UNIVAC®

418-III

REAL-TIME SYSTEM

SYSTEM DESCRIPTION

This manual describes a UNIVAC system. It is as complete and accurate at the time of publication as is feasible by current documentation techniques. The Univac Division will issue complete revisions of this manual when necessary. The Univac Division reserves the right to make such additions, corrections, and/or deletions as, in the judgment of the Univac Division, are required by the development of its systems. To assure that you have the current version of this manual and for the current status of the system, contact your local Univac Representative.

UNIVAC is a registered trademark of the Sperry Rand Corporation.

Other trademarks appearing in the text of this publication are:

UNISERVO
FASTRAND
UNISCOPE

PRINTED IN U.S.A.

CONTENTS

APPENDIXES

FIGURES

1. INTRODUCTION

1.1. UNIVAC 418-III REAL-TIME SYSTEM

The UNIVAC 418-III Real-Time System is a medium scale computer system designed to provide complete capability in the three major types of general purpose data processing -- batch, scientific, and real time. Its modular structure implements the latest advances in computer design, system organization, and programming technology. The UNIVAC 418-III Real-Time System offers the medium scale user multiprogramming capability and a real time operating system featuring an executive, a system's support library, and language processors.

1.2. MAJOR TYPES OF DATA PROCESSING APPLICATIONS

The early use of general purpose computers to perform batch, scientific,or real time data processing was accomplished on a computer which, while technically classified as general purpose, was indeed highly specialized. Originally, the only applications were scientific and commercial. Because effective data transmission was unavailable, the early commercial computers were used only for batch processing. With the advent of computer controlled data communication lines, applications were separated into batch, real time, and scientific.

The specialization of the general purpose computer for the requirements of commercial batch data processing include the simultaneous reading and writing of high speed magnetic tapes, a broad data path through the input/output channels, high speed storage, and decimal arithmetic capability.

The specialization of the general purpose computer for the requirements of real time data processing include sophisticated interrupt systems, comprehensive data communication subsystems, discrete buffering of input/output for each communication line, many types of direct access storage devices, and the capability of handling a mixed variety of character coding.

The specialization of the general purpose computer for the requirements of scientific data processing include high speed fixed-point and floating-point arithmetic, and binary word-oriented high speed storage.

Advances in computer technology, particularly in the area of real time capability, have made it possible to produce general purpose computers which meet the requirements of two of the three types of data processing, but seldom all three. Many of the computers available today still have this limitation. *The UNIVAC 418-III System offers its user all three types of data processing capabilities.*

1.3. SYSTEM AND PERIPHERAL CHARACTERISTICS

The UNIVAC 418-III System provides: up to 131,072 eighteen-bit words of 750 nanosecond cycle core storage; up to 2.66 million word-per-second input/output transfer rate; storage protection; externally specified indexing (discrete communication line buffering); a large instruction repertoire; and automatic tabling of communications interrupts.

The peripheral subsystems of the UNIVAC 418-III System include: magnetic tape with from 34,160 to 96,000 frames-per-second transfer rate with dual synchronizer available for simultaneous reading and writing; FH-432/1782 magnetic drums with 1.44 million character-per-second transfer rate and 4.25 ms and 17 ms average access time respectively; FASTRAND II Mass Storage with up to 1,056,964,608 characters of storage per subsystem; FASTRAND III Mass Storage with up to 1,585,446,912 characters of storage per subsystem; and UNIVAC 9000 Series computer subsystems for online and offline card reading, card punching, and printing as well as a high speed printer subsystem with a printing speed of up to 1600 lines per minute.

The communications subsystem for the UNIVAC 418-III System is the Communications Terminal Module Controller. The Controller and its modules have been enhanced to provide additional communication-oriented functions. These functions are included in order to relieve the Command/Arithmetic Section and the currently running program of the burden of bookkeeping previously associated with computer controlled communications. Among these functions are character and message parity encoding and checking, and End-of-Message recognition. The Communications Terminal Modules offer a full range of line speed, character size, and mode capability. The Communication Terminal Module Controller provides for termination of up to 32 duplex lines. Multiple controllers can be connected to the UNIVAC 418-III System.

While the UNIVAC 418-III System is in the medium price class, it can readily be seen that it provides large scale capabilities. The size and speed of the main storage section and the versatility of the Command/Arithmetic Unit assures complete capability for handling the most demanding requirements. Together with the full range of peripheral subsystems, the system provides a virtually unlimited range of applications capability. The UNIVAC 418-III System is equally capable of handling batch, scientific, or real time processing, and it adjusts dynamically to any one or a mixture of these environments. All system operations are coordinated and controlled by a versatile executive system having full real time and multiprogramming capabilities.

2. SYSTEM HARDWARE

CHARACTERISTICS

- Up to 131,072 eighteen-bit words
- 750 nanoseconds cycle time
- Up to 2.66 million words/second data transfer rate
- Write storage protection in 256 word increments
- Externally specified indexing (discrete communication line buffering) in full or halfword mode
- 108 standard operation codes
- Floating-point arithmetic option
- Decimal-to-binary and binary-to-decimal conversion options
- Twelve classes of interrupts
- Automatic tabling of communication buffer termination interrupt (ESI)

2.1. SYSTEM COMPONENTS

The UNIVAC 418-III System is constructed of six individual component classifications. The various components (or modules) of the UNIVAC 418-III System are presented in this manual under the following headings:

- Command/Arithmetic Section
- Input/Output Modules
- Main Storage
- Auxiliary Storage Subsystems
- Peripheral Subsystems
- Communications Subsystem

2.1.1. Command/Arithmetic Section

The Command/Arithmetic Section of the UNIVAC 418-III System contains all the arithmetic and control circuitry required to perform each instruction in the UNIVAC 418-III repertoire. The Command/Arithmetic Section performs all instruction decoding and execution sequences. Fixed-point addition and subtraction operations are performed on a single-word or double-word basis. Multiplication produces a double-word product. Division produces a single-word quotient. Double-word addition and subtraction operations are standard. The ability to perform floating-point arithmetic operations is available as an optional feature and provides double-word capability. Another optional feature provides decimal-to-binary and binary-to-decimal conversion.

2.1.2. Input/Output Modules

Operation of the Input/Output Modules (IOM) is independent of the Command/Arithmetic Section; these modules control input and output data transfers. One or two IOM's may be used in each system. Each IOM contains a set of high speed Integrated General Registers (IGR) which are used as buffer control words. Each IOM contains from eight to sixteen 18-bit input/output channels. Depending on the peripheral subsystem, these channels can be used singly or in pairs in order to provide a broader data path. Each IOM operates independently of the other giving two-way simultaneity of input/output and storage with an accumulative capacity of 1.33 million 18-bit word transfers per second per module. Main storage provides separate data paths to and from each I/O Module. In previous systems, all the peripheral devices competed with each other for the use of main storage cycles. The simultaneity achieved by the modular design of the UNIVAC 418-III System and the speed of each module makes it possible to handle high speed peripheral devices with no reduction in overall system performance.

2.1.3. Main Storage

The UNIVAC 418-III System provides a main storage expandable in 16,384 eighteen-bit word increments from a minimum of 32,768 words to a maximum of 131,072 words.

The main storage read/restore cycle time is 750 nanoseconds. A parity check is made on every storage cycle to ensure accuracy of operation. Up to four 32,768 word banks of storage are provided. Each bank has an access path for IOM #0, IOM #1, and the Command/Arithmetic Unit. Thus, three-way simultaneity can be accomplished to main storage. In previous systems, the input/output section and the Command/Arithmetic Section competed with each other for main storage cycles. Program execution was delayed for every input or output data transfer cycle. The speed of the main storage of the UNIVAC 418-III System and the separate data paths for the IOM's and the Command/Arithmetic Unit makes possible truly simultaneous operation with no delays in program execution.

2.1.4. Auxiliary Storage Subsystems

Up to eight FH-432 and/or FH-1782 magnetic drums may be attached to a single control unit. Each FH-432 drum can store 524,288 eighteen-bit computer words plus parity (1,572,864 alphanumeric characters). Each FH-1782 drum can store 4,194,304 eighteen-bit computer words plus parity (12,582,912 alphanumeric characters). Both types of drum transfer data at up to 1.44 million characters per second. The average access time of the FH-432 is 4.25 milliseconds and that of the FH-1782 is 17 milliseconds.

Up to eight FH-880 magnetic drums may be attached to a control unit. Each FH-880 drum can store 1,572,864 eighteen-bit computer words plus parity (4,718,592 alphanumeric characters) with a transfer rate of up to 360,000 characters per second. The average access time of the FH-880 is 17 milliseconds.

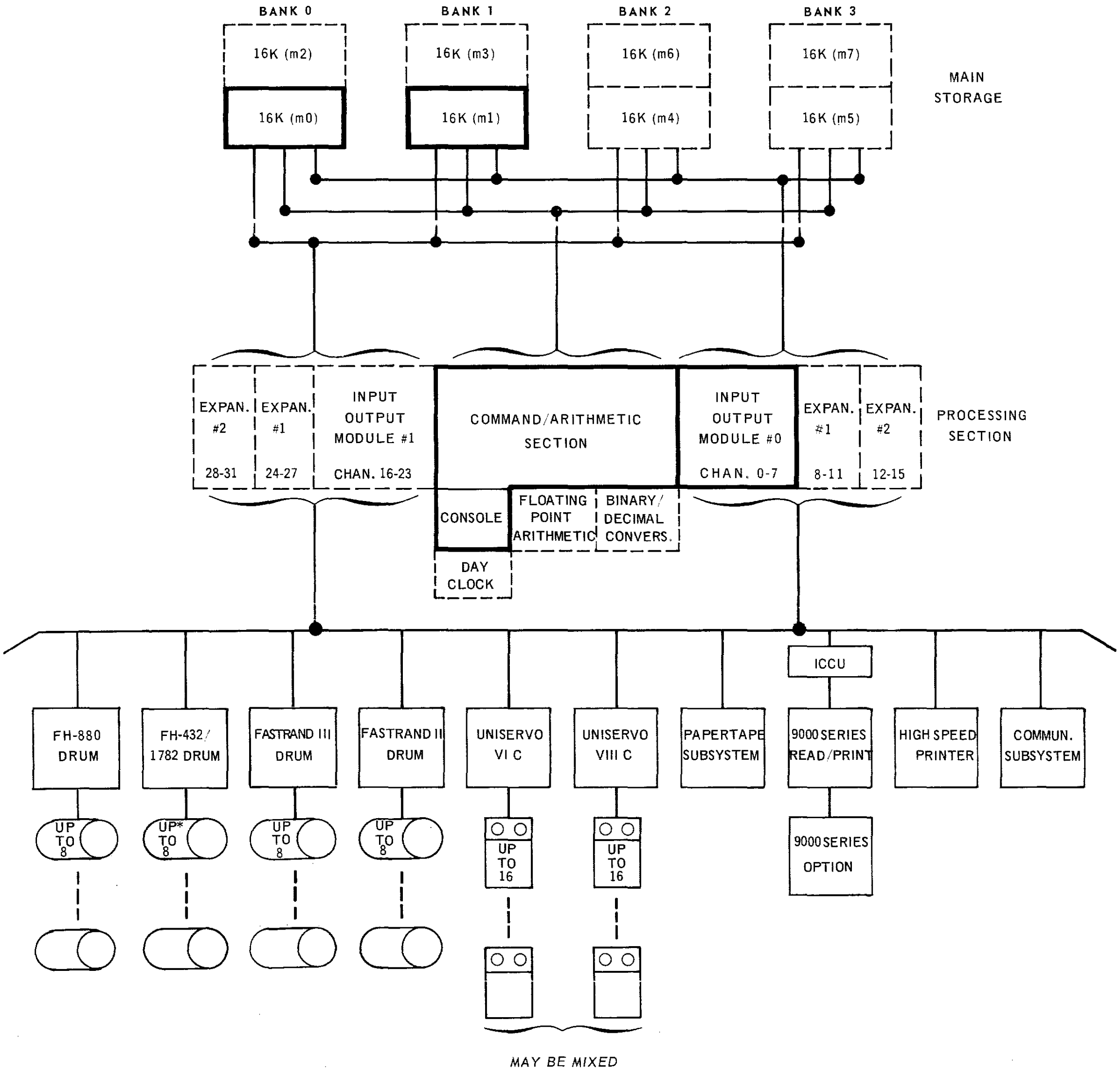
2.2. SYSTEM CONFIGURATION

Figure 2-1 shows a central site configuration for the UNIVAC 418-III System. The lower portion illustrates the standard peripheral subsystems which can be connected to the system. These peripheral subsystems are described in more detail in Sections 6 and 7 of this manual.

The upper portion of Figure 2-1 illustrates the components which constitute the equivalent of what, in the past, has been called the central processor unit. These components are the main storage section (described in Section 3), the Input/Output Modules (described in Section 4), and the Command/Arithmetic Section (described in Section 5).

The minimum set of these components: the Command/Arithmetic Section with the Console, one IOM with eight input/output channels, and 32,768 words of main storage, is shown inside the heavy lines. The Command/Arithmetic Unit may be expanded to include floating-point arithmetic, binary/decimal conversion instructions, and the Day Clock. The input/output to the UNIVAC 418-III System may be expanded by adding the expansion channels to IOM #0 and/or by the addition of a second IOM, or both. The addition of IOM #1 provides for three main storage referencing units operating on an independent basis.

Main storage may be expanded to 65,536 words horizontally, 16K (module (m)0), then 16K (m1), then 16K (m4), then 16K (m5), or vertically, 16K (m0), then 16K (m1), then 16K (m2), then 16K (m3). The expansion beyond 65,536 words depends on the manner in which the first 65,536 words were reached. The capability for three-way simultaneous and independent referencing of main storage is provided by the addition of IOM #1, and at least 49K in three banks of storage.



*FH 432 and 1782 drums may be mixed.

K = 1024 words

Minimum system in heavy lines; optional in broken lines

Figure 2–1. UNIVAC 418-III System Configuration

3. MAIN STORAGE

3.1. GENERAL

Main storage of the UNIVAC 418-III System is a high performance, immediate access repository for instructions, data, and input/output communications areas. Its design and construction fully supports the concepts of multiprogramming, modularity, and system integrity. Among the characteristics of the UNIVAC 418-III main storage are:

- 750 nanosecond read/restore cycle time
- 32,768 to 131,072 eighteen-bit words
- Parity checking on all storage references
- Three-way simultaneous access by two Input/Output Modules and the Command/ Arithmetic Section
- Modular expansion -- four banks with one or two 16,384 word modules per bank
- Hardware storage protection -- write lockout boundaries establishable in 256 word increments

While these features are all discussed generally as storage features, some of them such as three-way simultaneity and storage protection are functions of the entire system. The main storage, through proper organization, becomes a series of allocatable components of the system in the same manner as the peripheral devices. In realizing this objective, some departures from the traditional close integration of the processor and the storage element have taken place:

- The main storage is comprised of independently accessible banks, yet it presents a continuous addressing structure to the Command/Arithmetic Section.
- In order to service one or two Input/Output Modules and the Command/Arithmetic Section, a method of establishing relative priority between them at each bank is provided in the case that two or three of these referencing units attempt to reference the same bank simultaneously.

To assure that the requesting components will wait for storage access, communication between the bank and the component is done on a request/acknowledge basis.

3.2. BASIC STORAGE MODULE

The basic storage module is composed of 16,384 words of ferrite cores. Each word is 18 bits in length and carries two additional parity bits in nonaddressable levels, one bit for each halfword. One or two storage modules may be assigned to a bank of storage. The physical components of each bank are:

- An 18-bit read/restore register
- Parity checking circuits
- Request/acknowledge circuits
- Maintenance switches allowing each bank to be removed logically for servicing or testing

3.3. BANK PARITY ERROR

Parity is checked when reading or calculated when writing on each storage access. If a parity error is detected, the bank issues a parity error interrupt signal to the Command/Arithmetic Unit. Thus, in addition to the proven reliability of sub-microsecond Univac core memory, the main storage of UNIVAC 418-III System has parity checking for storage reference validation.

3.4. MULTIPLE BANK ACCESS

Each storage bank has three connection paths for storage references. Two or three of the referencing units (one or two Input/Output Modules and the Command/Arithmetic Unit) may attempt simultaneous reference to the same storage bank. The path with the highest priority is serviced first, then the second highest is serviced, and then the third highest is serviced. Should another request occur before the lowest priority path is serviced, that request waits until the lower priority path(s) have been serviced. This sequence prevents any one of the referencing units from monopolizing the use of the storage bank.

3.5. STORAGE PACKAGING

The first 16,384 words of both Bank 0 and Bank 1 and the power supplies for each bank are packaged in a Storage Cabinet. A Storage Expansion Cabinet is required to house the second 16,384 words for Banks 0(m2) and 1(m3). The same packaging arrangement is made for Banks 2 and 3.

3.6. STORAGE CONFIGURATIONS

Figure 3–1 illustrates the maximum storage unit and referencing unit configuration. The minimum configuration is the Command/Arithmetic (C/A) Unit, IOM #0 and 32K of storage in Bank 0(m0) and Bank 1(m1). This provides two-way simultaniety with 16K in each bank. The capacity of each bank may then be expanded by an additional 16K.

Three-way simultaneity capability is achieved by adding Bank 2 and IOM #1. The addition of Bank 3 provides greater flexibility of storage allocation to ensure as much as possible the achievement of three-way simultaneity.

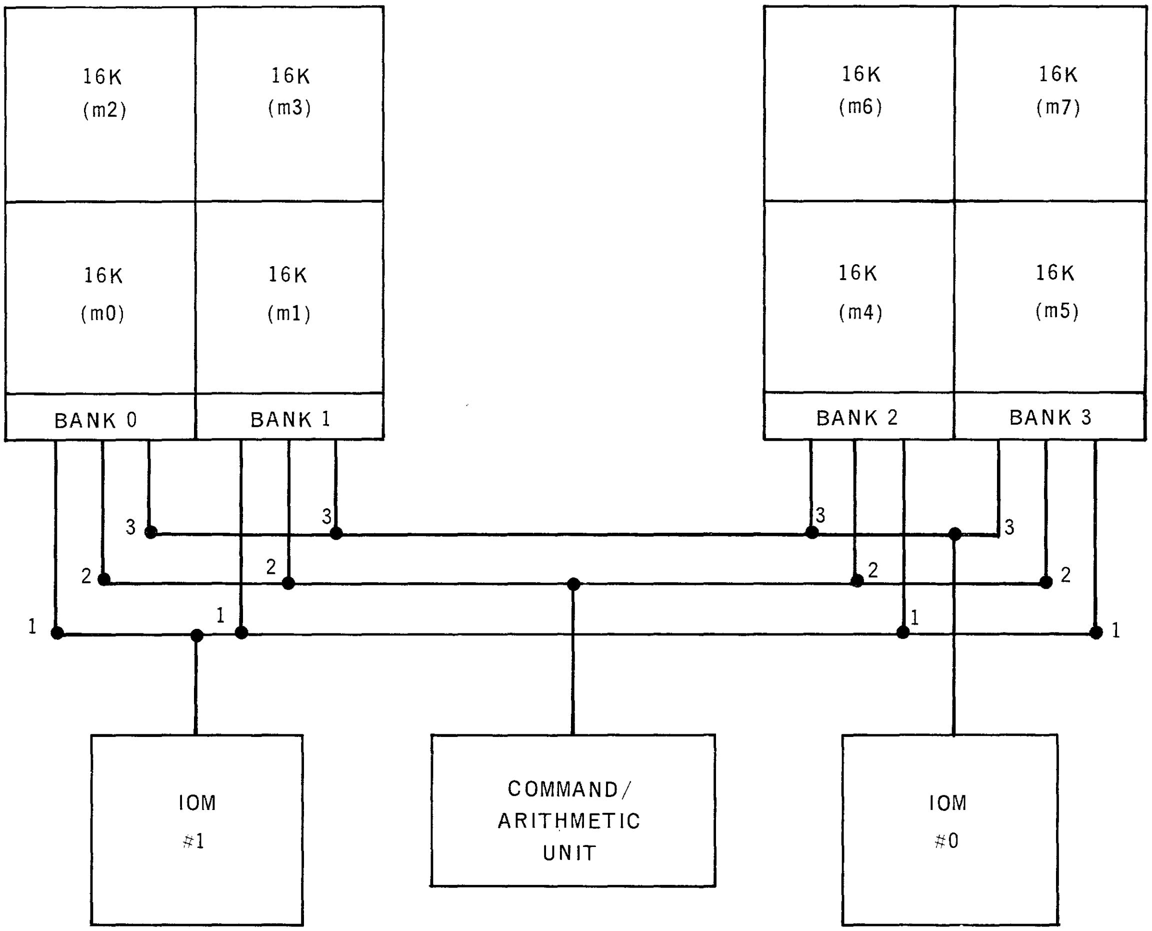


Figure 3–1. Storage Unit Assignment in a Maximum Storage Configuration

The modular addressability of main storage allows the user to expand his memory requirements either by adding banks to provide more simultaneity or expanding within a bank to provide more storage capacity, or both.

The main storage of the UNIVAC 418-III System is constructed so that the banks of storage have a physical unit/logical unit relationship. This is accomplished by a series of printed circuit cards mounted in each storage cabinet. The printed circuit cards are used to assign the bank number to each of the memory modules. Referring to Figure 3–2, the normal assignment of addresses with a full 131,072 words of storage would be:

ADDRESSES	BANK	MODULE
0-16383	0	m0
16384-32767	1	m1
32768-49151	0	m2
49152-65535	1	m3
65536-81919	2	m4
81920-98303	3	m5
98304-114687	2	m6
114688-131072	3	m7

Figure 3–2. Address Assignment

Three-way simultaneity is achieved with three banks of storage and three referencing units. This level of performance is normally achieved when the fifth storage module (m4) is acquired. However, if this increase in system performance is desired, the printed circuit cards make it possible for the user to elect to have three-way simultaneity with as little as 49,252 words of storage and three referencing units. The schematic for this configuration would show the first 16K (m0) in Bank 0; the second 16K in Bank 1(m1),and the third 16K in Bank 2(m4). The first and second 16K (m0 and m1) are housed in one storage cabinet, and the third 16K (m4) occupies half of another storage cabinet. The increase to 65,536 words for this configuration would be accomplished by adding a fourth 16K (m5) which would be housed in the other half of the second storage cabinet. Expansion of this configuration beyond 65,536 words requires storage expansion cabinets, with space for two 16K modules each.

3.7. SIMULTANEITY OF STORAGE ACCESS

Computer systems in which the main storage is an integrated component of the central processing unit utilize a storage referencing technique called Main Store Interrupt. Such systems have a single Storage Address Register. The purpose of the Storage Address Register is to precisely define the coordinates for the storage location to be referenced. With only a single Storage Address Register, the storage locations referenced by the input/output section and the Command/Arithmetic Section have to time-share the storage unit. Under Main Store Interrupt, the input/output section is given priority control of the Storage Address Register. When the input/output section requires a storage reference, the access and execution of instructions is delayed for the data transfer. Therefore, if more than one high speed I/O device is in operation at the same time, the program processing may be delayed (interlocked) until the rate of I/O transfer slackens. Most computer systems utilize the Main Store Interrupt technique.

In the UNIVAC 418-III System, each Input/Output Module (IOM) as well as the Command/Arithmetic Unit contains a Storage Address Register. When any one of these devices requires a reference to main storage, it presents a request signal to the appropriate storage bank. Bank selection is determined by the two high order bits of the Storage Address Register. When the request is granted by the storage bank, the low order 15 bits of the Storage Address Register are used to control the access of the specific storage location. Thus, if two (or three) of these devices make simultaneous requests of different storage banks, each storage bank cycles simultaneously with and independently of the others.

On this basis, one or two I/O data transfers occur between input/output devices and main storage at the same time that the Command/Arithmetic Unit is accessing an instruction or data. In a multiprogramming environment, the execution time of one program and the input/output time of one or more other programs can now be overlapped to provide simultaneous operation of several programs.

3.8. STORAGE PROTECTION

To prevent inadvertent program changes to out-of-range storage addresses, the UNIVAC 418-III System provides the Guard Limits Register for hardware storage protection. The Guard Limits Register contains the upper boundary limit and a lower boundary limit which is loaded by the Executive to establish allowable operating areas for each program. Before giving control to a particular program, the Executive loads the Guard Limits Register with the boundaries which were assigned to this program at load time and then sets Guard Mode. Storage protection is assigned in increments of 256 words of storage.

