive operating system without undue main storage utilization or loss of operating efficiency. The rapid access time of the UNIVAC 8411 Direct Access Storage Subsystem permits lesser used control segments to be stored outside of main storage, and read a section at a time only when required. This arrangement affords efficient usage of main storage. When magnetic disc packs are included, the handling, access, and transfer time required for compiling or assembling programs and for input/output operations can be reduced. The use of the disc subsystem frees the magnetic tape units for exclusive use to meet primary input/output data demands.

The UNIVAC 8411 Direct Access Storage Subsystem consists of one control unit with from one to eight UNIVAC 8411 Direct Access Storage Drives. The UNIVAC 8411 Direct Access Storage Drive(s) and control unit are contained in separate cabinets with each storage device connected to the control unit.

The UNIVAC 8411 Direct Access Storage Subsystem provides the UNIVAC 9400 System with storage for nonvolatile, variable length records to a maximum capacity of 7,250,000 bytes per disc pack with a total subsystem storage capacity of 58,000,000 bytes.

Each disc pack contains six discs with the data recorded on the ten inside surfaces. Ten read/write heads are mounted on a single accessor mechanism which moves the ten heads in unison between the periphery and the central area of the disc. The accessor mechanism can assume one of 203 positions across the disc surface; this simultaneous head movement on the ten disc surfaces creates 200 addressable data recording cylinders in the disc pack, with 3 cylinders reserved for replacement tracks. Each cylinder contains 10 tracks numbered 0 through 9. The addressing of an individual track in the cylinder is by cylinder number (000–199) and by read/write head number (0–9).

3. PROGRAMMED SYSTEMS SUPPORT

3.1. OPERATING SYSTEM

The software systems support provided for the UNIVAC 9400 System has been designed to meet the total computing and operating requirements of today's advanced data processing problems. The data processing capabilities of the system are carefully controlled and directed by means of comprehensive software packages which provide strong operating links between the UNIVAC 9400 System's power and the user's data processing problems. The degree of effective utilization of any computing system is in direct proportion to the scope and versatility of the software. In developing the 9400 System, Univac has drawn upon many years of experience in multiprogramming, and communications oriented systems. The result is a system easy to operate and easy to use; yet it is a system which ensures user program integrity in a demanding business environment.

3.1.1. Supervisor

The Supervisor control program within the operating system of the UNIVAC 9400 System is responsible for the administration of control commands to the computer system. It coordinates all input/output activity and other services provided by components of the operating system. It schedules processing time to the many jobs in a multiprogramming environment, then supervises the execution of these jobs.

The major unit of work in the 9400 System is considered a job; each job can be subdivided into job steps or individual programs. The Supervisor exercises a serial control over the job steps and at the same time applies a parallel control to the various jobs so that they may be executed concurrently.

The Supervisor control program is designed in such a way that it is easy to use in any data processing situation, provides maximum utilization of the computer's facilities, and handles problems directly, efficiently, and promptly with the major portion of time allotted to the user's programs.

3.1.1.1. Multiprogramming

Optimum utilization of computer time is achieved through multiprogramming. Up to five problem programs may be running concurrently with other Supervisory control functions.

Five program priorities are provided by the Supervisor control program, two of which are used by the operating system and three by problem programs. The three types of problem program and associated priority levels follow:

(1) Communications type programs. The highest priority level available to the user is intended for time critical programs such as those involving communications processing.

(2) Batch programs with high input/output utilization. The majority of batch type programs involving frequent input/output use are assigned to the second level of user priority.

(3) Batch programs with low input/output utilization. User programs which are primarily of the computational type with low input/output use are assigned to the lowest user priority level.

The user can designate priority levels in the job control stream by specifying priority levels 1, 2, or 3 according to the known requirements of the problem programs.

3.1.1.2. Interval Timer and Simulated Day Clock Services

The interval timer is a standard hardware feature on the UNIVAC 9400 Central Processor. A number of services in the operating system make use of the interval timer. The Supervisor provides the following timer affiliated services.

- Time allocation – this service is provided to distribute processing time among the various programs at each priority level. The length of the time intervals for program execution at each priority level is specified by the user at systems generation time.

- Job timeouts – this service is used with time allocation. Job time limits are user submitted in the job control stream to prevent unauthorized domination of the computing system by a single program.

- Time of day – a day clock is simulated by the Supervisor and is accessible to the programmer.

- Programmed timeouts – the programmer can request to be notified upon the expiration of an interval of time specified by him.

3.1.1.3. Time Allocation

The division of processing time to the jobs on a given program priority level is controlled through time allocation, that is, each job is given equal time on a cyclic basis so that all jobs on a given priority level may operate concurrently and share the full processing power of the computer.

Processing time is allocated to programs in intervals ranging from one millisecond to four seconds. If a program does not voluntarily give up control of the central processor before its allocated time interval expires, an interrupt is set and the Supervisor control program intervenes to evaluate the utilization of the computing system. An accumulation of processing time used for each program is maintained and compared against an estimated maximum time requirement before each issuance of processing time. If a program exceeds its estimated maximum time requirement, the operator is notified to either initiate abort procedures or allocate more time. As an optional feature, the Supervisor can automatically initiate abort procedures.

The length of the intervals allocated for program execution at each priority level are selected by the user at systems generation time; the estimated maximum time requirement for each program is selected by the user for that program when it is loaded.

3.1.1.4. Shared Routine Coding

Routines may be used by more than one problem program in the multiprogramming environment. Control and linkage is a function of the Supervisor, employing two techniques. These techniques are:

- Serially reusable routines – request for use are queued, then executed in sequence of job priority.

- Re-entrant coding – allows for concurrent execution of routines by two or more requesting jobs. This facility is available only for the data management portions of the operating system.

3.1.1.5. Operator Communication

Facilities are provided to permit communication between the operator and Supervisor, or problem program. All messages are printed at the operator's console and may be automatically time stamped by means of the operator communications functions of the Supervisor.

Supervisor messages to the operator are full text messages either stating that operator intervention or choice is required before processing can resume, or providing information for inclusion in the system's chronological log. Operator messages to the Supervisor are either replies to previous messages issued by the Supervisor, or commands directing the Supervisor in its operations. The format and content of the reply depends upon the requirements of the Supervisor message being answered.

