UNIVAC 90/70 System

System Description

SPERRY UNIVAC COMPUTER SYSTEMS

UP–8050

CONTENTS

APPENDIXES

FIGURES

TABLES

I. INTRODUCTION

The UNIVAC 90/70 System, shown in Figure 1–1, is the latest and most powerful member of the UNIVAC Series 90 computer family. It is a general purpose, disc-oriented computer designed to function in many different data processing environments with equal operating efficiency.

The UNIVAC 90/70 System includes dynamic program relocatability, the Operating System Storage Facility (OSSF), and the Communications Intelligence Channel (CIC). Two operating systems are available, OS/4, an enhanced version of the UNIVAC 9400 DOS, and OS/7, specifically designed to fully utilize the modern hardware of the UNIVAC 90/70 System. Up to 14 user programs may be concurrently executed while each is guaranteed its integrity by hardware storage protection. Another significant feature is the ease of transition from an IBM* 360/DOS to a UNIVAC 90/70 System.

The modular design of the UNIVAC 90/70 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 terminal devices can be readily incorporated into existing UNIVAC 90/60 Systems. This architecture extends the usefulness of the initial planning, programming, and operational procedures of a system.

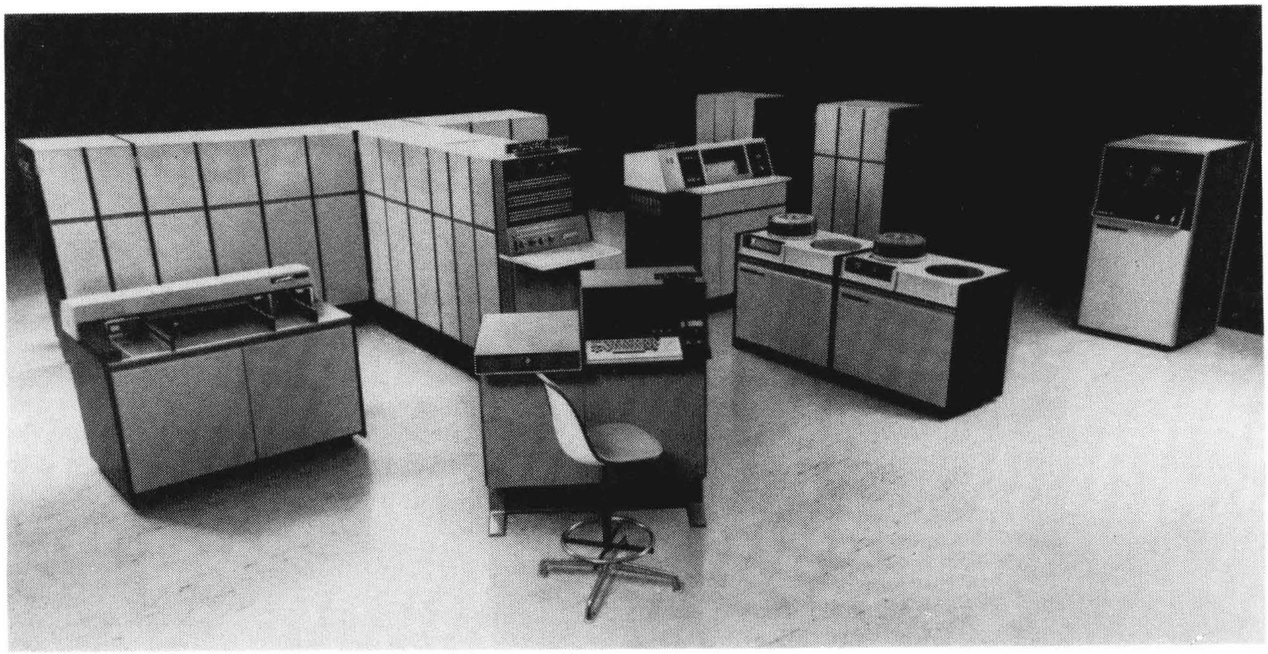


Figure 1–1. UNIVAC 90/70 System

**Trademark of International Business Machine Co.*

Flexible software generation procedures allow the user to construct an operating system tailored to his particular needs. System performance is dependent upon the available hardware and the operating system options selected.

The UNIVAC 90/70 System offers a choice of programming languages, many of which are compatible with IBM System/360 DOS. The job control language is extended over those of its competitors. Variable multitasking within a job allows more than one task to compete for processor time. The new Operating System Storage Facility offers a dedicated device and channel to further increase system performance.

2. SYSTEM DESCRIPTION

2.1. HIGHLIGHTS

The UNIVAC 90/70 System consists fo central hardware components and a wide range of peripheral subsystems; including extensive data communications capabilities. Also, software support is provided by two operating systems:

- OS/4

  A versatile multi-jobbing operating system patterned after the UNIVAC 9400 disc operating system. This system is designed for smaller configurations and for those users not requiring the extended capabilities provided by OS/7.

- OS/7

  A more sophisticated multi-tasking system with extended capabilities to meet the needs of larger configurations and more complex processing environments.

The highlights of the UNIVAC 90/70 System are shown in Figure 2–1. A detailed description of the hardware and software components in the UNIVAC 90/70 System is given in the following sections of this document.

UNIVAC 90/70 SYSTEM

HARDWARE

CENTRAL COMPONENTS

- PROCESSOR

  MICROPROGRAM CONTROL
  UNIVERSAL INSTRUCTION SET
  INTERVAL TIMER
  HARDWARE RELOCATION
  INDIRECT ADDRESSING

- MAIN STORAGE

  131 TO 1048K BYTES
  600 NANOSECOND CYCLE TIME

- OPERATING SYSTEM STORAGE FACILITY

  8.34 MILLISECOND AVERAGE ACCESS TIME
  3 MILLION BYTES
  DEDICATED CONTROL CHANNEL

- SELECTOR CHANNELS

  1 STANDARD; 3 OPTIONAL
  1.1 MEGABYTE TRANSFER RATE

- MULTIPLEXER CHANNEL

  175 KILOBYTE TRANSFER RATE

- COMMUNICATION INTELLIGENCE CHANNEL
- SYSTEM CONSOLE WITH VISUAL DISPLAY

PERIPHERALS

- DISC SUBSYSTEMS

  UP TO 954 MILLION BYTES/ SUBSYSTEM
  55 TO 30 MILLISECOND ACCESS TIME

- MAGNETIC TAPE SUBSYSTEMS

  UP TO 320 KILOBYTE TRANSFER RATES
  7 AND 9 TRACK OPTIONS
  NRZI AND PHASE ENCODING

- HIGH SPEED PRINTERS

  UP TO 3000 LINES/ MINUTE

- CARD SUBSYSTEMS
- DATA COMMUNICATIONS SUBSYSTEMS
- UNISCOPE DISPLAY TERMINALS
- OPTICAL DOCUMENT READER
- PAPER TAPE SUBSYSTEM
- OTHER 9000 SERIES SYSTEMS

SOFTWARE

OS/4

- MULTIJOBBING SUPERVISOR

  UP TO 5 JOBS CONCURRENTLY
  PRIORITY DISPATCHING
  TIME SLICING

- LANGUAGE PROCESSORS

  COBOL
  FORTRAN
  RPG
  ASSEMBLER

- DATA MANAGEMENT SYSTEM

  SAM
  DAM
  ISAM

- JOB CONTROL LANGUAGE
- STORED JOB STREAMS
- COMMUNICATIONS
- UTILITY PROGRAMS
- INPUT/OUTPUT SPOOLING

OS/7

- ALL THE FEATURES OF OS/4 PLUS:
- VARIABLE MULTITASKING

  UP TO 14 JOBS CONCURRENTLY
  NO FIXED LIMITS FOR TASKS WITHIN JOBS

- SYSTEM FILE CATALOG

  VOLUME LOCATION AND ACCESS CONTROL
  SIMPLIFIED SYSTEM/USER INTERFACE
  CATALOGED JOB STREAM PROCEDURES

- AUTOMATIC JOB SCHEDULING

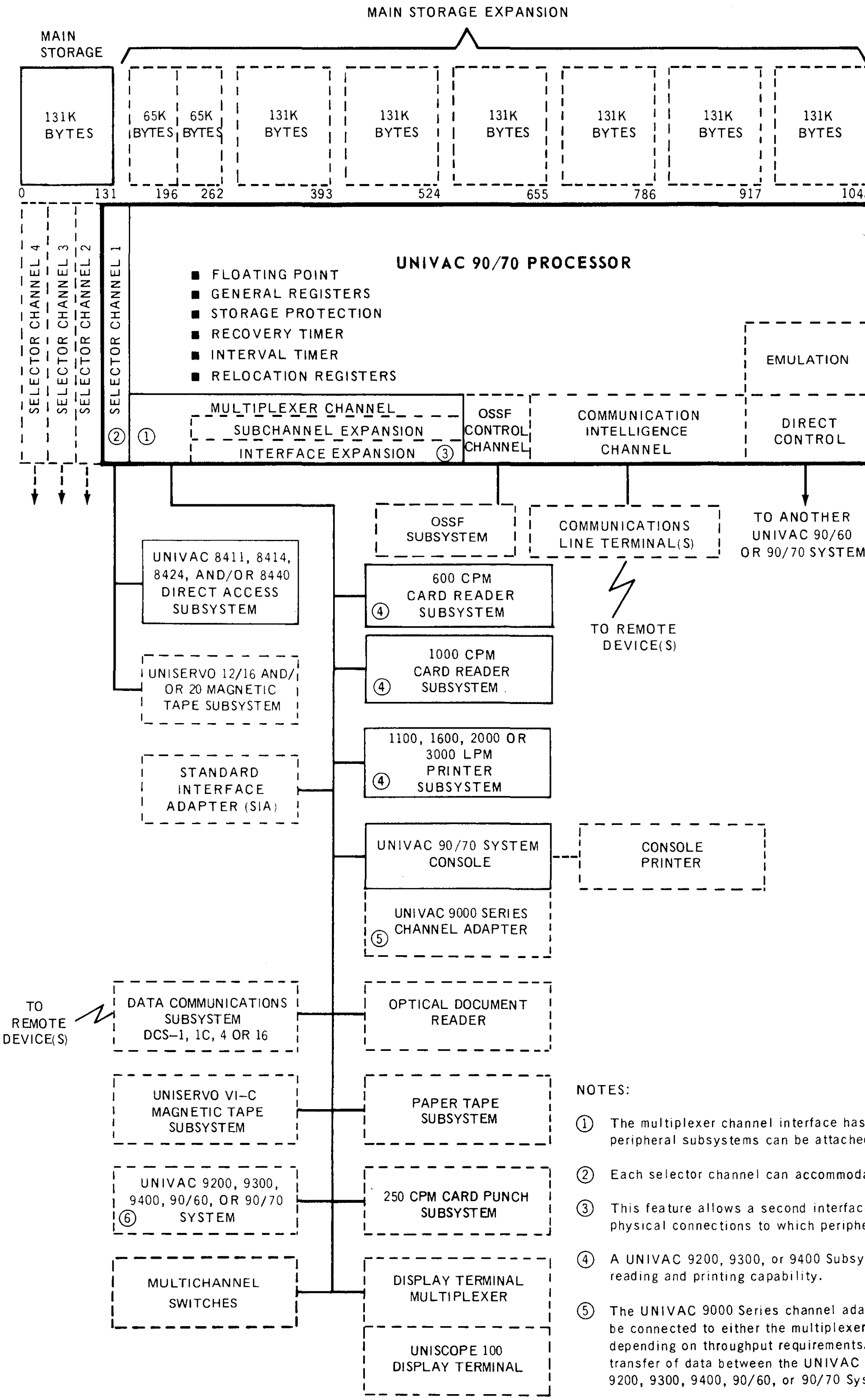
  INCREASED THROUGHPUT
  DECREASED OPERATOR INTERVENTION

- REMOTE JOB ENTRY
- ROLLOUT/ROLLIN WITH HARDWARE RELOCATION
- INFORMATION MANAGEMENT SYSTEM
- JOB ACCOUNTING
- DEVICE INDEPENDENT DATA MANAGEMENT SYSTEM
- ONLINE DIAGNOSTICS
- IBM 360 EMULATION
- IBM 1400 EMULATION
- SERIES 70 EMULATION

Figure 2–1. Highlights of UNIVAC 90/70 System Hardware and Software

2.2. SYSTEM HARDWARE COMPONENTS

The UNIVAC 90/70 System hardware consists of a processor, main storage, Operating System Storage Facility (OSSF), system console, and standard UNIVAC Series 90 I/O equipment. The processor is composed of the instruction processor and I/O channels. Main storage is byte-addressable and is expandable from 131K to 1048K bytes. The OSSF consists of a fixed head disc subsystem connected to a control channel. A full line of I/O subsystems is available for use with the UNIVAC 90/70 System onsite or at remote sites. An overall block diagram of the UNIVAC 90/70 System hardware and its range of options is given in Figure 2–2. This figure is followed by a summary of system characteristics.



NOTES:

① The multiplexer channel interface has eight physical connections to which peripheral subsystems can be attached.

② Each selector channel can accommodate eight high-speed subsystems.

③ This feature allows a second interface which has eight additional physical connections to which peripheral subsystems can be attached.

④ A UNIVAC 9200, 9300, or 9400 Subsystem may provide the necessary reading and printing capability.

⑤ The UNIVAC 9000 Series channel adapter mounted in the console may be connected to either the multiplexer channel or selector channel depending on throughput requirements. It provides for the control and transfer of data between the UNIVAC 90/70 System and a UNIVAC 9200, 9300, 9400, 90/60, or 90/70 System.

⑥ Any of these systems (if they have a 9000 Series or Series 90 channel adapter) may be connected to the multiplexer or selector channel of a UNIVAC 90/70 System.

LEGEND:
BASIC EQUIPMENT ————
OPTIONAL EXPANSION — — — — — —

Figure 2–2. UNIVAC 90/70 System Hardware

UNIVAC 90/70 SYSTEM CHARACTERISTICS	
SYSTEM ORIENTATION	Disc
PROCESSOR	Microprogram control Versatile instruction set Hardware relocation Indirect addressing
DATA ORGANIZATION	Eight-bit bytes; four bytes per main storage access
MAIN STORAGE CAPACITY	131K, 196K, 262K, 393K, 524K, 655K, 786K, 917K, or 1048K bytes
MAIN STORAGE PERFORMANCE	A cycle time of 600 nanoseconds for a 4-byte access
FLOATING-POINT ARITHMETIC	Standard
DIRECT CONTROL FEATURE	Specialized interprocessor access feature for multiple UNIVAC 90/60 or 90/70 systems
MULTIPLEXER CHANNEL	Up to 8 subsystems – standard Up to 8 additional subsystems – optional feature Up to 15 subchannel addresses – standard Up to 16 additional subchannel address – optional feature 175K bytes/second throughput rate
SELECTOR CHANNEL	1 – standard 3 additional – optional feature Up to 8 subsystems per channel 1.1M bytes/second throughput rate per channel
REGISTERS	16 for user program functions 16 for supervisor functions 8 for system working 1 for relocation 4 for floating-point arithmetic
DISC STORAGE: UNIVAC 8440 DISC SUBSYSTEM	119.28 million bytes/pack, maximum 954.24 million bytes/subsystem, maximum 624K bytes/second transfer rate 30 milliseconds average access time
UNIVAC 8424 DISC SUBSYSTEM	58.352 million bytes/pack, maximum 466.816 million bytes/subsystem, maximum 312 bytes/second transfer rate 30 milliseconds average access time
UNIVAC 8414 DISC SUBSYSTEM	29.17 million bytes/pack, maximum 233.36 million bytes/subsystem, maximum 312K bytes/second transfer rate 60 milliseconds average access time
UNIVAC 8411 DISC SUBSYSTEM	7.25 million bytes/pack, maximum 58 million bytes/subsystem, maximum 156K bytes/second transfer rate 75 milliseconds average access time
OPERATING SYSTEM STORAGE FACILITY	3.0 million bytes capacity 312K bytes/second transfer rate 8.34 milliseconds average access time Uses OSSF control channel
MAGNETIC TAPE:	7- and 9-track, phase and NRZI
UNISERVO 20 SUBSYSTEM UNISERVO 16 SUBSYSTEM UNISERVO 12 SUBSYSTEM UNISERVO VI-C SUBSYSTEM	320K bytes/second transfer rate 192K bytes/second transfer rate 68K bytes/second transfer rate 34K bytes/second transfer rate
CARD READER	615 or 1000 cards/minute
PRINTER	426 to 3000 lines/minutes; 96 to 24 characters
CARD PUNCH	250 cards/minute
PAPER TAPE	300 characters/second – read 110 characters/second – punch
OPTICAL DOCUMENT READER	300 or 600 documents/minute
COMMUNICATIONS	Up to 30 lines using a DCS-16 on the multiplexer channel or up to 128 half-duplex or 64 duplex lines on the communication intelligence channel.

2.2.1. Processor

The UNIVAC 90/70 Processor performs control, arithmentic, and input/output operations. The control operations include determination of the sequence in which instructions are executed and interpretation and control of the execution of each instruction. Arithmetic hardware performs all data manipulation including logical operations, aritmmetic operations, data comparisons, and shifting. The input/output hardware initiates, directs, and monitors the transfer of data between main storage and the I/O subsystems.

The processor operates under microprogram control and contains an interval timer, recovery timer, storage protection, register stack, floating-point arithmetic and I/O channels. It also has provisions for optional features such as: direct control for processor-to-processor communication, CIC for communications oriented processing, and OSSF dedicated to the operating system.

The processor register stack contains 16 registers for supervisor functions, 16 registers for user program functions, and 4 registers for floating-point arithmetic operations.

The UNIVAC 90/70 Processor uses industry-standard instructions. The instruction repertoire is an expansion of the repertoire of the UNIVAC 9400 System. The UNIVAC 90/70 System instruction repertoire also includes all of the nonprivileged instructions of the IBM System/360 Model 50.

2.2.2 I/O Channels

The UNIVAC 90/70 System is built with powerful high-speed I/O capabilities. From one to four selector channels are used for high-speed subsystems such as tape and disc. The multiplexer channel can service up to 16 low-speed subsystems such as printers and card readers. The CIC has programmable message control logic to minimize software overhead for communications and to provide the specialized control necessary for a variety of message disciplines.

2.2.2.1. Standard Interface Adapter (SIA)

The Standard Interface Adapter allows the connection of a Series 70 Communications Controller Multichannel (CCM) to a UNIVAC 90/70 Multiplexor Channel. Features are available that allow up to six CCMs to be attached to an SIA.

2.2.3. Communications

The UNIVAC 90/70 System offers added flexibility in communications-oriented applications. A wide variety of remote devices is available for use with this system. Connection to the processor is accomplished through the CIC or the multiplexer channel.

The UNIVAC 90/70 System may be used with DATA-PHONE* service, TWX networks, TELEX**, and Wideband or any combination of these services up to the maximum throughput of the system.

Communications capability is provided by means of the CIC or by a Data Communications Subsystem (DCS) connected to the multiplexer channel. A description of each follows:

2.2.3.1. Communication Intelligence Channel (CIC)

The CIC provides a flexible, communications-oriented I/O channel that handles a wide variety of message disciplines.

* *Trademark and service mark of AT&T Co.*
** *Trademark of Western Union Telegraph Co.*

2.2.3.2. Data Communications Subsystem (DCS)

Smaller communications requirements are satisfied by the DCS. Each DCS uses one of the physical connections of the multiplexer channel. A DCS-1 or DCS-1C provides capability for communicating with one duplex line; a DCS-4 or DCS-16 provides capability for communicating with up to 4, or 14 duplex lines respectively.

2.3. SOFTWARE FEATURES

The operating system for the UNIVAC 90/70 System consists of a comprehensive set of control programs, utility service, and programming aids. (See Figure 2–3.) It is modular in design to fulfill a wide range of data processing requirements. The user can select the precise level of software capability to match his configuration. Control programs provide for communications processing and random and sequential batch processing. Data to be processed can be introduced to the system from either central or remote locations.

2.3.1. Supervisor

The supervisor is the part of the operating system that operates with user programs to provide the control necessary for optimum utilization of the UNIVAC 90/70 System hardware and software.

The supervisor consists of those routines which control physical input/output operations, system resource allocation on a dynamic basis, task switching to achieve multijobbing, hardware interrupt servicing, communications with the systems operator, and in general, the interface of user programs with the hardware.

2.3.1.1. Multijobbing/Multitasking

The UNIVAC 90/70 System can concurrently process from 1 to 14 jobs. Each job consists of one or more job steps (programs) which are executed serially. Additionally, a job step may have one or more tasks which are executed concurrently. This structure allows the user complete flexibility in determining the use of the system and in scheduling the tasks to be done.

The allocation of processor time is based on a system switch list which contains information about switching priorities, time-slice values, and processor utilization.

The number of priorities and the initial time-slice values for each priority level are provided by the user at system generation. Priority for a given job may be changed by the system or the operator within predetermined limits.

2.3.1.2. Rollout/Rollin

All main storage addresses generated by problem programs are hardware restricted to being job-region-relative addresses. Hardware relocation registers which are managed by the operating system are used to locate job regions in main storage. This facility provides for effective rollout/rollin of jobs without regard to prior addressing constraints. For example, a high priority job may request additional main storage space to satisfy a critical dynamic requirement. In order to provide contiguous storage, it may be necessary to roll out a lower priority job to mass storage. This reduces main storage fragmentation, thus increasing main storage utilization. The lower priority job, when returned to main storage, can be loaded into a different absolute area of main storage.

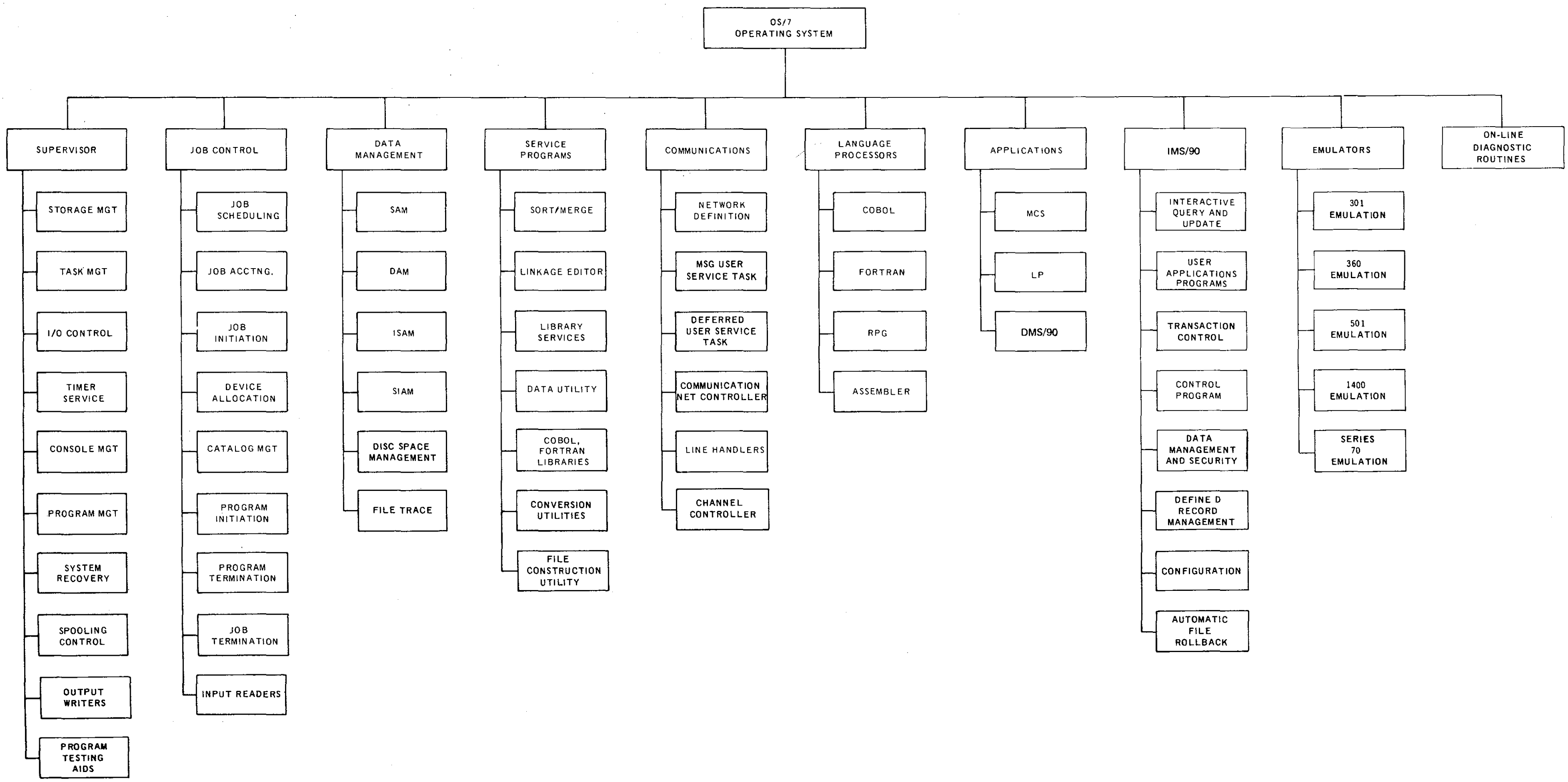


Figure 2–3. UNIVAC 90/70 Operating System (OS/7)

2.3.2. Data Management

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

Data management offers four methods of accessing files. These methods are:

- Sequential Access Method (SAM)
- Direct Access Method (DAM)
- Indexed-Sequential Access Method (ISAM)
- System Integrated Access Method (SIAM)

The first three access methods are IBM/360 DOS compatible. The fourth access method is designed for system usage.

2.3.3. Job Control

Job control is the nonresident component of the UNIVAC OS/7 Operating System (OS/7) that manages the system resources (main storage, software facilities, and peripheral devices), prepares jobs for processing, and initiates program execution. A job is a user-submitted task or unit of work to be performed. Each job can be divided into job steps to be executed serially. Job steps are made up of problem programs, with each job step containing only one request for program execution.

The services performed by OS/7 job control are directed by the user through control statements known as the job control language. These control statements convey information required by the operating system to initiate and control the processing of jobs, such as identifying the job and the programs that comprise it and specifying main storage requirements and peripheral device assignments necessary for job execution.

Figure 2–4 shows job flow and the facilities provided by job control.

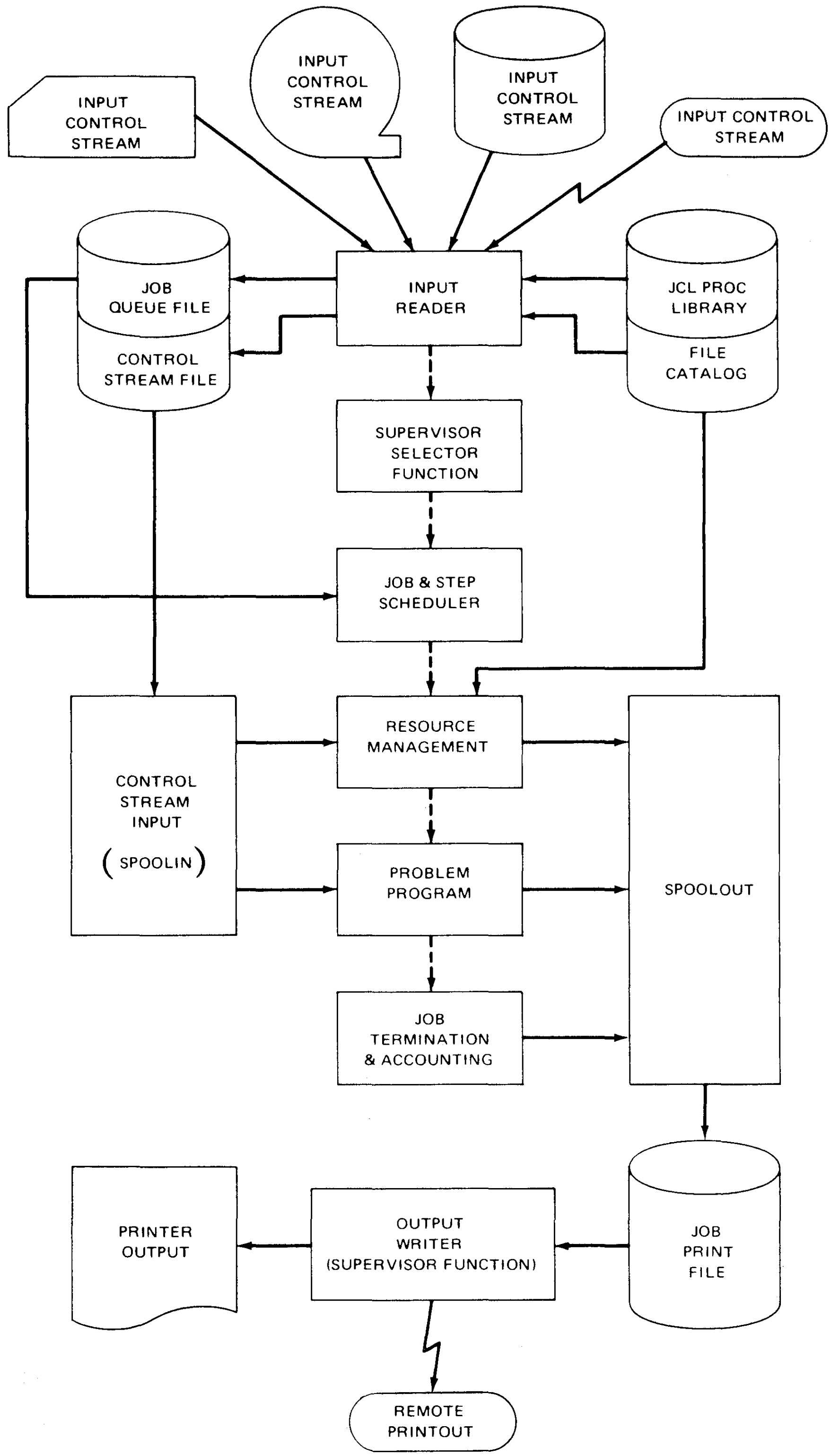
2.3.3.1. System File Catalog

Use of the system file catalog results in the ability to locate files automatically when only the file identifier is specified. Successive generations of data files may be cataloged and subsequently located by a relative generation number.

