SPERRY RAND

UNIVAC

1108

MULTI-PROCESSOR
SYSTEM

SYSTEM
DESCRIPTION

PRINTED IN U.S.A.

CONTENTS

APPENDIXES

FIGURES

TABLES

1. INTRODUCTION

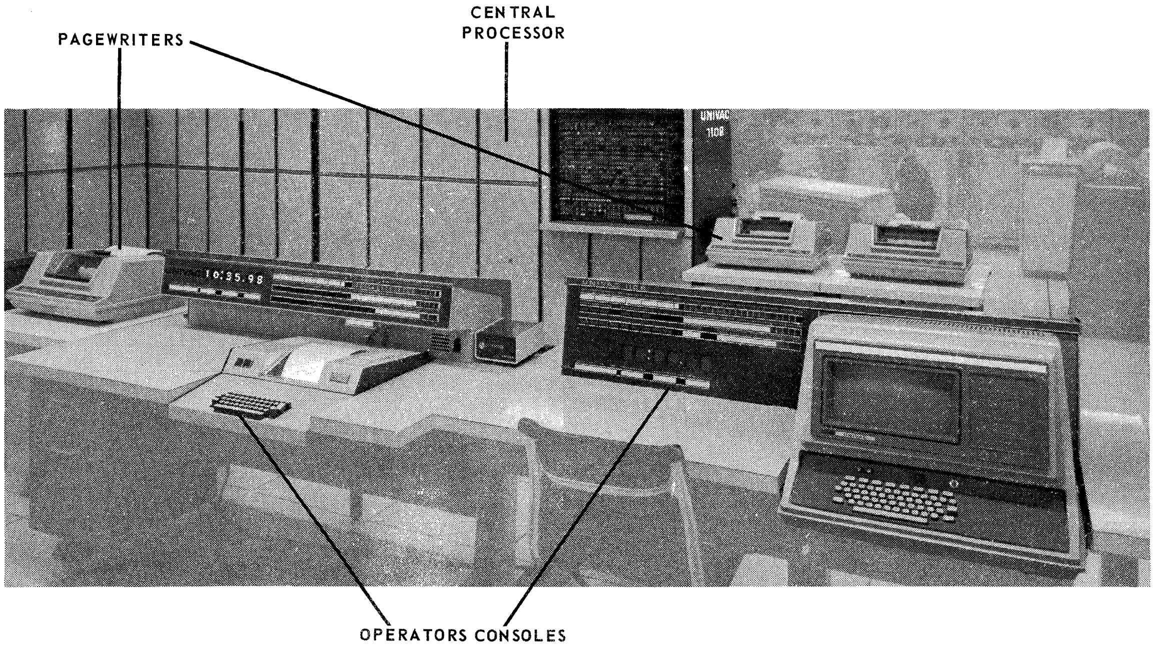


Figure 1–1. The Central Processor and Operator's Console

The UNIVAC 1108 Multi-Processor System – the logical, program compatible successor to the UNIVAC 1107 Thin-Film Computer – is an integration of system-oriented hardware design, imaginative development, and programming technology. The result is a system that effectively knows no application boundaries. It is equally effectual in real-time, scientific, or data processing environments, and is capable of adjusting 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, multiprogramming, and multiprocessing capabilities, but possessing the simplicity of a monitor system.

2. SYSTEM DESCRIPTION

2.1. GENERAL

The UNIVAC 1108 Multi-Processor System is a general purpose, high performance unit and multiprocessor system, incorporating the latest advances in computer design, systems organization, and programming technology. Its modular structure permits the selection of systems components to fulfill most efficiently the speed and capacity requirements for applications ranging from a basic job-shop system to the comprehensive public utility computing complex.