Before each change of storage by the Command/Arithmetic Unit, a limits check is performed on the address to ensure that it falls inclusively within the upper and lower limit portions of the Guard Limits Register. Should the address to be referenced *not* fall within those limits and the Guard Mode is set, a Guard Mode Interrupt is generated, preventing the erroneous reference and transferring control to the Executive for appropriate action.

Thus, in a multiprogramming environment, the accidental alteration of one program area by another program, even one being tested, is prevented. This is of particular significance in a real time multiprogramming environment, where the real time program cannot be halted or placed in jeopardy by program testing.

The Executive normally operates in open mode. When the Executive gains control, the Guard Mode is de-activated. This gives the Executive free access to the entire main storage. When finished with its task, the Executive reloads the Guard Limits Register and reactivates the Guard Mode before giving control to a worker program.

4. INPUT/OUTPUT MODULES

4.1. GENERAL

The Input/Output Modules (IOM) of the UNIVAC 418-III System provide the data paths and the control circuitry necessary for direct communication between main storage and the input/output devices. The portion of the circuitry and the data paths which are necessary to connect a peripheral subsystem with main storage is called a channel. Each channel allows bidirectional transfers between main storage and the devices on that channel. Each channel contains 36 data lines (18 for input and 18 for output).

The minimum UNIVAC 418-III System includes one Input/Output Module. A second IOM may be added to provide additional channels and an additional main storage referencing unit. As shown in Figure 4-1, each basic IOM contains a minimum of eight channels and can be expanded in increments of four channels to a maximum of sixteen per IOM -- 32 channels for an entire UNIVAC 418-III System.

MINIMUM	EXPANSION 1	EXPANSION 2
(8)	(4)	(4)
CHANNELS 0 THRU 7	CHANNELS 8-11	CHANNELS 12-15

Figure 4–1. Input/Output Module Channel Availability

Most peripheral subsystems utilize both the input and corresponding output lines of the same channel. Data transfers on these subsystems are bidirectional but may not occur at the same instant of time through a single channel. Similarly, data transfers may not occur at the same instant through two different channels on the same IOM. However, if the main storage buffer areas for input/output transfers through two channels of different IOM's are located in different storage banks, these data transfers can occur at the same instant. When two or more subsystems are requesting data transfers at the same time, each IOM performs as a multiplexer with a transfer rate of up to 1.33 million 18-bit words (approximately 4 million characters) per second. The transfer rate capability is 2.66 million 18-bit words if there are two IOM's operating in separate storage banks from each other and from the Command/Arithmetic Section.

Each IOM functions as a small processor. Programmed instructions executed by the Command/Arithmetic Unit load index values into the Buffer Control Words located in the IOM and establish desired peripheral subsystem activity. From that point on, the IOM scans the input/output channels automatically, accepting data from and passing data to the subsystem at the normal rate of the subsystem. When a peripheral subsystem requests a data transfer, the contents of the Buffer Control Word for that channel are used as a main storage address, and the transfer occurs. At the same time, the IOM updates the contents of the Buffer Control Word and tests for a terminal condition. This entire operation requires only the time for one memory cycle, 750 nanoseconds.

The Buffer Control Words are located in very high speed Integrated General Registers which are assigned main storage addresses. However, instead of making reference to the main storage addresses, reference is made to the Integrated General Registers.

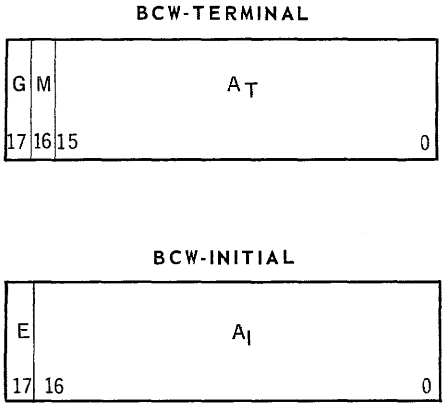
4.2. NORMAL INPUT/OUTPUT MODE

Each channel operates in one of three states: input, output, or function. The input and output states are employed when transferring data to or from main storage. The channel has the capability of alternating between the input and the output state on consecutive memory cycles. The function state is the means by which the Command/ Arithmetic Unit establishes initial communication with a peripheral subsystem and is basically an output state. During this state, the Command/Arithmetic Unit causes the IOM to transfer one or more function words to a peripheral subsystem. These function words direct the subsystem to perform the desired operation.

4.3. BUFFER CONTROL WORDS

The actual word-by-word transfer (regardless of the transfer state) through a given channel is governed by the Buffer Control Words (BCW) in the IOM. Two Integrated General Registers (IGR) are used to control output and two are used to control input for each channel. The first of these two words is the Buffer Control Word for Terminal Address (BCWT). The second is the Buffer Control Word for Initial Address (BCWI).

The format of the Buffer Control Words is shown in Figure 4-2.



A_I : The address of the first word in the buffer area (becomes the current address).

A_T : The address of the last word in the buffer area.

G : A_I Modifier

G = 0 A_I will be incremented by one for each buffer word transfer.

G = 1 A_I will be decremented by one for each buffer word transfer.

M : Monitor

M = 0 Channel transfer will terminate when bits 15-0 of the current address (A_I) equals bits 15-0 of the terminal address (A_T).

M = 1 Channel transfer will terminate when the current address equals the terminal address, and a monitor interrupt will be generated for that channel.

E : Used by the hardware. This bit must initially be zero.

Figure 4–2. ISI Buffer Control Words

The normal input/output mode is referred to as Internally Specified Indexing (ISI) because of the fact that the presence of a data request from a subsystem causes the input/output circuitry of the UNIVAC 418-III System to internally generate the address reference to the specific Buffer Control Words based on the channel through which the data request is received.

4.4. EXTERNALLY SPECIFIED INDEX MODE

The Externally Specified Index (ESI) feature in conjunction with Univac data communication subsystems allows multiple communication lines to transfer characters automatically to and from main storage through a pair of IOM channels. The pair of channels is required because of the manner in which the ESI mode functions.

When operating in ESI mode, each communication line has its data flow governed by its own unique pair of BCW's. Unlike the BCW's for normal mode, the ESI Buffer Control Words are located in main storage to allow accommodation of many multiplexed communication lines. As each line presents a character for transfer to main storage, the address of the unique pair of Buffer Control Words associated with that communication line is also presented to the channel; thus, the BCW address is externally specified. The BCW address is presented on the input lines of one of the paired channels, and the character to be transferred is presented on the input lines of the other of the paired channels. In this manner, the characters received from or transferred to a given communication line are stored in or retrieved from a discrete buffer area controlled by a unique pair of Buffer Control Words for the communication line. The flow of data from a communication line to and from main storage is accomplished under control of the Input/Output Module without the delay or intervention of the Command/Arithmetic Section.

Any adjacent even/odd pair of IOM channels except 0 and 1 can be specified to operate in ESI mode.

4.4.1. Fullword or Halfword ESI Storage of Characters

Paired channels which have been specified to operate in ESI mode must be further designated to operate in fullword or halfword storage mode. In fullword mode, each character is stored in a separate 18-bit word of storage. In halfword mode, two characters are stored in an 18-bit word. The halfword storage method provides twice the capacity in the same number of main storage locations. When storage mode is specified, all communication lines connected through that single pair of channels must operate in the same mode.

Characters stored in fullword mode are placed in the storage location beginning at bit position 0. The unentered bit portion of a storage location (which depends on character size) is set to binary zero. Characters stored in halfword mode are placed in the storage location beginning at bit position 0 or bit position 9. The unentered bit portion of each halfword is set to binary zero.

The ESI Buffer Control Words have a format similar to the normal (ISI) Buffer Control Words. The format is shown in Figure 4-3.

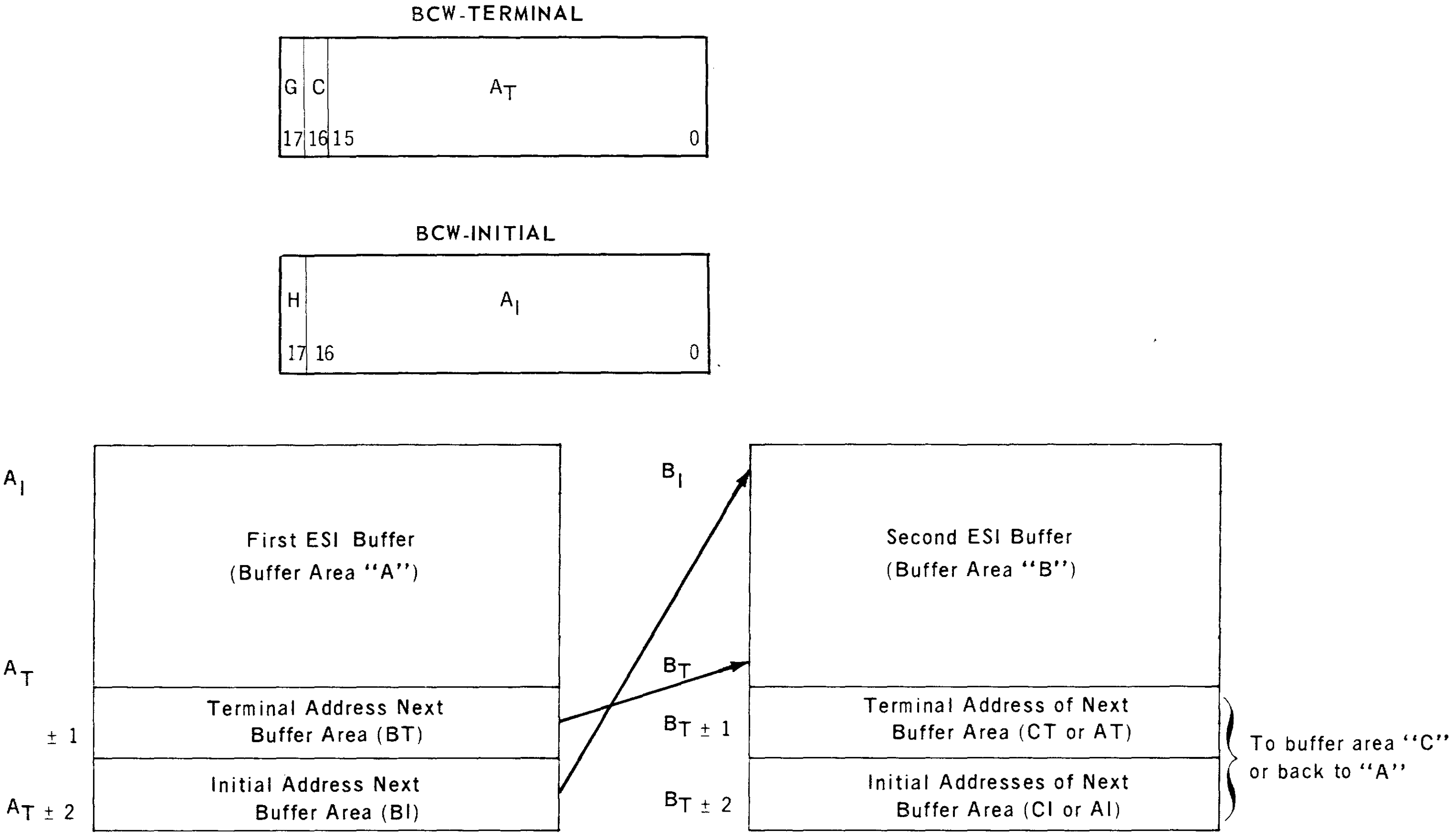
4.4.2. Hardware ESI Buffer Chaining

The hardware control of data communication lines in the UNIVAC 418-III System includes ESI buffer chaining. The purpose of ESI buffer chaining is to eliminate the programmed intervention usually required in order to provide additional buffer areas for communication messages. The buffer chaining technique provides the mechanism to include these specifications, as required, during the operation of the system rather than having them a near rigid specification to a supervisory software program. In addition to this flexibility, hardware buffer chaining also eliminates the storage locations usually required for software control of this function plus the time required to execute those instructions each time a buffer termination occurs.

ESI buffer chaining is controlled by a specific bit position in the ESI Buffer Control Word for Terminal Address (see Figure 4-3). When the current address in the Initial Address BCW is equal to the Terminal Address, the chaining control bit is examined. If this bit is set to 1, the hardware automatically causes the contents of the two storage locations following the address specified by the Terminal Address BCW to become the new Terminal and Initial Address BCW's and data transfer continues. If the chaining control bit is set to 0, no buffer chaining occurs and data transfer terminates. In either case, when the end of a buffer area is reached, the ESI interrupt status word is stored.

Figure 4-3 contains an example of automatic ESI buffer chaining. Assume the C bit in the Terminal BCW which specified Buffer A is set to 1. When A_I equals A_T, the hardware examines the C bit. Upon detecting that it is set to 1, A_T + 1 becomes the new Terminal BCW, A_T + 2 becomes the new Initial BCW, and data transfer continues. The occurrence of the end of Buffer A is automatically stored in the Interrupt Table (discussed in 4.4.3). The new Buffer Control Words specify Buffer B. When B_I equals B_T, the C bit is again examined; if it is set to 1, B_T + 1 and B_T + 2 become the new BCW's. These two words can specify a third buffer (buffer C) or a return to Buffer A.

Upon reaching the end of a chain (indicated by C = 0), the contents of the Initial BCW equals the address in the Terminal BCW, and the Terminal BCW is cleared to 0's. The purpose of clearing the Terminal BCW to 0's at the conclusion of a buffer sequence is to prevent any further data transfers to or from the particular communication line. Formerly, upon conclusion of a buffer area, the entire processor channel was terminated. This necessitated Executive intervention to reactivate the channel and allow data transfers to proceed on other communication lines. Using this new technique, the channel is not terminated. If the communication line inadvertently requests additional data transfers after the Terminal BCW has been cleared to 0's, this condition is detected and such requests are logically ignored. This not only prevents the occurrence of a runaway buffer but also eliminates the necessity for the programmed intervention normally required upon termination of a communications buffer in order to prevent the situation.



A_T : Terminal Address

A_I : Initial Address (becomes current)

G : = 0 Increment A_T
= 1 Decrement A_T

H : = 0 Lower Half
= 1 Upper Half

C : = 0 No Chaining
= 1 Chaining

Figure 4–3. ESI Buffer Control Words

4.4.3. Automatic Tabling of ESI Interrupts

Automatic tabling of ESI buffer termination and external interrupt status word has been included in the UNIVAC 418-III System. This feature precludes the necessity of a special hardware transfer of control (an internal interrupt) to the Executive each time a communication line buffer area is filled or exhausted as well as making unnecessary the execution of any user provided subroutine.

In the UNIVAC 418-III System, the occurrence of an internal interrupt is handled by a combination of ESI buffer chaining and the automatic tabling of ESI interrupts. The buffer chaining automatically provides an alternate buffer area for the continuity of the data transmission. The automatic tabling of the interrupt associated with the buffer termination provides a record of the occurrence of the event without Executive intervention.

The hardware permits the UNIVAC 418-III Executive to refuse or to accept control when an interrupt attempts to cause a transfer. When the Executive is in the process of providing a series of ESI interrupt status words to the communication control routine, the IOM hardware will continue to table ESI interrupts into main storage without disturbing the C/A unit. Any buffer termination or external interrupt which occurs under these circumstances continues to be stored in an Interrupt Status Table. Upon completion of its task, the Executive examines the tabling mechanism to determine if any interrupts have occurred during the period when interrupts were prevented. If it is determined that an ESI interrupt was tabled during the period when ESI interrupts were prevented, the Executive would again provide the communications control routine with the accumulated interrupt status words. The automatic tabling of interrupts prevents the unnecessary transfer of control and execution of instructions to inform the Executive that there has been an ESI interrupt when the Executive is busily engaged in passing previous ESI interrupts to the communications control program. If no ESI interrupts occur during the period of interrupt prevention, the Executive releases the ESI interrupt lockout.

Each IOM provides an Integrated General Register which is used as the Interrupt Table Pointer for the communications channels on the IOM. The Table Pointer Word provides for specifying the size of the table in increments of 64 words to a maximum size of 512 words. The Interrupt Table Pointer Word also contains an address portion which specifies the location in the table to be used to store the next interrupt status word; each time a status word is stored, this address is incremented by one. When it becomes the last address in the table, the status is stored, and the address is reset to point to the beginning of the table. See Figure 4–4.

INTERRUPT POINTER

TS		B		TP	
17	15	14	9	8	0

TS: Interrupt Table Size

This field specifies the size of the interrupt table as follows:

TS (Octal)	Table size in words (Decimal)
0	64
1	128
2	192
3	256
4	320
5	384
6	448
7	512

B: Address of the interrupt table location.

This address must be located in the first 32K of storage. (The address is in 512 word increments.)

TP: A specific address within the table in which the next interrupt will be stored. It must initially be all zeros.

Figure 4–4. Table Pointer Word

The combination of the automatic tabling of ESI interrupts and ESI automatic buffer chaining provides an extremely versatile method for the design and control of communication lines. Formerly, these conditions required critical design of programmed subroutines, the space required to store these subroutines, and the time required to execute the instructions. The new techniques in the UNIVAC 418-III System permit the user to devote his attention to the processing of the data for communications rather than to tedious detail.

5. COMMAND/ARITHMETIC SECTION

5.1. INTRODUCTION

The Command/Arithmetic Section (C/A) of the UNIVAC 418-III System is the unit which most nearly corresponds to a central processor unit. As previously described, the input/output and the main storage sections are now separated and perform independently of the Command/Arithmetic Section.

The Command/Arithmetic Section consists of two basic sections: The Command (or Control) Section and the Arithmetic (or Processing) Section.

The Command Section provides the logic for instruction decoding and execution. It includes the Instruction Address Register, which is used for the access of instructions and as a base register for operand access; the Special Register, used as a base register for operand access; and the Index Register Pointer, which is used to control operand address modification. The Command Section is responsible for the initiation of input/output functions but not for the handling of the input/output data transfers associated with these functions. Input/output data transfers occur independently between the Input/Output Modules and main storage under the control of the Input/Output Modules. The Command Section also is responsible for hardware servicing of interrupts.

The Arithmetic Section consists of the accumulators and control circuits necessary for the performance of fixed and (optional) floating-point and binary/decimal conversion arithmetic; shifting; logical operations; tests; and the control of the Block Transfer operation.

5.2. THE COMMAND SECTION

In addition to the previously discussed registers, the Command Section contains the circuitry necessary to reference the Integrated General Registers (IGR) located in the Input/Output Modules and the reserved locations in main storage. The function of the Integrated General Registers is explained in Section 4. The Special Register, the Instruction Address Register, and the Index Register Pointer are located in the Command/Arithmetic Unit but do not have main storage addresses.

5.2.1. Reserved Locations

Figure 5-1 shows the location and use of the reserved locations and main storage addresses. Locations 000302_8 to 000377_8 are Integrated General Registers which are physically located in Input/Output Module #0. Locations 000400_8 to 000477_8 are Integrated General Registers which are physically located in Input/Output Module #1.

5.2.2. Special Register

The Special Register (SR) is a six-bit register whose contents may be used as a base register for Type I instruction operand access. One of the bits in the Special Register is the SR Active bit. When this bit is set to 1, the other five bits become the high-order five bits of a 17-bit address whose low-order 12 are provided in the Type I instructions. Type I instructions are sensitive to the Special Register. The contents of the SR Active bit and the other five bits of the Special Register can be varied under program control.

5.2.3. Instruction Address Register

The Instruction Address Register (IAR) is a 17-bit register whose contents are used to control the access of instructions from main storage. The IAR has additive properties which normally specify incrementation by one to provide sequential access of instructions. The test commands specify an increment of two in order to perform branching (skipping). The entire contents of the IAR may also be changed by the jump commands in order to provide complete transfer of control.

In addition to its function of providing access to instructions, the high-order five bits of the IAR may also be used as a base register for operand access in certain of the Type I and Type II instructions. If the Type I instruction is not SR sensitive, or if it is SR sensitive and the SR Active bit is not set, then the high-order five bits of the IAR become the high-order five bits of the operand address whose low-order 12 bits are supplied within the instruction. Certain of the Type II instructions cannot be SR sensitive and always use the high-order five bits of the IAR for operand access. The effect of using the high-order five bits of IAR for operand access is to select an operand which is located in the same 4096 word segment as the instruction which calls for the access.

5.2.4. Index Register and Index Register Pointer

Eight main storage locations are reserved for use as Index Registers (IR). The function of the Index Register Pointer (IRP) is to control the storing and loading of one flip-flop type B-register to and from one of these Index Registers. The contents of the B-register are used for address modification in those instructions which call for address modification. The function of Index Register address modification takes place after and in addition to Special Register or Instruction Address Register Base modification.

The Index Register Pointer is a three-bit register whose contents can be set from 0 to 7 by a Load Index Register Pointer instruction. When this instruction is executed, the present contents of the IRP is the designation used to control the automatic storing of the B-register. After storing the contents of the B-register, the IRP is then loaded with the ''new'' value. This new value then designates the automatic loading of the B-register.

000000	RESERVED
000001 000010	INDEX REGISTERS
000011 000016	RESERVED
000017	SCALE FACTOR SHIFT COUNT
000020	SUPERVISOR CALL ENTRANCE
000021	PARITY INTERRUPT IOM #0
000022	PARITY ERROR ADDRESS IOM #0
000023	DAY CLOCK INTERRUPT
000024	DAY CLOCK COUNT
000025	REAL-TIME CLOCK INTERRUPT
000026	REAL-TIME CLOCK COUNT
000027	PARITY ERROR MAIN STORE
000030	GUARD MODE INTERRUPT
000031	PARITY ERROR IOM #1
000032	PARITY ERROR ADDRESS IOM #1
000033	POWER LOSS INTERRUPT
000034	FLOATING POINT UNDERFLOW INTERRUPT
000035	FLOATING POINT OVERFLOW INTERRUPT
000036	ESI INTERRUPT IOM #0
000037	ESI INTERRUPT IOM #1
000040	IGR/ARITHMETIC PARITY INTERRUPT
000041	INTERRUPT LOCKOUT TIMEOUT

000100 000117	INTERNAL INTERRUPT OUTPUT CHANNELS 0–15
000120 000137	INTERNAL INTERRUPT INPUT CHANNELS 0–15
000140 000157	EXTERNAL INTERRUPT CHANNELS 0–15
000160 000177	EXTERNAL INTERRUPT STATUS CHANNELS 0–15
000200 000277	PREVIOUS ASSIGNMENT REPEATED FOR CHANNELS 16–31
000300 000301	RESERVED
000302 000337	OUTPUT BUFFER CONTROL WORD CHANNELS 1–15
000340	INTERRUPT TABLE POINTER IOM #0
000341	RESERVED
000342 000377	INPUT BUFFER CONTROL WORDS CHANNELS 1–15
000400 000437	OUTPUT BUFFER CONTROL WORDS CHANNELS 16–31
000440	INTERRUPT TABLE POINTER IOM #1
000441	RESERVED
000442 000477	INPUT BUFFER CONTROL WORDS CHANNELS 17–31

Figure 5–1. Format of Reserved Locations

When Index Register modification is called for by an instruction (Type I, Type II, or Type III), the modification takes place at no increase in the execution time of the instruction because the modification value is in the very high speed B-register.

5.2.5. Real Time Clock and Day Clock

The very nature of real time operations makes it necessary for the computer system to be capable of responding to computer-time demands. Two clocking devices are available with the UNIVAC 418-III System, the real time clock, which is standard; and the day clock, which is an optional feature.

The circuitry for control of the real time clock is located in Input/Output Module #0. Every 200 microseconds this circuitry causes the contents of main storage location 000026_8 to be decremented by one. When the value of location 000026_8 is equal to 0, the circuitry creates a Command/Arithmetic Unit interrupt and causes control to be transferred to main storage location 000025_8. By loading the appropriate value into location 000026_8, the real time clock generates an interrupt after the lapse of a specified time. For example, a value of 5000_{10} would cause an interrupt at the end of one second. The real time clock is under control of the executive which provides the user program with the capability for establishing time-oriented initiation of programmed sequences.

The day clock, which is a 24-hour clock, is an optional feature of the UNIVAC 418-III System. A display on the console shows hours, minutes, tenths, and hundreths of minutes. The clocking mechanism updates a fixed location in main storage every six seconds and causes an interrupt transfer of control to location 000023_8 each minute. The time of day appears in the fixed main storage location 000024_8 in the following binary coded decimal format:

HOURS		MINUTES		
TENS	UNITS	TENS	UNITS	TENTHS
17 16	15 12	11 8	7 4	3 0

The executive uses the day clock to establish the time for its time/date control function. The time is available to the worker program via an executive request.