Also, control statement sequences can be prepared once and stored in a cataloged library; then the sequences can be called upon as frequently as desired. These control statement sequences are sometimes referred to as cataloged procedures.

2.3.3.2. Automatic Scheduling

The UNIVAC 90/70 System files all jobs submitted to the system into an input work queue based on a user-specified scheduling priority. The jobs may then be initiated automatically according to the scheduling priorities and the availability of system resources. The system operator is permitted to override the automatic job selection process of the system.



NOTES:

1. Solid arrows indicate data; broken arrows indicate control.

2. Neither SPOOLOUT files on magnetic tape nor card punch files are shown since neither are produced by job control.

Figure 2–4. Job Flow

2.3.4. Other Features

The UNIVAC 90/70 System software also has the following salient features:

- Automatic buffering (spooling) to disc or magnetic tape for such peripherals as the card reader or printer.
- Effective methods for recovery from device errors and system failures.
- Extensive job accounting information suitable to both local and remote batch environments.
- Use of reentrant routines to increase the utilization of main storage and decrease I/O traffic caused by program retrievals.
- Job scheduling and general operations support for a fixed-partitioned system can be optionally selected by the user.
- Emulation for the following Data Processing Systems in a multijobbing environment: IBM 360, 1401, 1440, 1460, and Series 70 real memory systems plus 301 and 501.
- Allocation algorithms for the use of the OSSF designed to increase system throughput and provide additional reliability for the system.

3. CENTRAL HARDWARE

3.1. GENERAL

The central hardware for the UNIVAC 90/70 System consists of the following components:

- Processor
- System Console
- Main Storage

Each of these components is discussed in the following paragraphs.

3.2. PROCESSOR

The UNIVAC 90/70 Processor is a medium-scale, high-performance processor with random or sequential batch processing, scientific processing, and communications processing capabilities. The processor operates under microprogram control. The micro code is resident in a separate semiconductor control storage which has a writable section.

The UNIVAC 90/70 Processor has the following characteristics:

- Industry-compatible instruction repertoire
- Hardware program relocation algorithm
- Indirect addressing
- Write or read/write storage protection
- Multiple control modes
- One multiplexer I/O channel
- One selector I/O channel
- A 15-level interrupt structure
- Program status word interrupt control
- Recovery timer facility

- Interval timer facility
- Parity generation and checking on the channel and storage buses
- System console
- Floating-point arithmetic control

The processor also has provisions for the following optional features:

- Up to three additional selector I/O channels
- Subchannel expansion and expanded interface features to enhance the multiplexer I/O channel capability
- Communication Intelligence Channel (CIC)
- Operating System Storage Facility (OSSF)
- Direct control and external interrupt

Arithmetic and input/output control comprise the major functions of the processor. The hardware associated with these functions is described in the following paragraphs.

3.2.1. Arithmetic Hardware

The arithmetic hardware performs all logical operations, arithmetic operations, data comparisons, and shifting. Fixed-point binary arithmetic uses the twos complement number representation. Floating-point and decimal arithmetic use signed absolute value number representation. This hardware also performs single or double indexing of operand addresses together with address relocation and indirect addressing.

3.2.1.1. Register Stack

The register stack contains general purpose registers in two sets of 16 each, four optional registers for floating-point arithmetic operations, eight working registers, and one relocation register.

- General Purpose Registers

  The processor can reference two sets of 16 general purpose registers. One set is reserved for the supervisor while the other set can be used by user programs. This design reduces the interrupt processing time overhead required when only a single set of general registers is used. When the processing mode is changed between user program and supervisor modes, the following steps which would be required in single register systems are unnecessary:

  1. Store the contents of user program registers.
  2. Load the executive routine data into the registers.
  3. Store the executive routine data.
  4. Reload the user program data back into the registers.

  These registers can be used for fixed-point arithmetic, logical arithmetic, and the indexing of instruction and operand addresses. The capacity of each register is 32 bits. The registers are identified by the hexadecimal numbers 0 through F. The general purpose registers are addressable only through the specific instruction fields provided for their access.

■ Floating-Point Registers

Four doubleword floating-point registers are provided to avoid unnecessary storing and loading operations for results and operand. These registers are numbered 0, 2, 4, and 6. For long format instructions, both words of the register are involved in the operation. For short format instructions, only the first word in the register is involved in the operation; the content of the second word in the register is ignored and not changed. If an odd-numbered register is specified, a program exception interrupt request in generated.

■ Working Registers

These registers are used for temporaty storage of operands and intermediate results when executing decimal arithemetic operations. Their primary function is to increase execution speed. These registers are not addressable by the user.

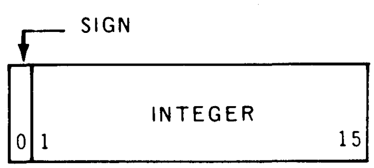
■ Relocation Register

This register is used by both the processor and the input/output channels to provide automatic program relocation.

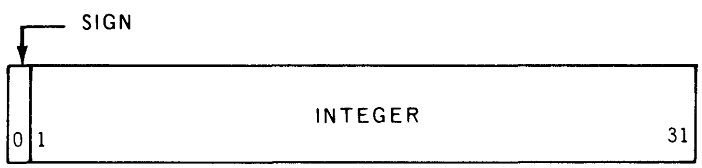
3.2.1.2. Fixed-Point Arithmetic

Fixed-point numbers have a fixed-length format comprising a sign bit followed by an integer field. When the sign bit is 1, the integer represents a negative value; when the sign bit is 0, the integer represents a positive value.

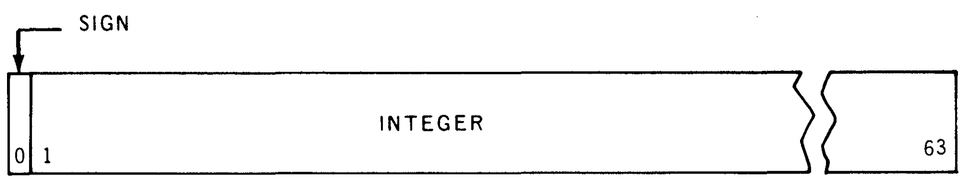
Halfword Format



Fullword Format



Doubleword Format



When held in one of the 16 general registers, a fixed-point number is generally treated as a 32-bit operand. When a halfword fixed-point number is called from storage and loaded into a register, the sign is extended to the left to fill the fullword register. The contents of the register are then handled as a fullword operand in fixed-point arithmetic operations.

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

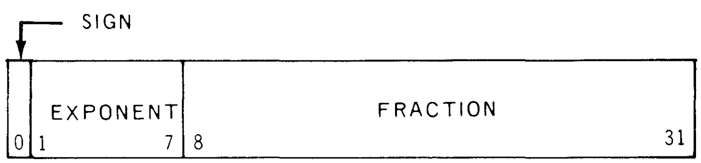
When fixed-point data is located in storage, it may be stored as a halfword, fullword, or doubleword. This data must be located on the integral storage boundary of its associated format.

3.2.1.3. Floating-Point Arithmetic

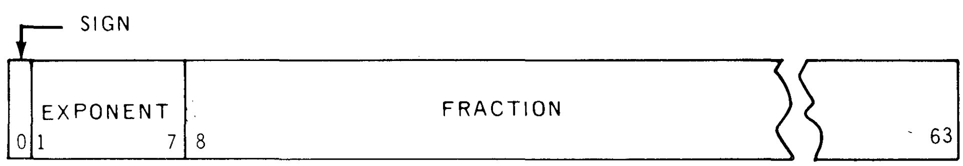
The UNIVAC 90/70 System can provide floating-point arithmetic operations as an optional feature. A floating-point number comprises a biased exponent (characteristic) and a signed fraction (mantissa). The biased exponent is expressed in excess 64 binary notation; the fraction is expressed as a hexadecimal number having a radix point to the left of the high-order digit. The quantity expressed by the full floating-point number is the product of the fraction and the number 16 raised to the power of the biased exponent minus 64.

Floating-point numbers are either a fullword (short format), or a doubleword (long format) in length. Both formats can be used in main storage or in the floating-point registers. In either format, bit 0 is the sign bit of the fraction, and bits 1 through 7 are exponent. The fraction field comprises bits 8 through 31 in the short format and bits 8 through 63 in the long format.

Short Format



Long Format



The floating-point instruction set provides for loading, adding, subtracting, comparing, multiplying, dividing, storing, and sign control of short or long format operands. Short format operands provide faster processing and require less storage space than long format operands. Long format operands provide greater precision in computation.

A normalized floating-point number has a nonzero high-order hexadecimal fraction digit and is the most significant representation of a given quantity. The process of normalization consists of shifting the fraction to the left until the high-order hexadecimal digit is nonzero, and reducing the characteristic by the number of shifts. A fraction with one or more high-order digits of zero is unnormalized. A zero fraction cannot be normalized.

3.2.1.4. 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 a multidigit 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 lease significant four bits. The zone portion of the lease 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 3-digit operand follows:

BYTE

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

UNPACKED DECIMAL OPERAND

In the packed format, each byte contains two digits. The least significant four bits of the least significant byte provide the sign of the number. The packed decimal format is used for all decimal arithmetic operations. The format of a 4-digit number follows:

BYTE

ZERO FILL	DIGIT	DIGIT	DIGIT	DIGIT	SIGN
0	7	8	15	16	23

PACKED DECIMAL OPERAND

Decimal numbers (0 through 9) are represented in the 4-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 National Standard Code for Information Interchange (ASCII) modified to eight bits, or Extended Binary Coded Decimal Interchange Code (EBCDIC). The codes 1100 (plus) and 1101 (minus) are hardware-generated in either the ASCII or EBCDIC mode. A control bit in the program status word determines whether the system is to operate in the ASCII or the EBCDIC mode. See Appendix C.

3.2.1.5. 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 in the general purpose registers or main storage. The instruction used determines whether the logical operation is to be performed in main storage or in a register.

3.2.2. Input/Output

The input/output (I/O) hardware of the UNIVAC 90/70 Processor initiates, directs, and monitors the transfer of data between storage and the peripheral subsystems. After the I/O instruction has been initiated, the data transfer is performed concurrently with other processor functions. The selector channels, the multiplexer channel, the OSSF, the CIC, and the processor can all operate concurrently.

The I/O hardware consists of I/O control logic and the input/output channels using a standard UNIVAC Series 90 I/O interface to connect channels with the unit controllers. This interface is identical for all the I/O control units (except for the CIC) designed for use with all I/O devices currently available on UNIVAC Series 90 Systems as well as for future devices.

3.2.2.1. Selector Channels

One selector channel is included in the basic system configuration; three selector channels are available as optional features to the basic configuration. High-speed devices, such as UNISERVO 12, 16, or 20 Magnetic Tape Subsystems, and UNIVAC 8411, 8414, 8424 and/or 8440 Disc Subsystems, are connected only 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 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 tranafer must be completed before another device on the channel can transfer data.

3.2.2.2. Multiplexer Channel

The processor has one multiplexer channel which has eight physical connections to which standard control units (for such devices as a card reader, card punch, or line printer, and a Data Communications Subsystem (DCS) can be attached. The address format for the multiplexer channel provides for 15 subchannel addresses. These can be a combination of standard and, if a DCS is conncted to the multiplexer channel, DCS subchannel addresses.

Subchannel expansion and expanded interface are two features that can be added to the basic multiplexer channel. The subchannel expansion feature provides an additional 16 subchannel addresses for a total of 31. The expanded interface feature provides eight additional physical connections if the subchannel expansion feature has been installed. Without the subchannel expansion feature, the limit of 15 subchannel addresses prevents expansion to more than 15 physical connections.

The standard control unit occupies one physical connection and uses one standard subchannel address. Devices such as the system console, card reader, line printer, or UNISERVO VI-C Magnetic Tape Subsystem are connected to the processor through the standard control unit. Each DCS occupies one physical connection but uses up to 28 DCS subchannel addresses. The DCS may be used for data communications over 1 to 14 communication lines.

3.2.3. Processor Modes

There are seven bits in the active program status word which determine processor modes of operation. These modes of operation are described in the following paragraphs.

- Selection of Problem or Supervisor Mode

  When operating in the problem mode, a program cannot execute privileged instructions. Also, the program performs only the main storage operations permitted by the storage protection key. When operating in the supervisor mode, there are no restrictions on the instruction usage, and storage protection still applies.

- Selection of Register Set

  The processor contains two sets of general purpose registers. The user program general registers can be used during the execution of instructions in the problem register mode. Similarly, the supervisor general registers can be used by the supervisor.

- Selection of ASCII or EBCDIC Code Interpretation

  Certain processor instructions interpret or generate code-sensitive operands. Zone fields and edit control characters are fields to which the processor is sensitive. Internal system code is intepreted as either ASCII or EBCDIC code as determined by the mode that is selected.

- Processor Wait Mode

  The processor wait mode is provided to allow the program to suspend instruction processing. This mode is particularly useful when exceptionally high I/O activity precludes a meaningful rate of job execution, when all jobs are waiting for I/O operations to be completed, and when no jobs are in the system.

- Program Trace Mode

  When operating in this mode, the execution of a successful branch instruction causes the address of the instruction following the branch instruction and the instruction length code of the branch instruction to be written into a table in main storage.

- Monitor Mode

  A program analysis level interrupt is generated prior to the execution of all instructions fetched while in this mode.

- 9400 Compatibility Mode

  This mode is provided to inform the software of UNIVAC 9400 System compatibility requirements.

- Special Emulation Mode

  This mode permits the UNIVAC 90/70 Processor to access microprograms stored in the writable section of control storage.

3.2.4. Interval Timer

An interval timer is included as an integral component of the UNIVAC 90/70 Processor. It provides the software with a relative running-time count and an incremental interrupt count. The running time counter and the interrupt counter are updated once every millisecond.

The running time counter allows for a continuous measure of elapsed time. The count may be incremented or decremented. The counter value changes from maximum to zero after incrementing, and from zero to maximum after decrementing without causing program intervention.

The interrupt counter allows the interval between interrupts to range from one millisecond to 64 seconds.

3.2.5. Interrupt Processing Control

The UNIVAC 90/70 System contains an efficient multilevel interrupt system. The processor can react to external and internal error conditions or monitoring conditions by means of this interrupt system. The hardware and associated software allow the processor to change from the user program state to the privileged or supervisor state. The types of interrupts employed in the UNIVAC 90/70 System are:

- Machine Checks

  This interrupt request occurs when a hardware malfunction is detected by the processor or when a hardware malfunction not identified as a subsystem fault is detected by the I/O section, or when a program exception is generated while being masked.

- Program Exception

  This interrupt request occurs when a program error is detected by the hardware. The interrupt is generated as a result of one of the following:

  - Operation Exception

    An illegal processor operation has been attempted or an operation using a nonexistent processor feature has been attempted.

  - Privileged Operation Exception

    A privileged operation has been encountered in a problem mode.

  - Execute Exception

    The object of an execute instruction is another execute instruction.

  - Protection Exception

    The key in key storage does not match the key in the program status word.

  - Addressing Exception

    Reference is made to a nonexistent storage location.

  - Specification Exception

    An integral boundary reference error has been made; general register pairs or floating-point registers have been specified incorrectly; the length of decimal fields is incorrect. The control indicators set by an LBR or BCRE instruction are invalid.

  - Data Exception

    The operands in decimal and editing operations contain incorrect digit and sign codes; decimal arithmetic fields are aligned incorrectly.

  - Fixed-Point Overflow Exception

    The result of a fixed-point arithmetic operation has caused a high-order carry, or a shift operation has caused the loss of significant bits.

  - Fixed-Point Divide Exception

    The quotient exceeds the size of the associated register, or the result of a decimal-to-binary conversion operation exceeds 31 bits.

  - Decimal Overflow Exception

    The capacity of the result field is exceeded during a decimal arithmetic operation.

  - Decimal Divide Exception

    The quotient field exceeds the capacity of the quotient part of the result field.

- Exponent Overflow Exception

  The characteristic result exceeds 127 during a floating-point arithmetic operation.

- Exponent Underflow Exception

  The characteristic result is less than zero during a floating-point operation.

- Significance Exception

  A floating-point addition or subtraction results in a zero fraction.

- Floating-Point Divide Exception

  Floating-point division is attempted with a zero divisor fraction.

- Indirect Address Exception

  An indirect address control word (IACW) with an incorrect format has been referenced.

- Indirect Address Specification Exception

  A main storage reference has exceeded the 8-level indirect addressing capability.

■ Program Analysis

this interrupt request occurs during program monitoring and certain program tracing operations.

■ Supervisor Call

This interrupt request occurs as a result of the execution of a supervisor call (SVC) instruction and may have up to 256 different states which are established by the software.

■ External

This interrupt is associated with maintenance trace, interrupt key, and the direct control and external interrupt feature. The direct control and external interrupt feature provides for the direct connection between two UNIVAC 90/70 Processors. The external interrupt request associated with this feature occurs when certain of the signal-in lines associated with the direct control interface of an object processor are enabled. The states within this level are:

Maintenance Trace

Interrupt Key

External Signal 2 State*

External Signal 3 State*

External Signal 4 State*

External Signal 5 State*

External Signal 6 State*

External Signal 7 State*

**This state is a part of the direct control and external interrupt feature.*

- Timer

  This interrupt request occurs when a present time interval expires.

- Communications Intelligence Channel

  This interrupt request occurs when a communications status word is tabled in main storage.

- Other Input/Output Channels

  An interrupt request occurs when status is generated or received from a subsystem during channel or subchannel operations. Each channel has its own interrupt to reduce software overhead associated with interrupt analysis.

  Selector Channel 1

  Selector Channel 2

  Selector Channel 3

  Selector Channel 4

  Operating System Storage Facility

  Multiplexer Channel Standard

  Multiplexer Channel DCS

3.2.6. Instruction Repertoire

The power and flexibility of the UNIVAC 90/70 System are reflected in the instruction repertoire and their execution times. The full repertoire includes all nonprivileged instructions of the IBM System/360 universal instruction set. The instruction repertoire has also been enhanced to facilitate dynamic program relocation and indirect addressing.

3.2.6.1. Instruction Types

Instructions can be two, four, or six bytes in length. All instructions have an even address. The object formats of the five instruction types are shown in Figure 3–1. The symbols used in Figure 3–1 are explained in Table 3–1.

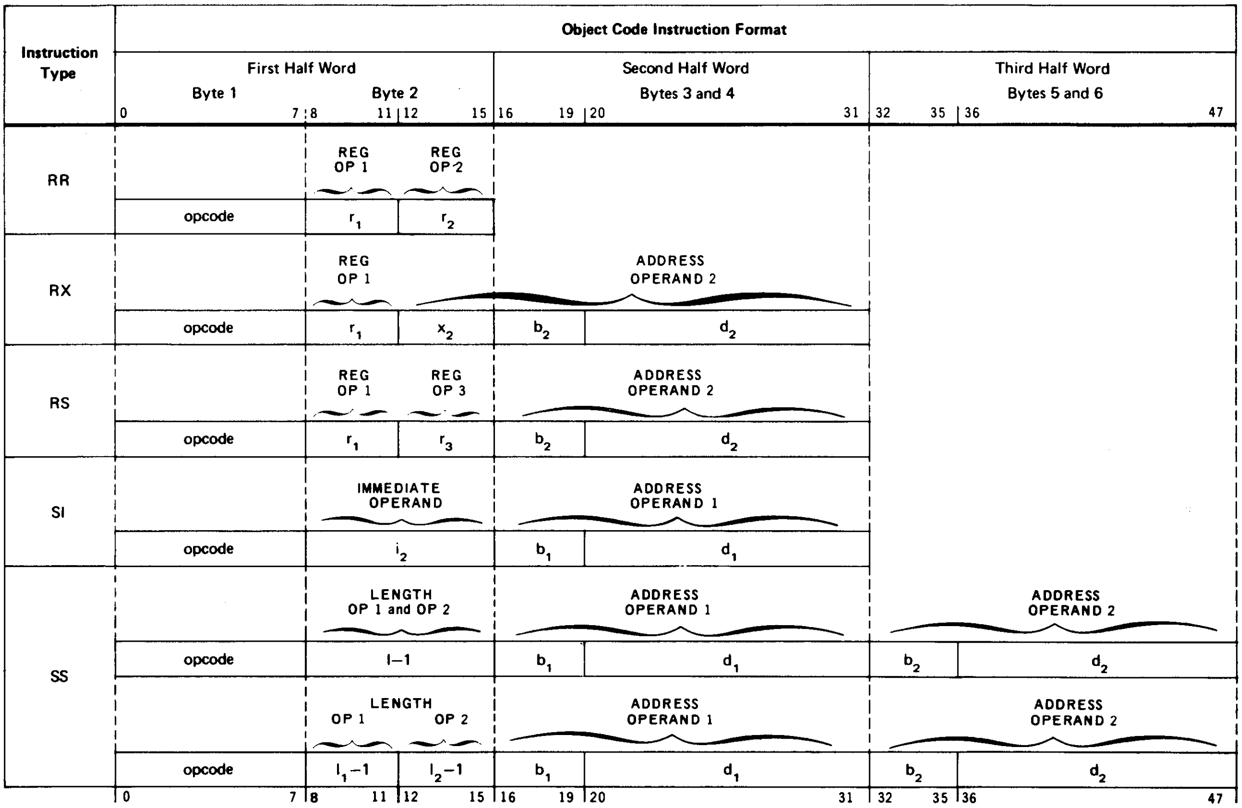


Figure 3–1. Basic Instruction Formats (Object Code Form)

- Register to Register (RR) Instructions

  The RR type instructions are used to process data contained in registers. The maximum length of the data that can be handled is a doubleword. The data may be a signed or unsigned binary number, a short or long format floating-point number, or a decimal number, depending on the specified operation. Operand 1 specifies either a register or a mask. Operand 2 specifies a register.

  Some RR type instructions use both operands 1 and 2 as an immediate data operand.

- Register and Indexed Storage (RX) Instructions The RX type instructions are used to process data between registers and indexed storage. The maximum length of the operand that can be handled is a doubleword. The data may be a signed or unsigned binary number, a shour or long format floating-point number, or a decimal number. Operand 1 specifies a register or a mask. Operand 2 specifies a main storage location, which may be further modified by a relocation and indirect addressing, and the number of a register containing an index value.

- Register to Storage (RS) Instructions

  The RS type instructions are used to perform multiple register and storage operations as well as data shifting. The first and third operands specify the numbers of two general registers or the boundaries for general register usage. Operand 2 specifies a main storage location, which may be further modified by relocation and indirect addressing, or maybe a shift count.

- Storage and Immediate Operand (SI) Instructions

  The SI type instructions are used to perform operations on an 8-bit value, called immediate data, and an operand in storage. Operand 2 specifies the immediate data or mask. Operand 1 specifies a 1-byte or halfword storage location which may be further modified by relocation and indirect addressing depending on the operation.

- Storage to Storage (SS) Instructions

  The SS type instructions are used to perform operations on two operands located in storage. In logical operations the operands are assumed to be equal in length and may be from 1 to 256 bytes. In decimal operations the operands may be different lengths and may be from 1 to 16 bytes.

Table 3–1. Symbols Used to Describe Operand Formats

SYMBOL	MEANING
OP CODE	Instruction operation code
R_1	Number of the register addressed as operand 1, a mask, or a register which is the first register of a multiregister group
R_2	Number of the register addressed as operand 2
R_3	An expression representing a register which is the last register in a multiregister group, an increment, an operand address, or a control storage address.
X_2	Number of the register to be used as an index for operand 2 of an RX instruction
I_2	Immediate data used as operand 2 of an SI instruction
L	Length of operands 1 and 2 as stated in source code*
L_1	Length of operand 1 as stated in source code*
L_2	Length of operand 2 as stated in source code*
B_1	Base register for operand 1
B_2	Base register for operand 2
D_1	Displacement for operand 1
D_2	Displacement for operand 2
OP1	Operand 1
OP2	Operand 2
OP3	Operand 3

*This is coded as the true source length of the operand, not the length less 1, as required in object code. The assembler makes a reduction of 1 in length when converting source to object code.

3.2.6.2. Nonprivileged Instruction Set

This instruction set includes the 132 nonprivileged instructions of the IBM System/360 universal instruction set, an add immediate (AI) instruction, three instructions for indirect addressing (BALE, LBR, and BCRE), and an emulation aid (EA) instruction. Instructions are provided to process a fixed-length binary numbers, floating-point numbers, packed and unpacked decimal numbers, and EBCDIC or ASCII characters. Data may be transferred between main storage and the user program set of general registers and from stroage to storage. The operations of shifting, branching, and logical functions are also included. All nonprivileged instructions are listed in Appendix A.

3.2.6.3. Privileged Instruction Set

The privileged instruction set is used by the software operating system when operating in the supervisory state. In this state, all instructions are valid and can be executed. This set of instructions includes the facility to load the current program status word (PSW), entirely or in parts; to load and store the contents of the key storage; to utilize the direct control and external interrupt feature; to control the I/O channels and subsystems; and to load the writable section of control storage. Instructions in the privileged set cannot be executed in a user program. The privileged instructions are listed in Appendix A.

3.2.6.4. Indirect Addressing

The UNIVAC 90/70 System nonprivileged instruction set is extended to include three additional instructions: load base relativize (LBR), branch and link external (BALE), and branch on condition to return external (BCRE). These instructions are the operating commands for the UNIVAC 90/70 indirect addressing function. The indirect addressing function is a unique design which has no equivalent in the current industry-compatible instruction set.

Indirect addressing provides uncomplicated entry to reentrant system subroutines under control of the supervisor. The user need not directly indicate the address of the subroutine in a supervisor request. Instead, the reference to the subroutine is implicitly indicated in the transfer address of the BALE instruction. The BALE instruction then references the associated indirect address control word (IACW) contained in a vector table in main storage. The IACW indicates the address of the reference; the IACW also has control bits which can indicate additional indirection (up to eight levels), switch from relative or absolute addressing, or cause program exception interrupts. The BALE instruction also provides a means for the user to examine the current control indicators.

Instructions used for operations such as shifting and constant generation are not sensitive to the operand control indicators.

The indirect addressing function in the UNIVAC 90/70 System has the following advantages when referencing system subroutines.

- No interrupt occurs (as with SVC), reducing system overhead, when the subroutine is in main storage.
- The supervisor is capable of moving subroutines without impacting user programs.
- User region sizes are reduced, since code with a low usage frequency is moved into main storage only when referenced.
- The user can place application subroutines or data bases, such as rate tables, under supervisor control.

3.2.6.5. Address Relocation

Another unique design feature in the UNIVAC 90/70 System is address relocation. The address relocation scheme utilizes a hardware relocating register that is dynamically loaded by hardware from a fixed main storage position each time the protection key of the program in control of the system changes.

These relocation registers define virtual address zero in the associated regions of main storage, and the contents of a register are automatically added to all the storage references generated by a user porgram.

This scheme provides a flexible and efficient rollout/rollin capability, because the supervisor can locate the rolled-in program in any portion of main storage by adjusting the associated relocation register.

Address relocation increases system throughout because:

- Storage utilization is improved. There is a decrease in checkerboarding and a resulting reduction of unused space. This unused space can be consolidated to provide additional main storage regions of useful size.
- Except in special cases, the program loader does not have to relocate programs as they are being loaded. This saves considerable processor time.

3.3. MAIN STORAGE

The main storage of the UNIVAC 90/70 System consists of high-speed, semi-conductor storage developed for the UNIVAC Series 90 computers. Main storage is contained in freestanding units with 600 nanosecond read or write cycle time for a fullword.

Minimum storage size is 131,072 bytes. This can be expanded by the following:

65,536-byte increments from 131,072 to 262,144 bytes; and

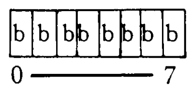
131,072-byte increments from 262,144 bytes to 1,048,576 bytes.

Address and data are checked for odd parity.

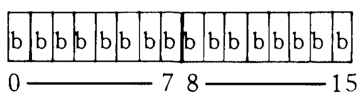
Each main storage cabinet is equipped with seven switches. These switches are used as the most significant bits of the cabinet starting address. If it becomes necessary to place a 131K storage module offline, the switches on the cabined with the highest addresses can be set to the starting address of the module removed from service. This permits the remaining online modules to have sequential addresses.