As the workload increases, this modularity also enables the addition of input/output subsystems and main storage and even additional processors to provide the full multiprocessor system. Among the principal features of the UNIVAC 1108 System are:

- Common resource systems organization
- Equality among multiple UNIVAC 1108 Central Processors
- Multiple input/output controllers
- Large, modular, parity-checked, high-speed main storage
- Overlapped and interleaved main storage access (for multimodule storage)
- Redundancy among systems components
- Program address relocation
- Storage protection
- Partial-word addressability in 6, 9, 12, and 18-bit portions as well as full-word (36 bits and double-word (72 bits) addressing
- High speed, random access, auxiliary storage
- Privileged mode for the Executive system
- Guard mode for user programs

2.2. SYSTEM COMPONENTS

The UNIVAC 1108 System is organized to allow multiple processors to perform a number of tasks simultaneously under the direction of a common Executive control system. A multiprocessor system requires a much higher degree of modularity in organization than does a unit-processor system. It must be divisible into individual logical components with the following properties:

- Each system component must have more than one access path.
- Priority logic must resolve possible access conflicts.
- The failure of any individual component must not prevent continued operation of the system.
- System components must be logically removable for servicing without disabling the system.

The system is constructed of six types of components:

- Central Processor Units (CPU)
- Input/Output Controllers (IOC)
- Main Storage Modules
- Auxiliary Storage Subsystems
- Systems Interconnection Components
- Peripheral Subsystems

A shared-processor system incorporates two processors (computational and input/output) which offer a considerably greater capacity than the unit processor.

2.2.1. Central Processor

Each processor can perform all functions required for the execution of instructions including arithmetic, input/output, and Executive control. In a multiprocessor configuration, all processors are equal – the test of a true multiprocessor system. Included in each processor is its own set of 125-nanosecond integrated circuit control registers providing multiple accumulators, index registers, input/output access control registers, and special-use registers.

In the shared processor system, the computational processor has only control and arithmetic sections for computation. The I/O processor consists of an I/O section for data transfer, and arithmetic and control sections for computations during idle input/output cycles. This increases throughput and provides a balanced system under direction of the Executive without special programming by the user. It can be expanded to a multiprocessor system without changes to software.

2.2.2. Input/Output Controller

The Input/Output Controller (Figure 2–1) is an independent processor utilized in multiprocessor systems to expand the input/output capabilities of the system. It includes:

- Up to 16 high-speed input/output channels
- Independent access to main storage
- Data chaining
- Sixteen pointer registers
- Sixty-four external function and ISI data access control registers
- 192 communications access control registers
- An optional 256 additional communications access control registers

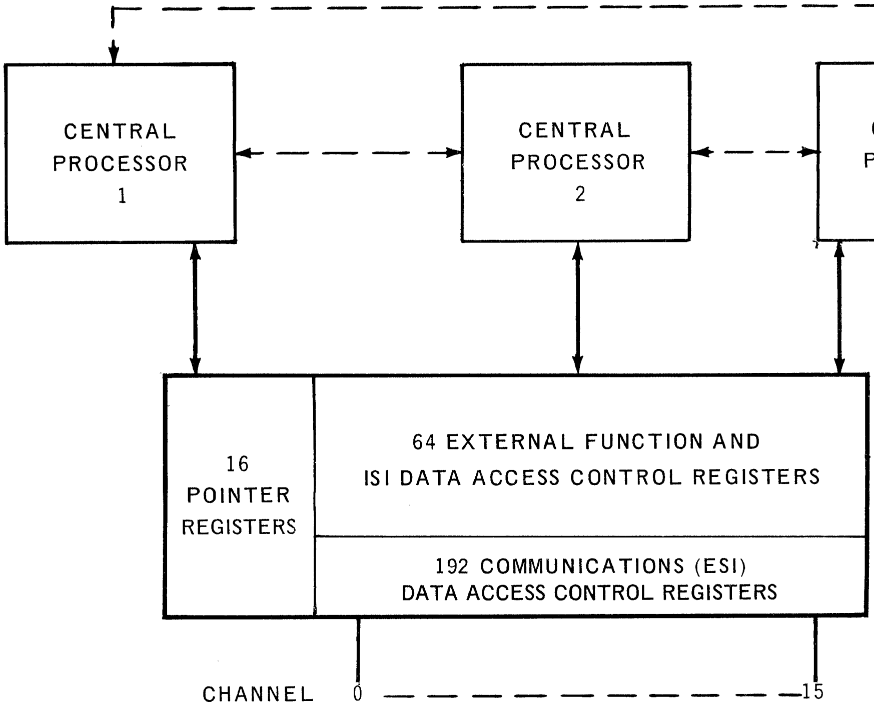

CENTRAL PROCESSOR 1
CENTRAL PROCESSOR 2
CENTRAL PROCESSOR 3
16 POINTER REGISTERS
64 EXTERNAL FUNCTION AND ISI DATA ACCESS CONTROL REGISTERS
192 COMMUNICATIONS (ESI) DATA ACCESS CONTROL REGISTERS
CHANNEL 0
15