5.3. THE ARITHMETIC SECTION

The Arithmetic Section consists of two 18-bit accumulators and a number of special purpose registers and circuits which are designed to control and perform the arithmetic, logical, transfer, shift, compare, test, etc., operations in the UNIVAC 418-III instruction repertoire. The two 18-bit registers are combined to form a single 36-bit register for certain of these operations. The 36-bit accumulator is called A. As shown in Figure 5–2, the two 18-bit registers are called A Upper (AU) for the high-order 18 bits of A and A Lower (AL) for the low-order 18 bits of A.

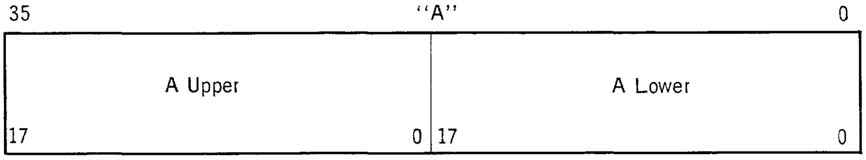


Figure 5–2. Fixed-Point Accumulator

5.3.1. Fixed-point Arithmetic

Fixed-point binary arithmetic add and subtract may be performed on an 18-bit (single length) basis in either AU or AL, or on a 36-bit (double length) basis in A. Multiplication of two 18-bit operands produces a 36-bit product in A. Division uses the contents of A as a 36-bit dividend which is replaced by an 18-bit quotient in AL and an 18-bit remainder in AU.

5.3.2. Floating-point Arithmetic

The optional floating-point feature of the UNIVAC 418-III System provides high-speed arithmetic operations for scientific and engineering applications using floating-point binary notation. The floating-point notation used consists of an eight bit characteristic (exponent), a 27-bit mantissa (fixed-point part), and a 1-bit sign position as shown in Figure 5–3.

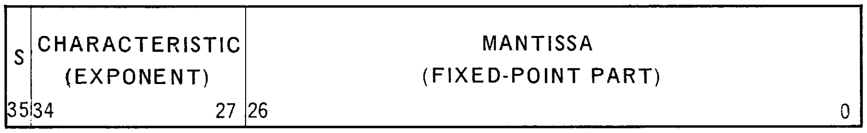


Figure 5–3. Floating-Point Accumulator

The eight-bit characteristic (exponent) provides a range of plus or minus $2^7 -1$, and the mantissa (fixed-point part) provides precision of $2^{27} -1$. In addition to the arithmetic functions of add, subtract, multiply, and divide, the floating-point feature provides a Pack and an Unpack instruction.

All floating-point operations are performed on normalized operands and produce normalized results. The speed of execution of floating-point arithmetic instructions in the UNIVAC 418-III System is greater than that of some higher priced systems. For example, the Multiply command requires only 12 microseconds including indexing of the operand and normalizing of the result.

5.3.3. Binary/Decimal Conversion Instructions

The binary/decimal conversion instructions are an optional feature of the UNIVAC 418-III System designed to enhance the operation in commercial data processing applications. Decimal arithmetic is inherently slow arithmetic and requires inefficient use of storage bits. Because decimal notation is ordinarily used, conversion instructions are provided to enable easy and rapid conversion of the data which passes through the man/machine interface. However, most of the data handled by the UNIVAC 418-III System does not require this decimal conversion. Nearly all of the quantitative data in master data files and program files can be stored in binary form in order to reduce the internal and external storage requirements and to speed the processing of the files.

The binary/decimal conversion instructions provided the programmer with an automatic means of converting decimal numbers, coded in six-bit Binary Coded Decimal, into binary numbers and for conversion of binary numbers into six-bit Binary Coded Decimal numbers.

Conversion from decimal to binary is made on a three-digit number placed in A Upper in the following format:

XXNNNN	XXNNNN	XXNNNN
17 12	11 6	5 0
D1	D2	D3

The bits indicated by X are ignored by the Decimal-to-Binary Conversion instruction. The resulting binary number is right justified in A Lower.

The Binary-to-Decimal Conversion instruction operates on a binary number which has been placed in A Lower and produces the format shown above with the exception that the bit positions indicated by X are binary 0's. The binary number to be converted should not produce a decimal value greater than 999. Values which are larger than three digits may be easily processed by using a factor of 10^3 to perform scaling by multiplication on the result of decimal-to-binary conversion or division on the binary number to be converted.

5.4. INSTRUCTION REPERTOIRE

The instructions of the UNIVAC 418-III System are of three types, and these types are also grouped in two classes – *privileged* and *nonprivileged.* The privileged instructions are reserved for use by the Executive and basically consist of the instructions associated with input/output and storage protection. The privileged instructions are listed in Section 5.4.4. The nonprivileged instructions may be executed by any program.

5.4.1. Type I Instructions

The Type I instructions are those commands which reference main storage and which are sensitive to the SR Active bit. The format of Type I instructions is:

f	u
17 12	11 0

f = six-bit function code (02-27, 32, 33, 40-47)

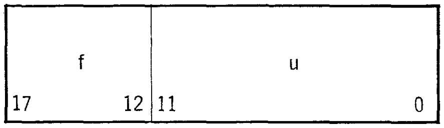
u = 12-bit displacement

When a Type I instruction is executed and the SR Active bit is set to 1, the high-order five bits required for a 17-bit main storage address are obtained from the Special Register. If the SR Active bit is not set to 1, the required five bits are obtained from the high-order five bits of the Instruction Address Register.

5.4.2. Type II Instructions

The Type II instructions are those commands which reference main storage and which are not sensitive to the SR Active bit, and those which supply an immediate operand.

The format of Type II instructions is:

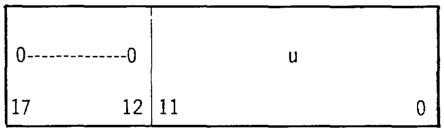


f = six-bit function code (30, 31, 34–37, 51–76)

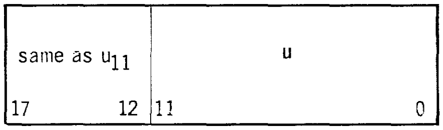
u = 12-bit displacement or immediate operand

When a Type II instruction is executed, the high-order five bits required for a 17-bit main storage address are obtained from the high-order five bits of the Instruction Address Register.

If the instruction is one which supplies an immediate operand, the u portion is handled in a manner specified by the function code. An immediate operand is a constant contained in the u portion of the instruction itself. The function code specifies the method of creating an 18-bit operand from the 12-bit u portion. Either zero extension or sign extension is used:



18-bit operand with zero extension



18-bit operand with sign extension

In the case of zero extension, the high-order six bits of the operand are arbitrarily set to 0's. In the case of sign extension, the high-order bit of the u portion (u_{11}) is used to fill the high-order six-bit portion of the operand.

5.4.3. Type III Instructions

Type III instructions are those instructions which contain special parameters which must be supplied to the internal circuitry for control of certain functions. The basic format of Type III instructions is:

f	m	k
17 12	11 6	5 0

f = six-bit function code (always 50)

m = six-bit minor function code (00 - 77)

k = special parameters (e.g. shift count, channel number, etc.)

Some Type III instructions require an additional one or two storage locations in order to convey parameters: For example, the commands which are used to supply Buffer Control Words to the IOM use the two storage locations immediately following the Type III instruction for these parameters. The optional Floating-Point and Binary/ Decimal instructions are Type III instructions.

5.4.4. Instruction Groups

The instructions of the UNIVAC 418-III System according to their fundamental purposes are:

- Arithmetic (Standard) Commands

  Add to Lower

  Add Negatively to Lower

  Add to Accumulator

  Add Negatively to Accumulator

  Multiply

  Divide

  Round Accumulator

  Add to Lower Accumulator Constant

- Floating-point Arithmetic Commands (optional feature)

  Floating-point Add

  Floating-point Subtract

  Floating-point Multiply

  Floating-point Divide

  Floating-point Pack

  Floating-point Unpack

- Binary/Decimal Conversion Commands (optional feature)

  Binary-to-Decimal Conversion

  Decimal-to-Binary Conversion

- Logical Commands

  Logical And

  Inclusive Or

  Exclusive Or

  Complement Accumulator, Lower

  Complement Accumulator, Upper

  Complement Accumulator, All

- Transfer Commands

  Load Accumulator, Upper

  Load Accumulator, Lower

  Store Accumulator, Upper

  Store Accumulator, Lower

  Load Accumulator, Lower, with Constant

  Masked Selective Load

  Load Index Register

  Store Index Register

  Load Index Register with Constant

  Store Address of Accumulator Lower

  Load Index Register Pointer

  Store Index Register Pointer

  Load Special Register

  Store Special Register and Inactivate

  Clear Main Storage

  Block Transfer

- Shift Commands

  Shift Right, Accumulator Upper

  Shift Right, Accumulator Lower

  Shift Right, Accumulator All

  Shift Left, Accumulator Upper

  Shift Left, Accumulator Lower

  Shift Left, Accumulator All

  Scale Accumulator

- Loop Control Commands

  Jump and Modify if Index Register Nonzero

  Test Index Register

  Test Any Location for Zero

- Compare Commands

  Compare Accumulator Lower

  Compare Accumulator Lower with Mask

- Comparison Jump Commands

  Jump on Equal

  Jump on Not Equal

  Jump on Less

  Jump on Not Less

- Arithmetic Jump Commands

  Jump on Accumulator Upper Zero

  Jump on Accumulator Upper Nonzero

  Jump on Accumulator Upper Positive

  Jump on Accumulator Upper Negative

  Jump on Accumulator Lower Zero

  Jump on Accumulator Lower Nonzero

  Jump on Accumulator Lower Positive

  Jump on Accumulator Lower Negative

- Unconditional Jump Commands

  Jump

  Jump Indirect

  Store Location and Jump Indirect

  Store Location and Jump

- Skip Commands

  Test Keys

  Test No Borrow

  Test Overflow

  Test No Overflow

  Test Odd Parity

  Test Even Parity

- Other

  Allow All Interrupts

  Prevent All Interrupts

  Load Special Designators

  Store Special Designators

- Executive Commands (Privileged Instructions)

  Input/Output Commands

  - Load Input Channel
  - Load Output Channel
  - Load External Function Channel
  - Terminate Input Mode
  - Terminate Output Mode
  - Skip if Input Mode Inactive
  - Skip if Output Mode Inactive
  - Skip if External Function Mode Inactive

Interrupt Control Commands

- Enable ESI Interrupts
- Wait for Interrupt
- Enable Interrupts and Jump Indirect

Storage Protection Commands

- Load Guard Mode

Stop Command

- Stop on Key Setting

Special Commands

- Read and Set
- Set Audible Alarm

Any attempt to execute a privileged instruction while in Guard Mode causes a fault interrupt which transfers control to the Executive. The Executive logs the event and terminates the program which caused the fault.

Appendix A contains lists of the instructions by operation code and functions which include their timings and description.

6. PERIPHERAL SUBSYSTEMS

6.1. GENERAL

Peripheral subsystems are connected to the UNIVAC 418-III System through one or more of the channels of an Input/Output Module. The number of channels required depends on the specific subsystem and the mode in which it operates. When two channels of an IOM are required for the subsystem, the channels must be an even-numbered channel and the adjacent next highest odd-numbered channel. This is called paired-channel mode. When the desired operation of the subsystem is such that it requires two of the paired channels, it is called dual paired-channel mode. Factors which dictate the mode of operation are the transfer rate of the device and whether or not two devices of the subsystem are to operate simultaneously.

The standard peripheral subsystems for the UNIVAC 418-III System are:

- *High Performance Drums*
  FH-432 Magnetic Drum Subsystem
  FH-880 Magnetic Drum Subsystem
  FH-1782 Magnetic Drum Subsystem

- *Mass Storage*
  FASTRAND II Subsystem
  FASTRAND III Subsystem

- *Magnetic Tape Subsystems*
  UNISERVO VI C Subsystem
  UNISERVO VIII C Subsystem

- *Auxiliary Subsystems*
  UNIVAC 9000 Series Online/Offline
  High Speed Printer
  Paper Tape

- *Communications Subsystems*
  Communications Terminal Module Controller (CTMC)
  Word Terminal Synchronous (WTS)

6.2. THE 'FH' SERIES OF MAGNETIC DRUMS

The UNIVAC Flying Head (FH) series of high speed medium capacity magnetic drum storage devices provides modular auxiliary storage useful to the operation of most general purpose systems. These devices vary from the ultra fast FH-432 drum which has an average access time of 4.25 milliseconds to the large capacity FH-1782 drum which has a capacity of 12.5 million alphanumeric characters and provides extensive fast access storage that can be used for large data files that have to be referenced frequently.

In each FH magnetic drum subsystem each track has an individual read-write head.

Any word of an FH series drum is available to the system in an average half revolution access time of 4.25 milliseconds (FH-432) and 17.0 milliseconds (FH-1782).

Every word in all FH subsystems is individually addressable so that the fullest use can be made of high performance storage. Offline search operations can be initiated by which the control unit matches the contents of any word of any drum area, up to the entire size of the subsystem, with a designated Identifier Word, and will, upon finding a match, supply the address of the match or commence reading to main storage. This entire process is carried out offline without any Command/Arithmetic unit attention once the input/output search function has been initiated and the Identifier Word designated. This feature is frequently used in the scanning of large data tables when the exact location of an item is unknown.

The transfer rate of data to and from the FH drum subsystems is in line with the ultra fast computing power available. The standard rate is 1,440,000 alphanumeric characters per second. By means of a field-installable option, drum transfer rates may be matched to system loads by interlacing to provide transfer rates of 720,000; 360,000; 180,000; or 90,000 alphanumeric characters per second. This ability to modify transfer rates avoids processing stoppage in those configurations unable to handle the extremely fast transfers.

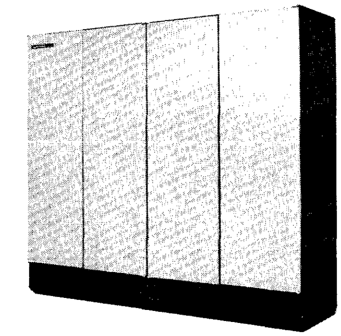
With all FH drum subsystems, either one or two control units are available using paired- or dual-channels. The dual-channel capability permits read/read, read/write, write/read and write/write operations simultaneously on any two drum units of the subsystem. If required, a search function may be substituted for any of the read functions. As an additional reliability measure, each control unit of a dual-channel subsystem has its own power supply. Thus, in case of a failure of one of the power supplies, the subsystem can still operate on a paired-channel basis.

The philosophy of the UNIVAC 418-III Real Time Operating System is to maximize performance by using drums. This reduces manual handling as well as access and transfer time when compiling and assembling.

These magnetic drum subsystems have many advantages in standard data processing as well as in real time operation. This is especially true in applications where rapid file processing and sort/merge routines are prevalent. Large capacity with rapid accessibility provides convenient intermediate storage. Instead of multiple magnetic tape units, the use of the drum subsystems frees the tape units for primary input/output demands.

Drum subsystems allow an extensive executive control system without undue main storage utilization or operating inefficiency. The high speed access permits lesser used control segments to be stored outside of main storage and read in to a common overlay area only when required. This arrangement greatly reduces the amount of main storage required for the Executive routine.

6.2.1. FH-432 Magnetic Drum Subsystem



CHARACTERISTICS

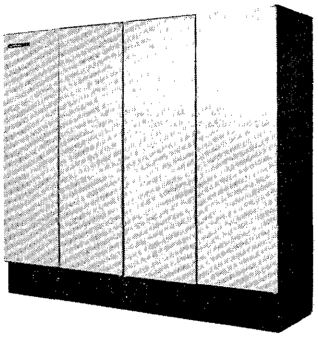
Storage Capacity	524,288 computer words of 18 data bits plus parity bits or 1,572,864 alphanumeric characters per drum
Average Access Time	4.25 milliseconds
Drum Speed	7,200 revolutions per minute
Number of Read/Write Heads	432 - one per track
Character Transfer Rate (Max.)	1,440,000, 720,000, 360,000, 180,000, 90,000 per second
Word Transfer Rate (Max.)	480.000, 240,000, 120,000, 60,000, 30,000 per second
I/O Channels Required	Two or four per subsystem
Number of Drums Per Subsystem	Eight (total of 12,582,912 characters)

A minimum FH-432 magnetic drum subsystem consists of a control unit and a drum unit with 524,288 eighteen-bit words of storage. Of the 432 tracks on each drum 384 are used for data purposes; the remaining tracks are used for spares, parity, and timing functions. There are 4,096 words of data for three tracks. The functions of reading and writing occur in three-bit-parallel mode on all three tracks of a band simultaneously at a maximum transfer rate of 480,000 words or 1,440,000 characters per second.

Up to eight FH-432 magnetic drums may be accommodated in a single subsystem using paired- or dual-control units, and by means of a Shared Peripheral Interface being attached to two Command/Arithmetic Units, maximum subsystem capacity is 4,194,304 words or 12,582,912 alphanumeric characters.

FH-432 drums may be intermixed with FH-1782 drums in the same subsystem to provide a powerful blend of ultra high speed and large capacity storage (see Section 6.2.3).

6.2.2. FH-1782 Magnetic Drum Subsystem



CHARACTERISTICS

Storage Capacity	4,194,304 computer words of 18 data bits plus parity bits or 12,582,912 alphanumeric characters per drum
Average Access Time	17 milliseconds
Drum Speed	1,800 revolutions per minute
Number of Read/Write Heads	1782 (33 blocks with 54 heads per block)
Character Transfer Rate	1,440,000, 720,000, 360,000, 180,000, 90,000 per second
Word Transfer Rate	480,000, 240,000, 120,000, 60,000, 30,000 per second
I/O Channels Required	Two or four per subsystem
Number of Drums Per Subsystem	Eight (total of 100,663,296 characters)

The Flying Head 1782 (FH-1782) magnetic drum is identical to the FH-432 drum but with eight times the storage capacity; achieved partly by an increase in the number of data tracks (to 1,536) and partly by an increase in the recording density. Each track has its own read/write head and average access time is 17 milliseconds.

A single FH-1782 drum holds 4,194,304 computer words, equivalent to 12,582,912 alphanumeric characters. Up to eight FH-1782 drums can be accommodated in a single subsystem giving a subsystem capacity of 100,663,296 characters.

In addition to the increased storage capacity available, the character transfer rates have been increased to be identical with those of the FH-432 drum; this arrangement enables FH-1782 drums to be associated on the same subsystem with FH-432 drums, as described in Section 6.2.3.

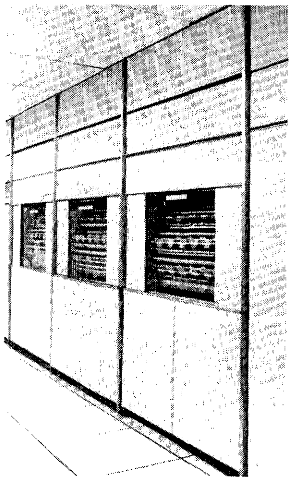
6.2.3. FH-432/FH-1782 Magnetic Drum Subsystem

A valuable option with UNIVAC 418-III drum subsystems is the ability to associate in the same subsystem the ultra high speed FH-432 drum with the fast high-capacity FH-1782 drum. Any combination of eight drums may be mixed on a subsystem.

This subsystem arrangement is of significant importance in the UNIVAC 418-III System storage configuration. An efficient blend can be made of high speed storage devices for rapidly required software, program segments, tables, and indexes, and greater access time but large capacity storage for less frequently used program segments, data files, and message assembly/disassembly areas. A mix that can be readily altered as requirements change can be judiciously planned for speed, capacity, and economy. Character transfer rates are identical for the FH-432 and FH-1782 drums; the only functional difference in a data transfer is the variation in access time.

This subsystem is available in both paired- and dual-channel versions to provide a hierarchy of auxiliary storage for general purpose and complex real time operations. The dual-channel version includes two electrically and logically independent control units each on a different paired I/O channel. This enables simultaneous operation of any two drums in the subsystem and provides the hardware redundancy necessary to highly critical real time operations.

6.3. FH-880 MAGNETIC DRUM SUBSYSTEM



CHARACTERISTICS	
Storage Capacity (per drum)	1,572,864 words 4,718,592 alphanumeric characters
Average Access Time	17 milliseconds
Drum Speed	1770 rpm
Number of Read/Write Heads	880 (one per track)
Transfer Rate	120,000 words per second 360,000 characters per second
I/O Channels Required	2
Number of Drums per Subsystem	1 to 8

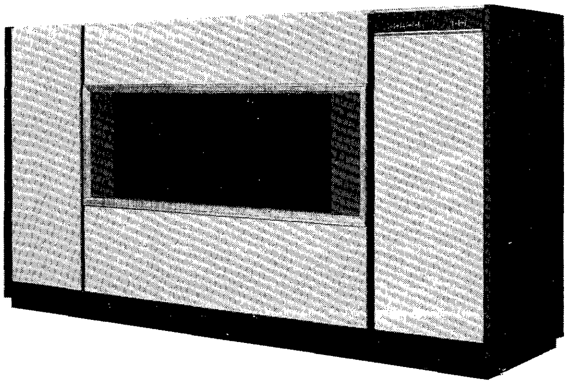
The FH-880 Magnetic Drum Subsystem provides the UNIVAC 418-III Real-Time System with a large-capacity, word-addressable, random-access storage medium. The subsystem consists of a control unit and from one to eight FH-880 magnetic drums, with each drum capable of storing 1,572,844 computer words of 18 data bits plus parity. The average access time for any word in the subsystem is 17 milliseconds. The FH-880 Drum Subsystem is connected to two I/O Channels of an IOM.

Of the 880 tracks on each drum, 768 are used for storing data, 32 for parity, and the remainder for spares and for timing purposes. The 768 tracks of data storage are organized into 128 bands of six tracks each. Each band has a capacity of 12,288 words. Reading and writing are performed in six-bit-parallel mode on all six tracks of a band simultaneously, at a maximum transfer rate of 120,000 words or 360,000 characters per second.

Odd parity checking is used to verify the accuracy of data recording. When data is recorded on the drum, the control unit generates one parity bit per word and stores this bit in a predetermined location on one of the parity tracks that is associated with the word being written. When data is read from the drum, the associated parity bit is also read, and parity is checked automatically. Should a parity error occur, an external interrupt signal is generated, and the C/A Unit is notified of the address of the word in which the error was detected.

Once initiated, all functions of the subsystem operate independently of the C/A Unit, including actual data transfers. In addition to the usual functions of reading and writing at known addresses, the FH-880 magnetic drum subsystem provides the ability to search offline through a drum area of any size. Once a search function has been initiated and an identifier word transferred, the subsystem performs all required operations without intervention from the C/A Unit until the function is terminated.

6.4. FASTRAND II MASS STORAGE SUBSYSTEM



CHARACTERISTICS

Storage Capacity (Per Unit)	44,040,192 words or 132,120,576 alphanumeric characters
Average Access Time	92 milliseconds
Recording Density	1,000 bits per inch
Tracks Per Inch	106
Drum Speed	870 revolutions per minute
Movable Read/Write Heads	64
Character Transfer Rate	153,600 characters per second
Word Transfer Rate	51,200 words per second
Fastbands (Fixed Read/Write Heads)*	24
Fastband Average Access Time	35 milliseconds
Fastband Storage Capacity (Per Unit)	258,048 characters
Write Lockout Protection*	Yes
I/O Channels	Two or four per subsystem
No. of Units Per Subsystem	Eight (1,056,964,608 characters per subsystem)

**Optional*

NOTE: Addition of Fastband increases capacity by 258,048 characters per unit (2,064,384 if all eight drums include Fastbands).

The FASTRAND II Mass Storage subsystem provides very large capacity random access storage. Usefulness of this subsystem is increased by providing the availability of both a paired and a dual-paired channel subsystem, and by the Fastband option, available on a unit basis, which provides fixed read/write heads on some tracks giving rapid access to these areas of storage.

FASTRAND II units include two large magnetic drums with flying heads similar to those used in the FH-432 and FH-1782 subsystems. However, to reduce cost, only a limited number of read/write heads which move laterally over 192 recording tracks are used.

There are 64 read/write heads per drum unit with all heads connected to a common positioning mechanism. As a result, the subsystem positions all of the heads in a drum unit with one movement of its positioning mechanism in an average time of 57 milliseconds. The maximum head positioning time is 86 milliseconds over all the tracks and the minimum in 30 milliseconds. Average latency is half a drum revolution time (35 milliseconds).

Access time, therefore, varies from less than one ms (when a head is already positioned over the desired track and minimum latency is required) to 156 ms (for maximum head movement and maximum latency), with an average of 92 milliseconds. The average time can usually be reduced by good systems design, data layout, and programming.