3.3.1. INFORMATION POSITIONING

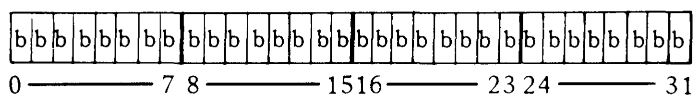
Locations in main storage are addressed consecutively from 0 through a maximum of 1,048,575. Bytes may be accessed separately or in groups. A group of bytes is addressed by the leftmost byte of the group. The bits in a byte are numbered from left to right starting with zero.



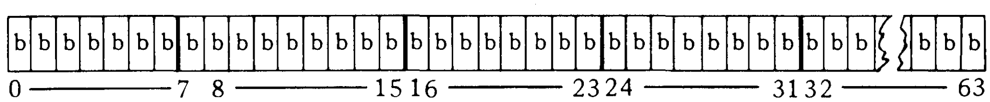
Halfword data formats consist of two consecutive bytes.



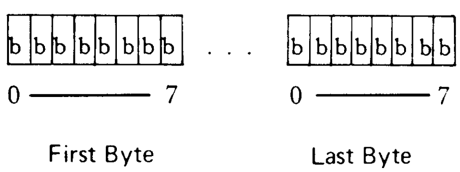
Fullword data formats consist of four consecutive bytes.



Doubleword data formats consist of eight consecutive bytes.



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



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 2, a fullword must have an address that is a multiple of 4, and a doubleword must have an address that is a multiple of 8.

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

3.3.2. Low Order Storage

The low order 640 bytes of main storage have been reserved to contain specific operating information. The information stored in these locations is accessed by the hardware and the operating system during the execution of appropriate functions. The operating system provides for the loading and protecting of the information in these locations.

3.3.3 Storage Protection

Program integrity in a multijobbing environment is guaranteed by the storage protection capability implemented in the processor and I/O channels. Sixteen keys are provided for this purpose. Write and read/write protection are provided.

Protection is provided in 2048-byte blocks of main storage. The supervisor assigns each block a storage protection key. A comparison is made between a requester key and the appropriate storage protection key. When the keys match, the storage operation specified is performed. When the keys do not match and the block specifies write storage protection, a read operation is allowed and a write operation is disallowed. When the keys do not match and the block specifies read/write storage protection, all operations are disallowed. In the case of disallowed operations, a program exception interrupt request is generated.

3.3.4. Program Relocation Registers

There are 16 program relocation registers located in low order main storage. These registers serve as base registers for the program modules in storage. A copy of the contents of the register associated with the current program status word is stored in a register in the control hardware to facilitate rapid storage addressing.

3.4. SYSTEM CONSOLE

The UNIVAC 90/70 System Console is a freestanding input/output device for directing and monitoring the operation of the system. (See Figure 3–2.) The system console provides a centralized location for initial load control, run/stop control, and system status monitoring. The system console consists of a keyboard and visual display unit, switches and indicators, which are housed in a cabinet that is separate from the processor. The system console communicates with the processor through the multiplexer channel.

Figure 3–2. UNIVAC 90/70 System Console

The keyboard and visual display unit, a UNISCOPE 100 Display Terminal, is capable of displaying 96 characters, including uppercase and lowercase alphabets, on the visual display screen. The visual display screen has a screen capacity of 1024 characters at a time. The total number of characters displayed is subdivided into 16 rows of 24 characters each. All communications between the operator and the operating system are displayed. The operator controls consist of an alphanumeric typewriter keyboard, cursor control keys, editing keys, data control keys, control knobs, and indicators. Thus, the operator can key-in a message, see it displayed on the visual display screen, edit it, and transmit it to the processor.

The system console includes all controls and indicators necessary to operate and monitor the operation of the system. A panel is provided to allow operator control of the switches, and to present an indication of the system configuration for the operator.

The system console cabinet also provides space for mounting the UNIVAC 9000 Series Channel Adapter.

As an optional feature, a UNIVAC incremental printer can be connected to the system console to provide additional hard copy output. It may be used to duplicate all messages displayed on the visual display screen. The UNIVAC incremental printer is mounted in a separate cabinet. This printer has a 94-character set (including uppercase and lowercase), a 132-position print line, prints up to 30 characters per second, and has a paper feed rate of 30 lines per second.

4. PERIPHERAL EQUIPMENT

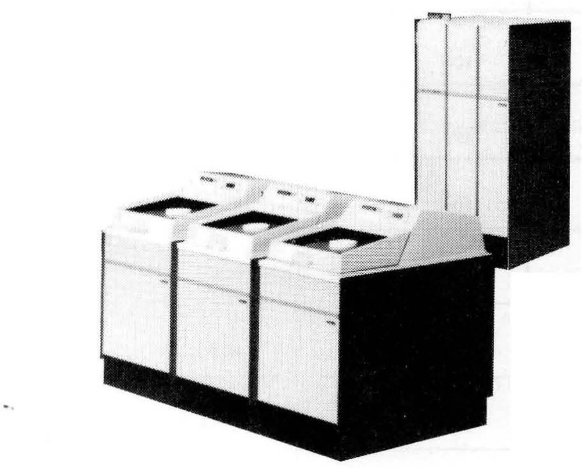
4.1. GENERAL

A full line of onsite peripheral equipment is available for use with the UNIVAC 90/70 System. These subsystems are:

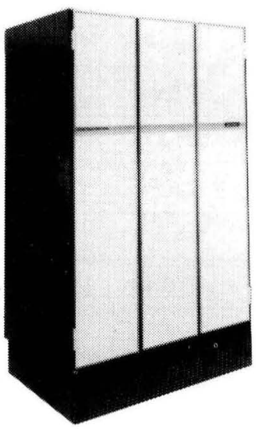
- UNIVAC 8411 Disc Subsystem
- UNIVAC 8414 Disc Subsystem
- UNIVAC 8424 Disc Subsystem
- UNIVAC 8440 Disc Subsystem
- UNISERVO VI-C Magnetic Tape Subsystem
- UNISERVO 12/16 Magnetic Tape Subsystem
- UNISERVO 20 Magnetic Tape Subsystem
- UNIVAC 0711 Card Reader Subsystem
- UNIVAC 0716 Card Reader Subsystem
- UNIVAC 0768 Printer Subsystem
- UNIVAC 0770 Printer Subsystem
- UNIVAC 0604 Card Punch Subsystem
- UNIVAC 0920 Paper Tape Subsystem
- UNIVAC 2703 Optical Document Reader
- UNISCOPE 100 Display Terminal
- UNIVAC 9000 Series Channel Adapter may be used to attach the following systems:
  - UNIVAC 9200 System
  - UNIVAC 9300 System

- UNIVAC 9400 System
- UNIVAC 90/60 System
- UNIVAC 90/70 System

4.2. DISC SUBSYSTEMS



UNIVAC 8411/8414 Disc Subsystem



CONTROL UNIT

DISC STORAGE UNIT

UNIVAC 8424/8440 Disc Subsystem

UNIVAC 8411/8414/8424/8440 DISC SUBSYSTEMS CHARACTERISTICS				
CHARACTERISTIC	**UNIVAC 8411**	**UNIVAC 8414**	**UNIVAC 8424**	**UNIVAC 8440**
NUMBER OF DRIVES	1–8	2–8	2–8	2–8
NUMBER OF DISC DRIVES PER CABINET	1	1	2	2
NUMBER OF R/W HEAD ACCESSOR MECHANISMS PER DISC DRIVE	1	1	1	1
NUMBER OF R/W HEADS PER DISC DRIVE	10	20	20	20
NUMBER OF TRACKS PER DISC SURFACE	203	203	406	406
NUMBER OF DATA RECORDING SURFACES PER DISC DRIVE	10	20	20	20
NUMBER OF USABLE TRACKS PER SURFACE	200	200	400	400
NUMBER OF USABLE TRACKS PER DISC DRIVE	2000	4000	8000	8120
MAXIMUM NUMBER OF BYTES PER TRACK	3625	7294	7294	14,910
CAPACITY (8-BIT BYTES PER DISC PACK)	7,250,000	29,176,000	58,352,000	119,280,000
MINIMUM ARM POSITIONING TIME	25 ms	25 ms	10 ms	10 ms
AVERAGE ARM POSITIONING TIME	75 ms	60 ms	30 ms	30 ms
MAXIMUM ARM POSITIONING TIME	135 ms	130 ms	55 ms	55 ms
AVERAGE LATENCY TIME	12.5 ms	12.5 ms	12.5 ms	12.5 ms
MAXIMUM LATENCY TIME	25 ms	25 ms	25 ms	25 ms
AVERAGE ACCESS TIME	87.5 ms	72.5 ms	42.5 ms	42.5 ms
MAXIMUM ACCESS TIME	160 ms	155 ms	55 ms	80 ms
DISC DRIVE SPEED	2400 rpm	2400 rpm	2400 rpm	2400 rpm
STORAGE TRANSFER RATE	156,000 bytes/ second	312,000 bytes/ second	312,000 bytes/ second	624,000 bytes/ second

The availability of four different disc subsystems enables the user to choose the disc storage facilities best suited to the installation. These facilities include the lower cost UNIVAC 8411 Disc Subsystem, the higher capacity UNIVAC 8414 Disc Subsystem, and the high performance large capacity of the UNIVAC 8424 and 8440 Disc Subsystems.

A disc subsystem is attached to the processor by means of a selector channel. These subsystems offer many advantages in standard data processing as well as in communications operations, especially in applications where rapid file processing is more prevalent. Large storage capacity is combined with rapid accessibility to provide convenient intermediate storage. Removable and interchangeable disc packs permit the user to store much of the total file capacity offline.

These subsystems allow 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 subsystems permits lesser used program segments to be stored externally and read into main storage only when required. This arragement affords efficient usage of main storage. Because the operating system is disc oriented, handling, access, and transfer time required for compiling or assembling programs and for input/output operations is reduced. This allows all the magnetic tape units to be used to meet primary input/output data demands.

4.2.1. UNIVAC 8411 Disc Subsystem

Each UNIVAC 8411 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 heads in unison between the periphery and the central area of the disc. The accessor mechanism can assume any 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 three cylinders reserved for replacement tracks. Each cylinder contains ten tracks numbered 0 through 9. The addressing of an individual track is by cylinder number (000–202) and by read/write head number (0–9). The storage transfer rate is 156K bytes per second.

4.2.2. UNIVAC 8414 Disc Subsystem

Each UNIVAC 8414 disc pack contains 11 discs with the data recorded on the 20 inside surfaces. Twenty read/write heads are mounted on a single accessor mechanism which moves the heads in unison between the periphery and the central area of the disc. The accessor mechanism can assume any one of 203 positions across the disc surface; this simultaneous head movement on the 20 disc surfaces creates 200 addressable data recording cylinders in the disc pack, with three cylinders reserved for replacement tracks. Each cylinder contains 20 tracks numbered 0 through 19. The addressing of an individual track is by cylinder number (000–202) and by read/write head number (0–19). The storage transfer rate is 312K bytes per second.

The 8414 Disc Subsystem has an optional feature which permits simultaneous operation of two disc drives.

4.2.3. UNIVAC 8424 Disc Subsystem

Each UNIVAC 8424 disc pack contains 11 discs with the data recorded on 20 of the inside surfaces. Twenty read/write heads are mounted on a single accessor mechanism which moves the heads in unison between the periphery and the central area of the disc. The accessor mechanism can assume any of 406 positions across the disc surface; this simultaneous head movement on the 20 disc surfaces creates 400 addressable data recording cylinders in the disc pack, with six cylinders reserved for replacement tracks. Each cylinder contains 20 tracks numbered 0 through 19. The addressing of an individual track is by cylinder number (000–405) and by read/write head number (0–19). The storage transfer rate is 312K bytes per second.

4.2.4. UNIVAC 8440 Disc Subsystem

Each UNIVAC 8440 disc pack contains 11 discs with the data recorded on 20 of the inside surfaces. Twenty read/write heads are mounted on a single accessor mechanism which moves the heads in unison between the periphery and the central area of the disc. The accessor mechanism can assume any of 406 positions across the disc surface; this simultaneous head movement on the 20 disc surfaces creates 400 addressable data recording cylinders in the disc pack, with six cylinders reserved for replacement tracks. Each cylinder contains 20 tracks numbered 0 through 19. The addressing of an individual track is by cylinder number (000–405) and by read/write head number (0–19). The storage transfer rate is 624K bytes per second.

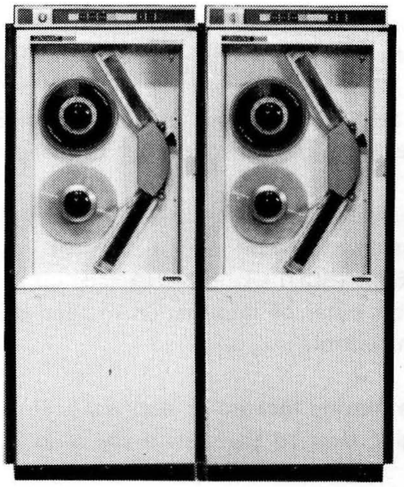
4.3. UNISERVO MAGNETIC TAPE SUBSYSTEMS

The availability of the UNISERVO VI-C, 12, 16, and 20 Magnetic Tape Subsystems provides the user with a wide choice of equipment cost and capability. This freedom of choice permits the user to select the subsystem or combination of subsystems that best suits installation requirements.

UNISERVO VI-C/12/16/20 SUBSYSTEMS CHARACTERISTICS

CHARACTERISTIC	UNISERVO VI-C	UNISERVO 12	UNISERVO 16	UNISERVO 20
TAPE UNITS PER SUBSYSTEM	2 to 8	1 to 16	1 to 16	1 to 16
DATA TRANSFER RATE (MAX.)	34,160 frames/ second	68,320 frames/ second	192,000 frames/ second	320,000 frames/ second
TAPE SPEED	42.7 inches/ second	42.7 inches/ second	120 inches/ second	200 inches/ second
TAPE DIRECTION:				
READING	Forward or backward	Forward or backward	Forward or backward	Forward or backward
WRITING	Forward	Forward	Forward	Forward
TAPE LENGTH (MAX.)	2400 feet	2400 feet	2400 feet	2400 feet
TAPE THICKNESS	1.5 mils	1.5 mils	1.5 mils	1.5 mils
BLOCK LENGTH	Variable	Variable	Variable	Variable
INTERBLOCK GAP:				
9-TRACK	0.6 inch	0.6 inch	0.6 inch	0.6 inch
7-TRACK		0.75 inch	0.75 inch	
INTERBLOCK GAP TIME (9 TRACK):				
NONSTOP	14.1 ms	14.1 ms	5.0 ms	3.0 ms
START-STOP	20.1 ms	20.1 ms	8.0 ms	5.0 ms
INTERBLOCK GAP TIME (7 TRACK):				
NONSTOP	Not available	17.6 ms	6.25 ms	
START-STOP		23.6 ms	9.25 ms	
REVERSAL TIME	25 ms	25 ms	10 ms	16 ms
REWIND TIME	3 minutes	3 minutes	2 minutes	1 minute
SIMULTANEOUS OPERATION	Available with two sub-systems	Optional	Optional	Optional
PULSE DENSITY:				
9-TRACK	800 ppi	1600 ppi 800 ppi	1600 ppi 800 ppi	1600 ppi
7-TRACK	800 ppi 556 ppi 200 ppi	800 ppi		
RECORDING MODE:				
9-TRACK	NRZI	Phase encoded	Phase encoded	Phase encoded
7-TRACK		NRZI	NRZI	

4.3.1. UNISERVO VI-C Magnetic Tape Subsystem



The UNISERVO VI-C Magnetic Tape Subsystem is a low cost magnetic tape subsystem which comprises a control and from two to eight magnetic tape units. The master/slave concept is employed in the logic of the UNISERVO VI-C Magnetic Tape 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.

The control, one master tape unit, and one slave unit are housed in a single cabinet. Additional tape units are added as needed. A second master unit is needed when more than four tape units are used. A subsystem of up to eight tape units is connected to the UNIVAC 90/70 System by means of the multiplexer channel. A separate multiplexer physical connection and shared subchannel address are required for each subsystem.

UNISERVO VI-C Magnetic Tape Units are available in the 9-track model only, and read and record data in 8-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 frames per inch, this being 34,160 frames per second.

UNISERVO VI-C Magnetic Tape Subsystems are employed in the following:

- System recovery tapes
- Software distribution tapes
- Audit trails and journal tapes
- Spooling I/O tapes

4.3.2. UNISERVO 12/16 Magnetic Tape Subsystem

A UNISERVO 12/16 Magnetic Tape Subsystem for the UNIVAC 90/70 System is available in two forms: UNISERVO 12 Magnetic Tape Subsystem and UNISERVO 12/16 Magnetic Tape Subsystem. The UNISERVO 12 Magnetic Tape Subsystem includes one UNISERVO 12 control and accommodates from 1 to 16 UNISERVO 12 Magnetic Tape Units. The UNISERVO 12/16 Magnetic Tape Subsystem includes one UNISERVO 12/16 control and accommodates from 1 to 16 UNISERVO 12 Magentic Tape Units, UNISERVO 16 Mangetic Tape Units, or any combination of both types of tape units. A UNISERVO 12/16 Subsystem is connected to the processor by means of a selector channel.

UNISERVO 12 and UNISERVO 16 Mangetic Tape Units are available in 9-track and 7-track models. The 9-track tape units produce a higher rate of throughput. At this recording level, data is phase encoded in 9-bit frames across the width of the tape. Each frame contains eight data bits plus one parity bit (one byte) at a density of 1600 frames per inch.

An optional dual density feature can be added to the 9-track units. This feature enables each UNISERVO 16 Magnetic Tape Unit, or UNISERVO 12 Master Magnetic Tape Unit and the 9-track slave units it controls, to read and write data in the nonreturn to zero (NRZI) mode at a density of 800 frames per inch. When this feature is included, the recording mode and density are controlled through the program instructions.

In the 7-track recording level, data is recorded in 7-bit frames (NRZI only) across the width of the tape. In this case, each frame contains six data bits plus one parity bit. Optionally the control automatically converts an 8-bit code of the processor (EBCDIC) to or from 6-bit (BCD) when 7-track tape units are used. A special data convert feature is provided to read and write three 8-bit bytes of information in four 6-bit frames. Either method of reading and writing can be selected under program control.

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

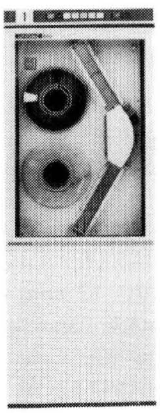
Writing can take place only when the tape is moving forward. The number of records to be included in each record block is established through directives to the operating system. A minimum block size of 18 bytes is imposed when recording in 7- or 9-track NRZI (compatible) mode to allow automatic detection of noise blocks.

Optional features are available to produce concurrent execution of multiple tape functions in addition to rewind. These simultaneous operation features must be added to the control unit, to each UNISERVO 12 Master Magnetic Tape Unit, and to each UNISERVO 16 Magnetic Tape Unit. When these features are included, the subsystem must be connected to the processor through two selector channels. In this configuration, a simultaneous read/read, read/write, and write/write capability is provided for UNISERVO 16 Magnetic Tape Units. Also, simultaneous read/read and read/write capabilities are provided for UNISERVO 12 Magnetic Tape Units; simultaneous write/write capability on two UNISERVO 12 Magnetic Tape Units is available through separate master units.

UNISERVO 12/16 Magnetic Tape Control Units can be accessed by two selector channels by means of the dual channel feature. Therefore, a second processor can be interfaced with the same bank of magnetic tape units. In this configuration, auxiliary storage is provided for two independent UNIVAC Series 90 Systems. However, the magnetic tape subsystem can be accessed by only one processor at a time.

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.

4.3.2.1. UNISERVO 12 Magnetic Tape Unit

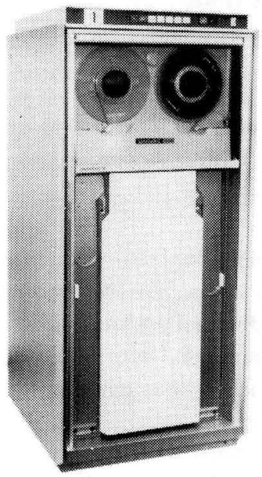


The UNISERVO 12 Magnetic Tape Unit is a low cost tape handler with moderate speed. The master/slave concept is employed in the logic of these units. The master unit, having the power supply and control circuitry, governs up to three additional slave units. There are no differences in programming for 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 9-track and 7-track models. The 9-track tape units read and write data in the phase encoded mode at a density of 1600 bytes per inch. If the optional dual density feature is added, the 9-track units can also write at 800 bytes per inch density. The 7-track UNISERVO 12 Magnetic Tape Units read and write at 800 bytes per inch density. The 7-track UNISERVO 12 Magnetic Tape Units read and write in NRZI mode only. These units can be programmed to read and write data at densities of 200, 556, or 800 frames per inch.

The physical tape speed is 42.7 inches per second, giving 9-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 7-track tape unit transfer data in NRZI mode at 34,160, 23,740, or 8540 characters per second depending upon the density selected.

4.3.2.2. UNISERVO 16 Magnetic Tape Unit



The UNISERVO 16 Magnetic Tape Unit is a higher performance tape unit and can be added to the subsystem individually to a maximum of 16. UNISERVO 16 Magnetic Tape Units are available in 9-track and 7-track models.

The UNISERVO 16 Magnetic Tape Unit reads and writes data in phase encoded mode at 1600 bytes per inch. If the optional dual density feature is added, the 9-track units can also write data in NRZI mode at 800 bytes per inch. The 7-track model reads and records data in NRZI mode only 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 9-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 has a data transfer rate of 96,000, 66,720, or 24,000 characters per second, depending on the density selected.

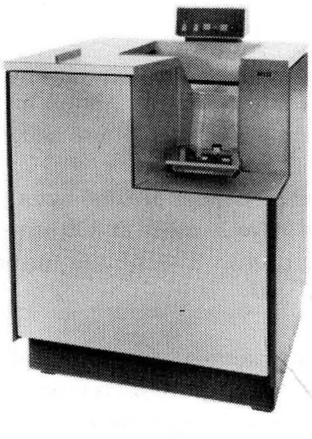
4.3.3. UNISERVO 20 Magnetic Tape Subsystem

The UNISERVO 20 Magnetic Tape Subsystem represents the highest performance tape handling capability offered by Sperry Univac. This subsystem comprises a control unit and from one to sixteen magnetic tape units. A second control unit may be used to achieve simultaneous dual access. The subsystem is connected to the processor by means of a selector channel. The UNISERVO 20 control unit may be configured with mixed tape units; that is, a mixture of UNISERVO 12, 16, and 20 Magnetic Tape Units may be configured with the UNISERVO 20 control unit, thereby providing tape handling flexibility by combining high-speed units with economic medium-speed units.

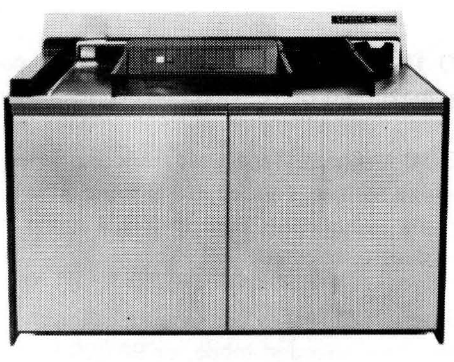
The UNISERVO 20 Magnetic Tape Unit reads and writes data in phase encoded mode at 1600 bytes per inch. Physical tape speed is 200 inches per second, giving the 9-track tape unit a maximum transfer rate of 320,000 bytes per second. The UNISERVO 20 Magnetic Tape Unit provides operational features such as power window, automatic tape threading, and a wrap-around tape cartridge.

The simultaneous dual access configuration provides for read/write, read/read, and write/write operation on any two individual UNISERVO 16 or 20 Magnetic Tape Units. In addition to doubling the performance of the subsystem, complete redundancy is achieved by virtue of individual power supplies for each control unit and independent access paths to each UNISERVO 16 or 20 Magnetic Tape Unit. Data validity-checking facilities include longitudinal redundancy check, vertical redundancy check, and cyclic redundancy check for 9-track NRZI tapes.

4.4. CARD READER SUBSYSTEMS



UNIVAC 0711 Card Reader Subsystem



UNIVAC 0716 Card Reader Subsystem

UNIVAC 0711/0716 CARD READER SUBSYSTEM CHARACTERISTICS			
CHARACTERISTIC	UNIVAC 0711	UNIVAC 0716	
		0716–00	0716–02
CARD READING SPEED	600 cards/minute	600 cards/ minute	1000 cards/ minute
INPUT HOPPER CAPACITY	1200 cards	2400 cards	
OUTPUT STACKER CAPACITY	1500 cards	2 stackers–2000 cards each	
READ MODES	Image mode–160 6-bit characters/card EBCDIC–80 characters card	Image mode–160 6-bit characters/card Translate mode: EBCDIC–80 characters/ card ASCII–80 characters/ card compressed code–80 characters/card	
OPTIONAL FEATURES	51- or 66-column short card feeds	51- or 66 column short card feeds Validity check Alternate stacker fill Dual translate Speed upgrade (0716-00 only)	

The UNIVAC 0711 or 0716 Card Reader Subsystem 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 90/70 System by means of the multiplexer channel. A separate multiplexer channel physical connection and standard subchannel address are required for each card reader subsystem. Either reader can have a 51- or a 66-column short card optional feature.

4.4.1. UNIVAC 0711 Card Reader Subsystem

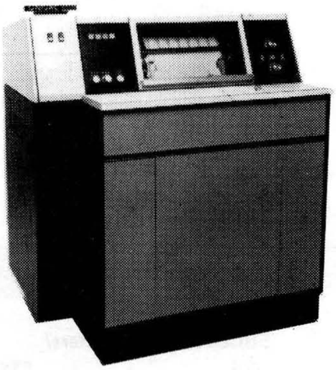
The UNIVAC 0711 Card Reader Subsystem operates at a rate of 600 cards per minute on a column-by-column basis. The read check feature is standard to ensure correct input. Information read from the card is transferred to the processor in either image mode or EBCDIC mode.

4.4.2. UNIVAC 0716 Card Reader Subsystem

The UNIVAC 0716 Card Reader Subsystem operate at a rate of 600 or 1000 cards per minute on a column-by-column basis. The standard read check feature ensures correct input. Information read from the card is transferred to the processor in image mode or translate mode, which includes EBCDIC, ASCII, or compressed code. Image mode and a selection of any one of the translate modes are standard features. The optional dual translate feature permits an additional selection from the two remaining translate mode choices. A speed upgrade feature permits the 600 card-per-minute reader to operate at 1000 cards per minute.

Two output stackers provide the means for error selection as a standard feature in addition to the capability of stopping on error. An optional feature, alternate stacker fill, provides the capability of stacking 4000 cards: when stacker A is filled, the reader automatically begins to fill stacker B. The stop-on-error feature may be used with alternate stacker fill. A unique carousel wheel decelerates and stacks the cards.

4.5. PRINTER SUBSYSTEMS



UNIVAC 0768 Printer Subsystem

UNIVAC 0770 Printer Subsystem

UNIVAC 0768 PRINTER SUBSYSTEM CHARACTERISTICS

CHARACTERISTIC	UNIVAC 0768–00/01	UNIVAC 0768–02/03
PRINTING SPEED (SINGLE LINE SPACING)	900 to 1100 lines/minute depending upon character contingencies: 900 lpm – 63 contiguous characters 1100 lom – 49 contiguous characters 1200 to 1600 lines/minute (Feature F1071–00 installed) depending upon character contingencies: 1200 lpm – 63 contiguous characters 1600 lpm – 43 contiguous characters	840 lpm – 94 contiguous characters 1000 lpm – 87 contiguous characters 2000 lpm – repeated subset of 14 contiguous characters
NUMBER OF CHARACTERS	Maximum of 63 characters consisting of alphabetic characters (A–Z), numeric characters (0–9), 27 punctuation marks and symbols	Maximum of 94 characters consisting of alphabetic characters (A–Z and a–z), numeric characters (0–9), 32 punctuation marks and symbols
PRINT POSITIONS PER LINE	132 print positions (including spaces)	
HORIZONTAL PRINT SPACING	10 print positions/inch	
VERTICAL PRINT SPACING	6 or 8 lines per inch as determined by form control tape	
FORM ADVANCE RATE	33 ips at 6 lines/inch spacing 22 ips at 8 lines/inch spacing	
FORM WIDTH	4 to 22 inches	
FORM LENGTH	1 to 22 inches	
NUMBER OF FORM COPIES	Up to six-part continuously sprocketed forms	
FORM ADVANCE	Loop control	
LINE ADVANCE	Single, double, or triple spacing under program control	

The UNIVAC 0768 Printer Subsystem is a freestanding self-contained unit. The controlling and synchronizing circuitry, including the 132-character print buffer and the print mechanisms, is housed within the cabinet. This complete printer subsystem is connected to the UNIVAC 90/70 System by means of the multiplexer channel. A separate multiplexer channel physical connection and shared subchannel address are required for each printer subsystem.

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.