The problem program may have two-way communications with the operator. The console can be treated as a peripheral device shared among problem programs. An operator communication interrupt routine must be provided and the address of this routine must be available to the Supervisor in order for the operator to direct an unsolicited message to the problem program.

3.1.1.6. Transient Area Management

Transient routines are filed in the main storage image library on the resident direct access device and loaded into main storage only when needed by the operating system or problem programs. An index is maintained in the auxiliary storage media of all transient routines stored in the library. The more frequently used transient routines are indexed in main storage. The transient area scheduling routine, a part of the resident portion of the Supervisor control program, maintains these indices and coordinates all activity between calling routines and transient routines.

Transient routines are considered as logical extensions of the calling programs and are executed at the same priority level as their respective calling programs. All transient routines are designed to operate within the space provided by one transient area. When a transient routine exceeds the size of the transient area, it loads in program overlays using the LOAD and FETCH macro instructions; therefore, the effective size of transient routines is virtually unlimited.

3.1.1.7. Interrupt Handling

The decoding and processing of all machine interrupts are handled automatically by the interrupt handler function of the resident Supervisor and are of no concern to the programmer when writing a problem program. There are seven types of interrupts and they are processed by the interrupt handler in the order of their priority:

- Supervisor Call (SVC) – initiated by encountering a Supervisor Call instruction in the problem program. The SVC instruction is used as a means of communication from the problem program to the Supervisor.

- Program Exception – this interrupt occurs under the following conditions:
  - Illegal operation code
  - Privileged operation attempted in the problem state
  - Limits Register check
  - Low order storage check
  - Binary arithmetic overflow
  - Decimal arithmetic overflow

- Interval Timer – when the interval timer is decremented to binary zeros, a hardware interrupt occurs and the timer continues to count through zero.

- Selector Channel 1 – the fourth level of interrupt and the highest associated with I/O operations. An interrupt occurs whenever channel 1 requires servicing by the software.

- Selector Channel 2 – the fifth level of interrupt and second for I/O operations. An interrupt occurs whenever channel 2 requires servicing by the software.

- Shared Multiplexer Subchannel – level six interrupt. An interrupt occurs whenever a low speed onsite peripheral device requires servicing by the software.

- Nonshared Multiplexer Subchannel – lowest level interrupt. An interrupt occurs whenever a remote line terminal requires servicing by the software.

3.1.1.8. Input/Output Control System (IOCS)

All activity between the central processor unit and its peripheral devices is controlled by a group of routines known as the channel scheduler, which provides I/O queuing, dispatching, posting, and error detecting services.

The programmer can communicate directly with the channel scheduler by using physical IOCS macro instruction. Whenever these macros are used, the programmer must supply the channel command words and provide any of the logical functions required by the problem program such as blocking and deblocking records, checking for wrong length records, swapping buffer areas, and detecting and bypassing checkpoint records if they are interspersed with data records.

3.1.1.9. Direct Access Auxiliary Storage

The availability of direct access (disc) auxiliary storage for use by the operating system has increased the processing power of the UNIVAC 9400 System. The designers of the operating system have, at every opportunity, taken advantage of this feature. The two functional types of routines in the Supervisor control program are:

- Resident Routines

  This category comprises those routines frequently used by the Supervisor, and require permanent residence in main storage. This category is referred to as the resident portion of the Supervisor control program.

- Transient Routines

  This category comprises those routines not frequently used and which can be kept in direct access auxiliary storage and loaded into main storage only when needed. Routines in this category are designed to function in special main storage transient areas reserved for the operating system. When needed, a transient routine is located in an index of transient routines, read from the resident direct access auxiliary storage into a main storage transient area, executed as an extension of the requesting program, and, if not needed immediately by another program, released from the transient area. Transient routines are serially reusable and self-initializing.

3.1.2. Data Management

Data management is part of the software that provides a convenient and easy to use interface between problem programs and the hardware oriented I/O portions of the Supervisor. The data management facilities provide organizational benefits such as record blocking and deblocking, buffering, data validation, and label processing.

There are many aspects of file processing that are common although the data contained in the files is vastly different. In order to exploit the similarity evident in so many file processing programs and utilize this similarity to a much greater degree than on other systems, the data management facilities consist of subprograms of common code that are part of the operating system. At systems generation time, the user selects those subprograms, or parts of subprograms that are needed to process the specific files. The result is a saving in storage space by having only one file processor program for all the files rather than several repetitive file processor programs tailored to each specific file. In order to further increase efficiency, specialized linkages are provided between the user's file table, generated by the DEFINE THE FILE macros, and the subprograms of common coding. These linkages are generated in sequences that are peculiar to the DEFINE THE FILE call. This eliminates large areas of slow interpretative code. Thus, the user has the advantage of common code balanced in an efficient combination of declarative and interpretative code.

The initiation and termination procedures for file processing (OPEN and CLOSE) are necessarily long and exacting, and involve a large amount of coding. In the UNIVAC 9400 System, these macros exist as system transient routines. The space used by the OPEN transient coding is used once and then overlaid with other Supervisor control coding; this coding is in turn overlaid with the CLOSE transient coding to terminate the file.

The user has several methods by which a file can be accessed in the UNIVAC 9400 System. The method chosen is determined by the overall use of the file in the application, the composition of the file to be processed, the time available to reach each record, and the device to be used as a storage medium. A discussion of the various access methods follow.

3.1.2.1. Sequential Access Method

All files whose records follow one another in a serial or physically adjacent manner are handled by a sequential access method. Devices that have sequential files include magnetic tapes, card readers, printers, punches, and discs.

A file definition macro is supplied for each appropriate device. Record or block level handling macros are also supplied.

3.1.2.2. Nonsequential Access Method

Nonsequential processing is suitable for direct access storage devices only. Specific access methods implemented for the UNIVAC 9400 System are the direct access method and the indexed sequential access method. File declaration macros are supplied for each access method as well as a variety of processing macros suitable for handling the data.

3.1.2.2.1. Direct Access Method

The direct access method permits a data file on a direct access storage device to be accessed by record where the record address is specified by the user. The record address is specified by means of an identification address or a key field. The macro calls permit records to be read, written, replaced, deleted, or inserted at specific addresses within the file.