Additionally, the FASTRAND II subsystem has an independent position control feature which allows greater flexibility and can be used to decrease the average access time. This is done in a multi-unit subsystem by concurrently pre-positioning the heads in each drum unit. Pre-positioning the heads saves time because once the position instructions have been transmitted to the FASTRAND II unit, the executive can immediately initiate another operation on a different drum unit without having to wait for the positioning operation to be completed. Then, whenever specifically instructed to read or write from a pre-positioned unit, the only time delay is for the latency and address circuit activation. These are the mechanical design features which contribute to the operating speed of the FASTRAND II subsystem.

If the circumstances are such that the head positioning time (resulting from prepositioning) and the latency are both zero, then the minimum access time is 0.8 milliseconds. This time interval allows certain addressing circuits to be activated.

Normally, such activation starts simultaneously with the other processes involved; in the best case situation described above, it is the only delay. In the case of mean (92 or 67.5 milliseconds) or maximum access time (156 or 102 milliseconds), the addressing circuit time need not be considered because it is overlapped by other time factors.

In any effort to reduce processing time, offline search represents an important advantage. This is the operation in which the computer instructs the FASTRAND II subsystem to locate a specific piece of data, and then goes on with the processing while the storage search takes place. Then, at the termination of this variable length search, the computer is notified and is given the results. The computer may take action as provided in the program.

Also, the FASTRAND II subsystem permits the computer to continue its work while records are being read from or written on the drums.

All data on a FASTRAND II drum is recorded with parity. These parity bits are automatically checked upon reading and created on writing.

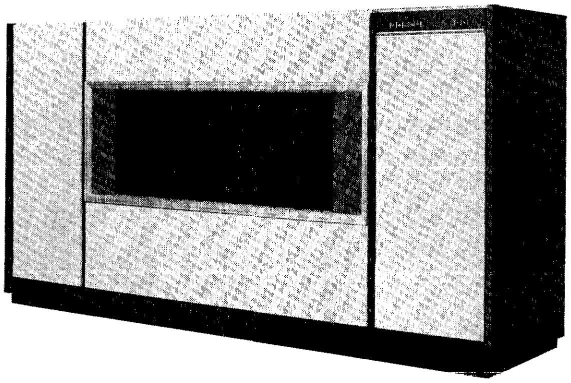
A single FASTRAND II subsystem can accommodate up to eight FASTRAND II drums.

As with the FH-1782 drum subsystem, either paired- or dual-channel operation is possible, providing full scale two-drum unit simultaneity.

The dual-channel subsystem includes two control units. The control units, being logically and electrically independent, provide the redundancy of data paths necessary for truly simultaneous operation. Simultaneous operations can take place on any two drums in the subsystem.

Optional features for FASTRAND II subsystems include 24 additional tracks with fixed read/write heads (Fastbands) for rapid access (35 ms average) and an incremental drum write lockout feature for data protection. By means of this lockout feature, the operator can manually prohibit writing on 1, 2, 4, 8, 16, 32, or all 192 tracks beginning with the first track of each head.

6.5. FASTRAND III MASS STORAGE SUBSYSTEM



CHARACTERISTICS

Storage Capacity (Per Unit)	66,060,288 words or 198,180,864 alphanumeric characters
Average Access Time	92 milliseconds
Recording Density	1600 bits per inch
Tracks Per Inch	106
Drum Speed	870 revolutions per minute
Movable Read/Write Heads	64
Character Transfer Rate	230,400 characters per second
Word Transfer Rate	76,800 words per second
Fastbands (Fixed Read/Write Heads)*	24
Fastband Average Access Time	35 milliseconds
Fastband Storage Capacity (Per Unit)	387,072 characters
Write Lockout Protection*	Yes
I/O Channels	Two or four per subsystem
No. of Units Per Subsystem	Eight (1,585,446,912 characters per subsystem)

**Optional*

NOTE: Addition of Fastband increases capacity by 387,072 characters per unit (3,096,576 if all eight drums include Fastbands).

The FASTRAND III Mass Storage subsystem provides very large capacity random access storage. Usefulness of this subsystem is increased by providing the availability of both a paired and a dual-paired channel subsystem, and by the Fastband option, available on a unit basis, which provides fixed read/write heads on some tracks giving rapid access to these areas of storage.

FASTRAND III units include two large magnetic drums with flying heads similar to those used in the FH-432 and FH-1782 subsystems. However, to reduce cost, only a limited number of read/write heads which move laterally over 192 recording tracks are used.

There are 64 read/write heads per drum unit with all heads connected to a common positioning mechanism. As a result, the subsystem positions all of the heads in a drum unit with one movement of its positioning mechanism in an average time of 57 milliseconds. The maximum head positioning time is 86 milliseconds over all the tracks and the minimum in 30 milliseconds. Average latency is half a drum revolution time (35 milliseconds).

Access time, therefore, varies from less than one ms (when a head is already positioned over the desired track and minimum latency is required) to 156 ms (for maximum head movement and maximum latency), with an average of 92 milliseconds. The average time can usually be reduced by good systems design, data layout, and programming.

Additionally, the FASTRAND III subsystem has an independent position control feature which allows greater flexibility and can be used to decrease the average access time. This is done in a multi-unit subsystem by concurrently pre-positioning the heads in each drum unit. Pre-positioning the heads saves time because once the position instructions have been transmitted to the FASTRAND III unit, the executive can immediately initiate another operation on a different drum unit without having to wait for the positioning operation to be completed. Then, whenever specifically instructed to read or write from a pre-positioned unit, the only time delay is for the latency and address circuit activation. These are the mechanical design features which contribute to the operating speed of the FASTRAND III subsystem.

If the circumstances are such that the head positioning time (resulting from pre-positioning) and the latency are both zero, then the minimum access time is 0.8 milliseconds. This time interval allows certain addressing circuits to be activated.

Normally, such activation starts simultaneously with the other processes involved; in the best case situation described above, it is the only delay. In the case of mean (92 or 67.5 milliseconds) or maximum access time (156 or 102 milliseconds), the addressing circuit time need not be considered because it is overlapped by other time factors.

In any effort to reduce processing time, offline search represents an important advantage. This is the operation in which the computer instructs the FASTRAND III subsystem to locate a specific piece of data, and then goes on with the processing while the storage search takes place. Then, at the termination of this variable length search, the computer is notified and is given the results. The computer may take action as provided in the program.

Also, the FASTRAND III subsystem permits the computer to continue its work while records are being read from or written on the drums.

All data on a FASTRAND III drum is recorded with parity. These parity bits are automatically checked upon reading and created on writing.

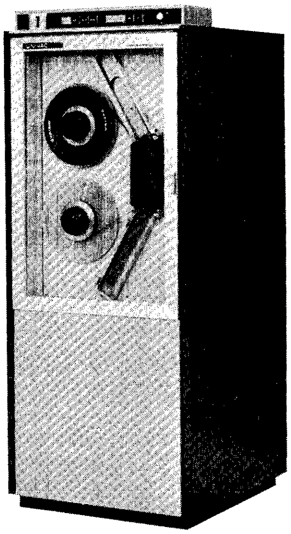
A single FASTRAND III subsystem can accommodate up to eight FASTRAND III drums.

As with the FH-1782 drum subsystem, either paired- or dual-channel operation is possible, providing full scale two-drum unit simultaneity.

The dual-channel subsystem includes two control units. The control units, being logically and electrically independent, provide the redundancy of data paths necessary for truly simultaneous operation. Simultaneous operations can take place on any two drums in the subsystem.

Optional features for FASTRAND III subsystems include 24 additional tracks with fixed read/write heads (Fastbands) for rapid access (35 ms average) and an incremental drum write lockout feature for data protection. By means of this lockout feature, the operator can manually prohibit·writing on 1, 2, 4, 8, 16, 32, or all 192 tracks beginning with the first track of each head.

6.6. UNISERVO VI C MAGNETIC TAPE UNIT



CHARACTERISTICS	
Transfer Rate	8,500/23,700 and 34,200 characters per second. Optional 8,500/23,700 and 34,200 bytes per second
Recording Density	200, 556, and 800 6-bit characters (optional 8-bit bytes) per inch
Tape Speed	42.7 inches per second
Tape Width	0.5 inch
Tape Length	2,400 feet
Thickness	1.5 mils
Block Length	Variable
Space Between Blocks	0.75 inch (7 track mode) 0.60 inch (9 track mode)
Tracks on Tape	7 tracks, 6 data, 1 parity Optional 9 tracks, 8 data, 1 parity
Maximum Number of Units in Subsystem	16
Special Feature	Backward Read
Channels Input/Output	One or two

The UNISERVO VI C Tape Unit is a low cost unit with moderate speed and transfer rates for applications which do not require massive file passing, extensive sorting, or other requirements that dictate the acquisition of higher speed magnetic tape subsystems.

A UNISERVO VI C subsystem can include up to 16 magnetic tape units connected to one or two input/output channels. Paired-channel operations permit a reading operation and a reading or writing operation to be performed simultaneously on any two magnetic tape units.

Backward read capability is standard on all units. The Master/Slave concept is employed in the logic of the UNISERVO VI C subsystem. That is, circuitry has been built into one of the UNISERVO VI C units which allows it to govern up to three other UNISERVO VI C units for certain electronic control functions. In a maximum subsystem of 16 units, there would be four Master units and twelve Slaves.

Data packing density is selectable for 200, 556, or 800 characters-per-inch. The physical tape passing speed is 42.7 inches-per-second, giving maximum transfer rates of 8,540; 23,741 and 34,160 characters-per-second respectively.

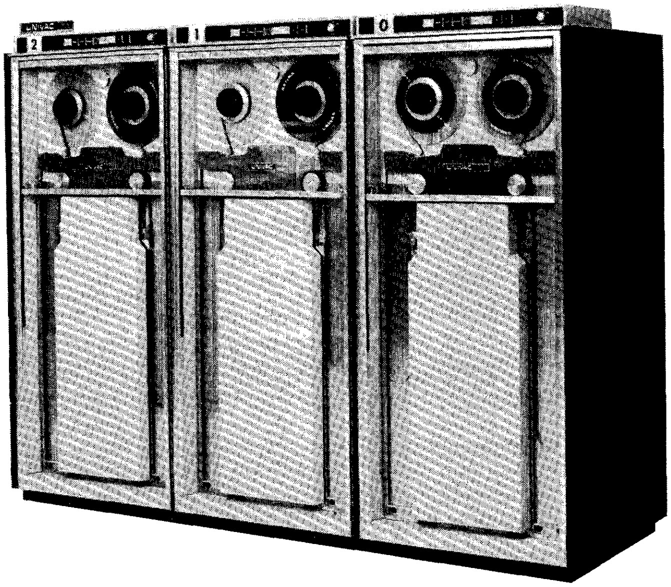
Rewind takes place at 160 inches-per-second, enabling a full reel of 2,400 feet to be rewound in 180 seconds. The 800 character-per-inch packing density normally is used; the 200 and 556 densities being used only for compatibility purposes. At this density more than 11,520,000 characters may be stored on a single reel with data held in 600 character blocks.

Data may be recorded in variable length blocks under program control, with character and block (i.e., horizontal and vertical) parity. A read after write head allows immediate verification of all data written, and under the control of the software Input/Output Handler, repeated read and write operations are automatically undertaken whenever read or write errors occur.

UNISERVO VI C magnetic tape units are fully compatible with IBM* 727, 729 Model I through VI, and 7330 units in seven-track mode, with IBM 2400 Series Models 1 through 3 units in nine-track mode, and with industry compatible units produced by other manufacturers. The UNISERVO VI C control unit can be fitted with a hardware translator to convert tape codes thereby assuring tape compatibility between installations.

* *Trademark of International Business Machine Corp.*

6.7. UNISERVO VIII C MAGNETIC TAPE UNIT



CHARACTERISTICS

Transfer Rate	24,000, 66,720 and 96,000 characters per second (7 track) Optional 24,000, 66,720 and 96,000 bytes per second (9 track)
Recording Density	200, 556, and 800 six-bit characters (eight-bit bytes optional) per inch
Tape Speed	120 inches per second
Tape Width	0.5 inch
Tape Length	2,400 feet (plastic)
Thickness	1.5 mils.
Block Length	Variable
Space Between Block	0.75 inch (7 track) 0.6 inch (9 track)
Tracks on Tape	7 tracks, 6 data, 1 parity Optional 9 tracks, 8 data, 1 parity
Units Per Control	16
Special Feature	Backward Read
Processor Input/Output Channels	One or two

A UNISERVO VIII C subsystem accommodates up to 16 magnetic tape units and may incorporate either one or two control units attached to one or two input/output channels.

UNISERVO VIII C units may be specified with seven- or nine-track mode. In seven-track mode one parity and six data bits are recorded in each frame across the width of the tape. A single 6-bit alphanumeric character or a 6-bit binary value may be stored per frame. In nine-track mode one parity and eight data bits are recorded in each frame across the width of the tape.

Density of 200, 556, or 800 frames-per-inch may be set manually by a switch located on each unit or by program control. Physical tape speed is 120 inches-per-second, giving maximum transfer rate of 24,000; 66,720; and 96,000 characters-per-second (in seven-track mode), or bytes-per-second (in nine-track mode). Rewind takes place at 240 inches-per-second, enabling a full reel of 2,400 feet to be rewound in 120 seconds. The 800 frame-per-inch packing density is normally used; the 200 and 556 densities are used only for compatibility purposes.

Reading may take place with the tape moving either forward or backward – the latter feature often being valuable by averting rewind time especially during sort/merge operations. Writing takes place when the tape is moving in a forward direction only.

Data may be recorded in variable length blocks under program control, with character and block (i.e., horizontal and vertical) parity. A read after write head allows immediate verification of all data written, and under the control of the software Input/Output Handler, repeated read and write operations are automatically undertaken whenever read or write errors occur in an attempt to recover from an error. Thus, programming problems with this tape subsystem are minimized since the circuitry of the control unit, combined with the software Input/Output Handler, deals with all operations except the system response to a nonrecoverable error.

UNISERVO VIII C magnetic tape units are fully compatible with IBM 727, 729 Model I through VI, and 7330 units in seven-track mode, and with IBM 2400 series Models 1 through 3 units in nine-track mode, and with industry compatible units produced by other manufacturers.

The UNISERVO VIII C control unit can be fitted with a hardware translator to convert between tape codes, thereby ensuring tape compatibility between installations.

6.7.1. Fully Simultaneous UNISERVO VIII C Subsystem

The fully simultaneous subsystem includes two UNISERVO VIII C control units each connected to a separate I/O channel of an I/O Module. With this dual-paired channel configuration, any two operations including write/write can be performed. This effectively doubles the maximum subsystem transfer rate provided by a single channel subsystem. In addition it provides a redundant path to the tape units for critical real-time systems.

To provide write/write capability, the tape units in the subsystem must have the fully simultaneous features. They may be either seven- or nine-track models. UNISERVO VI C units are not permitted.

With the addition of a Shared Peripheral Interface up to four C/A Units can access the subsystem.

6.8. UNISERVO VI C/VIII C SUBSYSTEM

The UNISERVO VI C/VIII C Subsystem permits any combination of UNISERVO VI C and UNISERVO VIII C magnetic tape units, up to a maximum of 16, to be intermixed on a single or paired input/output channel. A Shared Peripheral Interface may be attached to permit access from more than one I/O Module.

The UNISERVO VI C and UNISERVO VIII C units in a mixed subsystem may have seven- and nine-channel capability in nearly any unit combination.

The only restriction is that a Slave UNISERVO VI C unit may have the nine-track capability only if the associated Master unit has that capability.

This mixed magnetic tape subsystem enables a highly flexible blend of high speed and economic medium speed tape units to be used in conjunction with each other. In addition, the seven- and nine-track unit options permit data recorded in traditional industry compatible form to be handled while allowing the upgrading of these records in line with the USASCII code and faster decimal handling rates.

The UNISERVO VI C/VIII C subsystem is available in single and dual channel configurations. A dual channel configuration includes two UNISERVO control units. Each control unit requires a separate I/O channel of an IOM. With this configuration, two read operations or a read and a write operation can take place simultaneously. The subsystem is also available in a dual-paired channel configuration.

6.9. HIGH SPEED PRINTER SUBSYSTEM

CHARACTERISTICS

Printing Rate	1600 LPM (26 alpha, 10 numeric, 7 special characters)
	1200 LPM (26 alpha, 10 numeric, 27 special characters)
Maximum Characters/Line	132
Horizontal Spacing	10 characters per inch
Vertical Spacing	6 or 8 lines per inch
Form Advance Rate	33 ips at 6 lines per inch spacing 22 ips at 8 lines per inch spacing
Form Width	4 to 22 inches
Form Length	1 to 22 inches
Number of Form Copies	Up to 6 part continuously sprocketed forms
Form Advance	Loop control
Line Advance	Single and double spacing under program control
Forms Advance	Up to 132 lines per command
Speed of Form Advance	12.5 + 5.2 (N – 1) if set at 6 lpi 12.5 + 5.7 (N – 1) if set at 8 lpi N = number of lines advanced

The High Speed Printer subsystem provides a 1200/1600 LPM printer. This printer, attached to a printer control unit, forms a subsystem. The subsystem can be attached directly to a single-channel of an Input/Output Module or indirectly via a Shared Peripheral Interface. The control unit includes a full line buffer (132 characters) for storing information before printing.

The program controls the printing operation by sending function words to the subsystem specifying the operation and number of lines to be spaced. The subsystem responds by generating program interrupts and supplying the status of the subsystem to the Executive routine for analysis and action.

6.9.1. Line Printer

The printer speed varies from the lower rate for the full 63-character set to the higher rate for an abbreviated character set of 43 characters (A through Z, 0 through 9 and 7 special characters).

6.10. THE UNIVAC 9000 SERIES SUBSYSTEMS

The UNIVAC 9000 Series is a new computer family which embodies many bold, new design concepts in a unified and low-cost line of data processing equipment. The basic UNIVAC 9000 Series computers have been selected for use with the UNIVAC 418-III System for central site punched card reading and line printing peripherals. The basic UNIVAC 9000 Series computers used as online subsystems for the UNIVAC 418-III System are the UNIVAC 9200 and the 9300. The basic UNIVAC 9200 and 9300 may also operate as free-standing offline data processing systems. As such, they may be expanded in the manner of a UNIVAC 9200 and a 9300 to support the corresponding UNIVAC 9000 Series software.

6.10.1. Inter-Computer Control Unit

In order to provide for control signals and data transfer between UNIVAC 9000 Series subsystem and the UNIVAC 418-III System, an Inter-Computer Control Unit (ICCU) is required. The ICCU provides for communication of computer alert signals (interrupts) in either direction and also for a buffer for data transfer. The data transfer can be accomplished in one of the three modes under program control. Figure 6–1 shows these formats.

The ICCU connects to the UNIVAC 9000 Series subsystem through a subchannel on the Multiplex Channel. The ICCU connects to a single channel of an Input/Output Module of the UNIVAC 418-III System.

6.10.2. Modes of Operation

There are two basic modes of operation for the UNIVAC 9000 Series subsystem:

(1) as the card reader/line printer for the UNIVAC 418-III Real Time Operating System

(2) as a free-standing data processing system.

As the card reader/line printer for the Real Time Operating System, the UNIVAC 9000 Series subsystem performs its operation under control of the UNIVAC 418-III System software. The UNIVAC 9200 and 9300 may also be used as an independently operating data processing system. The offline use of the UNIVAC 9000 Series subsystem is accomplished by the fact that the program(s) operating in the UNIVAC 418-III System do not refer to the UNIVAC 9000 Series subsystem and vice versa.

In the basic UNIVAC 9200 System, the I/O units are combined with the processor to simplify installation and minimize the requirements for space, power, and cabling. The system is completely modular and easily expandable to include any of the options provided with the system.

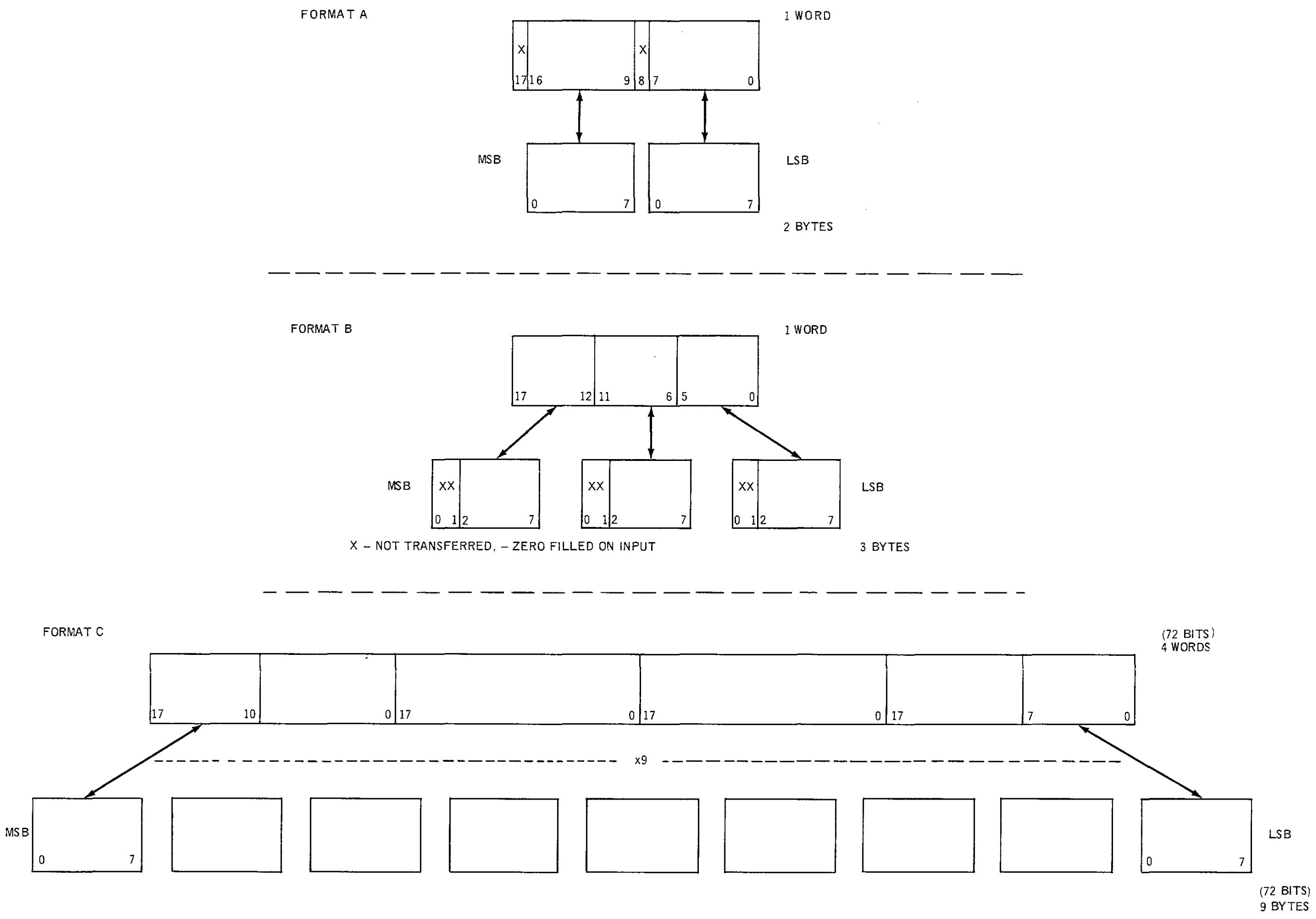
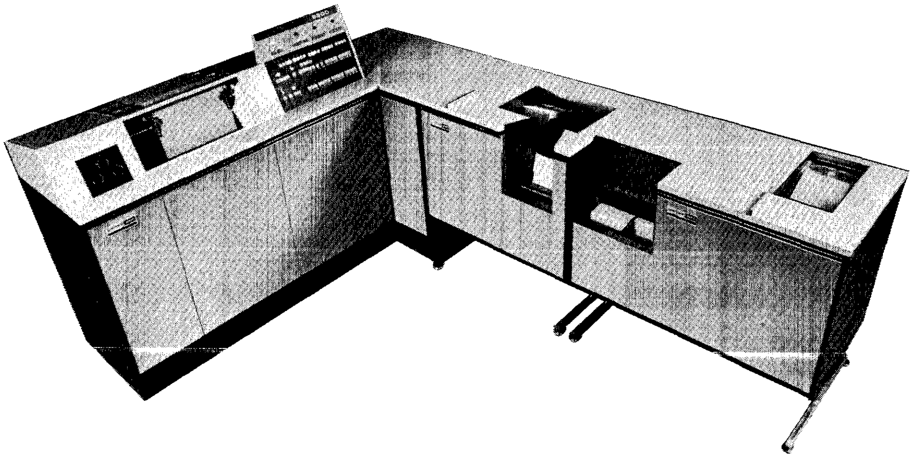


Figure 6–1. ICCU Data Formats

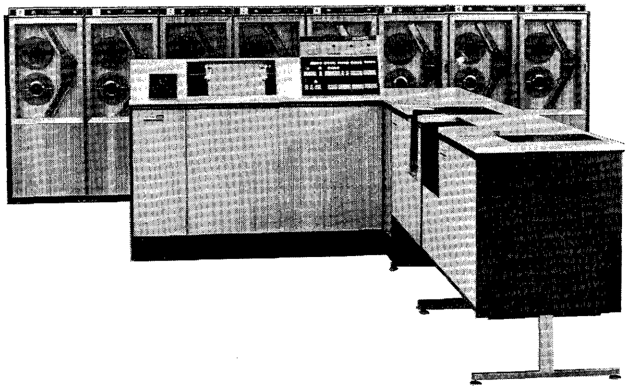
6.10.3. The UNIVAC 9200 Subsystem



CHARACTERISTICS	
System Orientation	Card
Basic Memory	8192 bytes
Memory Cycle Time	1.2μsec
Add (Decimal) Instruction Time (Two 5-Digit Fields)	104μsec
Card Read	400 cpm
Card Punch	75-200 cpm
Print Speed	250 lpm
Overlapped I/O Units	Standard
Multiplexer Channel Rate	85,000 bytes per second

The UNIVAC 9200 subsystem is a complete data processing system. A description is provided in "UNIVAC 9200 System, System Description," UP-4086 (current version).