4.5.1. UNIVAC 0768–00/01 Printer Subsystem

The UNIVAC 0768–00/01 Printer Subsystem is a drum printer that prints a maximum of 1100 lines per minute depending on the number of characters used. The full character set comprises 63 printable characters on a drum 3 inches in diameter. If all of the characters to be printed on a line are contained within a 49-contiguous-character subset on the drum, and single spacing is specified, printing is at 1100 lines per minute. If more than 49 contiguous characters are specified or if spacing other than single spacing is desired, printing speed decreases accordingly. A speed-up feature (F1071–00) is available which increases the line printing rate to a maximum of 1600 lines per minute. A 1600 lpm rate is obtained by using a single-spaced 43-contiguous-character subset. If more than 43 contiguous characters are specified and/or if spacing other than single spacing is desired, printing speed decreases accordingly.

4.5.2. UNIVAC 0768–02/03 Printer Subsystem

The UNIVAC 0768–02/03 Printer Subsystem is a drum printer that prints a maximum of 840, 1000, or 2000 lines per minute depending on the number of characters used. The full character set comprises 94 printable characters (including upper- and lowercase letters) on a drum 5 inches in diameter. Printing occurs at 840 lines per minute when 94 contiguous characters are used. Printing occurs at 1000 lines per minute when 87 contiguous characters are used. Printing occurs at 2000 lines per minute when a numeric set of 14 characters is used.

UNIVAC 0770 PRINTER SUBSYSTEM CHARACTERISTICS

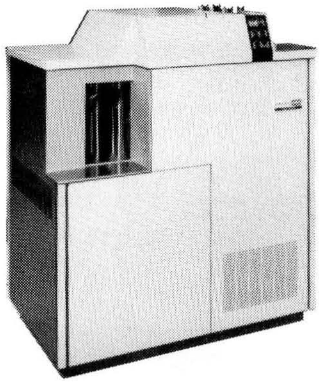
Characteristic	UNIVAC 0770–00/01	UNIVAC 0770–02/03	UNIVAC 0770–04/05
PRINTING RATE (SINGLE LINE SPACING)	1435 lpm for 24-character set 800 lpm for 48-character set 620 lpm for 64-character set 426 lpm for 96-character set	2320 lpm for 24-character set 1400 lpm for 48-character set 1111 lpm for 64-character set 780 lpm for 96-character set	3000 lpm for 24-character set 2000 lpm for 48-character set 1620 lpm for 64-character set 1170 lpm for 96-character set
FORMS ADVANCE RATE	After a continuous forms advance of 10 spaces, forms advance rate is approximately 50 inches (127,0 cm) per second.	After a continuous forms advance of 10 spaces, forms advance rate is approximately 75 inches (190,5 cm) per second.	After a continuous forms advance of 10 spaces, forms advance rate is approximately 100 inches (254,0 cm) per second.
HORIZONTAL PRINT SPACING	10 print positions per inch (2,54 mm/print position)	10 print positions per inch (2,54 mm/print position)	10 print positions per inch (2,54 mm/print position)
VERTICAL LINE SPACING	Either 6 lpi or 8 lpi, as determined by program (6 lpi = 4,23 mm/line; 8 lpi = 3,17 mm/line)	Either 6 lpi or 8 lpi, as determined by program (6 lpi = 4,23 mm/line)	Either 6 lpi or 8 lpi, as determined by program (6 lpi = 4,23 mm/line; 8 lpi = 3,17 mm/line)
NUMBER OF CHARACTERS RIBBON FEED	Standard 48-character set; any number of characters up to Bidirectional, self-reversing, self-correcting; ribbon removable without rewinding	Standard 48-character set; any number of characters up to Bidirectional, self-reversing self-correcting; ribbon removable without rewinding	Standard 48-character set; any number of characters up to Bidirectional, self-reversing self-correcting; ribbon removable without rewinding
RIBBON	Fabric and plastic film ribbons	Fabric and plastic film ribbons	Fabric and plastic film ribbons
CODES	EBCDIC, ASCII or any 6-, 7-, or 8-bit code	EBCDIC, ASCII or any 6-, 7-, or 8-bit code	EBCDIC, ASCII or any 6-, 7-, or 8-bit code
FORMS	Continuous single-part and multipart with standard edge sprocket holes, form widths from 3.5 inches (88,9 mm) to 22 inches (558,8 mm). Form depths (lengths) up to 24 inches (609,6 mm) maximum. Form depths up to 19 inches (482,6 mm) can be contained within printer casework. See *UNIVAC 0770 Printer Subsystem Operator Reference, UP–7938* (current version).	Continuous single-part and multipart with standard edge sprocket holes, form widths from 3.5 inches (88,9 mm) to 22 inches (558,8 mm). Form depths (lengths) up to 24 inches (609,6 mm) maximum. Form depths up to 19 inches (482,6 mm) can be contained within printer casework. See *UNIVAC 0770 Printer Subsystem Operator Reference, UP–7938* (current version).	Continuous single-part and multipart with standard edge sprocket holes, form widths from 3.5 inches (88,9 mm) to 22 inches (558,8 mm). Form depths (lengths) up to 24 inches (609,6 mm) maximum. Form depths up to 19 inches (482,6 mm) can be contained within printer casework. See *UNIVAC 0770 Printer Subsystem Operator Reference, UP–7938* (current version).
PRINT POSITIONS	132 print positions per line 160 print positions with factory F1533–00.	132 print positions per line 160 print positions with factory F1533–00.	132 print positions per line 160 print positions with factory F1533–00.

4.5.3. UNIVAC 0770 Printer Subsystem

The UNIVAC 0770 Printer Subsystem (printer) is a freestanding unit containing its own power supply and control logic. The printer prints alphanumeric data and responds to a 21-command repertoire to permit a versatile combination of functions. Line printing is accomplished with a replaceable print band.

The printer is capable of printing 132- or 160-character (columns) lines, at rates to 3000 lines per minute (lpm). Print rates of 800, 1400, or 2000 lpm are for a standard 48-character set, and rates of 1435, 2320, or 3000 lpm are for a 24-character set. A print line of 132 characters is on 0.1 inch (2,54 mm) centers or 10 characters per inch (25,4 mm). Vertical line spacing is at 6.0 (4,23 mm/line) or 8.0 (3,17 mm/line) lines per inch. Forms are advanced at a rate of 8.75 milliseconds per line for single line spacing.

4.6. UNIVAC 0604 CARD PUNCH SUBSYSTEM



UNIVAC 0604 CARD PUNCH SUBSYSTEM CHARACTERISTICS	
CARD PUNCHING SPEED	250 cpm (maximum)
INPUT HOPPER CAPACITY	1000 cards
OUTPUT STACKER CAPACITY NORMAL STACKER SELECT STACKER	 1000 cards 1000 cards
READ STATION SENSING	Row by row
PUNCH STATION PUNCHING	Row by row
OPTIONAL FEATURE	Read before punching
READ RATE (OPTIONAL)	250 cpm
PUNCH TRANSLATION IMAGE MODE COMPRESS MODE	 160 six-bit characters per card 80 characters per card

The UNIVAC 0604 Card Punch Subsystem is a freestanding self-contained unit. It feeds and punches 80-column cards, row by row, at a maximum rate of 250 cpm.

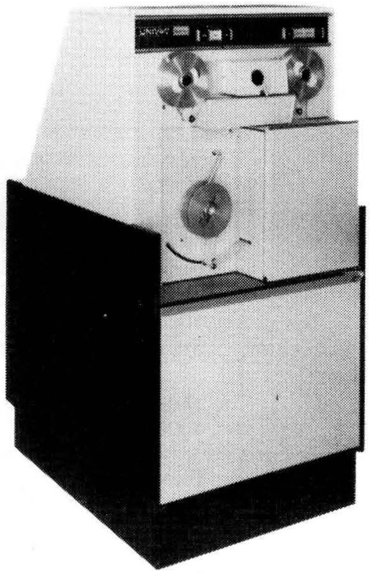
The card punch subsystem is connected to the UNIVAC 90/70 System by means of the multiplexer channel. A separate multiplexer channel physical connection and shared subchannel address are required for each card punch subsystem.

Punch installations having 4-card punch subsystems are restricted to physical connections within 100 feet of the processor cabinet, as measured along the cable distribution. More than four punches are not permitted.

The UNIVAC 0604 Card Punch Subsystem includes a control unit and synchronizer which regulates the flow of data and control signals to and from the punch mechanism. Data to be punched is transferred from the multiplexer channel to the punch buffer in the control unit which regulates the flow of data to and from the punch mechanism. Data received by the control unit is checked for validity, then sent in either the image mode or the compress mode of translation to the punch mechanism. To ensure that the card is punched correctly, the control unit senses the data punched on the card and compares it with the data initially sent to the punch mechanism. The punched cards are then directed by program control into one of the two output card stackers.

Feature 0875–00 (optional) is a prepunch read unit that reads data from a prepunched card, row by row, into the control unit. Additional data can then be punched into the card when the command is recieved and transferred to the control unit for a comparison check.

4.7. UNIVAC 0920 PAPER TAPE SUBSYSTEM



UNIVAC 0920 PAPER TAPE SUBSYSTEM CHARACTERISTICS

PAPER TAPE READER

READER MOUNTING	Mounted on a panel (7 inches high and 19 inches wide) with a pin spindle for handling reels containing up to 50 feet of tape (for tape reader without optional spooler).
TAPE READ	Unidirectional (right to left)
TAPE CHANNEL CAPACITY	Capable of reading 11/16-inch, 7/8-inch, or 1-inch paper tape. Three-position tape guide available to adjust to tape width used.
READ SPEED	300 characters/second at 10 characters/inch
TYPE OF TAPE	All conventional perforated tapes with a light transmissivity of 40% or less.
STOP AND START CAPABILITIES	Can stop on character or before next character. On START, unit reaches full speed within two characters.
TAPE SPOOLER	Up to five-inch reels can be used with the spooler to allow reeling of approximately 300 feet of paper tape.
TAPE LEADER	Approximately 3 feet of tape leader when the spooler mechanism is used.
TAPE TRAILER	A 12-inch trailer is needed to prevent false broken tape indication.

PAPER TAPE PUNCH

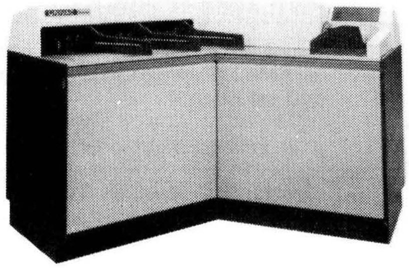
PUNCH MOUNTING	Mounted within a panel 14 inches high and 19 inches wide.
TAPE CHANNEL CAPACITY	Handles paper tape widths of 11/16-inch or 1-inch. Five levels of tape characters when using 11/16-inch paper tape or five, six, seven, or eight levels of tape characters when using 1-inch paper tape. Tape guide adjusts to conform to the paper tape width used.
PUNCH SPEED	110 characters/second at 10 characters/inch
TYPE OF TAPE	Oil base paper tape is recommended. A compatible tape utilizing a paper-MYLAR*-paper sandwich is available under UNIVAC part number 4956476. This is the only MYLAR tape authorized for use with this equipment.
STOP AND START CAPABILITIES	Punching is performed one character at a time. The tape punch is capable of stopping and starting between characters.
TAPE FEEDING	The tape punch handles a 1000-foot roll of paper tape. A tape supply sensing switch signals the control unit when the tape supply is low.

**DuPont trademark for its polyester film*

The UNIVAC 0920 Paper Tape Subsystem is a peripheral data processing device possessing the capability of punching data on paper tape and of reading data from punched paper tape. This subsystem, which is housed in a freestanding cabinet, reads perforated tape having five, six, seven, or eight channels at a rate of 300 characters per second and punches paper tape at a rate of 110 characters per second. The paper tape subsystem is connected to the UNIVAC 90/70 System by means of the multiplexer channel. A multiplexer channel physical connection and shared subchannel address are required for each paper tape subsystem.

The paper tape subsystem consists of a control unit, a tape reader and reader synchronizer, and/or a tape punch and punch synchronizer. The control unit provides the necessary synchronization and interface between the reader and punch synchronizers and the multiplexer channel. The synchronizer units regulate the transfer of data characters between the tape reader or tape punch and the control unit. The transfer rate is determined by the mechanical speed of the reader mechanism or punch mechanism. Tape parity checking or generation and data bit and character manipulations are performed within the synchronizers in accordance with the wiring of the program connector. Spooling features are available for both the tape reader and tape punch.

4.8. UNIVAC 2703 OPTICAL DOCUMENT READER



UNIVAC 2703 ODR CHARACTERISTICS	
DOCUMENT FEED RATE	300 documents/minute for OCR documents 6 inches in length
DOCUMENT SIZES AND WEIGHTS	Height – 2.75 to 4.25 inches Length – 3 to 8.75 inches Paper weights – 20- to 62-pound paper, based on a ream (500 sheets) of 17- by 22-inch paper
HOPPER	One provided – documents may be loaded while reader is operating
OPTICAL CHARACTER RECOGNITION	Reads a single printed line of numeric data and special symbols
FONT RECOGNITION	UNIVAC H–14 or numeric subset from ANSI X3.17–1966 at customer selection; standard for 0768 printing
STACKERS	Three provided – stacker selection under program and hardware control
FEATURES	600-document-per-minute speed upgrade Modulo 10 check digit verification Mark read Punch card read Validity check

The UNIVAC 2703 Optical Document Reader (ODR) provides UNIVAC 90/70 System users with an additional and unique input medium. It is unique in that it represents a major advancement in the handling of optically read documents. The ODR optically reads printed numeric data and manually inscribed marks on a variety of document sizes.

The ODR is connected to the UNIVAC 90/70 System processor by means of the multiplexer channel. A separate multiplexer channel physical connection and shared subchannel address are required for each ODR.

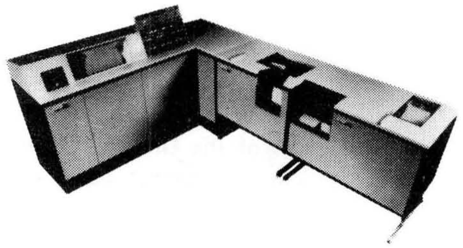
The ODR is capable of reading printed numeric and mark encoded data from documents ranging in size from 2.75 by 3.00 inches to 4.25 by 8.75 inches. The basic document reading speed is 300 documents per minute for documents six inches in length and up to 350 documents per minute for documents three inches in length. The combination of a dual-belt document feed, a solid state photoelectric sensing device, and carousel stackers provides for gentle document handling.

The UNIVAC H–14 font is used with the optical document reader. However, the American National Standard Character Set for Optical Character Recognition (OCR-A) size A is offered as an optional feature (for actual font sizes see American National Standard for Optical Character Recognition X3.17–1966). Other optional features offered with the ODR include a 600-document-per-minute speed upgrade, modulo 10 check digit verification, mark read, punch card read, and validity check.

4.9. UNISCOPE 100 DISPLAY TERMINAL

The UNISCOPE 100 Display Terminal can be easily utilized as an onsite peripheral device. Up to 31 UNISCOPE 100 Display Terminals may be connected to a UNIVAC 90/70 System multiplexer channel by means of the UNISCOPE 100 Display Terminal Multiplexer feature. The functional characteristics are given in 5.3.2.

4.10. UNIVAC 9200/9200 II/9300/9300 II/9400 SYSTEMS



UNIVAC 9200/9300 SYSTEM

UNIVAC 9400 SYSTEM

UNIVAC 9200/9200 II/9300/9300 II/9400 SYSTEMS CHARACTERISTICS

CHARACTERISTIC	UNIVAC 9200	UNIVAC 9200 II	UNIVAC 9300	UNIVAC 9300 II	UNIVAC 9400
SYSTEM ORIENTATION	Card/disc	Card/tape/disc	Card/tape/disc	Card/tape/disc	Tape/disc
BASIC MAIN STORAGE	8192 bytes	8192 bytes	8192 bytes	16,384 bytes	32,768 bytes
MAXIMUM MAIN STORAGE	16,384 bytes	32,768 bytes	32,768 bytes	32,768 bytes	262,144 bytes
MAIN STORAGE CYCLE TIME	1200 nanoseconds/ byte	1200 nanoseconds/ byte	600 nanoseconds/ byte	600 nanoseconds/ byte	600 nanoseconds/ 2 bytes
ADD (DECIMAL) INSTRUCTION TIME (TWO 5-DIGIT NUMBERS)	104 ms	104 ms	52 ms	52 ms	22.2 ms
MULTIPLY, DIVIDE, AND EDIT CAPABILITY	Optional	Optional	Standard	Standard	Standard
CARD READER					
BASIC READER	400 cpm	400/600 cpm	600 cpm	600 cpm	600 cpm
UNIVAC 1001 CARD CONTROLLER	1000/2000 cpm	1000/2000 cpm	1000/2000 cpm	1000/2000 cpm	Not available
UNIVAC 0716 CARD READER	1000 cpm	1000 cpm	1000 cpm	1000 cpm	1000 cpm
CARD PUNCH					
COLUMN	75–200 cpm	75–200 cpm	75–200 cpm	75–200 cpm	Not available
ROW	Not available	200 cpm 250 cpm	200 cpm 250 cpm	200 cpm 250 cpm	250 cpm
	Read/punch optional	Read/punch optional	Read/punch optional	Read/punch optional	Read/punch optional
AVAILABLE PRINT SPEEDS					
BAR PRINTER	250-500 lpm	250-500 lpm	600-1200 lpm	600-1200 lpm	
UNIVAC 0768 PRINTER		840-2000 lpm	840-2000 lpm	840-2000 lpm	840-2000 lpm
PAPER TAPE					
READ	300 cps	300 cps	300 cps	300 cps	300 cps
PUNCH	110 cps	110 cps	110 cps	110 cps	110 cps
MULTIPLEXER CHANNEL TRANSFER RATE	85,000 bytes/ second	85,000 bytes/ second	85,000 bytes/ second	85,000 bytes/ second	85,000 bytes/ second
SELECTOR CHANNEL TRANSFER RATE	Not applicable	350,000 bytes/ second	Not applicable	350,000 bytes/ second	330,000 bytes/ second
DATA COMMUNICATIONS SUBSYSTEM	Up to 8 duplex lines	Up to 8 duplex lines	Up to 8 duplex lines	Up to 8 duplex lines	Up to 64 duplex lines
REGISTERS	8 for processor functions 8 for I/O functions	8 for processor functions 8 for I/O functions	8 for processor functions 8 for I/O functions	8 for processor functions 8 for I/O functions	16 for supervisor functions 16 for user program functions

The UNIVAC 9200, 9200 II, 9300, or 9300 II System as a subsystem is a standard version of the UNIVAC 9200, 9200 II, 9300, or 9300 II System, respectively. It is connected to the UNIVAC 90/70 System by means of a channel adapter attached to the multiplexer channel. A UNIVAC 9000 Series System operating in an online mode serves as a slave reading, printing, or punching facility for the UNIVAC 90/70 System.

The UNIVAC 9400 System is a freestanding system which may be operated offline or online to the UNIVAC 90/70 System and which may include all peripheral units and hardware features available to the original system configuration.

When operating in the online mode, communication between the two processors through the channel adapter is under control of user programs residing in both systems. The supervisor of the UNIVAC 90/70 System acts as a master to coordinate all activity between the systems through the channel adapter.

For a description of the UNIVAC 9200, 9200 II, 9300, or 9300 II System, see *UNIVAC 9200/9200 II/9300/9300 II Systems System Description, UP–7806* (current version). For a description of the UNIVAC 9400 System, see *UNIVAC 9400 System System Description, UP–7566* (current version), and *UNIVAC 9480 System System Description, UP–8000* (current version).

4.11. UNIVAC 90/60 and 90/70 SYSTEMS

A UNIVAC 90/60 System or a UNIVAC 90/70 System may be connected to the UNIVAC 90/70 System by means of a UNIVAC 9000 Series channel adapter. Communications between the two processors are under control of the user programs residing in both systems. The supervisor of each system is used to coordinate all activity between the systems.

A UNIVAC 90/60 System or a UNIVAC 90/70 System may be connected directly to the UNIVAC 90/70 System by means of the direct control feature. To facilitate interprocessor data transfer, the direct control option offers two instructions. The read direct and write direct instructions are used to indicate solicited and unsolicited data transfers between the two systems.

5. COMMUNICATIONS

5.1. GENERAL

The UNIVAC 90/70 System offers flexibility in communications-oriented applications. A wide variety of remote devices is available for use with this system. Connection to the processor is accomplished through either a Data Communications subsystem (DCS), the multiplexer channel, or the Communication Intelligence Channel (CIC), which is especially designed to enhance communications line handling capabilities.

5.2. DATA COMMUNICATIONS SUBSYSTEM (DCS)

Limited communications requirements can be satisfied by the use of the DCS. The DCS provides the following capabilities:

- DCS-1 or DCS-1C services one line.

- DCS-4 services up to four lines.

- DCS-16 services up to 14 lines.

Each DCS uses one of the eight physical connections of the multiplexer channel and two DCS subchannel addresses for each line served.

A DCS-1 consists of a controller, a line terminal, an associated communications interface, and an asynchronous timing assembly if required.

A DCS-1C is identical to a DCS-1 except that the DCS-1C operates under binary synchronous communications (BSC) procedures.

The DCS-4 consists of a controller, up to four line terminals, associated communications interfaces, asynchronous timing assemblies, and a dialing adapter.

The DCS-16 consists of a controller, up to 14 line terminals (system allowable), associated communications interfaces, asynchronous timing assemblies, dialing adapters, and dual channel access.

For characteristics and possible configurations of the DCS see *UNIVAC 9000 Series Data Communications Subsystem (DCS) Programmer/Operator Reference Manual, UP–7613* (current version).

5.2.1. Line Terminal Controller

The line terminal controller provides control for the various line terminals and the automatic dialing adapter. The line terminal controller:

- coordinates and multiplexes data flow;
- monitors error conditions;
- interrupts subsystem commands;
- presents status and interrupts.

5.2.2. Line Terminal

One to 30 line terminals can be used in half- or full-duplex mode as data handlers for either sending or receiving information to or from the processor. Several types of line terminals are available to provide low and medium speed asynchronous, binary synchronous, or synchronous communication. Data characters may range from four to ten bits depending on the model and mode of the line terminal used.

5.2.3. Communications Interface

The communications interface is used to connect a line terminal with a communications line. The communications line may be telegraph wires, a modem, or a direct cable to a device. The available communications interfaces meet the EIA RS 232R (Industry Standard Interface), the MIL-STD-188B (Electrical Circuit Compatibility-Government) specifications, or CCITT. Each input/output line requires one communications interface.

5.2.4. Asynchronous Timing Assembly

The asynchronous timing assembly (ATA) is a clock source for asynchronous output line terminals. A single unit provides one speed for an entire DCS. Asynchronous timing assemblies are available in speeds up to 2400 bits per second. Each different speed asynchronous line terminal requires an asynchronous timing assembly. The maximum number of ATAs is eight in the case of the DCS-16.

5.3. COMMUNICATIONS INTELLIGENCE CHANNEL (CIC)

The Communication Intelligence Channel (CIC) is a sophisticated subsystem that interfaces the UNIVAC 90/70 to a wide assortment of remote terminal devices via commonly used communications lines. The CIC extends the 90/70 communication facilities beyond those offered by the Data Communication Subsystem (DCS) in terms of functional capability and the number of lines controlled.

The CIC is software supported by OS/7.

5.3.1. Benefits

In addition to offering advanced capabilities, the use of intelligence within a channel design offers a lower priced performance than that offered by most front-end processors (FEP). Flexibility offered in programmable procedural parameters, buffer chaining, and command chaining, provides FEP-type features without the cost incurred by redundant buffer storage and channel-to-channel transfer overhead. Among the benefits of the CIC are:

- dedicated communications channel;
- large communications network capability at low cost;
- reduction of much of the communications processing burden for the 90/70;
- complete redundancy available; and
- advanced communication capability, for example, automatic data rate detection and selection.

The CIC is designed to be a separately housed and powered unit that accommodates up to 64 full-duplex or 128 half-duplex lines. The transfer rate per line is up to 56,000 bits per second. The gross data transfer rate of the CIC is 60,000 bytes per second. The CIC is a micro-programmed channel that performs many of the functions of a front-end processor at a fraction of front-end processor prices. It independently references and controls a pool of buffers in 90/70 main storage instead of buffering messages in its own memory as would a front-end processor. The CIC is a dedicated channel having its own data path to storage, thereby eliminating the multiplexor contention and overhead previously found in a communications environment.

5.3.2. Capability

The CIC is engineered to acquire input buffers as required, store or retrieve data, chain buffers, maintain a count of available buffers, monitor that count to maintain a predetermined minimum number of buffers, and interact with the host processor only in the event of a completed task or fault condition. Other tasks performed by the CIC independent of the 90/70 processor include:

- automatic polling — in a sequence and frequency specified by the host processor;
- automatic dialing;
- embedding of polls or acknowledgments;
- generating and checking character and message parity (LRC or CRC);
- translation of characters;
- tables status in 90/70 storage;
- detection of illegal character sequences in output messages;
- chaining input buffers;
- acknowledging input messages;
- processing all status and nontext messages;
- recognizing control characters and control character sequences;

- accepting acknowledgments of output messages;
- automatic retransmission of negatively acknowledged or nonacknowledged output messages;
- automatic answering of incoming calls and interrupting of the processor;
- automatic disconnecting;
- automatic detection and adjusting to variable input data rate; and
- performance of timer functions for data messages, status messages, and acknowledgments.

The standard CIC hardware offers compatibility with the Direct Distant Dialing (DDD) network, the Wide-Area Telephone (WATS) network, the Teletypewriter Exchange (TWX) network, the wideband network, private line facilities, and equivalent international facilities. It offers a high level of flexibility allowing adaptation to all popular transmission procedures and data exchange techniques.

Standard CIC micrologic enables communications with all current UNIVAC terminals, TELETYPE*-compatible devices, and selected IBM terminals which conform to the binary synchronous (BSC) transmission conventions. Other terminal types are easily accommodated via program-supplied procedural parameters. Translate tables, message control codes, and error checking techniques are equally easy to specify.

5.3.3. Redundancy

Hardware redundancy can be as extensive as desired. Two CICs may share all line adapters through the use of the adapter sharing feature. This could provide redundancy for up to 128 lines.

Redundant host systems can be accommodated with the inclusion of the dual-channel interface expansion feature. This feature allows the CIC, while in control of an entire network of lines and devices, to be switched to the second or backup processor. Each level of redundancy is provided on a mutually exclusive basis; i.e., a line adapter is logically connected to one CIC or the other and the CIC is logically connected to one processor or the other.

5.4. REMOTE TERMINALS

A wide selection of devices can be chosen in a communications-oriented configuration. Key entry devices and batch processors are designed for communications applications. Some of these devices and subsystems are:

- TELETYPE*
- UNISCOPE 100 Display Terminal
- UNIVAC DCT 500 Data Communications Terminal
- UNIVAC DCT 1000 Data Communications Terminal

**Trademark of Teletype Corporation*

- UNIVAC DCT 2000 Data Communications Terminal
- UNIVAC 1004 or 1005 Card Processor System with appropriate communications terminal interface
- UNIVAC 9000 Series Systems
- IBM 2780 Binary synchronous Communications

5.4.1. TELETYPE EQUIPMENT

The UNIVAC 90/70 Systems software includes handlers that support TELETYPE modesl 28, 32, 33, 35, and 37.

5.4.2. UNISCOPE 100 Display Terminal

UNISCOPE 100 DISPLAY TERMINAL CHARACTERISTICS	
NUMBER OF TERMINALS PER STATION	1–31
DISPLAY CAPACITY	960 or 1024 characters
DISPLAY FORMAT	64 characters/line x 16 lines 80 characters/line x 12 lines
DISPLAY CHARACTER SET	64 standard character set expandable to 96 symbols
KEYBOARD	Numeric, alphanumeric, or combination of numeric and alphanumeric. 8 cursor control keys 5 editing keys For a complete description of the keyboard, see *UNIVAC UNISCOPE 100 Display Terminal General Description, UP-7701* (current version).
CONTROLLER STORAGE CAPACITY	Variable up to 9600 bits per second

The UNISCOPE 100 Display Terminal is a 2-way remote terminal device that makes it possible to hold direct data communications with the processor. Each terminal is keyboard operated and has a visual display screen for message display. The keyboard includes data keys, cursor control, and editing keys. The visual display screen displays processor output messages; it also allows input messages to be composed and edited before being transmitted to the processor. Each character entered by an operator is immediately displayed and stored in the control unit storage.

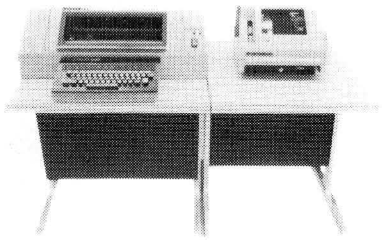
The UNISCOPE 100 Display Terminal provides input/output message buffering, refresh storage, character generation, and control logic. Special interfaces for direct computer connection and hard copy output are available. A variety of presentation formats are offered which provide a total display capacity of 960 or 1024 characters. Due to its modular construction, the UNISCOPE 100 Display Terminal can operate as a data entry or as a display device and can be conveniently located at the central computer site or at a remote station, where it is connected to the system by way of telephone lines. Up to 31 terminals may be connected to a communications line or directly to a DCS my means of the UNISCOPE 100 Terminal Multiplexer feature. This general purpose feature is available for use with any of the communication line interfaces provided on the UNISCOPE 100 Display Terminal, permitting a mixture of single and multiple units on one communications system. The feature also provides broadcasting of output messages to multiple devices.