3.1.2.2.2. Indexed Sequential Access Method

The indexed sequential access method is a method whereby the user specifies a key field to access his records. The file is created from presorted input records and stored in conjunction with a hierarchy of indices. The user key is stored along with the direct access storage device address in an index which is sequential on the basis of the key number. The index feature enables advantage to be taken of the direct access properties of the hardware in that any record can be accessed by finding its location in the index and then going to that location.

3.1.3. Job Control

Data and programs to be processed may be introduced to the UNIVAC 9400 System as a group of programs, each program containing its own control information. This approach to work submission is defined as a job control stream. The Job control program in the Supervisor controls the execution of these work tasks, transition between programs, restarting of programs, and termination of jobs.

As the job control streams are submitted to the UNIVAC 9400 System, they are stored in direct access storage files for subsequent selection and execution. The system operator selects jobs to be evaluated for available facilities. If facilities are sufficient and available, the job is selected and queued ready for processing time.

Transition from the normal termination of one job control stream to another is automatic with messages to the operator for mounting of tapes, setting up the printer and performing other physical adjustments to the system. A procedure is available to provide automatic checkpoints for restarting a job.

In addition to normal job termination, job suspension and cancellation are available through operator or program intervention.

3.1.4. Message Control Program

The Message control program is a generated input/output program that provides a logical input/output control system (IOCS) for communications processing. A problem program can use the Message control program to control the input/output from communication lines with the same ease as the input/output control for either tape or disc.

In addition to the normal GET and PUT type macros, macros are provided to construct a complete message control program that controls the flow of messages to and from a variety of commonly used remote terminals and the user processing program. Messages of fixed or variable length flow from the remote terminals to be queued in main storage. From main storage, messages are then used by the problem program. If direct access storage is used, the flow of messages is from remote terminals to direct access storage to be queued in order specified by the user. From direct access storage messages are then used by the problem program.

Since the Message control program is generated by macros, the user can specify the communications equipment configuration and the number and size of buffer areas to be used in both main storage and direct access storage. Also, the user can specify functions such as: message code translation, routing, time stamping, sequencing of messages, and error checking. The ability to specify modules of the Message control program allows the user to create a unique Message control program for a specific installation's requirements.

3.2. LANGUAGE PROCESSORS

Language Processors are provided to allow the user of the UNIVAC 9400 System a great deal of flexibility in preparing programs. Programs may be written in Assembly language, COBOL, and FORTRAN, or through the facilities provided by the Report Program Generator.

The symbolic language of the Assembler provides a simple and convenient method of writing programs through the use of mnemonic instruction codes, Assembler directives, data generation instructions, Assembly time modification instructions, and the powerful macro generation calls.

The user may elect to write programs in the language provided by COBOL or FORTRAN. COBOL provides the user with a language for data processing problem solutions involving maintenance and processing of large volumes of files while FORTRAN provides a language for solving scientific or mathematical problems.

The Report Program Generator provides for the automatic preparation of accurate report programs. The user describes the input records, the calculations to be performed, and the output records. The Report Program Generator produces an object program that will prepare the desired report.

3.2.1. Assembler

The symbolic language of the UNIVAC 9400 Assembler is an extension of the simple and convenient language used in the UNIVAC 9200 and 9300 Computing Systems. It includes a sophisticated variety of operators which allow the fabrication of desired memory address fields based on information generated at assembly time. The instruction codes are assigned mnemonics which represent the hardware function in each instruction.

Assembler directives provide the programmer with the ability to generate data, data words, values, addresses, and storage areas, and to provide him with the facilities to modify assembly control based on specific values at assembly time. A macro facility is provided to reduce coding errors by calling into the system precoded and pretested subprograms.

Output from the assembler run consists of a complete listing of symbolic coding, generated object coding diagnostic messages, and error indicators. A relocatable object module is produced which is suitable for linking to other modules prior to loading for subsequent execution. The output object module may be produced in punched cards, written on tape, or recorded on the disc file.

The symbolic format for writing the Assembler instructions consists of three basic symbolic fields; use of these fields requires conformity to simple rules in order that efficient translation of symbolic to object code may be performed.

- The label field may contain a symbolic name which is used to provide an entry point or a label for a block of data or a block of instructions.

- The operation field must contain the symbolic machine instruction code.

- The operand field provides for a variety of uses which may range from simple to complex specifications.

Combining names, grouping marks, arithmetic operators, logical operators, relational operators, and constants into operand expressions, make possible the solution of highly sophisticated coding problems. Operand expressions gain power by being able to include location counters references, and literals.

A wide range of data types are provided for constant generation and storage definition. Binary, hexadecimal, decimal, and character terms may be used to specify absolute values in the source code.

3.2.2. COBOL (COmmon Business Oriented Language)

COBOL is a programming language oriented towards the solving of data processing problems in business applications. The language is specified using a notation similar to the English language, rather than a notation which considers the technical aspects of a particular computing system. The source programs are therefore compatible among computer systems that accept the COBOL language. Each of these systems would provide a COBOL compiler to translate the COBOL source program into a machine oriented object program. The ability to advance from one generation of equipment to another in a logical, orderly, and rapid manner is assured through this ‘‘machine independence’’. Source programs written in COBOL consists of four major divisions as follows:

- Identification Division – contains information which identifies the source program and the output of a compilation. In addition, the author, installation, etc., may also be identified in this division.

- Environment Division – specifies a standard method of expressing those aspects of a data processing problem that are dependent upon the physical characteristics of a specific computer. This division also allows the specification of the compiling computer configuration, object computer configuration, special hardware characteristics, input/output control techniques, etc.

- Data Division – describes the data that the object program is to accept as input, manipulate, create, or produce as output. This division is further divided into sections to facilitate the description of data which is contained in input or output files, developed during the course of running the program, or which is preset or constant information to be used in the object program.

- Procedure Division – describes the logical steps that must be taken in the solution of the data processing problem. The verbs that describe these steps can be categorized as follows: input/output, arithmetic, data manipulating, sequence control, and compiler directing.

3.2.2.1. COBOL Compilers

The UNIVAC 9400 System provides two COBOL compiler programs. The first is intended for tape system configurations; the second, an enhanced version, is intended for direct access storage configurations. These compilers conform to specifications defined by CODASYL (COBOL Committee of the Conference on Data System Languages) publications. In addition, the features provided are in accordance with the proposed COBOL standards of USASI (United States of America Standards Institute).