6.10.4. The UNIVAC 9300 Subsystem



CHARACTERISTICS

System Orientation	Card/Tape
Basic Memory	8192 bytes
Maximum Memory	32,768 bytes
Memory Cycle Time	600 nanoseconds
Add (Decimal) Instruction Time (Two 5-Digit Fields)	52 microseconds
Multiply, Divide, and Edit	Standard
Card Read	600 cpm
Card Punch – Column (optional) – Row (optional)	75-200 cpm 200 cpm
Print Speed (Alpha)	600 lpm
Overlapped Input/Output Units	Standard
Multiplexer (Required) Channel Rate	85,000 bytes per second
Registers (Standard)	Eight for processor functions Eight for input/output functions

The UNIVAC 9300 subsystem is a complete data processing system. For a full description of this system refer to ''UNIVAC 9300 System, System Description,'' UP-4119 (current version).

6.11. HIGH SPEED PUNCHED TAPE SUBSYSTEM

The punched tape subsystem of the UNIVAC 418-III System provides a high speed punched tape photo-electric reader and a high speed punch.

The minimal punched tape subsystem consists of a reader with a speed of up to 1000 characters-per-second. The reader may be equipped with an optional spooling mechanism which provides for bi-directional reading as well as the spooling of tape.

The punch, which is an addition to the minimum reader, provides for tape punching at a rate of up to 240 characters-per-second. It is equipped with a photo-electric post punching read station for punching verification. The punch verifier may be disabled manually by a switch located on the punch unit.

Tape widths of 7/8 inches, 11/16 inches, and 1 inch can be used. The subsystem can process five-level through eight-level codes. The capability for checking odd and even parity or for non-checking of parity when reading or punching is provided.

CHARACTERISTICS	
Reader speed	Up to 1000 characters per second
Reader Tape Spooler	Optional feature
Punch Speed	Up to 240 characters per second
Post punch verification	Standard
Parity Check: odd, even, none	Standard: operator control
Tape Widths	7/8", 11/16", and 1"
Codes	5-level through 8-level
Number of I/O Channels	1

7. COMMUNICATIONS SUBSYSTEMS

7.1. INTRODUCTION

The two basic communications subsystems available for the UNIVAC 418-III System are the Word Terminal Synchronous and the Communication Terminal Subsystem. The Word Terminal Synchronous operates in Internally Specified Index mode and is used to connect a single high speed synchronous communication line to a single IOM channel. The Communication Terminal Subsystem operates in Externally Specified Index mode and is used to terminate a wide variety of low, medium, and/or high speed communication lines. The number of lines per Communication Terminal Subsystem depends on the speed of the lines connected. Each subsystem connects to a pair of IOM channels.

7.2. COMMUNICATION TERMINAL SUBSYSTEMS

The Communication Terminal Subsystem consists of the Communications Terminal Module (CTM) which makes direct connection with the communications facility and the Communications Terminal Module Controller (CTMC) through which the CTM's deliver data to and receive data from the UNIVAC 418-III System. The CTMC may be connected to any pair of channels on an Input/Output Module (except channels 0 and 1). The number (maximum 16) of CTM's which may be connected to a CTMC depends on the CTM and the speed and type of communication lines to which they connect.

7.2.1. Communications Terminal Module

The three basic types of Communications Terminal Module (CTM) are the low speed (up to 300 bps), medium speed (up to 1600 bps), and high speed (up to 50,000 bps). Each CTM is easily adjusted to the speed and other characteristics of the type of line with which it is to operate. The termination of a communication line requires a logical and electrical interface as well as buffering and control circuitry. The modularity provided in the subsystem is achieved by packaging and overlapping the use of common circuits and line clocks.

Each CTM provides for termination of a specific number of lines, depending upon the speed of the line and the line control capability required by the user. For example, the low speed CTM's provide for termination of two input and two output communication lines. These lines may operate in simplex, half-duplex, or full-duplex mode. The basic low speed CTM's provide for line termination with a minimum of line control capability. The enhanced low speed CTM's provide for end-of-transmission recognition, as well as other transmission control functions. There are basic and enhanced medium and high speed CTM's. Some of the enhanced control capabilities are character and message parity generation and checking, end-of-message recognition, and a CTM for automatic dialing control. These CTM's provide for connection to DATA-PHONE* service and Broadband** service, private line and private wire service, and Wide Band Data Service and Wide Band Channel and Service, as well as other available line speeds.

* *Service Mark of the Bell System*

** *Service Mark of Western Union*

7.2.2. Communications Terminal Module Controller

The function of Communications Terminal Module Controller (CTMC) is to provide housing and connection of up to 16 CTM's to a paired channel of an Input/Output Module of the UNIVAC 418-III System. More than one CTMC may be connected to an IOM. The function of the CTMC is to multiplex the CTM's to the paired channel of the IOM on a priority basis. The maximum capability of the CTMC is 80,000 character transfers per second. The number of lines which can be connected through a CTMC depends on the number and variety of lines which are terminated by the CTM's.

The CTM's may request access to the IOM via the CTMC in random order. The CTMC automatically assigns priority among the CTM's requesting access and identifies to the Input/Output Module the particular CTM being granted access. This identification is used by the IOM as the basis for Externally Specified Indexing. The process of data transfer is automatic and self-controlling among the CTMC, the IOM, and main storage and does not require the use or intervention of the Command/Arithmetic Unit.

7.3. WORD TERMINAL SYNCHRONOUS

The Word Terminal Synchronous (WTS) complements the UNIVAC 418-III communications subsystem by enabling it to be used more efficiently for high-speed data transmission over a single communication line. Since data characters are transferred to and from main storage on a fullword basis, three 6-bit characters per word, the I/O transfer time and the size of the area required for buffering are both considerably reduced. However, the most significant advantage of the WTS is that it minimizes manipulation of the data. This is accomplished by having the WTS, rather than the Command/Arithmetic Unit, add character and message parity to outgoing messages and checking character and message parity on incoming messages. If an error in parity is detected, an External Interrupt is presented to the C/A Unit.

This high-speed transmission capability, coupled with the ability to dial connect with any station on the public network complex, handles many of the data communications problems. The WTS allows the UNIVAC 418-III System to exchange data with a remote computer. This configuration can be used for order, entry, and inventory control,and more important, for remote processing capabilities.

7.3.1. Operational Modes

There are three basic modes of operation (leased line high speed, Dial network high speed, leased line Broadband) which can be used with the WTS.

- Leased Line High Speed
  Leased telephone line operation permits continuous data exchange between two fixed locations at a maximum transfer rate of 300 characters per second.

- Dial Network High Speed
  A number of high speed stations can be connected to a WTS, one at a time over dialed (DDD) network circuits. Dial network operation permits data exchange between any two subscriber locations, beginning with a connection procedure (dial) and ending with a disconnect procedure (hang up). Transfers may be initiated by either the WTS or the remote station. If the WTS initiates the trans-

fer, the connection is made automatically by the C/A Unit through the automatic calling option available on the WTS. The remote station automatically answers the call and data transfer can commence. If the remote station initiates the call, the WTS automatically answers the call utilizing the unattended answering option. Data transfers can then take place in the normal manner. All data transfers over dial high speed circuits are at a maximum transfer rate of 250 characters per second.

- Leased Line Broadband

  A single Broadband station is connected to a WTS by a leased Telpak A* line. Leased Telpak A line operation permits continuous data exchange between two fixed locations at a maximum transfer rate of 5,800 character per second. Data transfer can be initiated by either device, providing the remote hardware can handle such an operation. The data format is such that character parity, start and end-of-message limits, message parity and character synchronization are established. The WTS is prepared to receive data at all times.

These modes of operation reflect the influence of presently available communications tariff offerings. The WTS design and packaging philosophy allows for higher speed operation and for operation with communications hardware using more advanced techniques.

The Datasets (201-C, 201-D, 301-B) represent the latest versions of sets which have been on the market for some time. The 201-C succeeds the 201-A, the 201-D succeeds the 201-B, and the 301-B succeeds the 301-A. They are basically the same sets (speed, physical makeup, modulation techniques), these changes reflecting minor updating and modifications and, in the case of the 201-C, automatic dialing compatibility.

CHARACTERISTICS

Speed	2000, 2400, 4,800 bits/second
Data Coding	6 data bits per character
Control Coding	Control characters (Synch, Start-of-Message, End-of-Message, etc.) field selected by plugboard
Communications Facilities	Voice circuit dial or leased at 2000, or 2400 bps respectively. Broadband lease at 50,000 bps.
I/O Channels Required	1 per WTS
I/O Transfer Mode	Full word basis (three 6-bit characters per transfer)

* *Trademark of the Bell System*

8. UNIVAC 418-III REAL TIME OPERATING SYSTEM

8.1. GENERAL

The UNIVAC 418-III Real Time Operating System (RTOS) consists of a comprehensive set of control programs, programming aids, and utility services. It is modular in design to tultill a wide range ot data processing requirements in the area of batch, scientific, and real time applications. Both data and programs to be processed may be introduced to the system from either central or remote locations. The RTOS provides the user with a means to write programs in FORTRAN and COBOL (high-level programming languages) or in a symbolic assembly language.

The capabilities of the UNIVAC 418-III System cover the wide span commonly referred to as the intermediate scale system. The RTOS is designed to provide identical services to a wide variety of hardware configurations. However, system performance is dependent upon the type of facility available.

There are three basic sections in the RTOS. Each section performs a specific service in the overall system processing. The sections are as follows:

- Executive
- Language Processors
- Services and Library

8.2. SOFTWARE SYSTEM DESCRIPTION

To take maximum advantage of the speed and hardware capabilities of the UNIVAC 418-III System and to make effective use of a given hardware configuration, a comprehensive internal operating environment has been created.

This environment allows for the concurrent operation of many programs; it allows the system to react immediately to the inquiries and requests of many different users at local and remote stations; it allows for the stringent demands of real time applications; it is able to store, retrieve, and protect large blocks of data; and it makes optimum use of all available hardware facilities, while minimizing job turnaround time.

Only through central control of all activities of the UNIVAC 418-III System can this environment of the combined hardware and software systems be fully established and maintained to satisfy the requirements of all applications. The responsibility for efficient, flexible, centralized control is borne by the executive of the RTOS.

The executive, which controls and coordinates the functions of this complex internal environment, presents a relatively simple and uniform interface to the programmer. This permits easier use of the system and eliminates concern for the internal interaction between the user program and other coexistent programs.

8.2.1. Modularity

The technical capabilities of the UNIVAC 418-III executive span a broad spectrum of data processing activities. Its design is such that no penalties are imposed upon any one of these activities by the support provided for the other activities, and an installation not interested in utilizing the full spectrum may specify capabilities to be eliminated at system generation time.

8.2.2. Real Time Processing

The RTOS provides a variety of methods for transferring data between main storage and input/output devices. Foremost among these methods is the capability in support of real time processing. The Real Time Communications Control (RTCC) provides comprehensive controls for data transfers between main storage and remote terminals connected either to a Communications Terminal Modular Control (CTMC) Subsystem or to a Word Terminal Synchronous (WTS) Subsystem.

The RTCC is a generalized communications input/output system which offers the programmer several levels of interface to the communications environment. A user need no longer be concerned with the variety of characteristics of the many different terminal devices. To the programmer, data may now consist of queues of messages which are simply input or output. The user is thus free from the constraints of the normally unpredictable time and order of arrival and departure of messages to and from the central computer.

8.2.3. Batch and Scientific Processing

Complementing the real time capabilities of the RTOS are batch and scientific processing. Design emphasis has been placed upon the achievement of ease of run preparation and submission, minimization of job turnaround time, and minimization of operator intervention and decision requirements.

8.2.4. Multiprogramming

Run submissions may come from both central and remote stations. The coordination and construction of run decks in conjunction with the internal control of efficient multiprogramming techniques affords the user the best possible results from the multiprogramming environment. A run deck may contain one or more jobs. Jobs within a run are executed serially, while runs are processed concurrently, thus achieving multiprogramming.

It should be noted that batch, scientific, and real time programs are processed concurrently by the RTOS whenever sufficient facilities are available; hence, the user of any one mode experiences little variation in his turnaround time, regardless of the proportionate mix with other types of processing.

8.2.5. Use of Auxiliary Storage

The RTOS is drum-oriented, which means that libraries, temporary tables, processor scratch areas, run files, etc., are utilizing drum areas to further enhance the system's capabilities. Elements of both the RTOS and user programs are stored on the drum.

8.2.6. Facility Control

The executive provides for dynamic acquisition and release of all of the system's facilities. The user is given both an internal and an external interface for facilities control. It is through the efficient management of the system's facilities that a high utilization of these can be achieved. Many elements of RTOS take part in this facility control, yet all are channeled eventually to the allocators of the job control section (see Facilities Allocation 9.4.2).

Available facilities and their disposition are specified to the system at system generation time; thereafter, the executive assigns these facilities, as needed and as available, to fulfill the facilities requirements of all runs entering the system.

8.2.7. Utilization of Mass Storage and Drums

Provision is made for the creation and maintenance of permanent data files as well as permanent program files on mass storage and high performance drums. This includes facilities for updating, modification, and manipulation of these files.

8.2.8. Control Stream

In order to control the manner in which programs are executed, manipulated, assembled, or compiled, a series of control cards (control stream) has been defined to describe the type of work which is to be performed by the RTOS. These control cards are processed by a group of routines which are referred to as the job control routines of the RTOS. The job control routines determine the order in which programs are loaded and executed, the availability and assignment of facilities, and provide the user with control information of the concurring program(s).

8.2.9. Input/Output

The input/output control elements of the RTOS are responsible for controlling the activities of all ISI input/output channels. Basically, the user is provided with two types of interfaces. One is the input/output device handlers which control the operations of magnetic tapes, high performance drums, mass storage, console, paper tape, and intercomputer coupler, as well as the symbiont/cooperative handlers for card reading, printing and punching. The other provides for a higher level interface in form of a file control subsystem. File control has been designed to assist the user with his file handling on such devices as magnetic tapes, high performance drums, and mass storage.

8.2.10. Languages

Programs for the UNIVAC 418-III Real-Time Operating System may be written in three different languages: a powerful symbolic assembly language and for the nonmachine-oriented programmer, two higher level languages, COBOL and FORTRAN are provided.

8.2.11. Utilities

Utility and support routines comprise an important part of the RTOS. Debugging aids are provided in the form of main storage dumps, drum dumps, mass storage dumps, and tape dumps. There is a means to inspect storage and display on the operator's console allocated facilities. Through control card and unsolicited typein, an activity may be suspended and restarted.

A complete package of debugging aids is offered in the executive independent utilities. This is a highly useful group of services which are particularly helpful during initial checkout of programs. Support for file and data handling is provided through the sort/merge routines and the independent symbionts.

8.2.12. Operator Communications

The RTOS has been designed for operation with a minimum of operator intervention. However, it is recognized that some functions are beyond the scope of any of the elements of the system, while others demand operator concurrence. In addition, certain information must be presented automatically to the operator while other information requires definite answers from the operator to ensure continuation.

8.2.13. System Setup

The system setup consists of three basic parts: a parameter deck defining the hardware facilities, the operating system components, and the systems generation routines of the RTOS. The user is thus offered a means of generating an operating system tailored to his particular needs and desires. All this can be done with a minimum of effort and time.

9. EXECUTIVE

9.1. GENERAL

The purpose of the executive of the UNIVAC 418-III Real Time Operating System (RTOS) is to provide control routines which enable the users of the system to perform their tasks conveniently and efficiently. To that extent many standard facility routines are included in this executive. In order to make efficient use of the UNIVAC 418-III hardware the executive controls the use of all peripheral devices, controls the sequence and manner in which programs are entered into the machine, controls the processing of contingency errors, and determines the priority in which one or more programs are given control. The various control functions of the executive may therefore be divided into the following categories:

- Executive Control
- Input/Output Control
- Job Control
- Real Time Communication Control

9.2. EXECUTIVE CONTROL

Executive control consists of a series of interlinked routines which direct the flow of traffic within the executive. A multiprogramming capability is provided by scheduling work according to a priority scheme. An internal clock is maintained allowing time interval procedures. All interrupts are analyzed and scheduled for processing by the applicable routines. The interface with the user programs is accomplished through the use of specific calling sequences and the supervisor call interrupt feature.

In order to make the most efficient use of the hardware, the execution of instructions and the transfer of data through the input/output channels is overlapped whenever possible. Since most peripheral devices are not able to approach the data transfer rates of the IOM, input/output data transfers from and to different peripheral devices are also overlapped, or accomplished simultaneously. This method helps in utilizing the Command/Arithmetic Section and the Input/Output Section of the hardware to the fullest extent.

Since very few programs are written to allow extensive overlapping of the Command/Arithmetic and Input/Output Sections, the multiprogramming technique is used. Thus, while one or more programs are residing in main storage at the same time, only one has control at any one point in time. All others are temporarily suspended, either waiting for the completion of an I/O operation, or waiting until the executive scheduling mechanism returns control.

To support this multiprogramming capability a group of control programs is necessary. A functional division of these routines is as follows:

- interrupt control
- priority control
- real time clock control
- supervisor call interrupt control
- overlay control
- queue control
- contingency error control

9.2.1. Interrupt Control

The automatic interrupt is the very heart of the executive system. It is the vehicle for communication between the user program and the executive. Program malfunction, hardware malfunction, input/output termination and time interval interrupts permit effective monitoring of the system.

An interrupt is a signal which results in causing the central processor to transfer control to a specific location in main storage and to lockout further interrupts until released. The interrupt location is a hardware-fixed location which contains a Store Location And Jump instruction provided by the executive.

The executive is designed in such a way that it will receive control as the result of any hardware interrupt. The interrupt control routine ensures the integrity of the interrupted program before releasing control to another activity.

9.2.2. Priority Control

The priority control consists of those elements of the executive which determine the order in which the various jobs and activities in a multiprogramming environment are to be given control over the Command/Arithmetic Section of the hardware. A job or an activity may be a program or may consist of independent sections of code within a program. A program may be a part of the RTOS or it may be a user program. Priority control selects the next activity to be initiated whenever the activity currently in control is interrupted or terminated.

Basically, there are three types of programs that are scheduled by the priority control section:

- interrupt processing routines
- suspended programs
- scheduled programs

When one or more interrupts are registered by the interrupt control section, control is given to the corresponding interrupt processing routine of the executive with the highest priority. When no interrupt processing routines have to be scheduled, control is given first to any suspended program and then to the scheduled program with the highest priority.

9.2. 3. Real Time Clock Control

The real time clock control consists of the interrupt processing routine for real time clock interrupts and maintains the time of day. This routine receives control as a result of the real time clock interrupt. It determines the time elapsed since the last interrupt and adds this to the time of day. Time of day is kept in binary seconds and 200 microsecond intervals based either on the initial value established by the console day clock or by the operator upon initial load in the absence of the day clock.

The real time clock control determines if any time-dependent activities are ready to be scheduled. If so, clock control schedules these activities. The day clock interrupt is used to synchronize the real time clock with the day clock each minute.

9.2.4. Supervisor Call Interrupt Control

The execution of several special function codes causes a supervisor call interrupt. A certain number of these are recognized by the executive as special software requests. The supervisor call control section checks these functions and passes control to the various processing elements. The supervisor call interrupt is also generated when one of the optional instructions is executed in the absence of the required hardware.

Supervisor calls are divided into three principal types:

- Type one – dispatcher requests
- Type two – input/output requests
- Type three – optional hardware instructions

All others are considered illegal instructions and will cause termination of the offending program.

9.2.4.1. Dispatcher Requests

The use of the supervisor call as a special request may be thought of as an extended repertoire or perhaps an executive language. One category of this extended repertoire is called the dispatcher requests. Code generated for these dispatcher requests starts with the major function code of 077 and is followed by a system defined minor function code consisting of the remaining 12 bits (077xxxx).

The dispatcher requests provide the user with a method of communication with the executive and its services. These services help to:

- Relieve the user of rewriting often used routines.
- Provide a means to change priorities.
- Provide the user with a scheduling mechanism in relation to time and addresses in main storage.
- Provide the user with a dynamic method of facilities acquisition and release.
- Simplify the user/executive interface.

9.2.4.2. Input/Output Requests

Input/Output requests consist of the various parameter packets which provide the basic user interface to all peripheral equipment (on site or remote). All of the supported input/output equipment is controlled by what is generally called a handler. The term handler designates a routine whose sole function is to control input/output operations under the control of the executive.

9.2.4.3. Optional Hardware Instructions

Some of the optional hardware instructions (floating point arithmetic commands or binary/decimal conversion commands) might not be incorporated in a particular configuration. In that case the use of these instructions will cause supervisor call interrupts. When these interrupts occur, the executive passes control to a library overlay which performs an equivalent function.

9.2.5. Overlay Control

Executive overlays are those program elements which need not be resident at all times but must be available and loaded in the fastest possible way. Executive overlays are premapped based on the size of a particular system and occupy the same relative main storage areas. Overlay control consists of that area of the executive which controls the overlay or overlays that are currently in a specific area, loads new or different overlay programs, and controls the release of overlays. The main storage area occupied by one or more overlays is called an overlay area.

Executive overlay programs have the following general characteristics:

- They will not occupy an overlay area for any extensive length of time.
- They are not the type of program users will call upon frequently during the execution of a job.
- They should not be time critical.
- They are reusable upon completion and need not be reloaded.

In order to load overlay programs as fast as possible a directory is maintained which indicates where on the system's drum the overlay is stored.

9.2.6. Queue Control

Queue control is that element resident in the executive which allows variable-length buffers to be obtained, used, and released by system programs. Queue control maintains main storage buffers in blocks of 512 words which it obtains through the main storage allocator. Space in these buffers is allocated to various system routines which request buffers through special calling sequences. These buffers have a minimum length of 4 words and a maximum length of 508 words. Queue control keeps track of the unassigned areas which are maximized whenever a buffer is released.

9.2.7. Contingency Error Control

Contingency errors are errors detected by hardware or software which result in external interrupts at fixed addresses. Contingency error control, as a result ot these errors, receives control and appropriate action is taken for each type of interrupt.

Contingency error control attempts to preserve the integrity of the RTOS whenever a hardware or a program fault is detected. It also attempts to provide for the orderly shutdown of the active programs whenever possible.

The executive will not terminate a high priority real time program in the event of a contingency error if the program has a specific activity scheduled at the occurrence of the error.

The contingency error control routine provides one of the following basic functions upon detection of an error:

- Abort the offending program with an error message.
- Attempt an orderly shutdown of the RTOS.
- Provide a means for the user, so that he may specify an activity to be scheduled in the event that a program fault or recoverable hardware fault is detected.

There are several contingency errors that will cause contingency error control to attempt an orderly shutdown of the RTOS. Depending upon the severity of the contingency error and on the degree of success of the shutdown, there will be an attempt to restart the RTOS. In the instance of a power loss interrupt, input/output operations in progress will be allowed to continue, but no new operations will be initiated.