The UNISCOPE 100 Terminal Multiplexer is used in multistation configurations to reduce:

- line requirements to units within 5000 feet of the terminal multiplexer;
- polling time;
- modem requirements;
- the number of I/O channels necessary when operating with the processor.

Polling time is reduced in multiple or general polls because the multiplexer answers the poll on behalf of all associated units.

5.4.3. UNIVAC DCT 500 Data Communications Terminal



UNIVAC DCT 500 CHARACTERISTICS	
TRANMISSION CODE	8-level ASCII
INTERFACES	EIA standard RS-232/CCITT internal modem
TRANSMISSION MODE	Half-duplex or full-duplex (2- or 4-wire)
TRANSMISSION RATE	110, 150, or 300 bits/second (selectable)
PRINTING RATE	30 characters/second
CHARACTER SET	ASCII, EBCDIC A (Business)/H (Scientific)
PRINTABLE CHARACTERS	63 plus space
PRINT POSITIONS PER LINE	132 (adjustable tractor)
PAPER TAPE READER/PUNCH RATE	50 characters/second

The UNIVAC DCT 500 Data Communications Terminal is a low cost, unbuffered, asynchronous keyboard/printer terminal similar in operation to a teletypewriter, providing up to 132 print positions. The DCT 500 can replace existing teletypewriters with little or no change in the software handlers for point-to-point communications networks over voice-grade telephone toll lines or private lines. In a multiparty polled environment the DCT 500 operates in accordance with ASCII procedures.

The DCT 500 can operate in a receive-only mode, a keyboard send/receive mode, or an automatic send/receive mode. The basic printer system (minimum equipment) can be expanded to include a keyboard and a 1-inch paper tape read/punch unit at any time. Additional optional equipment is available to allow for multistation operation.

The UNIVAC DCT 500 also offers the following optional features:

- Automatic answering
- Master/slave operation
- Print monitor
- Internal modem
- Paper tape

5.4.4. UNIVAC DCT 1000 Data Communications Terminal

The UNIVAC DCT 1000 Data Communications Terminal is a fully buffered 30-characrer-per-second incremental printer which can be expanded to include a keyboard, card reader, card punch, paper tape reader/punch, and an auxiliary printer. The DCT 1000 transmits or receives data from a local or remote processor or to a remote DCT 1000 in a conversational or batch mode.

UNIVAC DCT 1000 CHARACTERISTICS	
CARD READING SPEED	40 cards/minute
CARD PUNCHING SPEED	35 cards/minute
PRINTING SPEED	30 characters/second
PRINTING POSITIONS PER LINE	132 (adjustable tractor)
PRINTABLE CHARACTERS	63 plus space
PAPER TAPE SPEEDS	50 characters/second
BUFFER STORAGE	320 character capacity in two buffers, 160 characters each
TRANSLATOR SELECTIONS	ASCII Code EBCDIC H (Scientific) Code EBCDIC A (Business) Code Binary with additional feature
TRANSMISSION METHOD	Block by block
TRANSMISSION MODE	Half-duplex; 2 or 4 wire (nonsimultaneous; two-way transmission)
TRANSMISSION FACILITIES	Voice-grade telephone toll exchange or private line
TRANSMISSION RATE	Asynchronous 300, 1200, or 1800 bits/second, synchronous up to 9600 bits/second

5.4.4.1. Data Buffers

Two 160-character buffers are standard on the DCT 1000. These buffers facilitate the following:

- Automatic Blocking

  This eliminates complicated and time-consuming operator functions and minimizes training.

- Automatic Error Correction

  This eliminates manual correction protection procedures such as reloading cards and retyping input data.

- Error Free Output

  All messages are completely checked for character errors, block errors, duplicate blocks, or lost blocks. The result is that no errors are entered into the output medium.

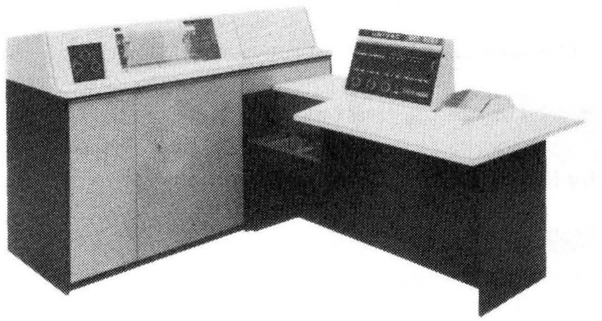
- High Transmission Speeds

  The full capability of the line can be utilized since the transmission rate can be much higher than the I/O rate. On party line systems, this yields data throughput on a line which is the sum of the throughputs of the individual terminals.

5.4.4.2. Polling System

The UNIVAC DCT 1000 has complete polling and address recognition capabilities which allow the processor to completely control up to 31 UNIVAC DCT 1000 terminals on a single line. The terminals may be connected in a series string in different geographical locations or at a single point on the UNIVAC Terminal Multiplexer.

5.4.5. UNIVAC DCT 2000 Data Communications Terminal



UNIVAC DCT 2000 CHARACTERISTICS	
CARD READING SPEED	200 cards/minute
CARD PUNCHING SPEED	75–200 cards/minute
PRINTING SPEED	250 lines/minute
PRINTING POSITIONS PER LINE	80 or 128
PRINTABLE CHARACTERS	63 plus space
BUFFER STORAGE	256 character capacity in two buffers, 128 characters each
TRANSLATION CAPABILITIES	Hollerith to ASCII Hollerith to XS-3 (DLT compatible)
TRANSMISSION METHOD	Block by block
TRANSMISSION FACILITIES	Voice-grade telephony toll exchange or private line
TRANSMISSION RATE	2400 bits per second (private line) 2000 bits per second (switched telephone network)
TRANSMISSION CODE	ASCII XS-3 (DLT compatible)

The UNIVAC DCT 2000 Data Communications Terminal is a combination printer and reader/punch designed to transfer large quantities of data efficiently over voice-grade facilities. This terminal can handle up to 250 blocks per minute. The DCT 2000 is also available without the combination card reader/punch for use as a printer terminal.

Ease of operation and the fact that no programming is required at the terminal location make the DCT 2000 simple to install and operate. Normally available ac power is all that is required and the common carrier can simply make connection to data communications facilities. Either a private line connection at a maximum rate of 2400 bits per second or a dial facility at a maximum rate of 2000 bits per second can be installed according to the user's requirements, since the DCT 2000 and the common carrier equipment both meet the EIA RS 232B standard communications interface for industry.

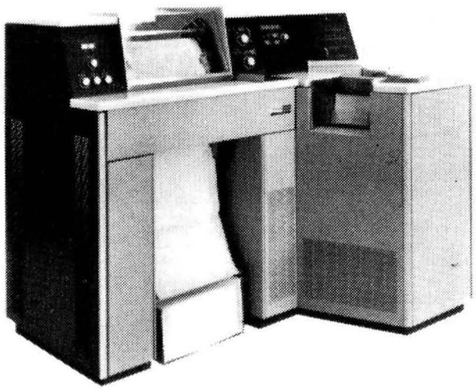
The UNIVAC DCT 2000 consists of a bar printer, a card reader/punch (not needed when used only as a printer terminal), a control unit, and an operator's console; it is designed for:

- Realiability – through the use of the latest monolithic integrated circuits. Monolithic integrated circuits far exceed the reliability of ordinary transistorized circuits; furthermore, they produce less heat, use less power, and are less affected by fluctuating environmental conditions.
- Expandability – through the use of an input/output channel. The input/output channel permits the use of four additional input or output devices; for example, a paper tape punch might be added.
- Flexibility – through the use of 11 field-installable features. (See Table 5–1.)

Table 5–1. UNIVAC DCT 2000 Field-Installable Options

OPTION NAME	DESCRIPTION
Punch Check and Alternate Stacker	Allows a check of the actual punch die movement and diverts incorrectly punched cards to an error stacker while automatically repunching the data into another card.
128 Print Positions	Allows the basic 80 print-position line to be expanded to 128 print positions.
Unattended Operation	Allows data to be transmitted or received with no operator intervention necessary at the DCT 2000.
Transmit/Receive Monitor	Allows data that is being punched or read to be printed simultaneously.
Offline Listing	Allows data to be printed from cards when the DCT 2000 is not transmitting or receiving.
Peripheral I/O Channel	Allows four additional input or output devices to be attached to the DCT 2000.
Short Block Capability	Allows shorter messages to be handled, thereby increasing the throughput and message efficiency. Punching can increase to a maximum rate of 200 cpm.
Select Character Capability	Allows a transmitting DCT 2000 to select the peripheral in the receiving DCT 2000.
Telephone Alert	Allows voice communications between locations over the data facilities by providing signals through which the operators can make connection.
Error Detection and Retransmission	Allows automatic retransmission of a message when a character or message parity error is detected.
Form Control	Allows multiple line spacing and form feed under control of a special character in a message and a paper tape loop.

5.4.6. UNIVAC 1004 or 1005 Subsystem



UNIVAC 1004 OR 1005 SUBSYSTEM CHARACTERISTICS	
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

The UNIVAC 1004 or 1005 Subsystem is a powerful processing unit 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 line 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.

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), paper tape punch (110 cps), and one or two UNISERVO VI-C Magnetic Tape Units.

This subsystem when attached to one of several types of line terminals can function as a remote data processor connected through a communications line to the UNIVAC 90/70 System. Transmission at speeds of 2000, 2400, or 40,800 bits per second is possible depending upon the types of line terminals and communications facility employed.

For detailed descriptions of this subsystem, see *UNIVAC 1005 System General Description, UP–4052* (current version); *UNIVAC 1004 System General Description, UP–3927* (current version); *UNIVAC 1004 Card Processor, 90 Column Reference, UT–2541* (current version); and *UNIVAC 1004 Card Processor, 80 Column Reference, UT–2543* (current version).

5.4.7. UNIVAC 9000 Series Systems

The UNIVAC 90/70 System has the capability to support the UNIVAC 9000 Series Systems as remote subsystems. These systems and their capabilities are discussed in Section 4. For additional information see *UNIVAC 9200/9200 II/9300/9300 II Systems System Description, UP–7806* (current version) or *UNIVAC 9400 System System Description, UP–7566* (current version).

6. OPERATING SYSTEM (OS/7)

6.1. GENERAL

The operating system (OS/7) provided for the UNIVAC 90/70 System has been designed to meet the total computing and operating requirements of advanced data processing problems. The development of the 90/70 System has drawn upon many years of experience in multijobbing 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 operating environment. All of the UNIVAC 90/70 software is proprietary to Sperry Univac and is available for use on Sperry Univac equipment on a lease basis at no extra charge.

OS/7 provides support for a flexible multijobbing environment which may consist of several types of user and system jobs:

- Batch production jobs submitted at the central site or from remote sites.
- Telecommunications jobs requiring fast and efficient response to remote users of the system.
- Online system diagnostics jobs to evaluate the performance of hardware subsystems.
- Program debugging and applications systems checkout.
- Data spooling.

System generation procedures permit the user to tailor the operating system to meet the prerequisites of his hardware configuration and system requirements.

The UNIVAC 90/70 System can process up to 14 jobs concurrently. Each job consists of one or more job steps (programs) which are executed serially. Additionally, a job step may have one or more tasks which are executed concurrently. This structure allows the user a high degree of flexibility in determining use of the system and in scheduling applications programs.

The operating system controls and coordinates the functions within this environment. It presents a simple interface to the programmer which facilitates easy use of the system and alleviates concern for interaction between the coexisting user programs.

The software system includes the industry-standard language processors: RPG, COBOL, FORTRAN, and a macro assembler. The operation of all language processors is controlled by the operating system. The operating system has the responsibility of executing processors as required, providing inputs to the processors, and storing and maintaining their outputs.

OS/7 supports:

- language processors for user program development;
- program diagnostic capabilities for use during the debugging phase of program implementation;
- program manipulation and utility routines for use during the linkage editing and integration stage;
- comprehensive data mangement capabilities for controlling data at the record, block, and file level;
- a flexible, easy-to-use job control language;
- IBM System/360 DOS compatiblity, including IBM 1400 emulation;
- UNIVAC 9400 System compatibility.

OS/7 has the capability to aid the installation management by:

- providing account number and project number checks for each job submitted for processing;
- providing accounting and facility utilization records;
- automatic job scheduling and initiation;
- collecting statistics concerning system performance to assist preventive maintenance and to provide job accounting records.

6.2. SUPERVISOR

The supervisor is that component of OS/7 which provides the centralized control necessary for proper utilization of the system hardware and software complex. The supervisor provides a comprehensive set of services to the user, including:

- Generalized physical input/output control system
- Resource allocation
- Task control
- Timer and day clock services
- Program management
- System console management
- Record and file protection
- Program error handling
- Subroutine linkage table
- Spooling operations
- Operating environment recovery
- Diagnostic and debugging aids

6.2.1. Physical Input/Output Control System

Activity between the central processor and its peripheral devices is controlled by a group of supervisory routines known as the physical input/output control system (IOCS). These routines provide for the checking, scheduling, starting, and queueing of I/O orders, the handling of I/O interrupts, and the processing for peripheral device error recovery.

The physical IOCS limits its scope to orders utilizing the system resident device channel, selector channels 1, 2, 3, and 4, and the standard multiplexer subchannels. The handling of orders for the Communication Intelligence Channel (CIC) and the nonstatus stacking multiplexer channel is performed by a group of routines known as the communications software system.

The physical IOCS is organized into two distinct sections of resident code; the channel scheduler and the error handler, each serially reusable.

6.2.2. Resource Allocation

The resource allocation routines are referenced by a set of supervisor requests. This allows these services to be used by job control, data management, supervisor, or user routines. Accounting information for the requesting user job is updated by the allocation routines so that an accurate representation of resource expenditure is available. Resources such as main storage, or temporary direct access storage may be allocated and/or controlled by the operating system.

Requests for main storage are available for use at two levels:

- Job control may request space for the duration of a job or a job step.

- A user program may request space for dynamic storage requirements.

A request for main storage space may cause rollout of a module of lower priority. In some cases, modules are rolled out to direct access storage and returned to a different location in order to provide an expanded main storage region.

Supervisor requests are provided for controlling access to system resources by independent tasks or jobs. A user may ask to be given exclusive use of a named resource for a period of time or use this resource concurrently with other users who also allow shared access. If any exclusive requester has control of the resource or if the requester requires exclusive use and the resource is in use, the requesting task may choose to be suspended until the resource is free or it may wish to accept the indication that the attempt to gain control of the resource was unsuccessful.

If a user program uses the system integrated access method (SIAM), temporary direct access storage will be available to the program. This service is intended for temporary files, and may be utilized by user programs or by operating system components. Space allocated by this request is always formatted to the standard system block size to improve system response and throughput.

Software operating efficiency may be increased by using the Operating System Storage Facility (OSSF). This optional feature consists of a fixed head disc subsystem connected to a control channel. Efficient utilization of this storage is managed by the supervisor. Files are allocated to the OSSF on the basis of file volatility, frequency of access, and size. Some of the files in this classification may be:

- operating system transient routines and overlays;

- parts of the system catalog;

- language processor and sort scratch areas;

- operating system checkpoint information.

OSSF files are accessed (as other DASD files can be) using the system integrated access method (SIAM).

While it is apparent that the fast access time of the fixed head disc subsystem improves system throughput, it should also be noted that the reliability and functional capability of the system are also extended. The system can be checkpointed at more frequent intervals, and time-critical functions provided by the system, such as communications message buffering, are improved.

6.2.3. Task Control

Up to 14 user jobs can be activated by the job scheduler for concurrent execution. Job steps can consist of one or more tasks that are scheduled concurrently for allocation of processor time.

Each job step has one task generated by the system. This task is deleted at the termination of the step. If it is desirable to establish additional tasks for the program, supervisor requests are available to establish additional tasks. Facilities to synchronize and delete tasks are also provided.

The allocation of processor time to a task is based on a system switch list which contains information about switching priorities, time-slice values, and time-slice utilization. The number of priorities and the initial time-slice values for each priority level are parameters in the supervisor generation (SYSGEN) procedure. Priority may be changed for a given task, at that task's request within predetermined limits.

6.2.4. Timer and Day Clock Services

The UNIVAC 90/70 System hardware contains a high-resolution timer. An interface is provided to allow a task to request an interrupt after any time period greater than 1 millisecond. The calling program may specify the wait interval in milliseconds or may specify a time of day at which the interrupt is to occur. A user program may request the time of day in various formats for any purpose and, as a SYSGEN option, the time can be displayed on the system console visual display screen.

The time of day is provided by a simulated day clock. In addition to providing the time to the programs upon request, this time is used by the supervisor for time-stamping of messages.

The maximum length of the time slice for each switching priority level is established at supervisor generation and is controlled by using the timer. This prevents misuse of processor resources which would result in an imbalance of system resources and degraded throughput. A time-slice optimization routine may be added to the supervisor.

6.2.5. Program Management

Conventions have been established for exit from a routine under abnormal conditions so that the calling routine may take alternate processing paths based on the results of the subroutine. If the subroutine is a job step, the exit condition is used to control conditional execution of the job control stream.

All addresses in a program are relative to the beginning of the main storage region in which the program resides, not the physical address in main storage. The base address for the program is set by the supervisor in a relocation register associated with the storage protection key. The hardware relocation register augments the relative address by the appropriate relocation value.

Program module rollout/rollin is utilized to provide contiguous storage for the region allocated to a job step. This provides for effective utilization of main storage. Storage allocation algorithms, which provide job storage contiguity, reduce the number of I/O operations associated with program module rollout/rollin.

Program rollout/rollin is an automatic function of OS/7. Figure 6–1 shows how rollout/rollin aids in maximizing storage utilization by compacting active regions when necessary. Figure 6–2 shows how main storage space is acquired for high priority programs such as communications applications.

The reentrant subroutines used by a job, such as data mangement modules, are shared by multiple jobs, and are loaded into storage only when needed by the first job request. These subroutines remain in main storage until the last job using them terminates.

Reentrant modules are treated as shareable resources. These modules may have multiple entry points, but may not reference externally defined labels which are not defined in another module.

6.2.6. System Console Management

Facilities are provided in the UNIVAC 90/70 System to permit communications between the operator and both the supervisor and user programs.

6.2.6.1. Input from Operator

The three basic inputs to the system are:

- Operator Commands

  Requests for a software component to perform some specific service (such as: DISPLAY, CANCEL, DUMP).

- Unsolicited Messages

  Data or control information for either a program or a component of the operating system. The program must be prepared to accept unsolicited input.

- Replies

  Data or control information for a user or operating system task which has requested input does not proceed until this response is provided.

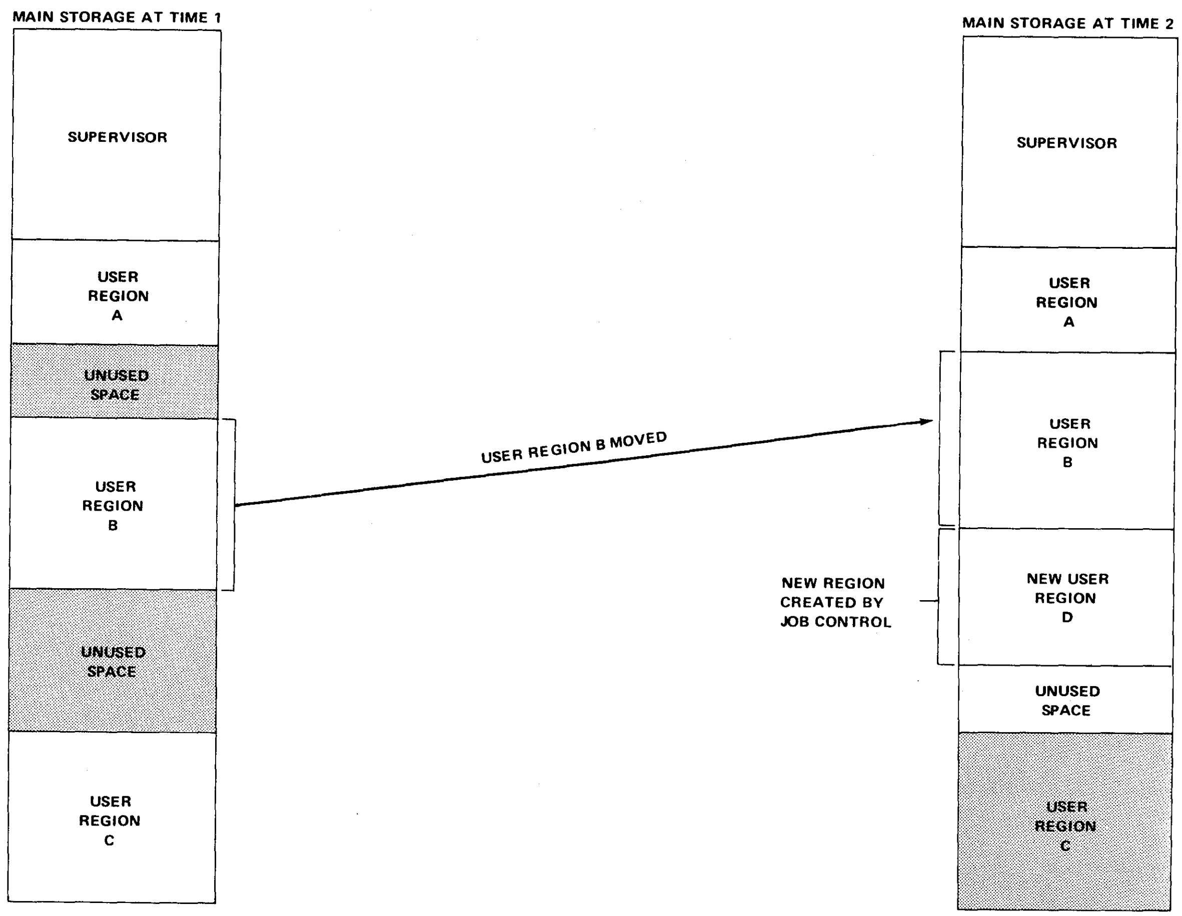


Figure 6–1. Main Storage Compaction by Rollout/Rollin (moving only)

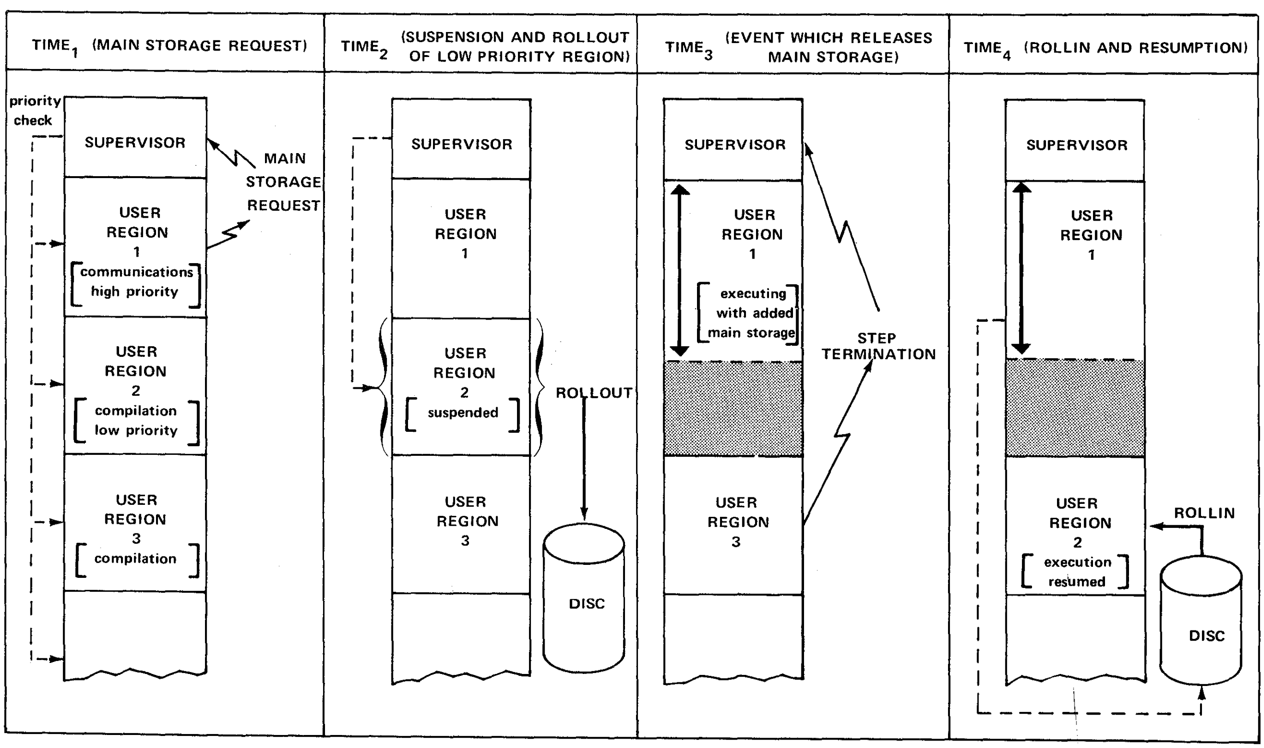


Figure 6–2. Use of Rollout/Rollin to Satisfy High-Priority Main Storage Requests

6.2.6.2. Output to Operator

Any program may direct information to the system console through a supervisor statement. This may be a directive which requires no response or may be an inquiry which requires a reply. System disciplines are available to reduce the number of messages sent to the system console. In a system of this size, it is usually preferable to have control stream parameters or data instead of operator input.

The UNIVAC 90/70 System provides disc storage space which is a file for permanent message information. These messages are preformatted and have separate keys which may be referenced. Any of the messages in the file may be retrieved by any program and displayed on the visual display screen of the system console. This permits the user to build a catalog of standard messages that are commonly used. In doing so, an installation may standardize console messages for similar operating conditions and may control the environment so that unnecessary messages do not appear on the system console visual display screen.

On request, the system provides detailed information concerning the system status. The information is provided on the visual display screen, unless otherwise specified either by the operator or by an option of the system generation procedure. System status may optionally be stored on a direct-access log. System status requests display the following kinds of information:

- Summary list of all active jobs in the system, including status information such as percentage of time used to estimated run time, and current running priority.
- Detailed information concerning a job such as number of tasks, resources assigned, and accounting information to date.
- Summary list of all onsite peripheral devices in the system.
- Summary list of all remote terminals online.
- Detailed information concerning a device such as allocation statistics, volume serial number, accounting data from present users, error and use counts.
- Summary list of all inactive jobs which have been submitted from onsite or remote terminals.

Messages which are directives to or requests from the operator are displayed on the visual display screen and, optionally, an abbreviated form of the message is recorded in the direct access log. Messages of a purely statistical or informative nature are not displayed on the visual display screen but are stored in the direct access log. If the optional console printer feature is provided, selected information from the direct access log is printed. The decision of which classes of messages will be sent to the system console visual display screen, the system log file, or optional printer, is made as a part of the SYSGEN procedure.

6.2.6.3. Message Display Management

The visual display screen for the system console has reserved space for operator input messages and separately reserved space for output messages from the operating system. When an operator keys in an entry at the system console the display screen is updated. The update procedure is an upward roll method which displays new inquiries and deletes serviced inquiries. (See Figure 6–3.)

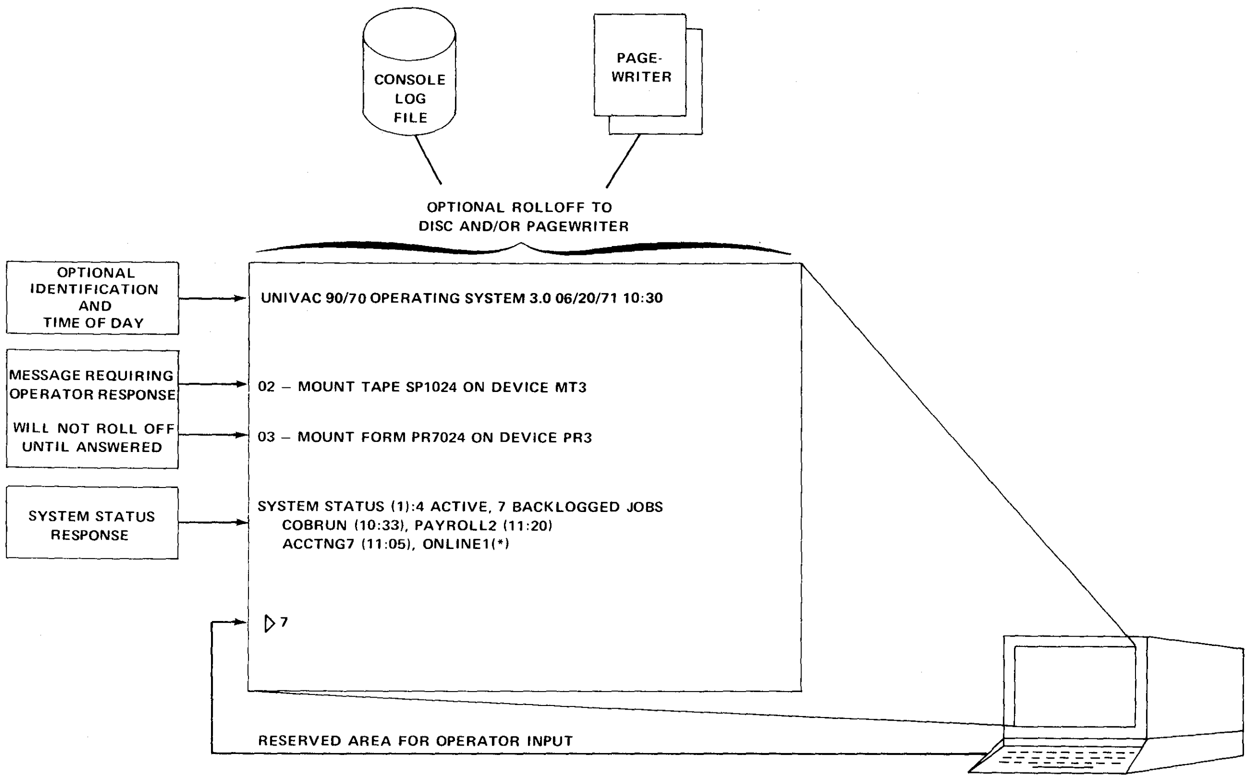


Figure 6–3. System Console Message Management

6.2.7. File Services

In shared direct access files, the physical IOCS provides lockout to prevent reading or writing a record while it is being updated. In addition, if the status of a file is to be changed dynamically from shared to nonshared while updating, this may be explicitly requested of the supervisor by requesting nonshared use of the file.