3.2.2.2. Tape Compiler

The tape compiler includes those features of COBOL required by USASI level one elements of the following modules: Nucleus, Sequential Access, Segmentation, and Table Handling. In some cases features provided are above level one.

The Nucleus module provides, in part, the following language features:

– data editing capabilities

– numeric literals of 18 digits in length

– nonnumeric literals of 132 characters in length

– use of figurative constants (ZERO, SPACE, HIGH-VALUE, LOW-VALUE, QUOTE)

– special register TALLY

– currency sign and decimal point substitution

– redefinition of data areas

– preset values in data description

– arithmetic verbs (ADD, SUBTRACT, MULTIPLY, DIVIDE)

– input/output verbs (ACCEPT, DISPLAY)

– data manipulation verbs (EXAMINE, MOVE)

– sequence control verbs (ALTER, EXIT, GO TO, PERFORM, IF, STOP)

– compiler directing verbs (ENTER, NOTE).

The Sequential Access module provides the following language features:

- specification of sequential input/output files
- provision for use of all standard I/O devices (card reader, card punch, printer, tapes)
- use of single or blocked record files
- use of alternate areas for input/output buffering
- file labeling
- input/output verbs (OPEN, CLOSE, READ, WRITE).

The Segmentation module provides the following language features:

- segmentation of the object program into a fixed control segment and secondary segment overlays.

The Table Handling module provides the following language features:

- description of data tables
- allows for table arrays of three dimensions
- use of subscripting and indexing for table referencing
- SET verb for establishing reference points within tables.

3.2.2.3. Direct Access Storage Compiler

The direct access storage compiler includes all of the features of the tape version as previously described. This compiler also contains additional features of COBOL as required in USASI level two of Nucleus and Table Handling modules, and level one of Random Access and Sort.

The additional features of the Nucleus module include:

- use of characters for arithmetic and relational operators (+, –, =, etc.)
- use of qualification for name uniqueness when necessary
- use of figurative constant form ALL and OR
- nested redefinition of data areas
- ability to RENAME data areas
- logical operators AND, OR, NOT
- abbreviated formats in relation condition
- the CORRESPONDING feature of ADD, SUBTRACT, and MOVE verbs
- expansion of ALTER verb to allow series of specifications
- the COMPUTE verb with rounding and size error features
- nested statements within conditional sentences
- expansion of PERFORM verb to allow VARYING and UNTIL options.

The additional features of the Table Handling module include:

- additional options of SET verb to facilitate increasing and decreasing reference points within a table.

The Random Access module provides the following language features:

– specification of sequential or random direct access storage files
– expansion of READ and WRITE verbs to allow for INVALID KEY actions
– SEEK verb to facilitate location of direct access storage records.

The Sort module provides the following language features:

– description in data division of physical structure, identification, and record names of sort files
– the RELEASE verb for transferring records to the sort operation
– the RETURN verb for accessing sorted records from the sort operation
– the SORT verb for specification of sort parameters (ascending keys and/or descending keys and input/output procedures (either coded in COBOL by user or done by compiler furnished code).

In addition to the above, both compilers (in conjunction with the operating system) provide features such as:

– sequence check of source images
– ability to combine compilation output with output from other language processors
– compilation listings including diagnostic information
– ability to maintain and/or update a library of source programs on magnetic tape.

3.2.3. FORTRAN

FORTRAN is a problem oriented programming language designed primarily for performing the mathematical computations required for the solutions of engineering and scientific problems. FORTRAN is also useful for many nonscientific data processing applications.

All of the elementary mathematical operations and functions are available to the FORTRAN programmer. Moreover, the programmer may combine elementary sequences of operations into a more complex procedure, and give this procedure a name. At any time, the procedure may be executed merely by referencing it by name. At the time of reference, actual parameters may be specified which take the place of formal parameters used within the procedure.

FORTRAN is problem oriented in the sense that the language is designed so that the user can express his solution in a way which is natural to his problem. The user need not consider the particular characteristics of the computer on which his program is executed.

The program written in the FORTRAN language is input to another program called the compiler. This compiler translates the FORTRAN program into a code which is more nearly in a form ready to be executed by the computer. (The code must be altered by one more program, the linkage editor, before it is ready for execution.)

Procedures defined outside of the FORTRAN program, and possibly written in a language other than FORTRAN, may be referenced by name and thereby be made implicitly part of the program.

There are actually two FORTRAN compilers for use with the 9400 System. One compiler, designed for the tape oriented system, includes the language as defined by the American Standard Basic FORTRAN. It also includes the following extensions:

– the ability to include commentary as part of individual FORTRAN statements

– the option of deleting one or more statement labels from an arithmetic IF statement

– the ability to re-edit with a new FORMAT the last logical record read from any device

– the use of the T format code for pointing to a particular character position in an input or an output record

– the use of the ''quote'' form of a literal, that is, a character string in apostrophes is a literal.

The other compiler is designed for use with the direct access storage oriented configuration of the 9400 System. This compiler includes the language features as defined in the American Standard FORTRAN. In addition it includes the following extensions:

– the NAMELIST statement which allows the variables to receive data to be specified as part of the data rather than by an input list in the program

– the IMPLICIT statement which allows the type of a variable to be specified implicitly by the first character of the variable's name

– the ability to specify initial data values for variables and arrays in TYPE statements

– Seven dimensional arrays

– generalized subscript expressions which may use any of the arithmetic operators and may include function references

– implicit typing of generic function names by the type of the function argument

– multiple entries to a subroutine of function subprogram by means of the ENTRY statement

– hexadecimal constants (for use in data statements)

– the Z format code for reading and writing hexadecimal numbers

– the use of the G format code for integer, logical, and complex data

– assignment statements in which complex values may be assigned to integer and real variables, and in which real and integer values may be assigned to a complex variable

– the PAUSE ''text'' statement, where text is a message not exceeding 255 characters

– the specification, in a READ statement, of statements to receive control on encountering an error or an end-of-file condition

– ''call-by-value'' and ''call-by-name'' actual arguments in subprogram calls

– the RETURN k statement is a subroutine, which returns control to the statement number in the calling program that is the actual argument corresponding to the kth marked dummy argument in the SUBROUTINE statement.

Every statement that is acceptable to the tape oriented compiler is also acceptable to the direct access storage compiler. Moreover, the output from each version has the same form and may be input to the linkage editor.