9.3. INPUT/OUTPUT CONTROL

Input/output control is designed to control and direct interactions between the processing unit and input/output equipment. It provides the user of the UNIVAC 418-III Real Time Operating System with routines for reading and writing data records. Its more important advantages are reduced programming, efficient routines, freedom from programming errors, and standardization of data handling. Input/Output control provides the user with two levels of interface to the input/output equipment of the system.

- handler level
- file control level

Each level of interface offers distinct advantages. The choice of one level or the other, or even a mixture of both depends on the user's requirements and on the degree of control he wishes to exert over the input/output equipment.

Functionally the elements of the input/output control consist of the following three components:

- input/output device handlers
- cooperative/symbionts
- file control

9.3.1. Input/Output Device Handlers

The input/output handlers are a set of routines which control input/output operation in the UNIVAC 418-III executive according to the specified parameters. Each handler is designed for a specific type of input/output device.

The purpose of the input/output handler is to simplify the task of tsansferring data between the computer and its peripheral equipment. This precludes the necessity of a user writing his own code for each input/output operation and reduces user code for error checking and analysis.

Each handler has the ability to initiate an input or output operation at the request of a program in control and to monitor this operation until its completion has been indicated by a processor interrupt. The handlers also interpret error conditions and attempt recovery whenever possible.

9.3.1.1. Value of a Handler

The use of a handler is economical in several ways. It prevents duplication of programming effort because only one routine is required to handle all input/output operations for each type of equipment. The use of one program for control saves main storage space when there are several programs running. The coding is in main storage only once and is not duplicated in each individual program.

The use of handlers has an additional value in a real time system. In many systems, program operation is often held up while awaiting I/O. In the UNIVAC 418-III System, a program gives up control to the executive when it requests an input or output operation, and another program may be allowed to run during the previous program's I/O time. The time at which a running program requests the use of a handler is a convenient point to transfer control to a second program. When a handler signals that the operation is complete, control can be returned to the first program.

9.3.1.2. Supported Input/Output Equipment

The following list of input/output equipment is currently being supported by I/O handlers in the RTOS.

- Console
- Magnetic Tape (VI C and VIII C)
- FH-type Drum (FH-432, FH-880 and FH-1782)
- FASTRAND (II and III) Mass Storage Subsystems
- Paper Tape
- Intercomputer Coupler

The handlers are basically similar in operation. Each operates in two phases, an initiation phase and an interrupt analysis phase.

9.3.1.3. Initiation Phase

An I/O operation is initiated when executive control transfers control to the handler with the address of the parameter packet. The handler decodes the user's packet, validates and stores the information necessary to monitor the operation, sends the function and initiates the data transfer before making exit with an indication that the operation has been initiated but is not complete.

9.3.1.4. Interrupt Analysis Phase

When an interrupt occurs on an input/output channel, control is given immediately to the related handler's interrupt analysis routine. A check is made to see if the interrupt can be processed immediately or whether it is to be logged for later processing due to the executive program (critical) being in control. When the interrupt is processed, the related status word is analyzed for successful completion or for an error condition. If the operation was successfully completed, the user's completion address is scheduled and control is given to the executive. If an error condition exists, the handlers will attempt to recover whenever possible. When recovery from the error condition is not possible, a user-assigned address is scheduled with an error status and control goes to the executive.

9.3.1.5. Console Handler

The console handler allows user programs to display information, communicate with the operator, request information, and receive unsolicited messages. The handler provides a simple means of initiating executive or support services or user programs on an unsolicited basis. It provides for the orderly keyboard and printer input/output in a multiprogramming environment.

9.3.1.6. Magnetic Tape Handler

The magnetic tape handler allows user programs to handle all magnetic tape input/output operations under control of the executive. A magnetic tape subsystem accomodates up to 16 tape units and may consist of one or two (dual IOM channel) control units. The magnetic tape handler in the RTOS has the capability of controlling one or more magnetic tape subsystems in any of the following configurations:

- UNISERVO VI C Subsystem
- UNISERVO VI C/VIII C Subsystem
- UNISERVO VIII C Subsystem

The interface between a magnetic tape subsystem and an IOM may have any one of the following configurations:

- Single channel interface (18 bits)

  This type of interface requires only one channel of an IOM and utilizes one magnetic tape control unit. Therefore, 18 bits of information are transferred at a time.

- Paired channel interface (36 bits)

  This type of interface requires two channels of an IOM and utilizes one magnetic tape control unit. Therefore, 36 bits of information are transferred at a time. The assigned IOM channels must be an even-numbered channel and the next highest odd-numbered channel.

- Dual-single channel interface (2 x 18 bits)

  This type of interface requires two channels of an IOM and utilizes two magnetic tape control units. One IOM channel is used for each magnetic tape control unit. This configuration permits simultaneous 18-bit transfer operations.

- Dual-paired channel interface (2 x 36 bits)

  This type of interface requires two paired IOM channels (four channels of an IOM) and utilizes two magnetic tape control units. One paired IOM channel is used for each magnetic tape control unit. This configuration permits simultaneous 36-bit transfer operations.

The magnetic tape handler provides for data to be written or read on seven or nine tracks across the width of a magnetic tape. The recording densities are 200, 556, and 800 bpi for seven track, and 800 bpi for nine track operation. Reading may be either in a forward direction or in a backward direction at normal or low gain. Writing is in a forward direction only and should be at low gain. The following list contains the commands available in the magnetic tape handler:

- read tape forward
- read tape backward
- write tape
- skip-write tape
- rewind tape
- rewind tape with interlock
- write end of file on tape

9.3.1.7. Magnetic Drum Handler

The magnetic drum handler provides the user with the ability to perform all input and output operations on high performance flying head type drums under the control of the executive. The magnetic drum subsystem serves also as the auxiliary storage device for the RTOS and contains the systems library, directories, reference tables, symbiont file storage, processor scratch areas and run libraries. All drum area, temporary or permanent, is allocated through the drum allocator. Even though this drum allocator is functionally a part of the job control routines, a close operational relationship exists between the drum allocator and the magnetic drum handler.

The magnetic drum handler provides control over operations on the following equipment:

- FH-432 drum subsystem
- FH-1782 drum subsystem
- FH-880 drum subsystem
- FH-432/1782 drum subsystem

These drum subsystems provide the RTOS with a large capacity, word addressable, random access storage medium. The subsystem consists of a control unit and from one to eight drum units.

The magnetic drum handler requires the standard I/O calling sequence for all drum operations. The handler interprets the parameters of the user's calling sequence and validates the contents of that information. The user is notified of the results of the operation by use of status codes. The user specifies logical drum addresses in the calling packet and the drum handler converts these logical addresses into absolute references before initiating any drum operations. If the requested operation references noncontiguous blocks on drum, the handler automatically divides the requested function into separate operations to be performed consecutively. The following is a list of the commands available in the magnetic drum handler:

- drum write
- drum read
- drum search
- drum block search
- drum search read
- drum block read
- drum block search read

9.3.1.8. FASTRAND Mass Storage Handler

The FASTRAND mass storage handler provides the user with the ability to perform all input and output operations on the FASTRAND mass storage subsystem under the control of the executive. The FASTRAND is of particular importance to the real time user who requires extensive storage space since it provides enough mass storage to allow for multiple levels of backup for transaction as well as history files. All FASTRAND area, temporary or permanent, is allocated and controlled by the FASTRAND element of the drum allocator.

The handler is designed to either support a FASTRAND II or a FASTRAND III mass storage subsystem.

A FASTRAND mass storage subsystem consists of a control unit and from one to eight storage units. In this large capacity random access storage medium, the smallest amount of addressable data is a sector which contains 56 UNIVAC 418-III words or 168 characters. When writing data to FASTRAND and using buffers which do not contain a multiple of 56 words, the remainder of the final sector will be padded with zeroes.

The handler requires the standard input/output calling sequence (packet) for all of its operations. The handler analyzes, interprets and validates the I/O packet parameters and notifies the user of the results of the operation through the use of status codes.

The handler is designed in such a way that FASTRAND mass storage addressing is binary continuous throughout a drum unit, with each address related to a separate sector of data.

The Fastbands are an extra cost option. These Fastbands are additional data tracks accessed by fixed data heads. Since no boom movement is required to position these heads, the access time for Fastbands is reduced to the latency time.

The following is a list of the commands available in the FASTRAND handler:

- continuous write
- continuous read
- position
- long search, first word
- long search, all words
- short search, first word
- short search, all words

9.3.1.9. Paper Tape Handler

The paper tape handler provides the user with the ability to perform all read and punch operations on the high speed paper tape subsystem, under control of the executive. The subsystem allows for the processing of 5 through 8 level paper tape code. The user interfaces to the subsystem through the standard I/O calling sequence. The paper tape handler analyzes, interprets and validates the call packet parameters, and notifies the user of the results of the operation through the use of status codes. Besides the various read and punch functions, the capability for checking odd and even parity or for nonchecking of parity when reading or punching is provided in the paper tape handler.

9.3.1.10. Intercomputer Coupler Handler

The intercomputer coupler handler provides an interface between programs running under control of the RTOS and programs operating in other computers or another UNIVAC 418-III System. The prime purpose for a computer coupler handler is to provide a standard interface to any user program in a multiprogramming environment and to achieve the highest possible rate of data transfer between the two participating systems. In general, computer systems that are candidates for such an intercomputer operation are free from the typical constraints which normally govern peripheral handler operation. Thus, a procedure has been defined as a basis for the design of the intercomputer coupler handler. This procedure is quite simple and provides for the following capabilities:

- defines data block formats
- defines character/word formats
- defines control functions
- defines a control procedure
- allows for data character translation

The design of this intercomputer coupler handler is not in any way constrained by any particular feature of the RTOS.

9.3.2. Cooperative/Symbiont Structure

The cooperative symbiont structure consists of a series of routines which provide a means for both the user and the operating system to interface with the card reader, printers, and card punches connected to a UNIVAC 418-III System. The idea of the cooperative operation is simply to buffer data from input peripherals (card reader) to the systems drum area and to transfer data files from the systems drum area to the output peripherals. A user request for an input card image, for example, causes a data transfer from the run file on the systems drum to the user's program. This method not only provides the user with faster access to the input data, but also ensures that multiple jobs have access to the associated input files without unnecessary delays for available peripherals. The same design logic is used for output operations. A user request for a punch or print operation causes a data transfer from the user's program to the run file on the systems drum, while the system symbionts output the run file to the peripherals. Normally, card read, card punch, and printer peripherals are part of the central site. The RTOS offers in addition to the central site peripherals a remote symbiont capability.

The cooperative/symbiont structure of the RTOS consists of the following elements:

- cooperative
- symbiont control
- symbiont device handlers
- remote symbionts
- independent or utility symbionts

9.3.2.1. Cooperative

The prime purpose of the cooperative is to react to user print, card read, and card punch functions. For output operations it builds a print and/or punch file on the systems drum for the run requesting the service. The cooperative controls the volume in the output files and causes the actual start of printing or punching. Input files are always established before any reference is allowed to be made.

The cooperative satisfies requests for card read operations by reading card images from the run file on the systems drum. Part of the cooperative control functions allow for the acceptance of unsolicited operator messages which cause the suspension, termination or restarting of print, punch, or read operations.

9.3.2.2. Symbiont Control

As the name implies, symbiont control is primarily responsible for the interface of the run files to the appropriate symbiont device handlers. On input files, symbiont control receives card images from the device handlers and builds input run files on the systems drum. For output files, symbiont control is responsible for the loading and termination of the symbiont device handlers. Symbiont control also provides for the initiation and control of the remote symbionts.

9.3.2.3. Symbiont Device Handlers

Operation of the various symbiont device handlers is controlled by symbiont control. There is a symbiont device handler for each type of peripheral device on the system regardless of the total number of such devices. These handlers perform the actual read, print, and/or punch operations.

The primary symbionts in the RTOS provide for the following functions:

- card reader to drum
- drum to high speed printer
- drum to punch

9.3.2.4. Remote Symbionts

To fully utilize the capability of the RTOS, a remote symbiont capability is included in this system, Under RTOS, the Real Time Communication Control (RTCC) elements provide handlers for remote batch terminal operations. This capability has been extended into the cooperative symbiont structure. The remote symbiont routines provide an interface between symbiont control and the RTCC remote batch terminal handlers. The remote symbionts format data received from either symbiont control or the RTCC batch terminal handler and pass the data to symbiont control or the remote batch terminal handlers. The remote symbionts in the RTOS provide for the following functions:

- card to drum
- drum to print
- drum to punch

9.3.2.5. Independent or Utility Symbionts

The independent or utility symbionts provide a second level of support in the cooperative/symbiont structure. These routines supplement the primary and remote symbionts in such areas as drum file overflow, peripheral equipment utilization during off periods, and offline run file preparation. In addition to the systems drum and the standard read, print and punch devices, the independent symbionts utilize available magnetic tapes. The following functions are provided in this second level of symbiont support:

- card reader to magnetic tape
- drum to magnetic tape
- magnetic tape to drum
- magnetic tape to printer
- magnetic tape to punch

9.3.3. The File Control Subsystem

The file control subsystem is a set of routines which exercises centralized control over operations on files within the system. It automatically overlaps input/output operations and processing. In addition, it provides a scheme for reading and writing groups of data (any number of machine words) from or into input/output files, irrespective of the type of storage device or of the physical arrangement of data in the file.

The file control subsystem of the RTOS assists the programmer in achieving maximum efficiency in handling the mass of data that is processed in an installation.

To attain this objective, file control facilities have been designed to provide systematic and effective means of handling, identifying, reading, manipulating, storing, and retrieving all data being handled or processed by the RTOS.

The file control facilities can be grouped into two major categories:

- file control module
- file access module

9.3.3.1. File Control Module

The file control module provides a comprehensive group of commands that features automatic and efficient control of many data processing operations previously performed by programming personnel as clerical tasks.

The primary function performed by the file control module is to provide an interface between the user program and the file access module.

In order to perform its function, the file control module requires a set of control information which describes the files to be processed.

The file specifications must be in the format required by the file access module using given information. Selected elements of the file access module are then mapped, collected, and loaded with the user's object program.

This method allows the user variation and flexibility in the file description and the systems processing procedure for each computer run. Hence, many different kinds of files can be processed by the same program with no changes except in the file descriptions furnished to the file control module.

9.3.3.2. File Access Module

The file access module provides functions which offer a major expansion in the file handling over competitive systems.

Input/output routines are provided to efficiently schedule and control the transfer of data between main storage, mass storage, and magnetic tape devices.

Input/output routines are available to perform the following functions:

- Randomly seek records in a file.
- Access given files randomly, sequentially, or through a combination of both.
- Locate records.
- Perform file maintenance in an index-sequential access method.
- Read data.
- Write data.
- Block and unblock records.
- Create/expand files.
- Overlap read/write and processing operations.
- Check for the presence or absence of labels on input magnetic tape files.
- Verify labels for prescribed content.
- Write prescribed labels on output magnetic tape files.
- Check file sentinel records appearing on input magnetic tape files.
- Write sentinel records on output magnetic tape files.
- Automatically position and reposition files.
- Detect error conditions and correct them when possible.
- Provide exits to user-written error, end of file, and label routines.

The programmer and the installation can select from among three methods of file access to obtain a group of facilities tailored to their processing requirements.

These access methods and the devices they support are:

- sequential – magnetic tape, FH-Drums, FASTRAND mass storage
- direct access – FH-Drums, FASTRAND mass storage
- index-sequential – FH-Drums, FASTRAND mass storage

Each access method supplies a comprehensive group of macro instructions that permit the programmer to specify input/output requests with a minimum of effort; the programmer need not be concerned with learning the individual access characteristics of the many input/output devices supported by the system.

9.4. JOB CONTROL

Job control provides the user with control of the sequence and manner in which programs are entered and the allocation of hardware facilities which are to be available during the operation of programs. It interfaces the user with the executive by directing the execution of the individual jobs of the run and by relaying operational information concerning the run and the job. This interface may be requested via job stream or program control or a combination of both. These requests may be for the manipulation of elements, allocation of facilities for the run or the job, and loading of user or system jobs for execution. If program control is chosen, simple dispatcher calls are available to the user. If the job stream is selected, parameter information is introduced to job control in the form of run control cards. Normally, jobs are loaded in the order in which they are submitted to the system. However, once a job is loaded, the priority indicator or the availability of facilities determines the sequence or urgency in which execution takes place with respect to other jobs in other job streams.

9.4.1. Job Stream

The job stream conveys information concerning a run to job control. It is prepared by the user of the system and provides a basic control for coordinating and executing system and user programs. The job stream consists of control information specifying operations to be performed, limited data, source code for the language processors, program parameters, and other information pertinent to the run. The job stream may originate from card devices, magnetic tape devices, remote devices or mass storage when prestored as an element of a permanent file. The I/O cooperatives feed the job stream to job control without regard to the origin. This enables modularity and flexibility in a multiprogramming environment.

9.4.2. Facilities Allocation

Job control assigns and controls all hardware facilities under the executive. It analyzes requests for peripheral equipment, main storage, and mass storage which may be assigned for the job or all subsequent jobs in the run. Mass storage may also be permanently assigned and catalogued by name for use by more than one run over a period of time. Job control returns released facilities to the inventory for reassignment. The allocation of facilities is available to both system programs and user programs and can be referenced internally or through the job stream.

9.4.3. Element Manipulation

The smallest logical unit of information which may be entered into the system is an element. Each run has its own run library which is available for accummulation of its elements. Job control enters the elements in the run library when requested by the control information specified in the job stream. These elements may be input from the job stream directly, magnetic tape, or permanent mass storage files.

The four basic types of elements are:

- Source Element

  A source element consists of any collection of card images. In addition to the programming languages processed by COBOL, FORTRAN and the 418-III Assembler, the job stream may be manipulated as a source element.

- Relocatable Element

  Relocatable elements are produced by the language processors in a format designed for speed and ease of loading. They represent processed source language programs or subprograms which may be complete or which may be dependent upon collection with other relocatable elements before utilization as a job execution.

- Procedure Element

  Procedure (PROC) elements contain one or more PROC's coded in the 418-III Assembler language. The 418-III Assembler references these PROC's when it has determined that they are not present in the source element.

- Map Element

  The map element is used by the job loader when preparing a job with the same name for execution. The map element specifies element grouping, segmentation and special entry points.

All of these elements may also be copied from tape or permanent mass storage files to tape or permanent mass storage files. With the exception of relocatable elements, these elements may be updated with corrections, insertions and deletions when being input or copied. Again this is dictated by the control information contained in the job stream.

Job control also outputs the elements as specified in the job stream. The destination of these elements may be a card punch device, a magnetic tape unit or a permanent mass storage file. No corrections may be performed when elements are being outputed. However, during any element manipulation, printed listings may be acquired.

9.4.4. Job Execution

Job control provides an efficient and flexible means for preparation and activation of the relocatable elements which comprise the job. Equally important is the method of removing the job from the active system once it has reached its termination point.

9.4.4.1. Job Loading

Job control provides a modular load control system, capable of handling relocatabl elements produced by the assembler and the FORTRAN and COBOL compilers in any combination of the three. The element(s) are prepared as a job for execution under control of RTOS. The user may control this preparation by the creation of an appropriate map element. This map element controls the elements which are to be grouped in the same bay, the elements which are in segments, and the special entry points of the segments which are to cause segment loading prior to transfer of control.

Before the load process begins, the desired environment of the run and the system have been established via job stream and system generation, respectively. Part of this environment may be the existing facilities of the run which are available to the job. The relocatable elements needed for the job and the map element (if desired) must be present in the run and/or system libraries. The libraries are reference in the stated order so that override may be controlled and predicted.

The load process is basically performed in two separate phases. The first phase is a planning one with the collection and mapping of the elements being performed in conjunction with the map element (if one exists). The results of this phase are saved in the run library for the second phase. The second phase is the actual retrieval of the elements from their respective libraries. This phase is not performed until the facilities needed by the job can be assigned. By having these two separate phases, much of the load process may be performed before the job is actually brought into the system.

9.4.4.2. End of Job

Job control provides the user with a method of removing his program from the active system once it has reached its termination point. The type of this termination may be abort, error, normal or successor job. When the type is abort, the entire run is terminated after the job is terminated normally. When the type is error, the contents of the main storage occupied by the job are printed out prior to the job terminating normally. When the type is normal, all outstanding activities (including I/O) are terminated, and the facilities acquired for the job are returned to the facilities inventory for reassignment. When the type is successor job, the load process is initiated just as if the next run control card in the job stream requested a job to be loaded. Regardless of the type of program termination all activities (except I/O cooperative) are terminated to ensure a smooth job-to-job transition.

9.4.5. Program Control

The user has the option of performing many of the functions of job control during the execution of a job in the run. This is accomplished by using dispatcher calls for the desired service to be performed. Job control services the job for these dispatcher calls in the same way as the control cards in the job stream. These similar services include requests for additional facilities, return of facilities no longer needed, and the execution of a successor job. In addition to these services the job may request segment loads as established by the map element. Also, the job being executed may request that another job be loaded and executed in parallel in another run of its own Finally, the job may request that a job stream which has been prestored on a permanent mass storage file be initiated by the I/O cooperative. This second job stream can then operate in parallel to the originating job stream. Thus, an existing job can use program control to control its facilities, serial execution of jobs, parallel execution of jobs, and parallel job streams in addition to segmentation of the job.

9.5. REAL TIME COMMUNICATION CONTROL

The purpose of the UNIVAC 418-III Real Time Communication Control (RTCC) System is to provide users with a flexible interface to remote communication facilities which is generally free of the constraining influences imposed by specific hardware characteristics. Flexibility is achieved by providing multiple levels of interface and standardizing the characteristics at each entrance level.

The CTMC handler and the WTS handler comprise the executive interface to the computer terminal equipment and are used by both system and user programs.

Additional components which make up the RTCC elements are:

- transaction routing
- queue processor
- remote device handlers
- computer block interchange technique
- communication services

The assignment of software line and device identification numbers in conjunction with a communication drum directory at system generation time simplifies the interface between user programs and the remote facility.

9.5.1. CTMC Handler

The lowest level at which a user program can communicate with a remote device is through the Communication Terminal Modular Control (CTMC) handler. The means of contact between a user program and the CTMC subsystem is a line control table and a supervisor call to the executive.

The major services provided by this first level of support are:

- Answers all ESI processor interrupts.
- Suspends any noncritical program that is in progress.
- Retrieves the hardware tabled interrupt.
- Performs an analysis of an interrupt.
- Schedules the user's activity upon the occurrence of an interrupt.
- Initiates or terminates line activity as indicated in the user's line control table.
- Provides a buffer timing service under user control.
- Issues auto-dial and remote release functions.
- Discards unsolicited interrupts.

9.5.2. WTS Handler

The Word Terminal Synchronous (WTS) operates in the Internally Specified Index (ISI) mode and interfaces with single or paired hardware I/O channels. The handler for this terminal provides the user with a means to simulate an online device at a remote location.

The design of this handler is modular, providing only those functions which are required by the remote device.

9.5.3. Transaction Routing

The transaction routing routine is the primary focal point within the executive for the assignment of all communication facilities, the analysis of all communication dispatcher requests and the scheduling of all associated RTCC activities.

Transaction routing identifies remote device handlers when a communication line is opened and if not currently active, initiates an internal load of the appropriate remote device handler from the systems library. Linkages are then established to provide a rapid access to the handler entry point for subsequent communication dispatcher requests.

All queue processor service requests are intercepted by transaction routing. An internal load is initiated from the systems library when the first data queue is opened. Subsequent calls are passed directly to the queue processor.

9.5.4. Queue Processor

The UNIVAC 418-III Queue Processor is designed to provide independent message buffering between a user program and the remote device handlers. In addition, it supplies the vehicle with which independent but collaborating user programs may exchange blocks of data.

The queue processor honors I/O service requests in an order based upon the caller's operating MID.

Special features of the processor include the ability to queue a message at a higher or lower priority level than any message which is currently on a specified queue and to designate multiple destinations for any one message.

In the event that a recipient of queued data encounters an operating contingency, a capability exists which permits the queue originator to retrieve all remaining data that is queued for that destination.

Inherent in the design of the queue processor is the ability for a user program to queue data in a random manner by multiple classifications and to retrieve that data in blocks by class.

The queue processor resides in the system library and is loaded automatically by transaction routing when the first data queue is opened.

The end of job routine is called when the last data queue is closed.

9.5.5. Remote Device Handlers

Each remote device handler that is supplied for the RTOS user is designed to operate with multiple communication lines and is reentrant for each line. The handlers interface with the CTMC handler and with transaction routing.