Files on direct access devices which are generated and maintained by the system are protected by the supervisor. Only authorized data management users are allowed to write on a device which contains a volume on which protected files are stored. Physical IOCS users are allowed to access these systems volumes if the program has declared a DTFPH and had executed an OPEN. In this manner, conversion efforts can be minimized for IBM 360/DOS and UNIVAC 9400 DOS programs which use physical IOCS without sacrificing systems reliability.

6.2.8. Program Error Handling

Any error which causes a program interrupt is examined to determine:

- The type of interrupt – such as program check and protection violation
- The type of job – user or system

An interface is provided for processing of error information by means of user-supplied island code. Island code is a closed subroutine, having the entry point defined to the supervisor by various action macro instructions and is given control upon the occurrence of certain contingencies. Standard actions are initiated in the absence of user code. If the unrecovered error is in the system, the system is brought to a quiescent state and a restart at this point is attempted. If recovery fails, information is collected for an orderly abnormal termination.

If any requester of a supervisor function provides a set of parameters which are inconsistent or invalid, the requester is notified.

6.2.9. Subroutine Linkage Table

Each protection key has an associated relocation register containing an absolute address. This allows all references to program and data to be job-relative. In the case of noncontiguous reentrant subroutines or supervisor tables, the supervisor vector address table is provided to enable a program to reference data or subroutines outside of its region. This is accomplished through indirect data addressing or the indirect branch and link instruction.

6.2.10. Spooling Operations

OS/7 uses a spooling technique which consits of a set of routines that buffer data files for low speed input and output devices to an intermediate storage device. There are four types of routines used for spooling operations:

- Input readers
- Output writers
- Spoolin routines
- Spoolout routines

The function of each routine during spooling operations is shown in Figure 6–4.

Forms control for printer buffers and recovery at line or page increments are provided. It is also possible to allow multiple copies of a buffered output file. Operator intervention to postpone or initiate output to the peripheral is allowed.

Output writers are provided for devices used in a remote batch environment. This allows a batch program to be unaware of whether it is servicing a local or remote user.

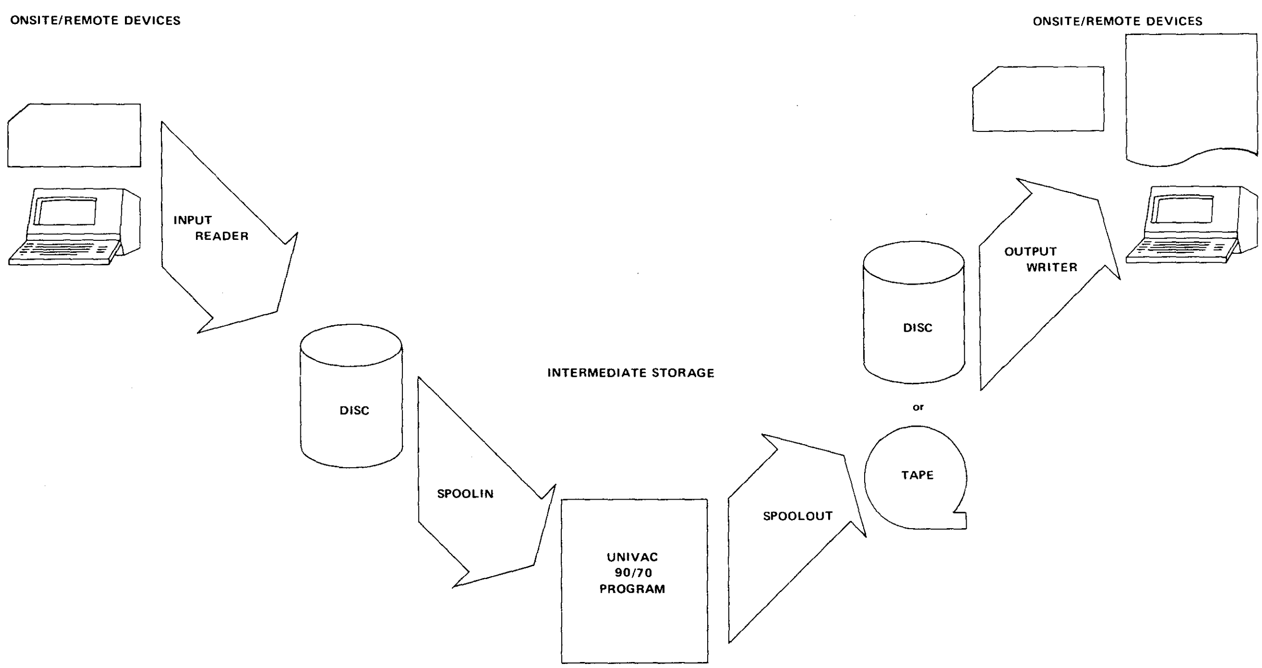
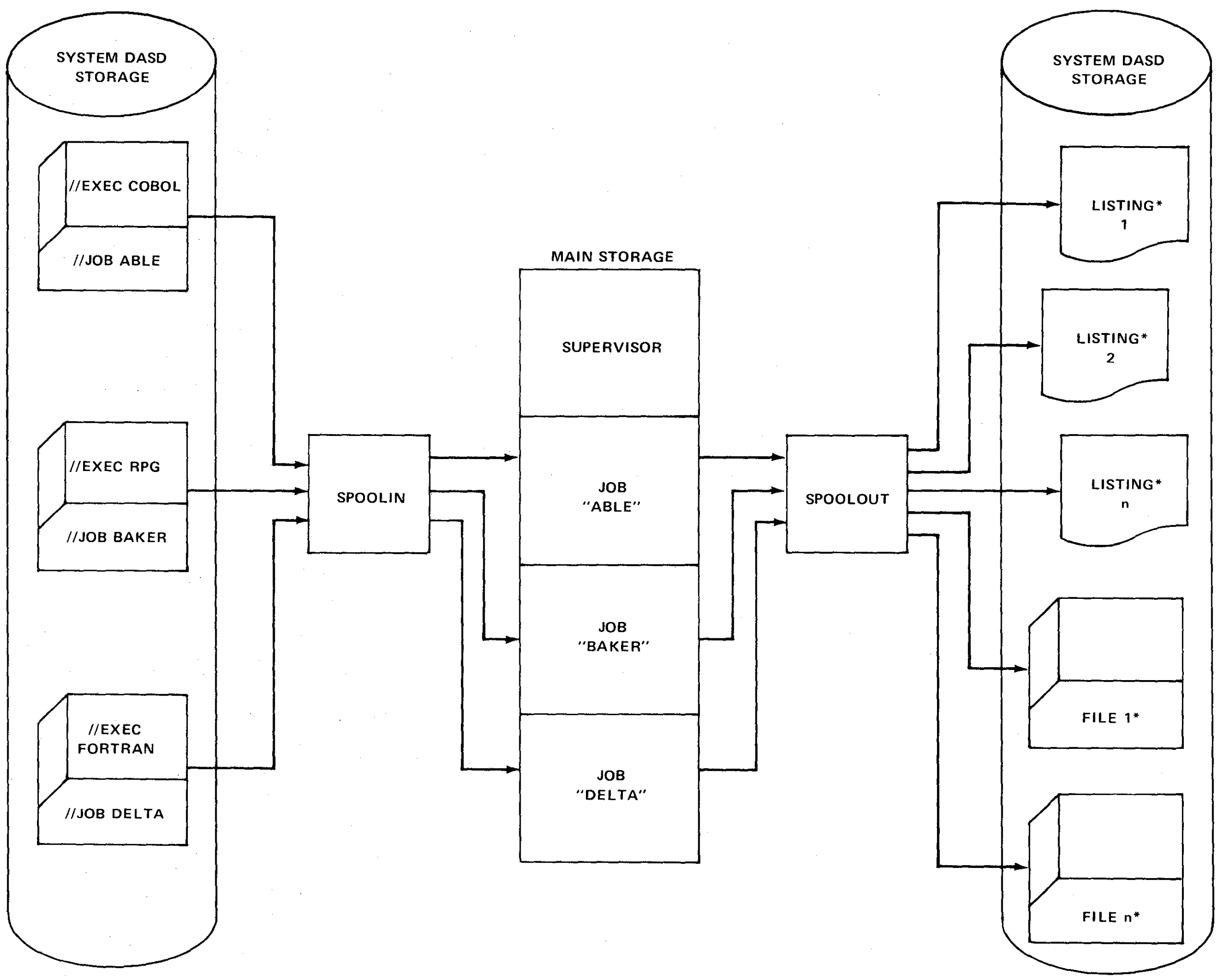


Figure 6–4. Spooling Operations

Input readers are provided for local and remote subsystems which are normally used as batch mode input devices. The program is not required to discern whether data files were submitted by the control stream or buffered input.

The standard device type used for intermediate storage and the allocation algorithm for disc space used for this purpose is determined as a part of the supervisor generation procedure. Disc space is normally used, although tape may be specified by the user. In addition, the system standard can be overridden through job control stream parameters for each output file requiring intermediate storage.

Figure 6–5 shows how each active job in the system has access to the concurrent spooling capability.



*These files may optionally be directed to magnetic tape instead of direct access storage.

Figure 6–5. Concurrent Job Control Streams

6.2.11. Operating Environment Recovery

Operating environment recovery is provided through physical IOCS services and checkpoint. Descriptions of these recovery methods follow.

6.2.11.1. Physical IOCS Services

The supervisor provides services at the physical IOCS level to assist data mangement or message control routines in data recovery and redundancy.

If a user program bypasses the standard access methods, these same basic IOCS services are provided.

6.2.11.2. Checkpoint

A checkpoint request instruction is provided for restarting a job with synchronization of all disc and tape sequential files. If unit record input files are buffered to disc, then the position of these may also be restored on restart.

6.2.12. Diagnostic/Debugging Aids

Diagnostic and debugging aids provided in OS/7 include trace mode, monitor mode snapshot display of main storage, main storage dumps, standard system error message interface, and uniform error responses to user programs. Descriptions of these aids are provided in the following paragraphs.

6.2.12.1. Trace Mode Support

The UNIVAC 90/70 processor provides branch and interrupt information when operating in the trace mode. This information is accumulated and stored in the calling program spoolout file to facilitate debugging. A branch trace capability is provided in addition to the complete instruction trace (monitor mode) provided by the program trace routine. The 90/70 hardware trace and monitor functions are used to support this capability.

- Branch Trace

  Provides a log on the printer of every successful branch instruction in a user program. The branch trace is facilitated by associated hardware capability and it is more efficient than other purely software trace methods.

6.2.12.2. Monitor Mode Support

The OS/7 program monitor facility provides an instruction-by-instruction chronological log of a user program or part of a program for subsequent printing. This log contains the relative address of the instruction in main storage, instruction mnemonic code, instruction machine code, index register, base register, operand 1 after instruction execution, operand 2 after instruction execution, operand 1 before instruction execution, and condition code setting.

Program monitor is initiated through the use of control stream parameters or by the trace macro. Optionally, the user can specify that only a particular segment of the program is to be traced. In this case, address limits specified in the control stream designate the segment to be traced.

6.2.12.3. Snapshot Display of Main Storage

The capability is provided for requesting a partial storage printout at given points in a program by means of a SNAP macro instruction within the program itself. It is also possible to specify by parameters at run time the portion of storage to be printed and at which points the printouts are desired. This enables a program to be tested without recompilation to include and alter SNAP requests.

6.2.12.4. Main Storage Dumps

There are three situations in which a main storage dump may be provided:

- Abnormal termination dump for user programs. This provides a main storage dump of the region in hexadecimal and alphanumeric plus a formatted display of error codes, job-oriented tables, and supervisor information to assist the user in debugging.
- Program or operator requested dump. This provides an orderly capability for the operator or any program to request a main storage dump in the same format as the above dump.
- System failure dump. This is a program intended for use when for some unexplained reason the operating system performs abnormally.

6.2.12.5. Standard Systems Error Message Interface

An error message service routine provides complete and specific error messages without requiring each system module to contain alphanumeric error information. This routine locates the message in a disc file and transfers control to the system console handler for message display or system logging.

6.2.12.6. Error Response to User Programs

Error codes which are returned by the supervisor to the calling program are standarized to provide a uniform interface for all system services.

If a user requires return of control after the detection of hardware failure or software exception, island code must be provided to handle this situation; otherwise, an orderly abnormal termination is invoked for the user job which optionally may include a main storage dump.

6.3. JOB CONTROL

Job control is the nonresident component of the OS/7 that manages the system resources (main storage, software facilities, and peripheral devices), prepares jobs for processing, and initiates program execution. A job is a user-submitted task or unit of work to be performed. Each job can be divided into job steps to be executed serially. Job steps are made up of problem programs, with each job step containing only one request for program execution. Additionally, a job step may have several tasks (See 6.3.3.)

Multijobbing is the concurrent execution of several jobs residing simultaneously within the system. Job control automatically schedules jobs according to priority and available resources and has the capability of initiating and managing up to 14 jobs for concurrent execution. Job control services are performed prior to the execution of the initial job step, during the transition between job steps, and at the conclusion of the job.

The services performed by OS/7 job control are directed by the user through control statements known as the job control language. These control statements convey information required by the operating system to initiate and control the processing of jobs, such as identifying the job and the programs that comprise it and specifying main storage requirements and peripheral device assignments necessary for job execution. The flexibility provided by these statements enables the user to define the job requirements for a variety of facilities and to be independent of many limitations imposed by system configurations. The number of required statements can be greatly reduced by the use of prestored control statement sequences or procedures and by default parameters supplied during system generation.

The services performed by OS/7 job control are:

- Analysis of the job input control stream
- Resource allocation, including main storage, peripheral devices, and reentrant software facilities
- Device assignment
- Volume and file label storage
- Retrieval of prestored control statement sequences for subsequent modification and execution.
- Maintenance of the system file catalog allowing automatic location of files by identifier
- Program restart from a checkpoint
- Object code debugging capability and parameter input to object modules
- Automatic job scheduling and initiation
- Requests to schedule additional jobs
- Requests to store control stream data

A control stream is a group of sequenced control statements, written in job control language, which defines a job and directs the processing of that job. The control stream acts as an interface between the user job and OS/7. The control statements contain information regarding:

- Job identification and accounting
- Names of the problem programs to be executed
- Job scheduling
- Cataloged files
- Device assignment
- File label information
- Storage allocation
- Magnetic tape operations between job steps
- Job termination
- Conditional execution of job control statements

6.3.1. System File Catalog

The system file catalog provides the capability of locating a file automatically when only a file identifier is given. The system file catalog permits the recording of file name, volume serial number, device type, file sequence number, and the file identifier. These items of information, necessary for file assignment, can be obtained from the catalog by specifying only the file identifier, thereby assuring accuracy and convenience.

Certain files are periodically updated or are logically part of a group of files, each of which is created at a different time. The system file catalog can retain information concerning successive generations or versions of such files. These generations may be assigned to a job by specifying only the file identifier and a number which is relative to the most recent generation.

6.3.2. Job Entry

As control streams enter the system from local and remote input devices, the input reader performs the following functions:

- recognizes calls on the procedure processor;
- expands the called procedures;
- checks the order and syntax of control statements;
- separates and files embedded data;
- summarizes facility requirements for subsequent use in scheduling the job or job step;
- recognizes job priorities;
- queues the job according to its priority; and
- deletes control streams of processed jobs.

All control statements in the input control stream, except those contained in embedded sets of data, are checked for correct order and syntax. If errors are detected, the control statements and the appropriate diagnostics are placed in the job output file for printing. When a syntax error is detected in the control statements, the input control stream and other information pertaining to the job are deleted. However, data which was embedded in the control stream can be retained.

When a procedure call statement is encountered in the input control stream, the requested procedure is retrieved from the procedure library and expanded in light of information specified in the procedure call. The expanded procedure is processed as if it had been supplied in the control stream. See Figure 6–6.

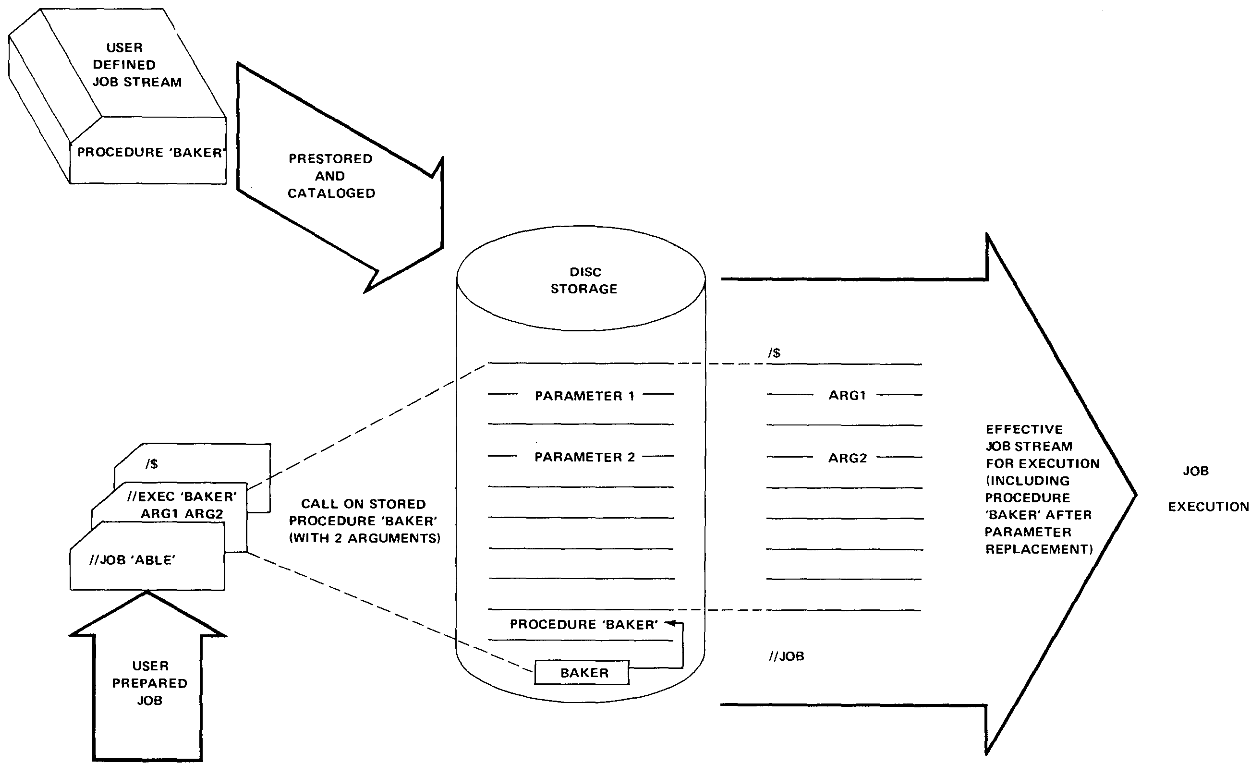


Figure 6–6. Information Flow – OS/7 Cataloged Job Control Procedures

6.3.3. Job Management

The job scheduling, initiation, and continuation functions of job control establish the operating environment for a job and provide the processing control required between job steps.

Jobs are not executed immediately upon entering the system. Instead, they are placed in an input work queue according to a user-specified scheduling algorithm.

Job scheduling is performed automatically by OS/7 and is based on the recognition of priorities assigned to each job by the user by means of the JOB control statement. At system generation time, an installation- determined number of priority levels, associated priority types, and job selection schemes are set up as a scheduling table. The job scheduler analyzes:

- the priority levels assigned to jobs by the user,
- the resource requirements of the jobs,
- the position of a job among all other jobs in the system which are not yet selected for processing,
- and schedules the jobs for processing based upon these characteristics and availability of resources in the system.

All priority levels are categorized into three functional divisions or priority types by the user. These priority types are explained in the following paragraphs.

- Normal Priority

  This is the priority type which is intended for use by the major portion of all jobs to be processed. Such jobs are characterized by the lack of need for urgent job completion. Jobs of the normal priority type are those which may be optionally declared as subject to preemption or rollout.

- Override Priority

  This is the second highest priority type in the system. If a job is assigned this priority, and no jobs having preemptive priority await resources, all available system resources necessary for completion of the job are allocated as soon as possible. If unallocated main storage is insufficient to satisfy the requirements for completion of a job having override priority, other jobs may be temporarily suspended and rolled out to disc storage in order to obtain the required main storage.

- Preemptive Priority

  This is the highest priority in the system. If a job is assigned this priority, all system resources necessary for completion of the job are allocated as soon as possible. If unallocated resources are insufficient to satisfy the requirements for completion of a job having this priority, other jobs (which have been specified as preemptable) are preempted (aborted) in order to obtain the required resources.

These priorities are used in conjunction with the priority levels determined by the user at system generation time and provide an efficient and workable job scheduling relationship.

At system generation time, a job selection scheme also may be determined and assigned for each priority level. The following job selection schemes may be applied to each priority level.

- FIFI (First in, first initiated)

- FIFF (First in, first fit)

The basic job selection strategy of the OS/7 job scheduling function involves an attempt to initiate jobs from succeeding, numerical priority levels. If a priority level contains no jobs which can be initiated, due to insufficient resources, initiation of a job in the next lower priority level is attempted.

Within each priority level, the user can specify the number of job initiations to be attempted.

The user may designate fixed job regions in main storage with assigned peripheral devices. Such a fixed job region may then be occupied by one job after another whose resource requirements are satisfied by the region. A job which is to occupy a fixed job region need not compete with all other jobs to resources.

During job initiation, the user-specified switching priority of the job is established. The new job step task is entered in the program switch list at that priority. Up to 14 concurrently operating jobs can be managed. This number usually includes one or two system jobs for supporting services such as job stream reading, printer output writing, and data communications control.

Peripheral devices which have been assigned for the duration of the job remain allocated to the job. Those devices which have been assigned only for the duration of the job step are released to the system. Job control then continues to process the control stream as specified by the user until the next job step is ready for execution. The user may specify that the execution of a subsequent job step is dependent on certain conditions: the job may be terminated or one or more job steps skipped. Job step termination will normally cause the scheduler to do one of the following:

- Roll in a job step which has been suspended due to preemption by a higher priority job.
- Initiate a job step of a job which is partially completed, but which has been suspended between steps due to higher priority jobs.
- Initiate the first step of a new job for which all resources are now available.

6.3.4. Job Termination

There are two forms of job termination, normal and abnormal. Both forms cause the job termination function of job control to deallocate resources previously allocated to the job, such as peripheral devices, main storage, and disc scratch areas. Any remaining data or control statements in the control stream associated with the terminated job are ignored following job termination.

6.3.5. Job Accounting

Accounting information for each job is accumulated upon termination of each job step and is reported at the end of the job print file.

The items of accounting information that are collected by job control at job step termination are:

- Processor time together with channel and device information
- Elapsed time
- Region size
- Time in main storage
- Number of tape or disc units assigned
- SIAM space used
- Number of cards read
- Number of lines printed and cards punched
- Account number

When a job step is initiated, the first portion of the job control transient routine is loaded and executed. This routine analyzes the amount of available space, selects space adequate to contain it, and allocates this space to the job step. Any space not required by a job step is considered to be available for another job. The remainder of job control is loaded into the allocated space after a test has been made to ensure that adequate space is available. Job control continues to process the control stream until it transfers control to the job step. The job step is read in, overlaying job control, and then is executed. For a multistep job, the job steps are sequentially loaded and executed.

When the execution of a job step is completed, job control is recalled to process subsequent control statements in the control stream. Job control determines whether the next job step to be executed has requested a change in resources and, if so, fulfills them.

6.3.6. Device Assignment

The job control language permits flexibility in assigning the devices required to execute a job. Peripheral devices are assigned to a job based on job control language information which may be specified in previously filed control streams, or established as default parameters at system generation.

The user may supply some or all of the following information when assigning devices:

- Logical Unit Designator

  The user may designate either a logical unit number or name to assign a device type.

- Volume Serial Number

  The user may request a device by specifying a particular volume. If the volume is mounted, the device may be assigned to the job; if the volume is not mounted, job control determines whether the device can be assigned.

- Duration of Device Assignment

  The user may specify whether a device is to be assigned for the duration of the entire job or on a job step basis. The user may dynamically deallocate a device during job execution or may deallocate the device at the termination of the job or job step.

- System Devices

  The user may assign certain devices that are available to all jobs executed in the system through operator commands at the system console.

- Shareable Devices/Volumes

  Devices or volumes may be assigned to more than one job at a time if the user so specifies.

- Alternate Devices

  Sequential files on tape or disc may occupy space on more than one volume. By allocating alternate devices to these files, the time for rewinding, demounting, and mounting may be overlapped with processing of the file.

- Device Groups

  Devices of varying types can be collected into groups. These groups are named by the user and the proper device relationship is established during device allocation procedures. Thus by referencing group names instead of device types or logical unit numbers during job execution, the user is allowed a considerable degree of device independence. This device-type independence is used in conjunction with all data management access methods within the device range as specified for each access method.

- Optional Devices

  The user may specify that the assignment of a particular device is optional to the execution of a job.

- File Characteristics

  The user may complete the file definition and specify parameters at job execution time.

6.3.7. Automatic Volume Recognition

Automatic volume recognition (AVR) is a feature which allows the operator to mount labeled volumes on available units before receiving the requesting message. The system is alerted with an attention interrupt when the operator mounts the volume on a unit. The system recognizes the volumes by their labels and later assigns these premounted volumes to the job steps calling for them. This reduces set-up time and increases system throughput.

6.4. Data Management

Data management is that part of OS/7 which provides a convenient interface between user programs and the hardware-oriented I/O portions of the supervisor. Data management facilities provide organizational benefits such as record blocking and deblocking, buffering, data validation, label processing, and device independence. Data management facilities consist of logical input/output control system (IOCS) modules, transient routines, and a convenient user interface. A functional diagram of data management is shown in Figure 6—7.

6.4.1. Logical IOCS Modules

Logical IOCS modules consist of reentrant subroutines that are controlled by OS/7. OS/7 loads these modules as they are needed to serve jobs that are active. One copy of a module can serve all files of the particular type instead of repeating the file processor coding for each active job in a program or for jobs using the same type of file. This is done to economize main storage space.

6.4.2. Transient Routines

Transient routines are used for infrequently requested functions. For example, the initiation and termination procedures for file processing (OPEN and CLOSE) involve seldom-used coding. The space used by transient coding is used and is then made available for other purposes.

6.4.3. User Interface

OS/7 data management provides a convenient user interface to allow definition and processing of files.

6.4.3.1. File Definition

Each file used by a program must be described to data management by a standard file definition. Different language processors accept different input from the programmer, and translate this input to the appropriately formatted table of parameters. In the assembler, a macro instruction called define the file (DTF) is used to define each file used by the program.

6.4.3.2. File Processing

A program communicates with logical IOCS in order to accomplish the processing of files that have been defined. This is accomplished by using simple macro instructions in the program, which communicate with logical IOCS. OPEN and CLOSE are examples of file control instructions; READ and WRITE are example of instructions used to input and output records of a file.