Both compilers and their object module outputs are executable under the UNIVAC 9400 Operating System.

3.2.4. Report Program Generator (RPG)

The UNIVAC 9400 Report Program Generator provides a convenient method of obtaining reports and report programs with full utilization of the tape and disc features of the system. With upward compatibility in mind, the UNIVAC 9400 RPG has been designed to accept UNIVAC 9200 or 9300 RPG source programs for program generation and execution on the UNIVAC 9400 System.

The RPG system provides two options in producing the object program:

- The generated RPG object program may be punched on cards or recorded on disc or tape. This feature results in saving time on frequently run reports by eliminating the generation phase on subsequent report runs.

- The generated RPG object program may be executed immediately after generation without producing the object program in punched cards. This approach is desirable for single use reports.

The object output program of the RPG run is relocatable and may be linked as a subprogram to other assembled or generated programs. In addition to producing the object code on punched cards, the program may be stored in the library on disc or magnetic tape.

The requirements for specifying a particular report run are listed on five specification forms. These requirements become the input parameters from which the RPG system generates the object program.

- File Description – assigns names to files and specifies input and output devices

- Input Format Specifications – identifies input files and records, defines sequence, data, data field format, and data locations

- Calculation Specifications – provides necessary information for setting and resetting indicators, linking to other programs, table look-up, looping, arithmetic operations, and definition of result fields

- Output Format Specifications – identifies output files and records, defines which are to be produced on the printer, punch, tape or disc, defines constants and editing

- File Extension Specifications – defines tables and table files, and the format of the entries in the tables.

The generated object program from the RPG utilizes an on/off selector concept to process data from input files. If a selector is ON, a group of instructions are carried out. If the selector is OFF, the group of instructions are skipped.

3.3. SERVICE AND UTILITY ROUTINES

Service and Utility programs are provided to remove the burden from the user for accomplishing common functions in operating the UNIVAC 9400 System. Some of these functions include sorting data according to a specified order and merging of data to facilitate processing, maintaining files on magnetic tape and/or direct access storage and linking output modules of language processors into a single executable program.

3.3.1. Sort and Merge

The Sort and Merge program provides the user with a highly efficient sorting and merging capability over a wide range of data processing requirements. The Sort and Merge program sequences files containing any volume of records.

- Scheduling

  The sorting of large files accumulated over a period of time can be expedited by sorting the input records in batches as they become available. The ordered subfiles can be held until the last batch is available. The subfiles can then be merged with the last batch after it has been sorted to produce a completely ordered file.

- Facility Requirements

  The Sort program uses main storage primarily for temporary storage of data. While the Sort program functions in the minimum main storage configuration, it automatically expands the size of its working storage areas in larger configurations. This expansion of sort work areas permits significantly faster operation of the Sort program when the facilities are not shared with other programs.

  The Sort program is modular in design; when more than one system of auxiliary storage (magnetic tape or disc) is available, the Sort program can be expanded to take advantage of the entire configuration. These facilities can be employed either as exclusive alternatives or in combination.

- Input/Output

  Records may be delivered directly to the Sort program as output from a problem program without first being written on an output file. The Sort program can receive the records from any input file and in any format permitted by the operating system. The size of the records to be sorted or merged depends upon the hardware facilities available, rather than the Sort and Merge program logic. The records may be of fixed size or they may be of variable length. Records can be delivered directly to a user program without first being written on an output file. Also, the ordered records can be written to any medium and in any format permitted by the operating system.

- Keys

  The ordering of the records is based on key fields specified by the user. Any choice between ascending and descending sequence may be specified separately for each key field. Key field types include:

  - alphanumeric
  - unsigned binary
  - signed zoned decimal
  - signed packed decimal
  - signed binary

  Should be desired ordering criterion be such that it cannot be defined by the specification of key fields, the user may elect to supply his own code to determine which of any given pair of records is to appear first in the final ordered sequence.

- Execution of Own Code

  The Sort and Merge program can transfer control to user supplied code when equality of specified key fields is detected. Thus, the user may consolidate the contents of like records to reduce the overall quantity of data to be processed.

- Parameter Specification

  The Sort and Merge is a fixed program which reacts to parameters at run time. However, the parameter elements, may be generated by the Assembler and filed in the library for repeated use.

- Restart

  A convenient restart procedure is available to reproduce, if necessary, any ordered subfile. Merging may be scheduled separately from the initial stages of sorting.

3.3.2. Library Services

The libraries are files usually described to the system at systems generation according to individual installation parameters. Any number of libraries may exist as long as the established formats are observed. A system library is required and must be defined as such to the system.

Input to the language processors, such as the Assembler, FORTRAN, COBOL, etc., may exist as part of a library. The processors, in turn, output information in prescribed format to be entered into a library. Specification of which library is to be used is accomplished in the job control stream.

UNIVAC 9400 System software includes the service routines necessary to establish, compress, display, and alter the contents of the libraries. The routines necessary for interface between the processors and the library is also provided.

Facilities in response to control cards are provided also for the user to dump selected elements of the specified library onto tape or cards and to re-enter the dumped elements later directly into the file.

3.3.3. Linkage Editor

The primary function of the Linkage Editor is to link object output modules (that is, the output of a complete processor run in a particular language) into one load module that is suitable for loading and execution by the Supervisor control program. In addition, the Linkage Editor also performs the following functions:

- searches the appropriate library and incorporates modules other than those in its primary input, either upon request or automatically
- performs program modification by deleting, rearranging, and replacing control sections as directed by the Linkage Editor control statements
- produces an overlay scheme for loading by the Supervisor control program
- reserves storage automatically for the common control sections generated by the assembler
- logs diagnostic messages.

3.3.4. Utility Programs and Program Testing Aids

The utility programs transfer data form one peripheral unit to another, or, from one area of a peripheral unit to another, and, provides for manipulating (creating, changing, deleting) files to best suit the users requirements. The control information required by some of these programs are furnished as parameters by means of the job control stream, and, in some cases, are supplied by the problem program.

The UNIVAC 9400 System software includes two program testing aids: a dynamic (snapshot) dump, and a terminal (postmortem) dump. The snapshot dump is used primarily as a program "debugging" aid. It provides (by user specification) a listing of register contents, control information, or the contents of any range of addresses in the users program. It permits the program to be continued after the specified information is listed. The postmortem dump is used when a program is terminated through an error or abort condition by the Supervisor. This dump provides complete listings of all aspects of the user's program. In this case the program must be reloaded before attempting to restart.