Figure 2–1. The Input/Output Controller

2.2.3. Main Storage

The main storage of the UNIVAC 1108 System is expandable in 65,536-word increments up to a maximum of 262,144 thirty-six bit words. The main storage read/restore cycle time is 750 nanoseconds. Up to four logical banks for instruction/data reference overlapping provide the capability for an effective cycle time of 375 nanoseconds. In addition, two-way interleaving of storage modules is provided to reduce the probability of access conflicts.

2.2.4. Auxiliary Storage

The auxiliary magnetic drum storage subsystems are an integral part of each UNIVAC 1108 System. Up to eight FH-432, FH-1782, or FH-880 magnetic drums, or any combination of the FH-432 and FH-1782 types, may be attached to one or two control units. Both the FH-432 and FH-1782 types of drum can transfer data at 1,440,000 characters per second. The type FH-880 transfer rate is 360,000 characters per second. The average access time of the FH-432 is 4.3 milliseconds; the access time of the FH-1782 and FH-880 is 17 milliseconds.

Additional storage may be obtained with UNIVAC Unitized Channel Storage. Up to four modules, each with a capacity of 262K 36-bit words, of Unitized Channel Storage may be used. The maximum word transfer rate is 2.25 microseconds, which can be tailored to 4.0 or 8.0 microseconds, according to the characteristics of the subsystem.

2.2.5. Interconnection Components

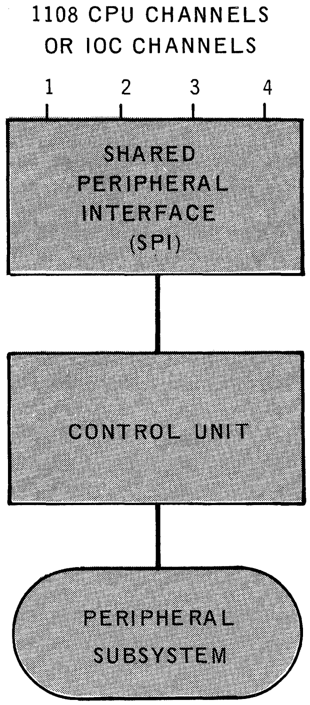
It is an essential characteristic of multiprocessor systems that there must be provision for sharing all of the main storage and all of the I/O subsystems by all processors in the system. This sharing must be on the basis of both established priorities and reactive priorities that may change during processing. In the 1108 System, Multi-Module Access Units (MMA) provide this access to the storage modules by several processors, and the Shared Peripheral Interface (SPI) similarly enables the sharing of I/O subsystems among processors.

2.2.5.1. Shared Peripheral Interface

The Shared Peripheral Interface (SPI) controls the access of up to four input/output channels to units in a shared peripheral subsystem (see Figure 2–2). Access to shared peripheral subsystems is determined primarily by time of request. If two requests are made simultaneously, the Central Processor or Input/Output Controller on the lower numbered SPI input/output channel receives priority. In case of a busy condition, or a priority conflict, the Executive automatically stacks the request until the SPI channel is available. First-level queuing is handled in the SPI itself. The Executive stacks longer queues and keeps track of the number of I/O function requests outstanding.

Input/output subsystems may be either single- or dual-channel as is illustrated in Figures 2–2 and 2–3. Single-channel subsystems perform one I/O operation at a time and therefore require one control unit, one I/O channel, and, in multiprocessor systems, only one SPI. Dual channel subsystems can execute two operations simultaneously using different peripheral units in the subsystem. Both of these operations may originate in the same processor or they may come from different processors.