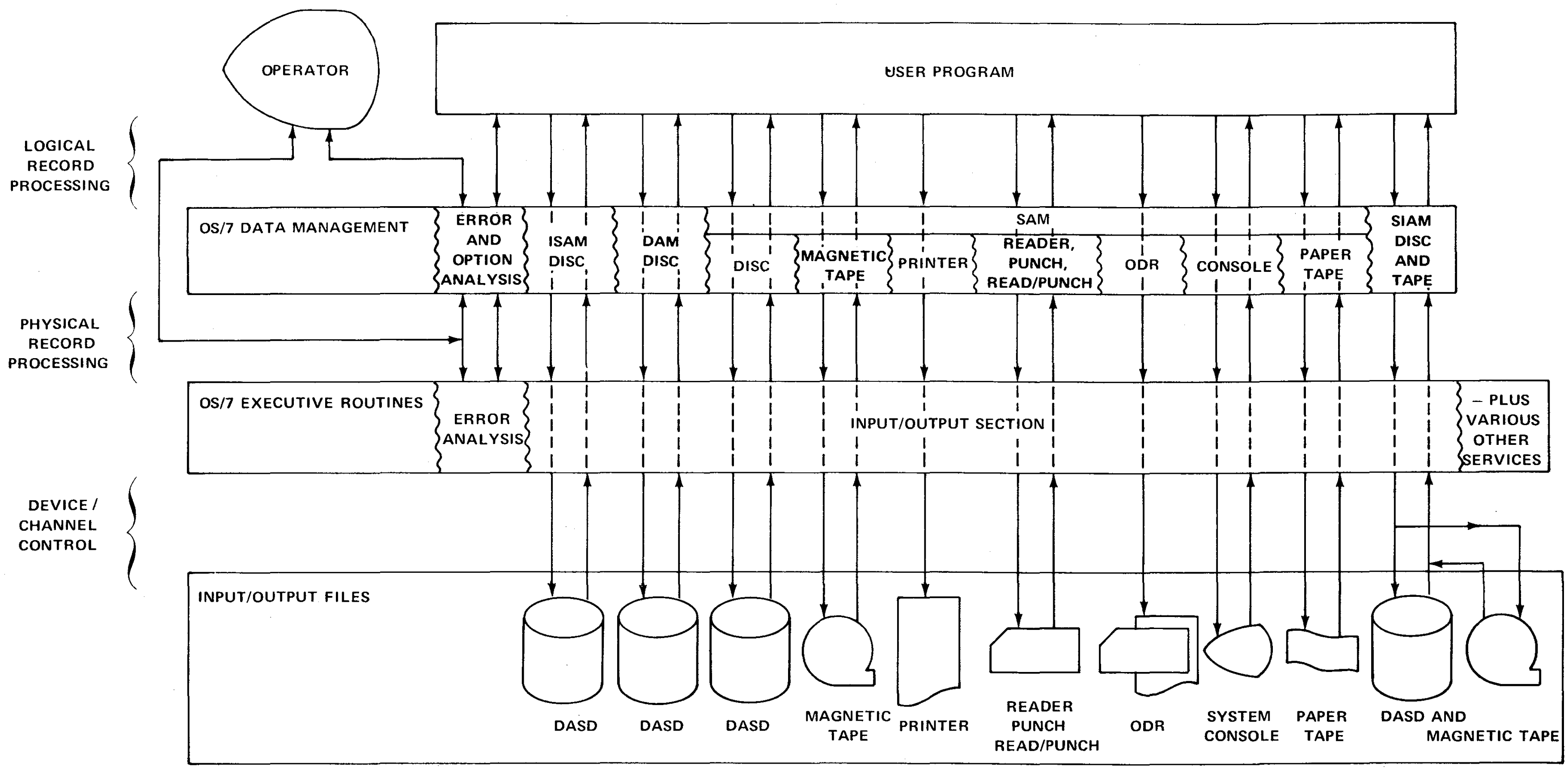


Figure 6–7. OS/7 Data Management

6.4.4. File Format Processing

One of the major areas of file processing is checking, creating, and validating label records, particularly in disc and tape processing. Standard label support is provided for all access methods. Provision is made for user label processing if desired.

6.4.5. File Table Completion

The file tables, which are generated by language processors, are the repositories for all file information. Parameters are needed to specify such characteristics as device type; record format; block and record sizes; and label, error, and exit options. In OS/7, most file parameters can be specified directly from the control stream by the programmer, from cataloged job information, or from labels on tape and disc volumes. In addition, a user exit is optionally available when the file is opened. This permits a last chance changing of the file table.

The priority order within the major sources of file parameter information is as follows:

- User OPEN exit
- Job control stream
- Catalog
- Macro-generated parameters

Due to the flexibility in supplying file parameter information, validation procedures provide for consistency checking when the table is generated and extensive rechecking when the file is opened.

6.4.6. Access Methods

The manner in which records are recorded on a storage medium depends on the characteristics of the storage involved. Four access methods for storage and retrieval are availabe in data mangement: the sequential access method (SAM), the direct access method (DAM), the indexed-sequential access method (ISAM), and the system integrated access method (SIAM). SIAM is used by the UNIVAC 90/70 System to support library and work files. Support for file protection in the form of track lockout is provided for disc files written under the control of DAM, SAM, and ISAM. Support for file protection in the form of segment lockout is provided for disc files written under control of SIAM.

6.4.6.1. Sequential Access Method (SAM)

Files with records that logically follow one another in a serial manner may be processed by the sequential access method. Some devices which may have sequential files include magnetic tape units, card readers, printers, punches, discs, the optical document reader, and paper tape.

An important facility in modern systems is the ability to substitute alternate devices for a primary device that for some reason is not available.

Device-independent sequential input files may be read from card readers, magnetic tapes, and discs. Card punches, printers, magnetic tapes and discs are interchangeable for output files.

The SAM for disc will dynamically attempt to acquire additional space for files if the original allocation is not sufficient to continue the file being created.

6.4.6.2. Direct Access Method (DAM)

The direct access method allows the user of a UNIVAC 8411, 8414, 8424, or 8440 Disc Subsystem to process a file in a random manner. A user may access a record by submitting to DAM the physical address of the record or by submitting a relative location which the software uses to determine the physical address. The value submitted may be relative to the first record of a file or to the first track of a file. Relative addressing is applicable only to fixed-length records.

This relative addressing promotes device independence because a record could be physically located in different track positions on each disc pack type but the relative record position would be the same. DAM also allows the storing and retrieval of records by name (key) within track or cylinder boundaries.

6.4.6.3. Indexed-Sequential Access Method (ISAM)

The indexed-sequential access method (ISAM) allows the user of a UNIVAC 8411, 8414, 8424, and 8440 Disc Subsystems to process a file in either a random or a sequential manner. Four file processing modes are provided:

- loading a file, which consists of writing presorted input records and their key fields onto the disc, while creating and writing out a set of indexes;
- retrieving and updating records sequentially;
- retrieving and updating records randomly;
- adding new records to an existing ISAM file.

In addition to the facilities previously mentioned, OS/7 data mangement offers an option to the ISAM cylinder overflow specification. A percentage of a cylinder may be defined for overflow records instead of a fixed number of tracks, thereby allowing a device-independent specification.

When inserting new records in an ISAM file, efficiency can be considerably improved by specifying a main storage area to contain track information. To allow device independence this specification can be deferred until the disc device type has been assigned.

When file processing is terminated, ISAM logs reorganization statistics in the user/system printer file. This is user information which helps to determine when the file should be reorganized.

There is an option which allows a new file to reuse the disc area assigned to an existing file precluding the need to scratch the old file and allocated disc space to the new file. This is particularly useful as a testing option.

6.4.6.4. System Integrated Access Methods (SIAM)

The system integrated access methods centralize and standardize all input/output done by system routines. SIAM is structured in much the same manner as other data management access methods with DTF tables, reentrant modules, processing macro instructions, and overlay routines. It is a disc-oriented method and offers device independence for users of UNIVAC 8411, 8414, 8424, or 8440 Disc Subsystems. Support is also provided for tape files with some degree of disc-type independence between tape and disc.

SIAM supports a user-oriented index structure. Names in the index can be fixed or variable in length.

Featured in SIAM is a fixed block size with writing and reading of multiple numbers of blocks to reduce disc accesses. Variable length records are supported. A calling program may organize SIAM files in the following ways:

- sequential
- sequential with subfiles
- partitioned
- random by name

6.4.7. Standard Direct Access VTOC Service

When executing jobs in a multijobbing environment, it is always difficult (and often impossible) for the programmer to know the exact organization of particular direct access volumes. There is, nevertheless, a need for most jobs to create and process files on these volumes. The creation and processing of direct access files require the means for allocating space, releasing unused space, scratching files when no longer needed, obtaining label and extent information, and renaming files. These procedures are required by data management, and various service and utility programs.

Disc space management routines for OS/7 provide an efficient and completely automatic space accounting and maintenance feature which relieves the user of the responsibility of knowing the precise contents of direct access volumes. These routines also permit the resolution of competing demands for allocation, and establish standard interfaces.

Disc space management consists of a set of service routines which allocate space to files on direct access volumes. This is accomplished by maintaining the volume table of contents (VTOC) through standard procedures for all files: system, temporary, and those designated permanent by the user.

The disc space management routines maintain the VTOC by creating format labels for new files and deleting format labels for files removed from the volume. When a file is created, unused space is found for it by searching the appropriate format labels in the VTOC, allocating the space as the extents of the file, and removing it from free space. When a file is deleted, the format labels for the file are removed from the VTOC; the extents previously assigned to the file are then available for allocation.

Disc space management includes four functional areas. Usually these functions are not called directly by the user but are the result of higher level requests. These functions are:

- ALLOCATE – allocates initial disc space to a file.
- SCRATCH – deletes a file or portion of a file by deleting its associated extents and/or blocks in the VTOC.
- RENAME – changes the name of a file.
- OBTAIN – enables direct access to any block in the VTOC.

6.4.8. Reconstruction and Recovery Procedures

OS/7 provides comprehensive procedures to assure the integrity of data files. Data management allows the user to take full advantage of the supervisor capabilities for checkpoint/restart. File reconstruction is also provided by way of the file trace routines provided within data management.

6.5. DATA COMMUNICATIONS

The UNIVAC 90/70 Operating System is designed to interface with programs having online requirements. A flexible communications control program combined with an efficient scheduling and interrupt processing supervisor provide the environment for online processing. The general message flow for communications processing is shown in Figure 6–8.

As stated in Section 2, communications capability is provided by means of a Data Communications Subsystem (DCS) connected to the multiplexer channel or the communication Intelligence Channel (CIC). Similar facilities are provided for both system configurations.

The communication control program consists of the following functional components:

- Physical I/O control system (IOCS)
- Message control program
- Message processing programs

These components are described in the following paragraphs.

6.5.1. Physical I/O Control System

The physical IOCS is a main storage resident extension of the supervisor and requires main storage to operate in the same processing state as the supervisor. Its primary function is to control the communcations environment of the operating system. It includes I/O dispatcher and interrupt handling routines.

6.5.2. Message Control Program

The message control program, which interfaces with the resident physical IOCS, is the major component of the communications control program. It operates as a separate job with prime priority on the supervisor switch list.

The responsibilities of the message control program include initiating and controlling:

- Message flow
- Line control
- Message acknowledgment
- Buffering
- Main and direct access storage queuing
- Error handling
- Message translation
- Polling sequences
- Operator-console communications

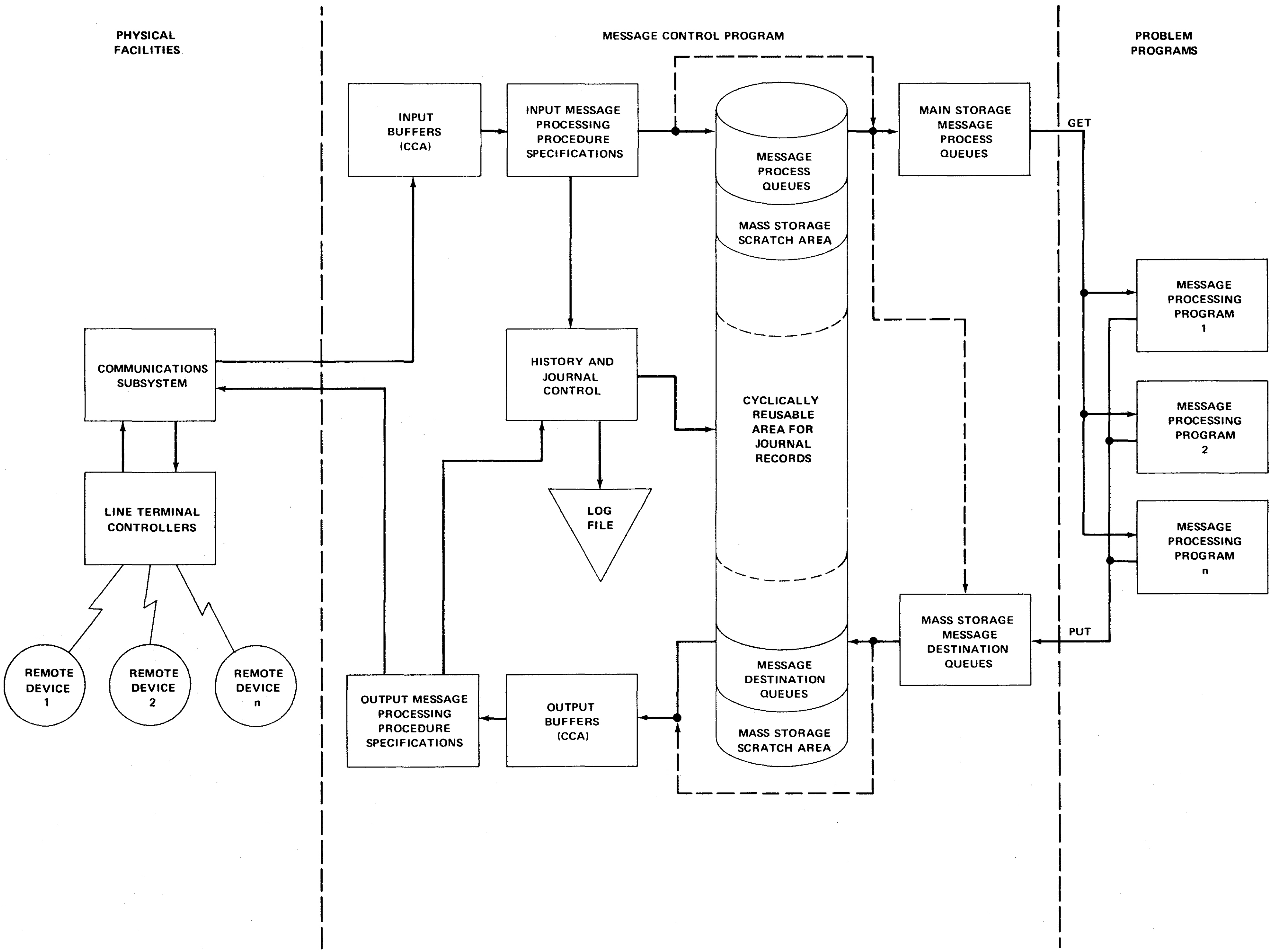


Figure 6–8. General Message Flow – Data Communications

Options enable users to audit error conditions, maintain error counts and header summaries, and provide traffic analysis information. Options are also provided to support audit trails, data integrity, and the various editing functions required in a communications environment. Support is provided for multiple destination routing, broadcast messages, rerouting, automatic store and forward of messages and message switching.

Users are permitted to examine messages after they have been received and before they are transmitted in order to perform special data conversion and editing (subject to systems considerations).

Queuing is supported on disc storage devices and main storage. Messages are queued on a priority basis by destination. Urgent messages are recognized as they are received and temporarily override other message priorities during transmission.

Message queue overflow from main storage to direct access storage is supported at the option of the user.

Support is provided for automatic polling, dial-out, automatic answering, and manual dialing.

The following terminals are supported:

- TELETYPE Models 28, 32, 33, 35, and 37.
- UNISCOPE 100 Display Terminal
- UNIVAC DCT 500, 1000, and 2000 Data Communications Terminals
- UNIVAC 9200/9300/9400 Systems
- UNIVAC 1004/1005 Subsystems
- IBM 2780 binary synchronous communications

The system handler interface has been standardized, facilitating the subsequent incorporation of new communications device handlers into the system. The above list does not preclude the addition of such device handlers as may be required.

6.5.3. Message Processing Program

The message processing program is user-generated coding which processes incoming messages and generates any applicable response messages. This program interfaces with the message control program through macro instructions provided for this purpose.

Macro instructions are provided to direct the message control program to control:

- Sending and receiving of messages
- Routing
- Message switching
- Time and data stamping
- Sequencing and sequence checking
- Source I.D. validation

- Message queue maintenance
- Destination validation

In addition, two macro instructions are provided to control:

- Length checking
- Priority control

6.5.4. Remote Job Entry

Communications input readers and output writers provide spoolin and spoolout of remote entry jobs independent of other system functions. The same communications networks which are used for such applications as inventory updating or inquiry and transaction processing may be alternately used to enter remote jobs into the system. Priority queues exist within the operating system for scheduling jobs. Job and system status are available for operator inquiry.

The remote job control language is identical to the standard onsite job control language with the exception that some additional information which uniquely identifies the remote station is required. An option is provided which allows outputs of remotely executed jobs to be delivered to central site peripherals rather than returned to the remote station.

6.5.5. Network Definition

Macro instructions are provided to define the system line groups and terminals. Common line sharing is permitted to provide diagnostics with the capability to test a given remote device while other devices are active. Dynamic reconfiguration of destination terminals is provided to allow user flexibility when a particular terminal is down or a particular communications environment requires a viable line terminal configuration.

6.5.6. User Own Code

Programs are permitted to use own code to perform special data conversion and editing on all messages entering and leaving the system.

6.5.7. Error Checking and Message Recovery

The communications control program provides automatic error recovery and message recovery, as needed. It provides for the interception, cancellation, and logging of messages in error.

Message retrieval and checkpoint/restart of message files can be requested on a message or file.

6.5.8. Multiple Message Processing Programs

A multiple number of message processing programs is permitted to operate concurrently under OS/7 subject to availability of system resources.

Program-oriented networks operating under the control of a message processing program are able to create files of information to be processed on another network by another message processing program, concurrently, if desired.

6.6. LANGUAGE PROCESSORS

Language processors are provided to allow the user of the UNIVAC 90/70 System flexibility in preparing programs. Programs may be written in COBOL, FORTRAN, Report Program Generator (RPG), or assembly language.

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 oriented toward computational problems.

The source program written in the COBOL or FORTRAN language is input to a COBOL or FORTRAN compiler. The compiler translates the COBOL or FORTRAN program into object code which is processed by the linkage editor and then may be executed.

RPG is a programming language which simplifies the preparation of business reports for an internally programmed data processing system. RPG translates a source language description and logical solution of report requirements into an object language program which accepts raw data and from it produces a report conforming in content and format to specified requirements.

The symbolic language of the assembler is a versatile and detailed method for writing programs through the use of mnemonic instruction codes, assembler directives, data generation instructions, and the powerful macro generation calls.

6.6.1. COBOL

COBOL (Common Business Oriented Language) is a programming language oriented toward problems in business applications. The language is similar to the English language, rather than a notation which considers the technical aspects of a particular data processing system. The source programs are easily transferable among systems that accept American National Standard COBOL X3.23–1968. Each of these systems provides 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 limited machine dependence. Source programs written in COBOL consist of four major divisions:

- Identification Division

  This division contains information which identifies the source program and the output of a compilation. In addition, the author, installation, and so forth, may also be identified.

- Environment Division

  This division specifies a standard method of expressing those aspects of a data processing problem that are dependent upon the physical characteristics of a specific system, and also allows the specifications of the compiling system hardware characteristics, input/output control techniques, and so forth.

- Data Division

  This division describes the data that the object program is to accept as input, manipulate, create, or produce as output. The 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

  This division describes the logical steps that must be taken in the solution of the data processing problem.

The UNIVAC 90/70 COBOL compiler conforms to American National Standard COBOL X3.23 – 1968. The compiler includes those features required by level 2 of the following modules: nucleus, sequential access, random access, sort, segmentation, and library; level 3 implementation is provided for the table handling module. An extension to the UNIVAC 90/70 COBOL compiler assists in the conversion of programs written for the IBM System/360 DOS COBOL.

6.6.2. FORTRAN

FORTRAN is a 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 appliations.

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 designed so that the user can express a solution in a way which is natural to the problem. The user need not consider the particular characteristics of the system on which the program is executed.

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.

The UNIVAC 90/70 FORTRAN compiler conforms to the specifications of American National Standard FORTRAN X3.9 – 1966. Programs written for the UNIVAC 9400 System can be compiled on the UNIVAC 90/70 System. This compiler is also source code compatible with IBM System/360 and System/370 FORTRAN IV.

6.6.3. Assembler

The symbolic language of the UNIVAC 90/70 assembler is a versatile and detailed language. It includes a sophisticated variety of operators which allow the fabrication of desired storage address fields based on information generated at assembly time. The instruction codes are assigned mnemonics which represent the hardware function in each instruction. The instruction repertoire includes all of the nonprivileged instructions of the IBM System/360 Model 50.

A utility will be provided so that the UNIVAC 90/70 assembler can assemble programs written for the UNIVAC 9400 assembler, including procedure (PROC) definitions. Source code programs written for the IBM System/360 DOS assembler can be assembled directly by the UNIVAC 90/70 assembler. User macro definitions written for the IBM System/360 DOS assembler are accepted by the UNIVAC 90/70 assembler.

UNIVAC OS/7 and IBM System/360 DOS provide the user with a comprehensive selection of system macro instructions which interface with data management, the supervisor, and other elements of the operating system. Many codes are the same; however, the code generated may vary. Programs in which the user is dependent on the generated code may not produce the desired results.

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 a mnemonic instruction code or the name of a macro instruction.

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

Combining names, parentheses, arithmetic operators, logical operators, relational operators, and self-defining terms into operand expressions makes possible the solution of high sophisticated coding problems. Operand expressions gain power by being able to include location counter references.

A wide range of data types is provided for constant generation and storage definition. Binary, hexadecimal, decimal, fixed-point, floating-point, and character formats may be used to specify absolute values in the source code.

Output from the assembler run consists of a complete listing of symbolic coding and generated object coding diagnostic messages. A relocatable object module is produced which is suitable for linking to other modules prior to loading for subsequent execution.

6.6.4. Report Program Generator (RPG)

The UNIVAC 90/70 Report Program Generator (RPG) is a problem-oriented language designed to provide a convenient method of obtaining report programs with full utilization of the tape and disc features of the system. The RPG system can:

- obtain records from single or multiple input fields
- write reports
- perform calculations on data taken from input records or RPG-literals
- use table looking to search a table contained in storage
- exit to a user's subroutine written in a language other than RPG
- branch within calculations
- sequence check input records
- update files
- use tag files for pointers to random access files on discs.

The output object program of the RPG run is relocatable and may be linked as a subprogram to other assembled, compiled, or generated programs.

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

- File Description

  This form is used to assign a unique name to each file associated with an input or output unit used in the program, and to provide certain basic information about the files that are used by the program.

- Input Format Specifications

  This form is used to specify input records within an input file, to define input record sequence, and to describe data field formats and locations within the input record.

- Calculation Specifications

  This form is used to provide necessary information for operations which are to be performed by the object program on input data and upon data obtained as a result of other calculations. These operations include setting and resetting indicators, linking to other programs, table look-up, looping, arithmetic operations, and definition of result fields.

- Output Format Specifications

  This form is used to define the type of outputs which are to be produced on the printer, punch, tape, or disc, and to specify constants and editing desired on output records and reports.

- File Extension Specifications

  This form provides information to RPG about record address, chaining and tag files, and tables used in the object program.

  RPG also provides a table handling capability which enables a programmer to coordinate tables described by file extension specifications.

- Line Counter Specifications

  This form is used to store reports on an intermediate storage device such as tape or disc. These reports can be directly printed from the storage device at the convenience of the user.

The UNIVAC 90/70 RPG accepts UNIVAC 9200, 9300, or 9400 RPG source programs for compilation and execution on the UNIVAC 90/70 System. The UNIVAC 90/70 RPG is also source code compatible with IBM System/360 DOS RPG I for models 30 and 40.

The report program generator produces an object program that may be processed by the linkage editor. The RPG module may be the entire program, a subroutine to another program, or the calling program to a subroutine written separately.

6.7. PROGRAM PROCESSORS

The program processing facilities provided in OS/7 include the linkage editor and the library service routines. A description of each follows.

6.7.1. Linkage Editor

The linkage editor combines the object modules produced as the output of various runs of language processors into one program that can be loaded and executed by the supervisor. The role of the linkage editor in program preparation in shown in Figure 6–9.

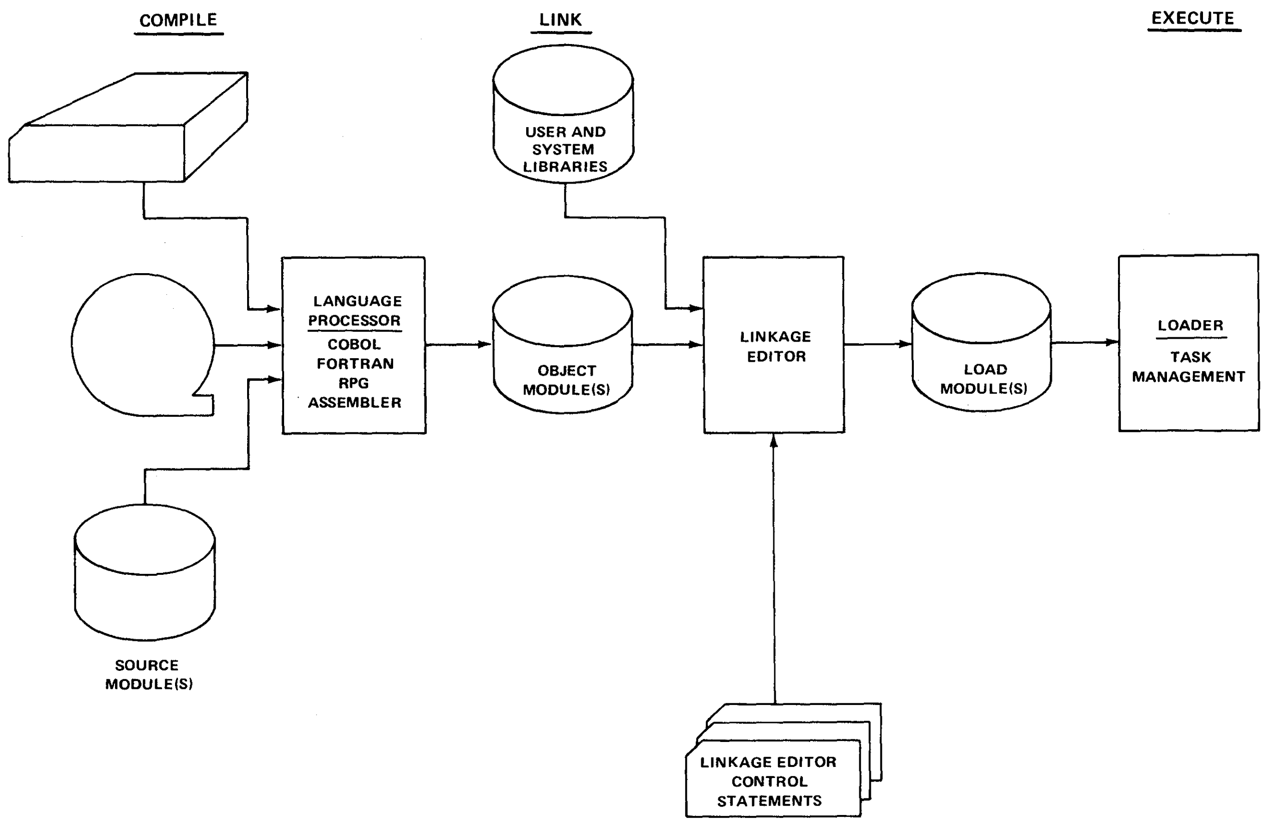


Figure 6–9. Program Preparation

The ability of the linkage editor to construct a single executable load module from several object modules has the following advantages:

- If a change is required in one of the object modules, only the changed object module must be reassembled.
- The various object modules may be written in the appropriate language, such as COBOL, FORTRAN, RPG, or assembly language and combined in a single executable program.
- Routines which are common to two or more object modules may be assembled or compiled only once and the resulting object code linked as needed, thereby reducing the total time required to assemble.

In addition to the linking function, the linkage editor performs the following:

- Searches the appropriate library and incoporates object modules other than those in its primary input, either upon request or automatically.
- Performs program modification by deleting and rearranging control sections of an object module as directed.
- Produces an overlay structure to be used by the supervisor during loading.
- Reserves storage automatically for common storage requests generated by the language processors.

A single execution of the linkage editor combines one or more object modules into a single executable unit referred to as a load module. Some of the object modules being linked may be standard subroutines from the system library. The linkage editor resolves all cross-references among the object modules being linked. It recognizes requests for all common storage and allocates the necessary space. It also analyzes external references to enable it to automatically obtain object modules from the library, for inclusion in the load module it is creating.

The linkage editor can prepare programs with an overlay structure; that is, a load module may consist of more than one phase. A phase is a portion of a load module which can be loaded as an overlay by a single execution of a supervisor macro instruction. The first phase is called the root phase. It is loaded by job control; all other phases are loaded by a previously loaded phase. The name and relative position of the phase within the load module are specified by linkage editor control statements. One or more load modules may be constructed in a single job step.

The linkage editor can also produce a multiphase load module with the necessary system information and system subroutines to load the phases automatically.

Input to the linkage editor consists of control statements and one or more object modules. The control statements direct the construction of a load module from the object modules. The linkage editor control statements that define the load module are contained in the control stream. Object modules to be included in the load module may be disc resident, on magnetic tape, or interspersed among the control statements.

Linkage editor control statements may be placed before, between, or after the control sections in an object module, but may not be placed within a control section.

If the control stream contains no linkage editor control statements defining the load module to be constructed, the linkage editor constructs a single load module from the first object module in the file in which the language processors write their output.

6.7.2. Library Services

A comprehensive system of tape and disc library service routines is a part of OS/7. This system provides for storage of source modules, COBOL and assembler copy, object, load and assembler procedure (PROC) modules. Facilities for adding, copying, deleting, compressing, correcting, creating, and renaming are provided. In addition, print and punch functions are available for all libraries. Support for program libraries on disc or tape is provided in OS/7.