3.4. SOFTWARE CONFIGURATOR

OPERATING SYSTEM	BASIC TAPE OPERATING SYSTEM	TAPE OPERATING SYSTEM	BASIC DISC OPERATING SYSTEM	DISC OPERATING SYSTEM
Minimum Hardware Requirements	24K Bytes Card Reader① Printer② 4 Tape Units	32K Bytes Card Reader① Printer② 4 Tape Units	24K Bytes Card Reader① Printer② 2 Discs (8411)	32K Bytes Card Reader① Printer② 2 Discs (8411)
Supervisor				
Multiprogramming	No	No③	Yes④	Yes④
Time Allocation	No	Yes	Yes	Yes
Simulated Day Clock	No	Yes	No	Yes
Interval Timer Services	Limited services available	Yes	Limited services available for one program	Full capability for all programs
Storage Protection⑤	No	Yes	No	Yes
Transient Area Management	Limited services available	Same	Yes	Yes
Automatic Buffering	No	No	No	Yes
Program Check Point	Yes	Yes	Yes	Yes
Data Management				
Sequential Access Method	Yes	Yes	Yes	Yes
Direct Access Method	Yes ⑥	Yes⑥	Yes	Yes
Index Sequential Access Method	Yes ⑥	Yes⑥	Yes	Yes
Job Control				
Control Stream Buffering	Yes – tape only	Yes	Yes – Disc only	Yes – Disc only
Program Restart	Yes	Yes	Yes	Yes
Message Control Program⑦	Limited services available	Yes	Limited services available	Queued communications control
Assembler	Yes	Yes	Yes	Yes
COBOL	No	Yes⑧	No	Yes⑧
FORTRAN	No	Yes	No	Yes
Report Program Generator	Yes	Yes	Yes	Yes
Sort/Merge	Yes	Yes	Yes	Yes
Library Services	Yes	Yes	Yes	Yes
Linkage Editor	Yes	Yes	Yes	Yes
Utility Programs	Yes	Yes	Yes	Yes
Program Testing Aids	Yes	Yes	Yes	Yes

① Card Reader may be standard online card reader or card reader on 9200/9300 Subsystem or 1004/1005 Subsystem.

② Printer may be standard online printer or printer on 9200/9300 Subsystem or 1004/1005 Subsystem.

③ Tape Operating System is capable of running one main program with auxiliary data conversion routines.

④ From 1 to 5 user programs depending on user selection at Systems Generation Time – note that with basic systems, running more than one program at a time may not be possible due to limited main storage.

⑤ Memory protect hardware feature required.

⑥ UNIVAC 8411 Direct Access Storage Subsystem required.

⑦ Data Communications Subsystem hardware required.

⑧ 32K bytes required for basic COBOL – 65K bytes and disc files required for extended language features.

APPENDIX A. UNIVAC 9400 SYSTEM INSTRUCTIONS

The following is a listing of all the UNIVAC 9400 Instructions by type, showing the mnemonic source code, the hexadecimal operation code and the timing in microseconds.

RR TYPE INSTRUCTIONS

DESCRIPTION	MNEMONIC CODE	OPERATION CODE	TIMING IN MICROSECONDS
Add	AR	1A	6.0
AND	NR	14	6.0
Branch and Link	BALR	05	6.0
Branch on Condition	BCR	07	4.2
Branch on Count	BCTR	06	7.2
Compare	CR	19	6.0
Compare Logical	CLR	15	6.0
Exclusive OR	XR	17	6.0
Load	LR	18	4.8
Load and Test	LTR	12	4.8
OR	OR	16	6.0
Set Program Mask	SPM	04	6.0
Subtract	SR	1B	6.0
Supervisor Call	SVC	0A	6.0

RX TYPE INSTRUCTIONS

Add	A	5A	6.0
Add Halfword	AH	4A	6.0
AND	N	54	6.0
Branch and Link	BAL	45	4.8
Branch on Condition	BC	47	3.0
Branch on Count	BCT	46	6.0
Compare	C	59	6.0
Compare Halfword	CH	49	6.0
Compare Logical	CL	55	6.0
Exclusive OR	X	57	6.0
Insert Character	IC	43	4.2
Load	L	58	4.8
Load Address	LA	41	4.8
Load Halfword	LH	48	6.0
OR	O	56	6.0
Store	ST	50	6.0
Store Character	STC	42	4.2
Store Halfword	STH	40	4.2
Subtract	S	5B	6.0
Subtract Halfword	SH	4B	6.0

NOTE: In the RX instructions the timing is increased by 1.2 μs when the index (X) field is not equal to zero.

RS TYPE INSTRUCTIONS

DESCRIPTION	MNEMONIC CODE	OPERATION CODE	TIMING IN MICROSECONDS
Load Limits Register	LLR	81	6.0
Load Multiple	LM	98	2.4 + (2.4 × f)
Shift Left Single Logical	SLL	89	10.8 + (1.2 × c)
Shift Right Single Logical	SRL	88	13.8 + (2.4 × c)
Store Multiple	STM	90	2.4 + (2.4 × f)

NOTE: c equals the contents of the base register plus displacement.
f is the number of full words.

SI TYPE INSTRUCTIONS

Add Immediate	AI	93	4.2
AND	NI	94	4.2
Compare Logical	CLI	95	4.2
Exclusive OR	XI	97	4.2
Halt and Proceed	HPR	99	6.0
Load Program Status Word	LPSW	82	7.2
Move	MVI	92	4.2
OR	OI	96	4.2
Set System Mask	SSM	80	6.0
Start I/O	SIO	9C	6.0 + CU Time †
Test under Mask	TM	91	4.2

NOTE: † yields an approximate time factor

SS1 TYPE INSTRUCTIONS

AND	NC	D4	16.2 + (2.4 × n)
Compare Logical Character	CLC	D5	16.2 + (2.4 × n)
Edit	ED	DE	15.0 + (3.6 × n) + (1.2 × ns)
Exclusive OR	XC	D7	16.2 + (2.4 × n)
Move Characters	MVC	D2	16.2 + (2.4 × n)
Move Numeric	MVN	D1	16.2 + (2.4 × n)
Move Zones	MVZ	D3	16.2 + (2.4 × n)
OR	OC	D6	16.2 + (2.4 × n)
Translate	TR	DC	15.0 + (4.8 × n1)