For outgoing messages, packed buffers of data are received from the user program. These are unpacked, translated to the device code, and formatted into a line buffer suitable for transmission to the remote facility.

For incoming messages, each text character is translated and packed into a pack buffer. As each buffer is filled, the transaction routing routine is notified. The buffer may or may not contain a complete message.

The characteristics of the remote device handler are:

- Removes or inserts the proper framing characters.
- Checks or generates character and block parity, if required.
- Provides for retransmission of faulty messages.
- Validates incoming message formats.
- Initiates messages indicating line contingencies or nonrecoverable errors.
- Provides for main storage to main storage or drum buffering of data.

The following remote device handlers are supplied in the RTOS:

- UNIVAC DCT 2000 Data Communications Terminal
- UNIVAC UNISCOPE 100 Display Terminal
- UNIVAC UNISCOPE 300 Visual Communications Terminal
- UNIVAC 1004 System
- UNIVAC 9200 System
- UNIVAC 9300 System
- TELETYPE* ASR/KSR

The support of the UNIVAC 9200/9300 as a remote device may be accomplished in two modes. One is the UNIVAC DCT 2000 mode (the UNIVAC 9200/9300 operates under its DCT 2000 handler). The other is the COMBIT mode (a definition of COMBIT is given in the following paragraph).

9.5.6. Computer Block Interchange Technique

The UNIVAC 418-III Computer Block Interchange Technique (COMBIT) Handler is designed to provide remote communication between two or more computer systems over a common data link. A single point and multipoint network capability is provided.

The design is based upon the proposed standard on communication control procedures provided by the Univac Division of Sperry Rand and the United States of America Standards Institute. The standard provided by the Univac Division is specified where a conflict may exist between the two proposed standards.

Special operating characteristics are provided to enhance computer-to-computer operation in the areas not defined in either standard or where current hardware is unable to support a specific function.

Trademark of the Teletype Corporation

The following special features are provided:

- Transparent data transmission
- Blocking of long messages to ensure efficient retransmission on data links with high levels of noise.

9.5.7. Communication Services

The communication services are a series of subroutines and PROC's which a user may incorporate into an applications-oriented program. The following general types of service are provided:

- communication core buffer pools
- communication drum buffer pools
- special numbers conversion
- display or canned message generator
- journal control
- message routing functions
- network supervisory routines
- data queueing
- general header validation
- file control interface module

The design of some of the above routines is such that they work only in a transaction-oriented environment. However, these services are not intended to make up a user message processing program. They are designed to assist the user with some common service functions.

10. LANGUAGE PROCESSORS

10.1. GENERAL

The UNIVAC 418-III Real Time Operating System (RTOS) provides two common high level programming languages (FORTRAN and COBOL) and a symbolic assembly language (UNIVAC 418-III Assembler) for translation of programs into machine code.

The FORTRAN processor translates language statements mostly for scientific and mathematical problems. The assembler translates the symbolic equivalents of the UNIVAC 418-III instruction repertoire into loadable object code. The COBOL processor lends itself to a wide variety of data processing applications such as maintenance or processing of large files or reports.

10.2. THE UNIVAC 418-III ASSEMBLER

The assembler provides the user of the system with a simple method of writing UNIVAC 418-III instructions in symbolic form. In addition, the assembler offers a number of assembly-directing commands designed to aid the user in his program organization, in directing the course of the assembly, and with the input/output interface to the operating system.

10.2.1. Symbolic Language

The symbolic assembler statement consists of four basic fields: a label field, an operation field, an operand field, and an optional comment field. When programming in this language, the user is not bound by fixed–length fields. All fields in this language are in free form. In addition, the language contains a wide choice of operators which provide the ability to fabricate fields during assembly without restrictions on the programmer. Another highly useful feature in the assembler is the provision for specifying up to sixteen location counters which help the user in program organization. The assembly-directing commands offered with the assembler may be classified as follows:

- definitive statements
- incremental statements
- repetitive statements
- procedural statements
- control statements
- generative statements

Included in the repertoire of directives is the procedure (PROC). The PROC is comparable, but generally superior, to a variable-length macro in other systems. It offers the ability to write a routine or a number of routines consisting of repetitive coding not necessarily identical but similar enough so that coding of the routine becomes mechanical. In using a PROC the user has a device within the assembler that generates this coding. When the assembler finds a PROC sample in the source code, it stores all the lines of coding within the PROC in the symbol storage. It then generates this coding during the course of the assembly, whenever the PROC is referenced, by substituting the parameters supplied in the reference line. A PROC must have a label which is its name. PROC's can then be called by referencing that label.

10.2.2. Assembler Organization and Operation

The assembler is a two-pass processor accepting symbolic statements as input and producing relocatable binary code as systems output. Relocatable binary code is the standard UNIVAC 418-III object code format usable by the loader of the executive. It also supplies a parallel listing of the original symbolic coding, together with an octal representation of the code generated as well as error diagnostics when necessary. The assembler consists basically of three parts:

- *The Main Processor* – a group of interlinked routines which analyze and assemble the individual statements.

- *The Symbol Table* – a variable-size buffer, anywhere in main storage, for the purpose of storing symbolic labels and the associated assignments, coded paraforms, and the PROC samples in symbolic form.

- *The Input/Output Section* – provides the assembler with source statements from the system's drum. All output (object code and listing) from the assembler is written to the system's drum.

10.3. FORTRAN

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

FORTRAN is a problem-oriented language that permits a user to express the solution to a problem in a language in terms of the problem rather than in terms peculiar to a specific computer.

10.3.1. Language Characteristics

The basic element in the FORTRAN language is the statement which expresses a complete idea. A statement specifying action is called an executable statement while a statement providing descriptive information, such as the characteristics of data, is called nonexecutable.

An executable statement may take two forms. It may resemble a formula allowing the programmer to express computation or a word or words defined by the FORTRAN language which results in the same program whenever interpreted. This kind of statement is simply a notation for indicating the number of times a group of statements is to be reiterated and provides for such things as decision-making statements.

Nonexecutable statements inform the compiler how data or groups of data are to be handled. This provides for tables (arrays) of data to be constructed. Other nonexecutable statements inform the compiler of data to be transferred to storage and the decimal point location.

To simplify writing a program, an independent group of statements, called a procedure, may be used. A procedure is useful when the same program action is necessary many times throughout the execution of a program, but with the parameters changed. A FORTRAN system supplies a number of procedures commonly used; however, the programmer is provided with a means of writing his own procedures and having them easily incorporated into his program.

The UNIVAC 418-III FORTRAN language is a subset of USA standard FORTRAN dated March 7, 1966, (USASX 3.9 – 1966). The major differences are listed below:

- exceptions
  - no complex type data-handling capabilities
  - data storage unit size for integer, real, and double precision data types is not the same as specified by USASI
  - real constants that contain more than eight significant digits are automatically taken to be a double precision type
- extensions
  - return K statement
  - implied DO list in DATA statement
  - mixed type arithmetic
  - literal descriptor (' ') in FORMAT statement

10.3.2. Compiler Characteristics

The UNIVAC 418-III FORTRAN processor (compiler) exists as an element of the RTOS and operates under control of the executive. The processor is referenced upon the detection of the "FOR" control card.

The FORTRAN processor is a one-pass through the source code processor. The input to the FORTRAN processor consists of symbolic images written in the FORTRAN source language. The source input is supplied to the processor via the system input routines (cooperatives). The output of the processor consists of relocatable object code. The processor also produces a 132-character print file which gives the user the option of printing out source statements and object code. The relocatable object code and the print file are outputted to the system drum via the system output routines.

10.3.3. FORTRAN Support Library

This package is utilized primarily by the UNIVAC 418-III FORTRAN language during execution of user programs. It contains an input/output package, a math package, a trigonometric package, and common routines used by the above packages. The support library fulfills the requirements of USASI with the following additional routines:

(1) trigonometric tangent (real and extended precision)

(2) trigonometric cotangent (real and extended precision)

(3) trigonometric secant (real and extended precision)

(4) trigonometric cosecant (real and extended precision)

(5) arcsine (real and extended precision)

(6) arccosine (real and extended precision)

(7) arccotangent (real and extended precision)

(8) arcsecant (real and extended precision)

(9) arccosecant (real and extended precision)

(10) hyperbolic tangent (extended precision)

10.4. COBOL

The UNIVAC 418-III COBOL language comprises a basic set of English language directives, words, and symbols used to define and create a program. Directed primarily at those unfamiliar with the UNIVAC 418-III computer-oriented language, the UNIVAC 418-III COBOL provides a method for problem analysis and program design that is largely independent of the characteristics of the computer system. Its purpose is to allow the user to remain relatively machine independent.

The particular language characteristics provided with the UNIVAC 418-III COBOL are in accordance with the requirements of the USA COBOL standard X3.23–1968.

The following levels of USA Standard COBOL modules are included:

- level 2
  - nucleus
  - sequential access
  - random access
  - table handling
- level 1
  - segmentation
  - library

10.4.1. Language Organization

Source programs written in COBOL consist of four major divisions:

- Identification Division

  The Identification Division contains information which identifies the source program and the output of compilation. In addition, it may also contain the author, installation, etc.

- Environment Division

  The Environment Division specifies those aspects of a data processing problem which are dependent upon the physical characteristics of a specific computer. This division also allows the specification of the compiling computer configuration and object computer configuration; special hardware characteristics, input/ output control techniques, etc.

- Data Division

  The Data Division describes the data involved during the object program execution. The division is divided into sections to facilitate the description of data contained in input or output files, working-storage requirements, and entries describing the communications environment and data.

- Procedure Division

  The Procedure Division describes the logical steps that must be taken in the solution of the data processing problem. The verbs which describe these steps are categorized as follows:

  - input-output
  - arithmetic
  - data manipulation
  - sequence control
  - compiler directing

10.4.2. COBOL Compiler Characteristics

The UNIVAC 418-III COBOL processor (compiler) exists as an element of the RTOS and operates under control of the executive. The processor is referenced upon the detection of the "COB" control card.

The input to the COBOL processor consists of symbolic images written in the COBOL source language. The source input is supplied to the processor via the system input routines (cooperatives). The output of the processor consists of relocatable object code. The processor also produces a 132-character print file which gives the user the option of printing out source and diagnostics, object code, map, and cross reference of object program. The relocatable object code and the print file are outputted to the system drum via the system output routines.

10.4.3. Processor Organization

The UNIVAC 418-III COBOL compiler is organized into five distinct processing phases, for the purposes of description and reference.

In general, a phase constitutes multiple core overlays; the execution of many of these being conditional upon the compilation options specified or upon source program size. A phase also may involve a complete pass, a partial pass or multiple passes over some or all of the (encoded) source program.

- Phase 1 accomplishes the initial scanning, syntax checking and encoding of the source program. Also during phase 1, each unique combination of characters which identifies a noun (data-name or procedure-name) is replaced by a unique identifying number (name number) which is employed throughout the remainder of the compilation process. Space limitations may require more than a single pass over the data to resolve all names.

- Phase 2 builds a dictionary (symbol table) of all source program definitions. Data division processing is largely completed during this phase and that part of the object program which represents the formal data descriptions is produced at this time.

- Phase 3 passes the encoded Procedure Division against the dictionary so that each reference is augmented by the definition information. This operation, too, may require multiple passes to complete.

- Phase 4 is the code generation phase wherein the compiler selects the appropriate sequences of machine instructions to perform the actions specified by the Procedure Division.

- Phase 5 comprises several optional overlays which accomplish the following:

  (a) Produces the procedure portion of the object program.

  (b) Prepares the source program and diagnostic listing of the compilation.

  (c) Prepares the data and/or procedure map of the object program.

  (d) Prepares the object program listing.

  (e) Produces a cross-reference listing of the source program.

In addition to these processing phases there is a significant portion of the compiler which is known as phase 0 and which is permanently resident in memory while the compiler is executing. Phase 0 contains the compiler's input-output system for the manipulation of its work-files, interphase communication storage and the mechanism for loading the individual overlays. When the compiler is invoked, control is transferred to phase 0 where, in an overlay called phase 0.0, the control card is interpreted before phase 1 is called.

10.5. COBOL LIBRARY

Associated with the COBOL compiler is the COBOL library. It contains entries available to a source program at compilation time. The effect of the compilation of library entries is the same as if the text were actually written as part of the source program. The COBOL library may contain the following three types of entries:

(1) Entries for the Environment Division which consists of equipment-oriented information available through the use of the COPY statement.

(2) Entries for the Data Division, which consists of information pertaining to file and data description entries available through the use of the COPY statement.

(3) Entries for the Procedure Division, which consists of sequences, procedures, paragraphs, and sections available through the use of the COPY statement.

Each division is capable of withdrawing from the COBOL library only that material pertaining to itself.

11. SERVICES AND LIBRARY

11.1. GENERAL

The services and library elements provide an important service function directly to the user and to the other elements of the RTOS. The services offered are in the areas of data file sorting and merging, program maintenance, and various useful debugging aids. The library consists of basically two sections, a procedure library and a subroutine library. Thus, the user can select from a comprehensive set of library routines and services during the design as well as during the checkout of his problem programs.

11.2. SORT

The sort program for the RTOS is a generalized sort program which sorts a file of randomly ordered records composed of characters into a sequence defined by a key. The sort provides the user with a highly efficient sorting capability over a wide range of data processing requirements.

The sort is a generalized program which is specialized for a particular sort through the use of input parameters. These parameters define the equipment to be utilized, the source and format of the input, the source and format of the output, the composition of the key, and the character set.

The sort also provides the facility for the introduction of own code which consists of the user's routines or subroutines.

The sort is executed in four phases.

- parameter processing
- input and distribution
- merge
- output

11.2.1. Parameter Processing

This phase of the program interprets the sort parameters, checks them for detectable errors, and inspects the facilities assigned by the executive to see if they are sufficient to perform a sort using the options that have been requested.

11.2.2. Input and Distribution

During this phase, the input is read in main storage and, if necessary, the items are unblocked. Subsequent to unblocking, the items are processed by own code, if specified, and their keys encoded (key to be sorted is placed at the beginning of the record). They are then written in sequenced strings. Sequence determination and string construction is done by the internal sort routine. The replacement-selection technique is used for the internal sort to produce the longest strings of items possible. The string distribution algorithm sets up the first pass output for polyphase merging when magnetic tape is used for external sort. If FASTRAND mass storage is used for the external sort, the strings are distributed as a reverse threaded list (each string is accompanied by the address of the first block of the preceding string).

11.2.3. Merge

The function of the merge segment is to combine into N number of long strings, the ordered strings of items that were distributed by the first pass. When FASTRAND mass storage is used for external sorting, N = 6; for tape sorting N = the number of available tape units minus one.

11.2.4. Output

The output phase is accomplished by using an N-way merge. As each item is selected in the final merge process, the keys are decoded (key moved to its original position), the item is processed by own code, if specified, and the items formed into block size specified by the user.

11.2.5. Sorting Facilities

UNISERVO VI C or VIII C magnetic tape units or FASTRAND mass storage may be used as an input device, as an intermediate storage device for external sorting, and as an output device. A minimum of four tape units is required, if FASTRAND mass storage is not used. A completely FASTRAND mass storage contained sort does not require any tape units.

11.3. MERGE

The UNIVAC 418-III Merge is a generalized Tape/FASTRAND mass storage merging program written for use on a UNIVAC 418-III Real Time Operating System equipped with UNIVAC VI C and/or VIII C magnetic tape units, or FASTRAND mass storage. In addition to merging, this routine may be used for sequence checking.

This merge may be considered as a generalized program because it is capable of being altered by the use of optional information within the parameter cards. The type of information that may be specified in the parameter cards is as follows:

- peripheral devices for input and output
- amount of main storage
- key description
- data description
- own code (if desired)

The minimum amount of storage required by the merge is 8 K; if buffer swapping is desired the minimum is 12 K. There is no logical limitation as to the total volume this program may handle. Eight files is the maximum number of input files that the merge can handle. When using tape each of the files may be composed of multiple reels.

In regard to tape, the merge performs reading, checking, preparation, and writing of header and trailer blocks on all input and output files when applicable. The items within the file may be either fixed-or variable-length. Both conventional and non-conventional blocks may be input to the merge.

The merge provides the facility for linking with a user's own code segment, which would be mapped with the merge for collection and loading. Own code may perform all input/output functions during the merge and may also be used without taking away the I/O processing from the merge, by defining how to handle the data only.

11.4. PROGRAM MAINTENANCE

The Program Utility Routine (PUR) and the Element Processor (ELT) enable the user to manipulate and update programs, program elements, and program libraries. The user is provided with a repertoire of control cards which are included in the control stream. The format of these control cards conforms to the standard RTOS control card conventions. Elements may be entered into a user run library from cards, from tape and from permanent mass storage libraries. Conversely, elements may be output from a run library to cards, to tape, and to permanent mass storage libraries. Updating functions such as corrections, insertions, and deletions may be made while elements are being input from tape or permanent mass storage into the run library or while being copied.

A program library may contain one or more program elements. The following types of program elements are processed:

- source element (S)
- relocatable element (R)
- map element (M)
- procedure element (P)
- library of elements (L)

Each program element starts with a header block which contains the type indicator and the element name. A program library starts with a library header block which contains the library name and may consist of one or more program elements. A particular program element may belong to one or more program libraries.

A program element is terminated with an end of file (EOF) indicator. Program libraries are terminated with a library trailer block which contains the library name.

11.5. PROGRAM CHECKOUT SERVICES

A number of service routines are provided which aid the user in checking programs. These routines allow the user to examine the contents of mass storage areas and files, dump storage to the printer or to magnetic tape, etc. There are basically two types of service routines. One type operates strictly under control of the operating system and the other type operates independently from the executive. The executive independent utilities are primarily used for initial program testing or systems software expansion. A more detailed description of the checkout services is given in the following paragraphs.

11.5.1. Dump Main Storage

The main storage dump routine provides a means of printing out main storage locations based on a specified translation key and specified storage limits. With this dump a number of operational options are offered. For instance, a calling sequence for such a dump may be permanently included in a program, yet based on an option character in the execute control card. The user has the choice of executing the dump or bypassing it. Through another option the user can specify that only the operating environment be printed. A user may include one or more calling sequences for the storage dump in his program, or he may specify that a storage dump be printed in case of error termination. The main storage dump routine operates under control of the executive.

11.5.2. Dump Mass Storage

The mass storage dump routine provides the user with a means to print files or areas of high performance drums, FASTRAND mass storage, and magnetic tape units. The resultant printout may be either in XS–3 or octal. Two methods are provided through which this dump routine may be called. One is an internal calling sequence, the other is through the use of a control card in the control stream. Regardless of the type of call the user must specify parameters which define the file number (or tape unit number) and the storage limits. The mass storage dump routine operates under control of the executive.

11.5.3. Inspect Main Storage

The inspect main storage routine provides a method of inspecting specified locations of main storage. This routine is operated through the operator's console and runs under control of the executive.

11.5.4. Facility Display

The facility display routine allows the user to display on the operator's console the names of resident programs, main storage assigned, mass storage assigned, as well as allocated tape units. The facility display routine operates under control of the executive.

11.5.5. Date and Time

The date and time routine provides for clock calibration and setting of current time and date through operator type in.

11.6. EXECUTIVE INDEPENDENT UTILITIES

The executive independent utilities are designed to interface directly with the hardware. Only one of the routines may be used at any one time.

The executive independent utilities provide the user with an easy method of initial program testing. The routines included in the executive independent utilities package are:

- Main Storage Inspection and Change

  This routine permits the operator to display any main storage address and change the contents of that address if so desired.

- Main Storage Fill

  This routine permits the operator to place a specific data pattern in consecutive addresses in any area of main storage. The parameters for this routine are the data pattern to be entered in storage and the beginning and ending addresses of the storage area to be filled.

- Utilities Move

  This routine permits the operator to move the entire executive independent utilities package from one storage area to another. The utility package resides within one storage area and is designed to operate in any storage area.

- Program Loader

  This routine makes it possible to load programs that have been created in the packed, relocatable, pseudo-binary format. Magnetic tape or punched cards may be used as the load medium. The program being loaded must not contain references to multiple location counters or use external reference or external definition techniques. The parameters for this routine consist of the start address where the program is to be loaded in storage and the program name. If the load medium is tape, the load channel number (octal) is also included as a parameter. If the load medium is card, the code for the reader device must be defined.

- Main Storage Dump

  This routine permits the operator to dump any area of main storage to the system's printer. This routine is controlled by specifying the beginning and ending addresses of the storage area to be printed and the printer device code.

- Main Storage Search

  This routine permits the operator to search any area of main storage for a specific bit pattern. This routine is controlled by specifying a mask containing the bit pattern desired, the beginning and ending addresses of the storage area to be searched, and the printer device code. If an equal comparison between the mask and a word of storage within the parameters is found, the address and contents are printed.

- Mass Storage Dump

  This routine permits the operator to print any area of any drum or FASTRAND mass storage on the UNIVAC 9200/9300, UNIVAC 1004, or high speed printer. The area to be dumped is specified by parameters which define the starting address and the number of words or sectors to be printed. The printer device code must also be specified.

- Main Storage to Tape Dump

  This routine permits the operator to dump an area of main storage to magnetic tape. The parameters for this routine consist of the main storage ending address and the tape channel (the starting address is always zero). The area specified is written as one block.

- Tape to Main Storage Dump

  This routine permits the operator to read the contents of a magnetic tape written by the main storage to tape dump routine back into main storage. The end address of the storage area needs to be specified as well as the tape channel number.

- Tape Copy

  This routine permits the operator to copy any magnetic tape to another magnetic tape. The maximum block size allowed is 4095 words. The tape channel number must be specified.

- Tape to Printer

  This routine permits the operator to dump the contents of any magnetic tape to any of the system's printers. The maximum allowed tape block size is 4095 words. The parameters for this routine consist of the tape channel number and the printer device code.

- Console Go Routine

  This routine permits the operator to transfer control to any absolute main storage address. The operator has the option of modifying the external interrupt location for the console channel so that control may easily be returned to the EIU package.

Each routine in the executive independent utilities package may be controlled through the console keyboard or from the processor maintenance panel. When loaded, the entire set of programs resides within one storage bay (4096 words) but may be moved to any other bay to meet the storage requirements of any other program. Each routine can be restarted after having been previously used. The routines are designed to operate in the same relative position of any storage bay. The density specifications for all magnetic tape operations are set manually.

11.7. SYSTEMS LIBRARY PROCEDURES

The systems library procedures provide the user with an inventory of commonly and frequently used procedures (PROC's). This procedure library is openended to allow for the addition of newly developed PROC's as they become available. To simplify the user interface with the many executive functions, procedures have been designed for all of the dispatcher requests, all available input/output calling sequences (I/O handler calls, file control calls, cooperative calls, and RTCC calls) and for all facility definitions. In addition to these PROC's, this library contains a number of common service procedures which provide assistance in code conversion, double precision arithmetic, editing and formatting. The user need not be concerned with the construction of the procedure; the only requirement is to write the appropriate PROC reference.

11.8. SYSTEMS LIBRARY SUBROUTINES

The UNIVAC 418-III System user is provided with a comprehensive library of support subroutines. These subroutines may be used by the user or by the system or by both. All of these subroutines conform to a standard format for the definition of the environment at the point of entrance and at the point of exit.

All entry points are externally defined to allow for a consistent calling sequence. The subroutine library is openended to allow for the addition of new routines as they become available. The type of subroutines supported are as follows:

- intrinsic and mathematical
- code translation
- editing
- common service routines
- communications support

The interface to these subroutines is quite simple, A user selects the required routines, includes the calling sequences in his program and requests a load operation. Subroutines are attached to the calling routine and thus become part of it.