Different processors can be connected to such a dual-channel subsystem through two SPI units. Two input/output channels from each Central Processor or Input/Output Controller take separate paths. The failure of an SPI or one of the pair of control units affects only one of the two paths to a peripheral subsystem. Therefore, all peripheral units are still accessible through the second SPI and control unit.

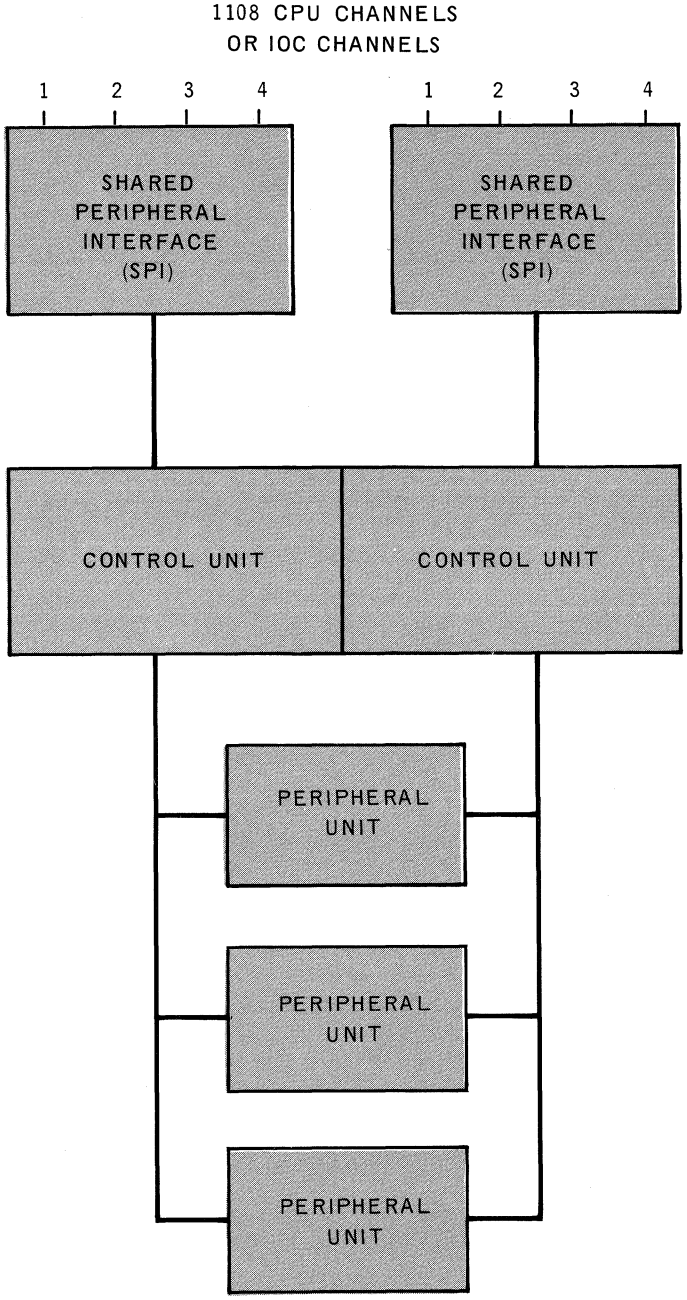


PERIPHERAL DEVICES

- FH-432/1782 Drums
- FH-880 Drums
- FASTRAND Mass Storage
- 8414 Disc Subsystem*
- Unitized Channel Storage
- UNISERVO VI-C Magnetic Tape Units
- UNISERVO VIII-C Magnetic Tape Units
- UNISERVO 12 Magnetic Tape Units*
- UNISERVO 16 Magnetic Tape Units*
- Card Read/Punch
- Printers
- UNIVAC 9200/9300 Subsystem
- Communications Terminal Module Controller
- Communication Terminal Synchronous
- Word Terminal Synchronous

*Must be connected to CPU I/O channels only

Figure 2–2. Shared Peripheral Interface (SPI), Single-Channel Subsystem



DUAL-CHANNEL SUBSYSTEMS

- FH-432/1782 Drums
- FASTRAND Mass Storage
- UNISERVO VIII-C Magnetic Tape Units
- UNISERVO 12 Magnetic Tape Units
- UNISERVO 16 Magnetic Tape Units
- 8414 Disc