NOTE: n is the number of result bytes
n1 is the number of bytes in operand one
ns is the number of signs in operand two

SS2 TYPE INSTRUCTIONS

DESCRIPTION	MNEMONIC CODE	OPERATION CODE	TIMING IN MICROSECONDS
Add Packed Decimal	AP	FA	16.2 + (2.4 × n1)*
Compare Packed Decimal	CP	F9	16.2 + (2.4 × n1)*
Divide Packed Decimal	DP	FD	21.6 × (n1 − n2) × (3.5 + n2)†
Move with Offset	MVO	F1	16.2 + (2.4 × n1)
Multiply Packed Decimal	MP	FC	21.6 × (n1 − n2) × (2.8 + n2)†
Pack	PACK	F2	13.8 + (4.8 × n1)
Subtract Packed Decimal	SP	FB	16.2 + (2.4 × n1)*
Unpack	UNPK	F3	16.2 + (2.4 × n1)
Zero and Add	ZAP	F8	16.2 + (2.4 × n1)*

NOTE: n1 is the number of bytes in operand one
n2 is the number of bytes in operand two
** plus 6.0 + (2.4 × n1) when the result must be recomplemented*
† yields an approximate time factor

APPENDIX B. ASCII-8 AND EBCDIC CODE CHARTS

The convention used for numbering the bit positions within a character differ for each of the codes. The conventions are:

	BIT POSITIONS							
EBCDIC	0	1	2	3	4	5	6	7
ASCII	7	6	X	5	4	3	2	1

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII) EXTENDED TO EIGHT BITS

		BIT POSITIONS 7, 6, X, 5															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
BIT POSITIONS 4, 3, 2, 1	0000	NUL	DLE			b̸ blank	0					@	P			'	p
	0001	SOH	DC1			!	1					A	Q			a	q
	0010	STX	DC2			"	2					B	R			b	r
	0011	ETX	DC3			#	3					C	S			c	s
	0100	EOT	DC4 STOP			$	4					D	T			d	t
	0101	ENQ	NAK			%	5					E	U			e	u
	0110	ACK	SYN			&	6					F	V			f	v
	0111	BEL	ETB			'	7					G	W			g	w
	1000	BS	CAN			(	8					H	X			h	x
	1001	HT	EM			)	9					I	Y			i	y
	1010	LF	SUB			*	:					J	Z			j	z
	1011	VT	ESC			+	;					K	[			k	{
	1100	FF	FS			,	<					L	\			l	|
	1101	CR	GS			–	=					M	]			m	}
	1110	SO	RS			.	>					N	^			n	~
	1111	SI	US			/	?					O	—			o	DEL

EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE (EBCDIC)

4 5 6 7	0 1 2 3		0 1 2 3		0 1 2 3		0 1 2 3		4 5 6 7
	0 0 0 0		0 0 1 0		0 1 0 0		0 1 1 0		
	SYMBOL	80. COL. CD. CODE	SYMBOL	80 COL. CD. CODE	SYMBOL	80 COL. CD. CODE	SYMBOL	80 COL. CD. CODE	
0000	NULL	12, 0, 9, 8, 1	DS	11, 0, 9, 8, 1	SP	No Punches	– (Minus)	11	0000
0001		12, 9, 1	SOS	0, 9, 1		12, 0, 9, 1	/ (Slash)	0, 1	0001
0010		12, 9, 2	FS	0, 9, 2		12, 0, 9, 2		11, 0, 9, 2	0010
0011		12, 9, 3		0, 9, 3		12, 0, 9, 3		11, 0, 9, 3	0011
0100	PF	12, 9, 4	BYP	0, 9, 4		12, 0, 9, 4		11, 0, 9, 4	0100
0101	HT	12, 9, 5	LF	0, 9, 5		12, 0, 9, 5		11, 0, 9, 5	0101
0110	LC	12, 9, 6	EOB	0, 9, 6		12, 0, 9, 6		11, 0, 9, 6	0110
0111	DEL	12, 9, 7	PRE	0, 9, 7		12, 0, 9, 7		11, 0, 9, 7	0111
1000		12, 9, 8		0, 9, 8		12, 0, 9, 8		11, 0, 9, 8	1000
1001		12, 9, 8, 1		0, 9, 8, 1		12, 8, 1		0, 8, 1	1001
1010		12, 9, 8, 2	SM	0, 9, 8, 2	¢	12, 8, 2	\	12, 11	1010
1011		12, 9, 8, 3		0, 9, 8, 3	. (Period)	12, 8, 3	, (Comma)	0, 8, 3	1011
1100		12, 9, 8, 4		0, 9, 8, 4	<	12, 8, 4	%	0, 8, 4	1100
1101		12, 9, 8, 5		0, 9, 8, 5	(	12, 8, 5	_(Undersc.)	0, 8, 5	1101
1110		12, 9, 8, 6		0, 9, 8, 6	+	12, 8, 6	[illegible]	0, 8, 6	1110
1111		12, 9, 8, 7		0, 9, 8, 7	¦ (Vert. Bar)	12, 8, 7	?	0, 8, 7	1111
	0 0 0 1		0 0 1 1		0 1 0 1		0 1 1 1		
0000		12, 11, 9, 8, 1		12, 11, 0, 9, 8, 1	&	12		12, 11, 0	0000
0001		11, 9, 1		9, 1		12, 11, 9, 1		12, 11, 0, 9, 1	0001
0010		11, 9, 2		9, 2		12, 11, 9, 2		12, 11, 0, 9, 2	0010
0011		11, 9, 3		9, 3		12, 11, 9, 3		12, 11, 0, 9, 3	0011
0100	RES	11, 9, 4	PN	9, 4		12, 11, 9, 4		12, 11, 0, 9, 4	0100
0101	NL	11, 9, 5	RS	9, 5		12, 11, 9, 5		12, 11, 0, 9, 5	0101
0110	BS	11, 9, 6	UC	9, 6		12, 11, 9, 6		12, 11, 0, 9, 6	0110
0111	IL	11, 9, 7	EOT	9, 7		12, 11, 9, 7		12, 11, 0, 9, 7	0111
1000		11, 9, 8		9, 8		12, 11, 9, 8		12, 11, 0, 9, 8	1000
1001		11, 9, 8, 1		9, 8, 1		11, 8, 1		8, 1	1001
1010		11, 9, 8, 2		9, 8, 2	! (Exclam.Pt.)	11, 8, 2	: (Colon)	8, 2	1010
1011		11, 9, 8, 3		9, 8, 3	$	11, 8, 3	#	8, 3	1011
1100		11, 9, 8, 4		9, 8, 4	*	11, 8, 4	@	8, 4	1100
1101		11, 9, 8, 5		9, 8, 5	)	11, 8, 5	' (Prime)	8, 5	1101
1110		11, 9, 8, 6		9, 8, 6	; (Semicolon)	11, 8, 6	= (Equals)	8, 6	1110
1111		11, 9, 8, 7		9, 8, 7	¬ (not)	11, 8, 7	'' (Quote)	8, 7	1111