APPENDIX A. INSTRUCTION REPERTOIRE

A1. OPERATION CODE SEQUENCE

OPERATION CODE	MNEMONIC	INSTRUCTION	DESCRIPTION	TIMING ① IN μ SECONDS
00		Supervisor Call		0.75
01		Supervisor Call		0.75
02	CL	Compare A Lower	(AL):(Y); Set CD Accordingly	1.50
03	CL*	Compare A Lower		1.50
04	MSL	Masked Selective Load	(Y_N)→AL for (AU_N)=1	1.50
05	MSL*	Masked Selective Load		1.50
06	CLM	Compare A Lower Masked by A Upper	[(AU) AND (AL)] :[(AU) AND (Y)] ; set CD Accordingly	2.0
07	CLM*	Compare A Lower Masked by A Upper		2.0
10	LU	Load A Upper	(Y)→AU	1.50
11	LU*	Load A Upper		1.50
12	LL	Load A Lower	(Y)→AL	1.50
13	LL*	Load A Lower		1.50
14	AL	Add to Lower	(AL)+(Y)→AL	1.50
15	AL*	Add to Lower		1.50
16	ANL	Add Negatively to Lower	(AL)−(Y)→AL	1.50
17	ANL*	Add Negatively to Lower		1.50
20	AA	Add to A	(A)+(Y−1,Y)→A	3.0
21	AA*	Add to A		3.0
22	ANA	Add Negatively to A	(A)−(Y−1, Y)→A	3.0
23	ANA*	Add Negatively to A		3.0
24	M	Multiply	(AL) · (Y)→A	6.5 ② 7.375③
25	M*	Multiply		6.5 ② 7.375③
26	D	Divide	(AL)÷(Y)→AL; Remainder→AU	6.5 ④ 7.375⑤
27	D*	Divide		6.5 ④ 7.375⑤
30	SLJI	Store Location and Jump Indirect	(IAR)+1→Location in(Y); (Y)+1→ IAR	2.25
31	SLJI*	Store Location and Jump Indirect		2.25
32	LB	Load Index Register	(Y)→IR	1.5
33	LB*	Load Index Register		1.5
34	J	Jump	Y→IAR	0.75
35	J*	Jump		0.75
36	LBK	Load Index Register with "Konstant"	Z→IR	0.75

OPERATION CODE	MNEMONIC	INSTRUCTION	DESCRIPTION	TIMING ① IN μ SECONDS
37	LBK*	Load Index Register with "Konstant"		0.75
40	CY	Clear Y	0→Y	1.50
41	CY*	Clear Y		1.50
42	SB	Store Index Register	(IR)→Y	1.50
43	SB*	Store Index Register		1.50
44	SL	Store A Lower	(AL)→Y	1.50
45	SL*	Store A Lower		1.50
46	SU	Store A Upper	(AU)→Y	1.50
47	SU*	Store A Upper		1.50
51	OR	Inclusive OR	(AL) OR (Y)→AL	1.50
52	AND	Logical AND	(AL) AND (Y)→AL	1.50
53	XOR	Exclusive OR	(AL) XOR (Y)→AL	1.50
54	EJI	Enable Interrupts and Jump Indirect	$(Y_{16-0})\rightarrow$ IAR; enables interrupts	1.50
55	JI	Jump Indirect	$(Y_{16-0})\rightarrow$IAR	1.50
56	TB	Test B-Register for Equality	If (IR)=Y,(IAR)+2→IAR If(IR)≠Y,(IR)+1→IR	2.5
57	TZ	Test Any Location for Zero	If(Y)=0,(IAR)+2→IAR If(Y)≠0,(Y)-1→Y	2.25
COMPARISON DESIGNATOR SET 60,61	JE	Jump on Equal	If CD equal condition set; Y→IAR	0.75
62,63	JNE	Jump on Not Equal	If CD equal condition clear, Y→IAR	0.75
64,65	JNLS	Jump on Not Less	If CD not less than condition, Y→IAR	0.75
66,67	JLS	Jump on Less	If CD less than condition, Y→IAR	0.75
60	JUZ	Jump on A Upper Zero	If (AU)=0, Y→IAR	0.75
61	JLZ	Jump on A Lower Zero	If (AL)=0, Y→IAR	0.75
62	JUNZ	Jump on A Upper Non-Zero	If (AU)≠0, Y→IAR	0.75
63	JLNZ	Jump on A Lower Non-Zero	If (AL)≠0, Y→IAR	0.75
64	JUP	Jump on A Upper Positive	If (AU) is positive, Y→IAR	0.75
65	JLP	Jump on A Lower Positive	If (AL) is positive, Y→IAR	0.75
66	JUN	Jump on A Upper Negative	If (AU) is negative, Y→IAR	0.75
67	JLN	Jump on A Lower Negative	If (AL) is negative, Y→IAR	0.75
70	LLK	Load A Lower with "Konstant"	Z→AL	1.0
71	ALK	Add to Lower A "Konstant"	(AL)+Z→AL	1.0
72	SIR	Store Index Register Pointer	$(IRP)\rightarrow Y_{2-0}\left\{\begin{array}{l}\text{If (IRP)=0,}\\ 001\rightarrow Y_{5-3}\\ \text{If (IRP)}\neq 0,\\ 000\rightarrow Y_{5-3}\end{array}\right\}$	3.0

OPERATION CODE	MNEMONIC	INSTRUCTION	DESCRIPTION	TIMING ① IN μ SECONDS
73	JBNZ	Jump and Modify If Index Register Non-Zero	If (IR)≠0, (IR)−1→IR and Y→IAR If (IR)≈0, (IAR)+1→IAR	1.75
74	SAD	Store Address of A Lower	$(AL_{11-0}) \rightarrow Y_{11-0}$	3.0
75	SSR	Store Special Register and Inactivate	$(SR) \rightarrow Y_{5-0}$; $0 \rightarrow SR_4$	3.0
76	SLJ	Store Location and Jump	(IAR)+1→Y; Y+1→IAR	2.0
77		Supervisor Call		0.75
5000		Supervisor Call		1.0
5001		Supervisor Call		1.0
5002**	FA	Floating-point Add	(A)+(Y−1, Y)→A	4.35 +x
5003**	FS	Floating-point Subtract	(A)−(Y−1, Y)→A	4.35 +x
5004**	FM	Floating-point Multiply	(A) . (Y−1, Y)→A	12.0
5005**	FD	Floating-point Divide	(A)÷(Y−1, Y)→A	12.0
5006**	FP	Floating-point Pack	Normalize (A),pack with biased characteristic from Y and store A.	3.5 +x
5007**	FU	Floating-point Unpack	Unpack A, leave mantissa in A, store characteristic in Y	3.5
5010**	RS	Read and Set	$(Y) \rightarrow AL$, $1 \rightarrow Y_{17}$	2.5
5011	LIC	Load Input Channel	Load I/O Channel K from (IAR)+1 and (IAR)+2, initiate input; then (IAR)+3→IAR	5.3 minimum
5012	LOC	Load Output Channel	Same as LIC except that output is initiated	5.3 minimum
5013	LFC	Load External Function Channel	Same as LIC except that external function is initiated	5.6 minimum
5015	STIC	Stop Input on Channel	Stop Input on Channel K	2.15 minimum
5016	STOC	Stop Output on Channel	Stop Output on Channel K	2.15 minimum
5017	SSD	Store Special Designators	(SD) → IAR+1	2.5
5020	LSD	Load Special Designators	(IAR+1) → SD	2.5
5021	TIC	Test Input on Channel	If input Channel K idle, (IAR)+ 2→IAR	1.0
5022	TOC	Test Output on Channel	If output Channel K idle, (IAR)+2→IAR	1.0
5023	TFC	Test External Function on Channel	If external function Channel K idle, (IAR)+2→IAR	1.0
5024 5025	WFI	Wait for Interrupt	Stop C/A unit (not I/O) until interrupt	1.0
5026	NOP	No Operation	(IAR)+1 → IAR	1.0
5030 5031	AAI	Allow All Interrupts	Allow all interrupts	1.0
5034 5035	PAI	Prevent All Interrupts	Prevent all interrupts	1.0

OPERATION CODE	MNEMONIC	INSTRUCTION	DESCRIPTION	TIMING ① IN μ SECONDS
5041	SRU	Shift Right A Upper	Shift AU right (END-OFF) K bit positions	1 +x
5042	SRL	Shift Right A Lower	Shift AL right (END-OFF) K bit positions	1 +x
5043	SRA	Shift Right A	Shift A right (END-OFF) K bit positions	1 +x
5044	SCA	Scale A	Shift A left (END AROUND) K places or until normalized; K less shift→ 00017_8	2 +x
5045	SLU	Shift Left A Upper	Shift AU left (END AROUND) K bit positions	1 +x
5046	SLL	Shift Left A Lower	Shift AL left (END AROUND) K bit positions	1 +x
5047	SLA	Shift Left A	Shift A left (END AROUND) K bit positions	1.25 +x
5050	TK	Test Keys	Skip if keys designated by K are set. (IAR)+2→IAR	1.0
5051	TNB	Test No Borrow	If borrow indicator off, (IAR)+2→IAR	1.0
5052	TOF	Test Overflow	If overflow indicator on, (IAR)+2→IAR	1.0
5053	TNO	Test No Overflow	If overflow indicator off, (IAR)+2→IAR	1.0
5054	TOP	Test Odd Parity	If sum of 1's in (AU) AND (AL) is ODD, (IAR)+2→IAR	2.4 minimum
5055	TEP	Test Even Parity	If sum of 1's in (AU) AND (AL) is EVEN, (IAR)+2→IAR	2.4 minimum
5056	SK	Stop on Key Settings (if not in Guard Mode)	Stop, if keys designated by K are set; if in Guard Mode, (IAR)+1→IAR	1.0
5060	RND	Round A	If (A) is positive and $(AL_{17})=1$, AU+1→AL; If (A) is negative and $(AL_{17})=0$, (AU)−1→AL; otherwise (AU)→AL	1.625
5061	CPL	Complement A Lower	The complement of (AL)→AL	1.0
5062	CPU	Complement A Upper	The complement of (AU)→AU	1.0
5063	CPA	Complement A	The complement of (A)→A	1.875
5065	LGM	Load Guard Mode	$((IAR)+1)_{17-9}$→Upper Limit $((IAR)+1)_{8-0}$→Lower Limit, Guard Mode is set and (IAR)+2 → IAR	1.75
5066	SAA	Set Audible Alarm		1.0
5067	EEI	Enable ESI Interrupt	If K=0, allow ESI Interrupts, IOM #0; If K=20_8, allow ESI Interrupts, IOM #1	1.0

OPERATION CODE	MNEMONIC	INSTRUCTION	DESCRIPTION	TIMING ① IN μ SECONDS
5070	BT	Block Transfer	Transfer K words from $ADR_{AU} \rightarrow ADR_{AL}$	1.750+1.5n
5072	LIR	Load Index Register Pointer	$K_{2-0} \rightarrow IRP$	2.5
5073	LSR	Load Special Register	$K_{5-0} \rightarrow SR$	1.0
5074	DB	Decimal-to-Binary Conversion	$(AU_{15-12,\ 9-6,\ 3-0})_{10} \rightarrow AL_2$	7.735
5075	BD	Binary-to-Decimal Conversion	$AL_2 \rightarrow (AU_{15-12,\ 9-6,\ 3-0})_{10}$	8.250
5077		Supervisor Call		1.0

A2. FUNCTIONAL GROUP SEQUENCE

OPERATION CODE	MNEMONIC	INSTRUCTION	DESCRIPTION	TIMING ① IN μ SECONDS
ARITHMETIC COMMANDS				
5060	RND	Round A	If (A) is positive and (AL$_{17}$)=1, AU+1→AL; If (A) is negative and (AL$_{17}$)= 0, (AU)−1→AL; otherwise (AU)→AL.	1.625
14	AL	Add to Lower	(AL)+(Y)→AL	1.50
15	AL*	Add to Lower		1.50
16	ANL	Add Negatively to Lower	(AL)−(Y)→AL	1.50
17	ANL*	Add Negatively to Lower		1.50
20	AA	Add to A	(A)+(Y−1,Y)→A	3.0
21	AA*	Add to A		3.0
22	ANA	Add Negatively to A	(A)−(Y−1,Y)→A	3.0
23	ANA*	Add Negatively to A		3.0
24	M	Multiply	(AL).(Y)→A	6.5② 7.375③
25	M*	Multiply		6.5② 7.375③
26	D	Divide	(AL)÷(Y)→AL; Remainder→AU	6.5④ 7.375⑤
27	D*	Divide		6.5④ 7.375⑤
71	ALK	Add to Lower A "Konstant"	(AL)+Z→AL	1.0
FLOATING-POINT ARITHMETIC COMMANDS				
5002**	FA	Floating Point Add	(A)+(Y−1,Y)→A	4.35+x
5003**	FS	Floating Point Subtract	(A)−(Y−1,Y)→A	4.35+x
5004**	FM	Floating Point Multiply	(A).(Y−1,Y)→A	12.0
5005**	FD	Floating Point Divide	(A)÷(Y−1,Y)→A	12.0
5006**	FP	Floating Point Pack	Normalize (A),pack with biased characteristic from (Y), and store in A.	3.5+x 3.5
5007**	FU	Floating Point Unpack	Unpack A, leave mantissa in A. Store characteristic in Y.	3.5
BINARY/DECIMAL CONVERSION COMMANDS				
5074	DB	Decimal-to-Binary Conversion	$(AU_{15-12,\ 9-6,\ 3-0})\rightarrow AL_2$	7.375
5075	BD	Binary-to-Decimal Conversion	$AL\rightarrow (AU_{15-12,\ 9-6,\ 3-0})_{10}$	8.250

OPERATION CODE	MNEMONIC	INSTRUCTION	DESCRIPTION	TIMING ① IN μ SECONDS
LOGICAL COMMANDS				
51	OR	Inclusive OR	(AL) OR (Y)→AL	1.50
52	AND	Logical AND	(AL) AND (Y)→AL	1.50
53	XOR	Exclusive OR	(AL) XOR (Y)→AL	1.50
5061	CPL	Complement A Lower	The complement of (AL)→AL	1.0
5062	CPU	Complement A Upper	The complement of (AU)→AU	1.0
5063	CPA	Complement A	The complement of (A)→A	1.875
TRANSFER COMMANDS				
10	LU	Load A Upper	(Y)→AU	1.50
11	LU*	Load A Upper		1.50
12	LL	Load A Lower	(Y)→AL	1.50
13	LL*	Load A Lower		1.50
44	SL	Store A Lower	(AL)→Y	1.50
45	SL*	Store A Lower		1.50
46	SU	Store A Upper	(AU)→Y	1.50
47	SU*	Store A Upper		1.50
70	LLK	Load A Lower with "Konstant"	Z→AL	1.0
04	MSL	Masked Selective Load	$(Y_N)\rightarrow AL$ for $(AU_N)=1$	1.50
05	MSL*	Masked Selective Load		1.50
32	LB	Load Index Register	(Y)→IR	1.5
33	LB*	Load Index Register		1.5
42	SB	Store Index Register	(IR)→Y	1.50
43	SB*	Store Index Register		1.50
36	LBK	Load Index Register with "Konstant"	Z→IR	0.75
37	LBK*	Load Index Register with "Konstant"		0.75
74	SAD	Store Address of A Lower	$(AL_{11-0})\rightarrow Y_{11-0}$	3.0
5072	LIR	Load Index Register Pointer	$K_{2-0}\rightarrow IRP$	2.5
5073	LSR	Load Special Register	$K_{5-0}\rightarrow SR$	1.0
72	SIR	Store Index Register Pointer	$(IRP)\rightarrow Y_{2-0}\left\{\begin{array}{l}\text{If }(IRP)=0,\\ \quad 001\rightarrow Y_{5-3}\\ \text{If }(IRP)\neq 0,\\ \quad 000\rightarrow Y_{5-3}\end{array}\right\}$	3.0
75	SSR	Store Special Register and Inactivate	$(SR)\rightarrow Y_{5-0}$, $0\rightarrow SR_4$	3.0
40	CY	Clear Y	0→Y	1.50
41	CY*	Clear Y		1.50
5070	BT	Block Transfer	Transfer K words from $ADR_{AU}\rightarrow ADR_{AL}$	1.750+1.5n
5017	SSD	Store Special Designators	(SD) → IAR+1	2.5
5020	LSD	Load Special Designators	(IAR+1) → SD	2.5

OPERATION CODE	MNEMONIC	INSTRUCTION	DESCRIPTION	TIMING① IN μ SECONDS
SHIFT COMMANDS				
5041	SRU	Shift Right A Upper	Shift AU right (END-OFF) K bit positions	1 +x
5042	SRL	Shift Right A Lower	Shift AL right (END-OFF) K bit positions	1+x
5043	SRA	Shift Right A	Shift A right (END-OFF) K bit positions	1+x
5044	SCA	Scale A	Shift A left (END AROUND) K places or until normalized K less shift→ 00017_8	2+x
5045	SLU	Shift Left A Upper	Shift AU left (END AROUND) K bit positions	1 +x
5046	SLL	Shift Left A Lower	Shift AL left (END AROUND) K bit positions	
5047	SLA	Shift Left A	Shift A left (END AROUND) K bit positions	1 +x
LOOP CONTROL COMMANDS				
73	JBNZ	Jump and Modify if Index Register Non-Zero	If (IR)≠0, (IR)−1→IR and Y→IAR If (IR)=0, (IAR)+1→IAR	1.75
56	TB	Test B-Register for Equality	If(IR)=Y, (IAR)+2→ IAR If(IR)≠Y,(IR)+1→IR	2.5
57	TZ	Test Any Location for Zero	If(Y)=0,(IAR)+2→ IAR If(Y)≠0,(Y)-1→Y	2.25
COMPARE COMMANDS				
02	CL	Compare A Lower	(AL): (Y) set CD accordingly	1.50
03	CL*	Compare A Lower		1.50
06	CLM	Compare A Lower Masked by A Upper	[(AU) AND (AL)] : [(AU) AND (Y)]; set CD accordingly	2.0
07	CLM*	Compare A Lower Masked by A Upper		2.0
COMPARISON JUMP COMMANDS (COMPARE DESIGNATOR SET)				
60,61	JE	Jump on Equal	If CD equal condition set, Y→IAR	0.75
62,63	JNE	Jump on Not Equal	If CD equal condition clear, Y→IAR	0.75
64,65	JNLS	Jump on Not Less	If CD not less than condition, Y→IAR	0.75
66,67	JLS	Jump on Less	If CD less than condition, Y→IAR	0.75

OPERATION CODE	MNEMONIC	INSTRUCTION	DESCRIPTION	TIMING ① IN μ SECONDS
ARITHMETIC JUMP COMMANDS (COMPARE DESIGNATOR NOT SET)				
60	JUZ	Jump on A Upper Zero	If (AU)=0, Y→IAR	0.75
61	JLZ	Jump on A Lower Zero	If (AL)=0, Y→IAR	0.75
62	JUNZ	Jump on A Upper Non-Zero	If (AU)≠0, Y→IAR	0.75
63	JLNZ	Jump on A Lower Non-Zero	If (AL)≠0, Y→IAR	0.75
64	JUP	Jump on A Upper Positive	If (AU) is positive, Y→IAR	0.75
65	JLP	Jump on A Lower Positive	If (AL) is positive, Y→IAR	0.75
66	JUN	Jump on A Upper Negative	If (AU) is negative, Y→IAR	0.75
67	JLN	Jump on A Lower Negative	If (AL) is negative, Y→IAR	0.75
UNCONDITIONAL JUMP COMMANDS				
34	J	Jump	Y→IAR	0.75
35	J*	Jump		0.75
55	JI	Jump Indirect	(Y_{16-0})→IAR	1.50
30	SLJI	Store Location and Jump Indirect	(IAR)+1→Location in (Y); (Y)+1→IAR	2.25
31	SLJI*	Store Location and Jump Indirect		2.25
76	SLJ	Store Location and Jump	(IAR)+1→Y; Y+1→IAR	2.0
SKIP COMMANDS				
5050	TK	Test Keys	Skip if keys designated by K are set. (IAR)+2→IAR	1.0
5051	TNB	Test No Borrow	If borrow indicator off, (IAR)+2→IAR	1.0
5052	TOF	Test Overflow	If overflow indicator on, (IAR)+2→IAR	1.0
5053	TNO	Test No Overflow	If overflow indicator off, (IAR)+2→IAR	1.0
5054	TOP	Test Odd Parity	If sum of 1's in (AU) AND (AL) is ODD, (IAR)+2→IAR	2.4 minimum
5055	TEP	Test Even Parity	If sum of 1's in (AU) AND (AL) is EVEN, (IAR)+2→IAR	2.4 minimum
EXECUTIVE COMMANDS (INTERRUPT CONTROL)				
5024 5025	WFI	Wait for Interrupt	Stop C/A Unit (not I/O) until Interrupt	1.0
5030 5031	AAI	Allow All Interrupts	Allow all Interrupts	1.0
5034 5035	PAI	Prevent All Interrupts	Prevent all Interrupts	1.0
54	EJI	Enable Interrupts and Jump Indirect	(Y_{16-0})→IAR; enables Interrupts	1.50
5067	EEI	Enable ESI Interrupt	If K=0 allows ESI Interrupts, IOM #0; If K=20_8, allow ESI Interrupts, IOM #1	1.0

OPERATION CODE	MNEMONIC	INSTRUCTION	DESCRIPTION	TIMING[1] IN μ SECONDS
EXECUTIVE COMMANDS (I/O)				
5011	LIC	Load Input Channel	Load I/O Channel K from (IAR)+1 and (IAR)+2, initiate input; then (IAR)+3→IAR	5.3 minimum
5012	LOC	Load Output Channel	Same as LIC except that output is initiated	5.3 minimum
5013	LFC	Load External Function Channel	Same as LIC except that External Function is initiated	5.6 minimum
5015	STIC	Stop Input on Channel	Stop Input on Channel K	2.15 minimum
5016	STOC	Stop Output on Channel	Stop Output on Channel K	2.15 minimum
5021	TIC	Test Input on Channel	If Input Channel K idle, (IAR)+2→IAR	1.0
5022	TOC	Test Output on Channel	If Output Channel K idle, (IAR)+2→IAR	1.0
5023	TFC	Test External Function on Channel	If External Function Channel K idle, (IAR)+2→IAR	1.0
EXECUTIVE COMMANDS (STORAGE PROTECTION)				
5065	LGM	Load Guard Mode	$((IAR)+1)_{17-9} \rightarrow$ Upper Limit $((IAR)+1)_{8-0} \rightarrow$ Lower Limit, Guard Mode is set and (IAR)+2 → IAR	1.75
EXECUTIVE COMMANDS (STOP)				
5056	SK	Stop on Key Settings (if not in Guard Mode)	Stop, if Keys designated by K are set; if in Guard Mode, (IAR)+1→IAR	1.0
EXECUTIVE COMMAND (SPECIAL)				
5010**	RS	Read and Set	$(Y) \rightarrow AL, 1 \rightarrow Y_{17}$	2.5
5026	NOP	No Operation		1.0
5066	SAA	Set Audible Alarm		1.0
SUPERVISOR CALL COMMANDS				
00		Supervisor Call		0.75
01		Supervisor Call		0.75
77		Supervisor Call		0.75
5000		Supervisor Call		1.0
5001		Supervisor Call		1.0
5077		Supervisor Call		1.0

LEGEND

Subscripts specify bit positions in the register or quantity subscripted.
N represents each bit position.

* = To index an Assembler instruction, prefix operand with *(asterisk). The Assembler will add 1 to octal operation code, or set $2^{12} = 1$ of IAR + 1.

x = $\frac{\text{number of shifts}}{8}$ μ seconds

n = number of words in the block

** = IR sensitive if 2^{12} of IAR + 1 is set to 1; indexing is indicated by prefixing the operand with an asterisk (*).

AL = Lower Accumulator

AU = Upper Accumulator

A = Upper and Lower Accumulators acting as one register

IR = The active Index Register

IAR = Instruction Address Register

CD = Compare Designator

Y = On the left of the → symbol, the storage address in the low-order 12 bits of the instruction (bits 11–0); on the right of the → symbol, the storage location specified by that address.

K = The unsigned integer or bit configuration in the low-order 6 bits of the instruction (bits 5–0).

Z = The low-order 12 bits of the instruction, extended to 18 bits by repetition of bit 11 in bit positions 17–12, and treated as a constant in the range – 3777 to + 3777 octal.

( ) = Contents of the register named in the parentheses; e.g., (Y) = contents of Y.

→ = Replaces the contents of

: = Compare algebraically the quantities on either side of this symbol.

XOR = Exclusive OR

AND = Logical AND

OR = Logical OR

① Several timing figures shown are different from those published in a previous edition of this document.

② Multiplying numbers of like signs.

③ Multiplying numbers of unlike signs.

④ Dividing positive numbers.

⑤ Dividing numbers of unlike signs or negative signs.

APPENDIX B. UNIVAC 418-III RTOS MINIMUM HARDWARE REQUIREMENTS

1 418-III Processor Unit with 32 K Main Storage

8 I/O Channels

1 Drum Subsystem consisting of:
  2 FH-432 drums;
    or
  1 FH-1782 drum;
    or
  1 FH-880 drum;
    or
  1 FASTRAND II Subsystem;
    or
  1 FASTRAND III Subsystem

2 Tape units (VI C or VIII C) with Synchronizer

1 9000 Series Subsystem

1 Communications Subsystem (CTMC or WTS) is required for real time users

UNIVAC

SPERRY RAND