6.8. UTILITY AND SERVICE PROGRAMS

A number of utility and service programs is provided in OS/7 to assist the user in the organization and maintenance of data. These program are described in the following paragraphs.

6.8.1. Sort/Merge

The OS/7 sort/merge is a multiphase, multimodule program which includes the following features:

- An interface that permits disc only, tape only, or tape/disc sorts of large or small volume files.
- A multicycle capability that allows an unlimited volume of data to be sorted automatically or in separate phases.
- The ability to sort either fixed-length or variable-length records.

- The handling of five types of key field formats:
  - character
  - EBCDIC data in ASCII collating sequence
  - binary (signed or unassigned)
  - ASCII numeric (leading and trailing sign)
  - decimal (packed, signed zone, leading sign, trailing sign, leading signed overpunched and trailing signed overpunched)
  - floating-point (single or double precision)
- Up to 255 key fields can be specified.
- Sorting of noncontiguous key fields in ascending or descending sequence, in any combination.
- User specification of the collation sequence.
- Execution of output own code.
- Execution of input own code.
- Record sequencing or data reduction by means of user own code.
- Shared input and reserved output devices.
- Tag sorting. Merging files of previously sequenced records.
- Data checksum facility (via SIAM).
- Rerun/restart facilities for tape and multicycle sorts.

6.8.1.1. Facilities Utilization

For a disc-only sort, the disc storage available must be large enough to contain the entire file of records plus sort control information of approximately 5 or 10%. A maximum of eight standard disc filenames may be assigned.

For a tape-only sort, a minimum of three tape units is required. A maximum of 14 tape units may be assigned. A maximum of six tapes is used by the sort subroutine for string collating. The sort allows a user to execute a multicycle tape sort, using the shared input and reserved output devices.

For a tape/disc sort, the use of disc and tape auxiliary storage in combination assumes a multicycle sort. The minimum amount of disc space required is ten million bytes of storage. The minimum number of tape units which may be assigned is 3, although more may be advisable.

6.8.1.2. User Control

Two versions of the sort/merge program are available:

- The independent sort/merge is an efficient, comprehensive utility operable in a minimum configuration. This particular sort/merge is a freestanding, parameterized processor which is initiated by means of job control. This sort can process data entered from any file which can be read sequentially.

- The sort/merge subroutine is designed to meet the following goals:

  – provide an efficient, comprehensive sort facility

  – provide a sort/merge subroutine callable in assembler language

  – provide a sort/merge subroutine to the 90/70 COBOL compiler

  – provide a sort/merge subroutine as the basic sort for the independent sort/merge

6.8.2. Data Utility

A comprehensive set of data utility programs provides for transferring of data from any input device to any output device. Support is provided for sequential, indexed-sequential, and direct access files. Provisions are made for reblocking, rearranging fields, deleting items and fields, checking or creating card sequence numbers, comparing files and various print options. Routines are provided to simplify the user interface.

6.8.3. Miscellaneous Utilities

Among the programs available as program aids in OS/7 are:

- Disc preparation (prep) for the UNIVAC 8411, 8414, 8424, and 8440 Disc Subsystems
- Disc display
- Tape display
- Tape volume initializaiton (tape prep)
- Clear disc
- Assign alternate track
- Catalog and VTOC displays
- Copy/restore
- Tape compare

6.8.4. Macro Utilities

The utility macro instruciton provides the user with a direct and easy means of generating efficient file- to-file utility programs tailored to the users specific needs. These macros can be combined with problem programs to produce generalized or specific file processing programs. The utility macro set includes:

- INCARD
- INTAPE
- INDISC
- INLOG
- OUTCARD
- OUTTAPE
- OUTDISC
- OUTPRT
- OUTLOG

6.9. DIAGNOSTIC PROGRAMS

The Diagnostic Software Library for the UNIVAC 90/70 System provides the software aids required to facilitate preventive and emergency maintenance by customer engineers. The library is fully compatible with the standard system software library and many of the programs interact with the system supervisor. The system software and diagnostic software are thoroughly coordinated to assure maximum efficiency in the detection and isolation of failures with respect to system availability.

The diagnostic library includes programs necessary for testing in both off line and on line environments. (Off line and on line are defined as environments with respect to user production.) Those diagnostic programs capable of execution as standard worker programs can be executed during normal production. The same programs are used for off line testing when no normal production is in progress; that is, when the system is dedicated to maintenance.

In addition to the on line and off line worker type programs which are dedicated to peripheral subsystem testing, the library contains free standing diagnostics which run independently of the operating system. These programs provide maintenance aids for the central complex which includes the processor, the memory, the console, the selector channel and the multiplexer channel.

6.9.1. Error Logging

During an operating session, recoverable errors and their cause and status are logged by OS/7. These statistics are maintained and are available to the customer engineer. These error statistics serve as a warning of potential marginal conditions with a device or devices.

6.9.2. Test Scheduling

Each test is selected and entered by the operator. Test scheduling is recommended at intervals consistent with user production and device usage.

The test results are used as diagnostic tools by maintenance personnel to determine marginal conditions. Maintenance is then scheduled as needed, according to customer production and usage.

6.9.3. Special Diagnostic Features

A capability is provided to assign unallocated or malfunctioning devices to a diagnostic routine.

Software partitioning allows the oeprating system to deallocate direct access devices when suspected of being unreliable. These deallocated elements can then be assigned to diagnostic tests to determine whether they should be restored to the system or removed for extensive offline testing.

6.10. APPLICATION PROGRAMS

In addition to the standard software package, the programmed support for the UNIVAC 90/60 system includes a number of applications programs. Each of these programs specializes in problems distinctive to a particular type of user. The specific features of these applications programs are given in separately published brochures.

6.11. INFORMATION MANAGEMENT SYSTEM

The UNIVAC 90/70 Information Management System (IMS/90) facilitates access to information stored in data files. (See Figure 6–10.) The IMS/90 provides a terminal-oriented data retrieval and update capability for managerial or clerical personnel. An authorized user can use IMS/90 to retrieve current information from data files, add data to files, and alter data within files. Programmers can use the IMS/90 functions to minimize the effort and time required to implement transaction processing programs.

The IMS/90 is integrated with OS/7 utilizing the message control program and data management. Standard file organizations are supported by IMS/90. A simple easy-to-learn language is provided to quickly train all authorized personnel. IMS/90 allows the user to go online quickly and easily expand communications capabilities as requirements increase.

IMS/90 can process files which are organized under the rules and conventions of OS/7 data management. IMS/90 provides defined record management which enables users to selectively display and/or access data from one or more files by way of one call.

6.11.1. Interactive Use

IMS/90 provides functions that allow authorized users to:

- retrieve and display information from files;
- add, delete, or change information in files, subject to considerations of optional data validation and security;
- request that lists be generated and displayed.

The inquiry/update language is:

- freeform;
- easily used from a visual display terminal or a keyboard/printer device.

Passwords, which are known only to authorized personnel, are used to control access to restricted information. Output displayed by IMS/90 is formatted for ease of interpretation by the user.

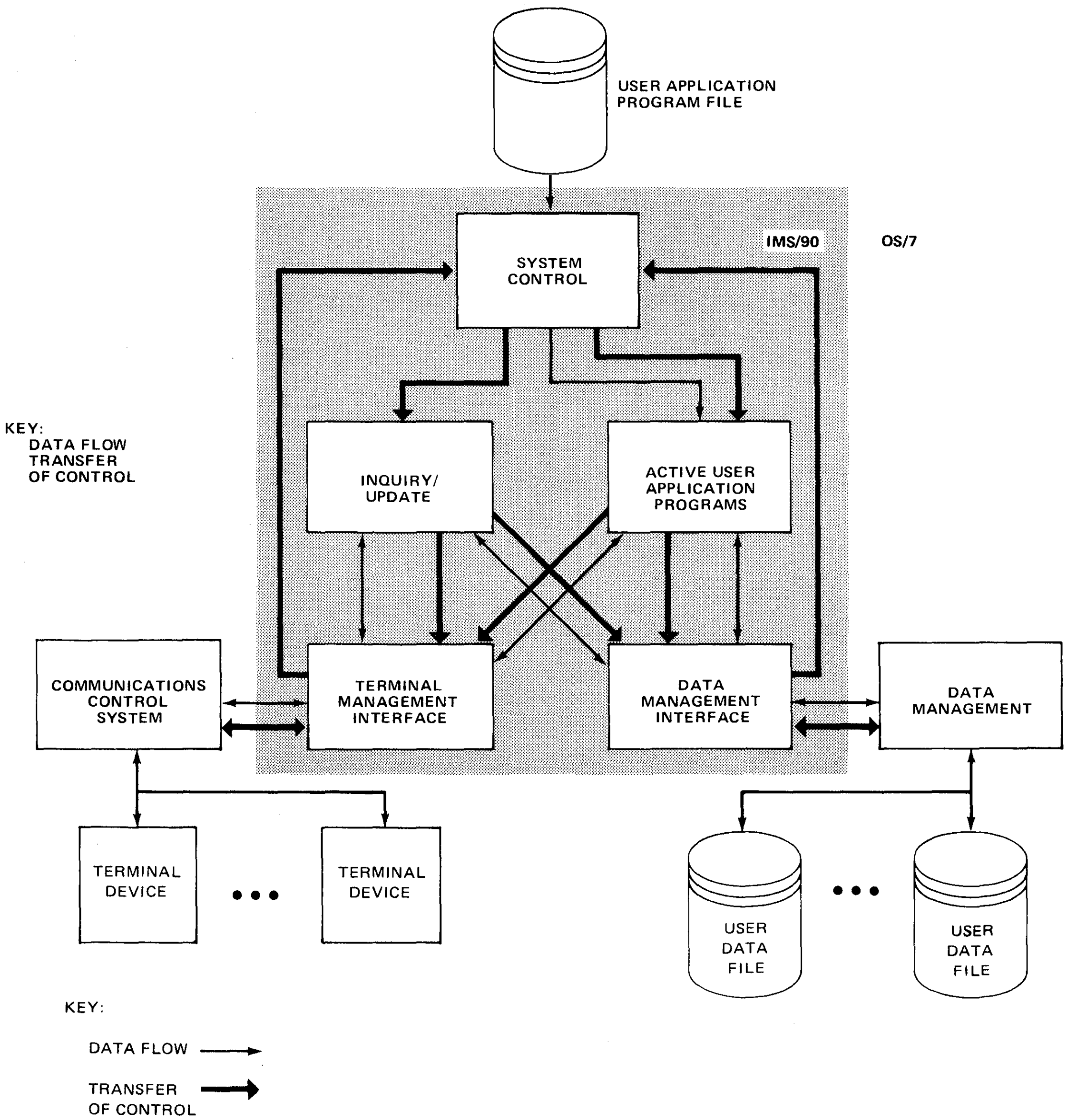


Figure 6–10. OS/7 Information Management System (IMS/90)

6.11.2. Application Program Support

IMS/90 provides functions which make it easier for programmers to write transaction processing programs for applications beyond the scope of the inquiry/update language.

The areas of support are:

- Communications interface
- Application program scheduling
- Data management
- System security
- Recovery of files and messages.

6.12. EMULATION CAPABILITIES

One of the benefits of microprogramming is the capability of making one machine process like another. This allows the 90/70 system to provide emulation capabilities for most of the IBM 360 Series and the IBM 1400 Series. Emulators are also available for the Series 70, TDOS, DOS, and 301 operating environments.

The emulators are an integrated part of the operating environment allowing concurrent operation within the OS/7 operating system. A stand-alone version is offered as well. Two processor emulator features are available allowing one or two separate emulators to be run, depending on the user's requirements.

Unique file and data handling requirements will be enhanced on the 90/70 system. In fact, the user's complete operating environment will be duplicated thus allowing the user to move between the emulated machine and the 90/70 system without change.

6.13. MANAGEMENT CONTROL SYSTEM

The UNIVAC Management Control System (MCS) approaches, in a unique way, the problem of scheduling tasks, costing, and the allocation of resources within complex multiproject contracts. The designation MCS denotes that this system is broader in content than a PERT/TIME or a CPM application package. It does, however, include these functions. MCS performs resource allocation and uses the the results to modify activity schedules, permits alternate description and processing of activity-on-node or activity-on-arrow networks, offers network interfacing multilevel summarization, is consistent with government directives, and complies with current American National Standards Institute (ANSI) specifications.

A typical MCS offers planning and scheduling, cost control, and reporting as major features.

6.13.1. Planning and Scheduling

- Extensive input editing
- Data base file organization to aid in the easy retrieval and updating of data
- Processing traditional PERT/CPM networks in activity-on-arrow or activity-on-node notation
- Multiple start and end events
- Alphanumeric, randomly named event and work item codes
- Assignment of schedule, and acutal dates to both the start and finish of each work item
- Input of percentage completion to date for in-progress activities
- Incorporation of arbitrary nonwork days in the calendar
- Assignment of a work week length, start day, and continuity code to each activity and work item.

6.13.2. Cost Control

- Use and maintenance of resource rate tables
- Cost summarization

- Attachment of resources and cost to activities
- Parallel cost control and reporting based on Work Breakdown Structure (WBS) and Organizational Accounting Structure (OAS)
- Projection of cost plan and resource requirements.

6.13.3. Reports

Upon user's request the following reports will be furnished.

- Activity oriented reports
- Event oriented reports
- Network summarization

Management oriented reports

- Cost summarization
- Project status
- Financial plan and status
- Organization status
- Manpower loading

Resource Scheduling

- Resource status
- Resource plan requirements

6.14. MATHEMATICAL PROGRAMMING SYSTEM

Mathematical Programming System (MPS) is a practical management and business-oriented discipline that is applied to a wide variety of scheduling, allocation, and planning problems. MPS gives decision makers – plant, project and policy managers, as well as industrial engineers – a plan of action. This plan of action guarantees least cost or maximum profit under condition that the problem can be modeled by linear constraints, but under other conditions as well. For this purpose MPS prints a best-operation level for every activity with which the user is concerned; at the same time, it strictly satisfies policy decisions with all interfacing managers.

Mathematical Programming System, MPS 90, contains Linear Programming System capabilities under the call LPS 90. LPS 90 contains the following important features:

- Bounded variables
- Range constraints on rows
- Multiple sets of bounds and ranges

- Multiple objective-function rows
- Multiple right-hand sides
- Use of double-precision arithmetic
- Automatic inversion as required
- Product form of the inverse used
- A flexible control language
- Share and extended 8-character input data formats with additional formats provided to permit input of bounds and ranges
- Extensive capability REVISE
- Save status and restore procedures
- Automatic output

Mathematical Programming System, MPS 90, provides a significant extension of capabilities over the Linear Programming System. Extended capabilities are:

- A state-of-the-art, class driven, matrix generator and report writer
- Mixed Integer Programming, a capacity by which designated activities are allowed to take on only integer values (e.g. site selection, yes-no activities, fixed charge models)
- A large-scale, fast reinversion
- A FORTRAN/COBOL file editing interface

MPS 90 will solve greater than 10,000 row problems in the largest configuration UNIVAC 90/70 System.

Mathematical Programming System, in particular the Linear Programming System, has proven itself cost-effective over a period of decades and is an accepted optimization tool in a wide variety of industries. Use of an input model derived for one industry is a common starting point for preparation of input for another industry or discipline. Generic models are constructed for such applications as production/distribution scheduling, material allocation, ingredient blending, portfolio selection, capital budgeting, plant and warehouse site location, work force allocation, machine loading, and other management areas. MPS solves these difficult problems in a systematic and cost-efficient way and guarantees that the very best solution is obtained. It automatically indicates when alternate, equally good, solutions may be applied. In addition, MPS can demonstrate how profit will vary with changes in capacities or costs. MPS provides:

- Extensive error analysis of all database operations
- Database performance and storage analysis
- Database rollback/recovery facilities to recover integrity of the database following either a software or hardware failure

6.15. DATABASE MANAGEMENT SYSTEM

Database Management System (DMS/90) is a subset of the April 1971 CODASYL Data Base Task Group Language specification. DMS is designed to provide database facilities for ANS COBOL programs oeprating in an OS/7 environment. It provides Data Manipulation Language (DML) statements which are used to store, retrieve and manipulate data stored on an 8411/8414/8424/8440 disc. DML statements may be included anywhere within the COBOL procedure coding of a program. Before compilation, a DML processor validates and converts all imperative DML statements into COBOL CALL statements. COBOL Working Storage is used to establish database record I/O areas and communications between the user and DMS/90.

Several functions provided by DMS/90 are:

- Permits the user to specify a variety of data structures including hierarchical, bill of material, network or a combination
- Provides multiple user-specified entry points into the database
- Provides control over physical placement of records to optimize performance
- Provides dynamic allocation and deallocation of physical storage space
- Eliminates need for designated overflow storage
- Eliminates the need for periodic reorganization of the database

APPENDIX A. UNIVAC 90/70 SYSTEM INSTRUCTIONS

The following is a listing of the UNIVAC 90/70 System instructions by type, showing the mnemonic source code and hexadecimal operation code.

RR TYPE INSTRUCTIONS

DESCRIPTION	MNEMONIC CODE	OPERATION CODE
Add	AR	1A
Add logical	ALR	1E
Add normalized (long format)	ADR	2A
Add normalized (short format)	AER	3A
Add unnormalized (long format)	AWR	2E
Add unnormalized (short format)	AUR	3E
AND	NR	14
Branch and link	BALR	05
Branch on condition	BCR	07
Branch on condition to return external	BCRE	0C
Branch on count	BCTR	06
Compare	CR	19
Compare (long format)	CDR	29
Compare (short format)	CER	39
Compare logical	CLR	15
Divide	DR	1D
Divide (long format)	DDR	2D
Divide (short format)	DER	3D
Exclusive OR	XR	17
Halve (long format)	HDR	24
Halve (short format)	HER	34
Insert storage key (privileged instruction)	ISK	09
Load	LR	18
Load (long format)	LDR	28
Load (short format)	LER	38
Load and test	LTR	12
Load and test (long format)	LTDR	22
Load and test (short format)	LTER	32
Load complement	LCR	13
Load complement (long format)	LCDR	23
Load complement (short format)	LCER	33
Load negative	LNR	11
Load negative (long format)	LNDR	21
Load negative (short format)	LNER	31
Load positive	LPR	10
Load positive (long format)	LPDR	20
Load positive (short format)	LPER	30
Multiply	MR	1C
Multiply (long format)	MDR	2C
Multiply (short format)	MER	3C
OR	OR	16
Set program mask	SPM	04
Set relocation flags	SRF	0B
Set storage key (privileged instruction)	SSK	08
Subtract	SR	1B
Subtract logical	SLR	1F
Subtract normalized (long format)	SDR	2B
Subtract normalized (short format)	SER	3B
Subtract unnormalized (long format)	SWR	2F
Subtract unnormalized (short format)	SUR	3F
Supervisor call	SVC	0A

RS TYPE INSTRUCTIONS

DESCRIPTION	MNEMONIC CODE	OPERATION CODE	DESCRIPTION	MNEMONIC CODE	OPERATION CODE
Branch on index high	BXH	86	Shift right double	SRDA	8E
Branch on index low or equal	BXLE	87	Shift right double logical	SRDL	8C
Load control storage (privileged instruction)	LCS	B1	Shift right single	SRA	8A
Load multiple	LM	98	Shift right single logical	SRL	88
Shift left double	SLDA	8F	Store multiple	STM	90
Shift left double logical	SLDL	8D	Supervisor load ~~multiple~~ (privileged instruction)	SLM	B8
Shift left single	SLA	8B	Supervisor store multiple (privileged instruction)	SSTM	B0
Shift left single logical	SLL	89			

RX TYPE INSTRUCTIONS

DESCRIPTION	MNEMONIC CODE	OPERATION CODE	DESCRIPTION	MNEMONIC CODE	OPERATION CODE
Add	A	5A	Insert character	IC	43
Add half word	AH	4A	Load	L	58
Add logical	AL	5E	Load (long format)	LD	68
Add normalized (long format)	AD	6A	Load (short format)	LE	78
Add normalized (short format)	AE	7A	Load address	LA	41
Add unnormalized (long format)	AW	6E	Load half word	LH	48
Add unnormalized (short format)	AU	7E	Multiply	M	5C
AND	N	54	Multiply (long format)	MD	6C
Branch and link	BAL	45	Multiply (short format)	ME	7C
Branch and link external	BALE	4D	Multiply half word	MH	4C
Branch on condition	BC	47	OR	O	56
Branch on count	BCT	46	Store	ST	50
Compare	C	59	Store (long format)	STD	60
Compare halfword	CH	49	Store (short format)	STE	70
Compare (long format)	CD	69	Store character	STC	42
Compare (short format)	CE	79	Store halfword	STH	40
Compare logical	CL	55	Subtract	S	5B
Convert to binary	CVB	4F	Subtract half word	SH	4B
Convert to decimal	CVD	4E	Subtract logical	SL	5F
Divide	D	5D	Subtract normalized (long format)	SD	6B
Divide (long format)	DD	6D	Subtract normalized (short format)	SE	7B
Divide (short format)	DE	7D	Subtract unnormalized (long format)	SW	6F
Exclusive OR	X	57	Subtract unnormalized (short format)	SU	7F
Execute	EX	44			

SI TYPE INSTRUCTIONS

DESCRIPTION	MNEMONIC CODE	OPERATION CODE	DESCRIPTION	MNEMONIC CODE	OPERATION CODE
Add immediate	AI	9A	Read direct (privileged instruction)	RDD	85
AND	NI	94	Set system mask (privileged instruction)	SSM	80
Compare logical	CLI	95	Start I/O (privileged instruction)	SIO	9C
Diagnose (privileged instruction)	DIAG	83	Store channel register (privileged instruction)	SCHR	AC
Exclusive OR	XI	97	Test and set	TS	93
Halt and proceed (privileged instruction)	HPR	99	Test channel (privileged instruction)	TCH	9F
Halt I/O (privileged instruction)	HIO	9E	Test I/O (privileged instruction)	TIO	9D
Load channel register (privileged instruction)	LCHR	AD	Test under mask	TM	91
Load program status word (privileged instruction)	LPSW	82	Write direct (privileged instruction)	WRD	84
Move	MVI	92			
OR	OI	96			

SS TYPE INSTRUCTIONS

DESCRIPTION	MNEMONIC CODE	OPERATION CODE	DESCRIPTION	MNEMONIC CODE	OPERATION CODE
Add decimal	AP	FA	Move numerics	MVN	D1
AND	NC	D4	Move with offset	MVO	F1
Compare decimal	CP	F9	Multiply decimal	MP	FC
Compare logical	CLC	D5	OR	OC	D6
Divide decimal	DP	FD	Pack	PACK	F2
Edit	ED	DE	Subtract decimal	SP	FB
Edit and mark	EDMK	DF	Translate	TR	DC
Exclusive OR	XC	D7	Translate and test	TRT	DD
Move	MVC	D2	Unpack	UNPK	F3
Move zones	MVZ	D3	Zero and add	ZAP	F8

SPECIAL INSTRUCTION

DESCRIPTION	MNEMONIC CODE	OPERATION CODE
Emulation aid	EA	EA

APPENDIX B. I/O CHANNEL ASSIGNMENT

INPUT/OUTPUT EQUIPMENT	PROCESSOR CHANNEL TYPE ASSIGNMENT	NUMBER OF PHYSICAL CONNECTIONS REQUIRED	NUMBER OF SUBCHANNEL ADDRESSES REQUIRED
UNIVAC 8411 Disc Subsystem	Selector	1	1
UNIVAC 8414 Disc Subsystem	Selector	1	1
UNIVAC 8424 Disc Subsystem	Selector	1	1
UNIVAC 8440 Disc Subsystem	Selector	1	1
UNISERVO VI-C Magnetic Tape Subsystem	Multiplexer	1	1
UNISERVO 12/16 Magnetic Tape Subsystem	Selector	1	1
UNISERVO 12/16/20 Magnetic Tape Subsystem	Selector	1	1
UNIVAC Printer Subsystem	Multiplexer	1	1
UNIVAC Card Punch Subsystem	Multiplexer	1	1
UNIVAC Paper Tape Subsystem	Multiplexer	1	1
UNIVAC 2703 Optical Document Reader	Multiplexer	1	1
UNIVAC 9200/9300 System (9000 Series Channel Adapter)	Either Type	1	1
UNIVAC 9400 System (9000 Series Channel Adapter)	Either Type	1	1
UNIVAC 90/60 System (Series 90 Channel Adapter)	Either Type	1	1
UNIVAC 90/70 System (9000 Series Channel Adapter)	Either Type	1	1
UNIVAC DCS-1 or DCS-1C	Multiplexer	1	2
UNIVAC DCS-4	Multiplexer	1	up to 8
UNIVAC DCS-16	Multiplexer	1	up to 28
UNIVAC 90/70 System Console	Multiplexer	1	1
UNIVAC Card Reader Subsystem	Multiplexer	1	1

APPENDIX C. ASCII AND EBCDIC CODE CHARTS

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

AMERICAN NATIONAL STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII) X3.4 – 1968, MODIFIED TO EIGHT BITS

BIT POSITIONS 4, 3, 2, 1	BIT POSITIONS 7, 6, X, 5															
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NUL	DLE			space	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			$	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	NUL	12, 0, 9, 8, 1		11, 0, 9, 8, 1	SP	No Punches	– (Minus)	11	0000
0001	SOH	12, 9, 1		0, 9, 1		12, 0, 9, 1	\	0, 1	0001
0010	STX	12, 9, 2		0, 9, 2		12, 0, 9, 2		11, 0, 9, 2	0010
0011	ETX	12, 9, 3		0, 9, 3		12, 0, 9, 3		11, 0, 9, 3	0011
0100		12, 9, 4		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		12, 9, 6	ETB	0, 9, 6		12, 0, 9, 6		11, 0, 9, 6	0110
0111	DEL	12, 9, 7	ESC	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		0, 9, 8, 2	[	12, 8, 2	¦ (Vert. bar)	12, 11	1010
1011	VT	12, 9, 8, 3		0, 9, 8, 3	. (Period)	12, 8, 3	, (Comma)	0, 8, 3	1011
1100	FF	12, 9, 8, 4		0, 9, 8, 4	<	12, 8, 4	%	0, 8, 4	1100
1101	CR	12, 9, 8, 5	ENQ	0, 9, 8, 5	(	12, 8, 5	_ (Undersc.)	0, 8, 5	1101
1110	SO	12, 9, 8, 6	ACK	0, 9, 8, 6	+	12, 8, 6	>	0, 8, 6	1110
1111	SI	12, 9, 8, 7	BEL	0, 9, 8, 7	!	12, 8, 7	?	0, 8, 7	1111
	0 0 0 1		0 0 1 1		0 1 0 1		0 1 1 1		
0000	DLE	12, 11, 9, 8, 1		12, 11, 0, 9, 8, 1	&	12		12, 11, 0	0000
0001	DC1	11, 9, 1		9, 1		12, 11, 9, 1		12, 11, 0, 9, 1	0001
0010	DC2	11, 9, 2	SYN	9, 2		12, 11, 9, 2		12, 11, 0, 9, 2	0010
0011	DC3	11, 9, 3		9, 3		12, 11, 9, 3		12, 11, 0, 9, 3	0011
0100		11, 9, 4		9, 4		12, 11, 9, 4		12, 11, 0, 9, 4	0100
0101		11, 9, 5		9, 5		12, 11, 9, 5		12, 11, 0, 9, 5	0101
0110	BS	11, 9, 6		9, 6		12, 11, 9, 6		12, 11, 0, 9, 6	0110
0111		11, 9, 7	EOT	9, 7		12, 11, 9, 7		12, 11, 0, 9, 7	0111
1000	CAN	11, 9, 8		9, 8		12, 11, 9, 8		12, 11, 0, 9, 8	1000
1001	EM	11, 9, 8, 1		9, 8, 1		11, 8, 1	′ (Grave)	8, 1	1001
1010		11, 9, 8, 2		9, 8, 2	]	11, 8, 2	: (Colon)	8, 2	1010
1011		11, 9, 8, 3		9, 8, 3	$	11, 8, 3	#	8, 3	1011
1100	FS	11, 9, 8, 4	DC4	9, 8, 4	*	11, 8, 4	@	8, 4	1100
1101	GS	11, 9, 8, 5	NAK	9, 8, 5	)	11, 8, 5	' (Prime)	8, 5	1101
1110	RS	11, 9, 8, 6		9, 8, 6	; (Semicolon)	11, 8, 6	= (Equals)	8, 6	1110
1111	US	11, 9, 8, 7	SUB	9, 8, 7	∧	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		0 1 2 3 1 0 1 0		0 1 2 3 1 1 0 0		0 1 2 3 1 1 1 0		4 5 6 7
	SYMBOL	80 COL. CD. CODE	SYMBOL	80 COL. CD. CODE	SYMBOL	80 COL. CD. CODE	SYMBOL	80 COL. CD. CODE	
0000		12, 0, 8, 1		11, 0, 8, 1	{	12, 0	/ (slash)	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	}	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	EO	12, 11, 0, 9, 8, 7	1111

NOTE: Lowercase letters are industry standard and printable only on the UNIVAC 0768-02/03 Printer.