NOTE: DOUBLE AND TRIPLE LETTER SYMBOLIC CODES ARE FOR USE WITH COMMUNICATIONS DEVICES.

EXTENDED BINARY-CODED DECIMAL INTERCHANGE CODE (EBCDIC)

4 5 6 7	0 1 2 3 / 1 0 0 0 SYMBOL	80 COL. CD. CODE	0 1 2 3 / 1 0 1 0 SYMBOL	80 COL. CD. CODE	0 1 2 3 / 1 1 0 0 SYMBOL	80 COL. CD. CODE	0 1 2 3 / 1 1 1 0 SYMBOL	80 COL. CD. CODE	4 5 6 7
0000		12, 0, 8, 1		11, 0, 8, 1	PZ	12, 0		0, 8, 2	0000
0001	a	12, 0, 1		11, 0, 1	A	12, 1		11, 0, 9, 1	0001
0010	b	12, 0, 2	s	11, 0, 2	B	12, 2	S	0, 2	0010
0011	c	12, 0, 3	t	11, 0, 3	C	12, 3	T	0, 3	0011
0100	d	12, 0, 4	u	11, 0, 4	D	12, 4	U	0, 4	0100
0101	e	12, 0, 5	v	11, 0, 5	E	12, 5	V	0, 5	0101
0110	f	12, 0, 6	w	11, 0, 6	F	12, 6	W	0, 6	0110
0111	g	12, 0, 7	x	11, 0, 7	G	12, 7	X	0, 7	0111
1000	h	12, 0, 8	y	11, 0, 8	H	12, 8	Y	0, 8	1000
1001	i	12, 0, 9	z	11, 0, 9	I	12, 9	Z	0, 9	1001
1010		12, 0, 8, 2		11, 0, 8, 2		12, 0, 9, 8, 2		11, 0, 9, 8, 2	1010
1011		12, 0, 8, 3		11, 0, 8, 3		12, 0, 9, 8, 3		11, 0, 9, 8, 3	1011
1100		12, 0, 8, 4		11, 0, 8, 4		12, 0, 9, 8, 4		11, 0, 9, 8, 4	1100
1101		12, 0, 8, 5		11, 0, 8, 5		12, 0, 9, 8, 5		11, 0, 9, 8, 5	1101
1110		12, 0, 8, 6		11, 0, 8, 6		12, 0, 9, 8, 6		11, 0, 9, 8, 6	1110
1111		12, 0, 8, 7		11, 0, 8, 7		12, 0, 9, 8, 7		11, 0, 9, 8, 7	1111
	1 0 0 1		1 0 1 1		1 1 0 1		1 1 1 1		
0000		12, 11, 8, 1		12, 11, 0, 8, 1	MZ	11, 0	0	0	0000
0001	j	12, 11, 1		12, 11, 0, 1	J	11, 1	1	1	0001
0010	k	12, 11, 2		12, 11, 0, 2	K	11, 2	2	2	0010
0011	l	12, 11, 3		12, 11, 0, 3	L	11, 3	3	3	0011
0100	m	12, 11, 4		12, 11, 0, 4	M	11, 4	4	4	0100
0101	n	12, 11, 5		12, 11, 0, 5	N	11, 5	5	5	0101
0110	o	12, 11, 6		12, 11, 0, 6	O	11, 6	6	6	0110
0111	p	12, 11, 7		12, 11, 0, 7	P	11, 7	7	7	0111
1000	q	12, 11, 8		12, 11, 0, 8	Q	11, 8	8	8	1000
1001	r	12, 11, 9		12, 11, 0, 9	R	11, 9	9	9	1001
1010		12, 11, 8, 2		12, 11, 0, 8, 2		12, 11, 9, 8, 2		12, 11, 0, 9, 8, 2	1010
1011		12, 11, 8, 3		12, 11, 0, 8, 3		12, 11, 9, 8, 3		12, 11, 0, 9, 8, 3	1011
1100		12, 11, 8, 4		12, 11, 0, 8, 4		12, 11, 9, 8, 4		12, 11, 0, 9, 8, 4	1100
1101		12, 11, 8, 5		12, 11, 0, 8, 5		12, 11, 9, 8, 5		12, 11, 0, 9, 8, 5	1101
1110		12, 11, 8, 6		12, 11, 0, 8, 6		12, 11, 9, 8, 6		12, 11, 0, 9, 8, 6	1110
1111		12, 11, 8, 7		12, 11, 0, 8, 7		12, 11, 9, 8, 7		12, 11, 0, 9, 8, 7	1111

NOTE: Lower case letters are industry standard — not printable on the UNIVAC 9400 Printer.

APPENDIX C. PRINTER CHARACTER SET

LEAST SIGNIFICANT BITS	MOST SIGNIFICANT BITS 00	01	10	11
0000	Blank	&	–	0
0001	A	J	/	1
0010	B	K	S	2
0011	C	L	T	3
0100	D	M	U	4
0101	E	N	V	5
0110	F	O	W	6
0111	G	P	X	7
1000	H	Q	Y	8
1001	I	R	Z	9
1010	¢	!	\	:
1011	.	$	(COMMA) ,	#
1100	<	*	%	@
1101	(	)	_	' (PRIME)
1110	+	;	>	=
1111	$\vert$	¬	?	"

One character is printed per byte, the two most significants bits of each byte are ignored

UNIVAC

SPERRY RAND

UP-7566