Figure 2–3. Shared Peripheral Interface (SPI), Dual-Channel Subsystem

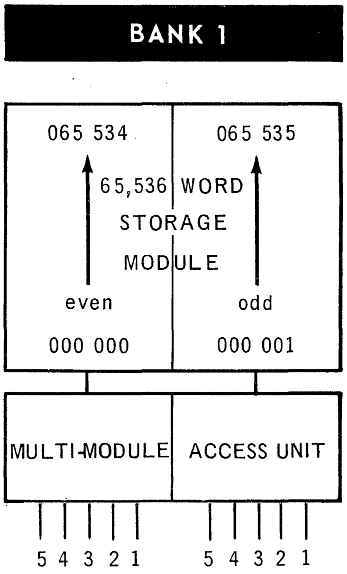


Figure 2–4. Multi-Module Access Unit

2.2.5.2. Multi-Module Access Unit

The Multi-Module Access Units (MMA) allow the sharing of individual storage modules by up to three Central Processors and two Input/Output Controllers on a fixed priority basis. (See Figure 2–4.)

The MMA recognizes the storage access requests on a priority basis with the lower channel retaining the higher priority. Input/Output Controllers which require higher priority are therefore connected to the lower numbered interfaces. Upon recognition of a storage access request, the MMA connects the address lines, the CPU or IOC data lines, and the write control signals to the storage module. The MMA then sends the storage acknowledgement back to the recognized processor and provides the drive necessary to transfer the data to the processor requesting it.

2.2.5.3. Availability Control Unit

Because of the system availability requirement, the multiprocessor system must have a means for partitioning the system for specific jobs or for maintenance purposes. The Availability Control Unit (ACU) performs this and related functions as follows:

- Partitions the multiprocessor system hardware into independent systems;
- Takes units offline for maintenance without disrupting operation of the rest of the system;
- Protects main storage in event of a power failure in the CPU or IOC;
- Automatically initiates a recovery sequence after a failure.

The ACU partitions the hardware into specific configurations by disabling and enabling the interface between units. It can set up as many as three logically independent configurations which run concurrently under control of the Executive system. The possible configurations can be prespecified for a given site. At the same time the ACU can take units offline for maintenance.

The ACU is an independent unit with its own power supply which is logically situated between the peripheral subsystems, the central processors, input/output controllers, and main storage. (See Figure 2–7.) It can interface with three Central Processors through one I/O channel of each processor, two IOC's, four banks of main storage, and six multiple-access peripheral subsystems. Additional peripheral subsystems, to a maximum of 24, can be added in groups of six. The ACU includes a control panel, physically located at the operator's console, that indicates all partitioning currently in effect and also shows which units are off line. It also has manual controls to switch units on or off line.

The automatic recovery sequence is based on a resettable system timer in the ACU. The period of this timer can be set to times varying from one to fifteen seconds. Unless the Executive system resets this timer within its period, the ACU assumes that a catastrophic malfunction has occurred and it initiates an automatic recovery sequence. The processor can interrogate the ACU to determine which units are online and available for use.

2.2.6. Display Console

The UNIVAC 1108 Display Console Subsystem is a freestanding input/output device for directing and monitoring the operation of the CPU. A multiprocessor configuration includes one console subsystem for each CPU. The various activities of the system can be apportioned among the available consoles so that the total system will be utilized to best advantage. The console is always connected to input/output channel 15 of the CPU.

The basic Display Console includes the following components:

- Keyboard and CRT Display

  The keyboard and CRT display enable the operator to monitor the performance of the system. The keyboard is a standard four-bank keyboard which can generate 63 Fieldata codes. A row of eight interrupt keys is located immediately above it. The CRT can display 16 lines of 64 characters each.

- UNIVAC PAGEWRITER Printer

  The UNIVAC PAGEWRITER printer provides a hard copy of all messages for a permanent record of all completed transactions between the operator and the Executive system. The PAGEWRITER printer prints lines of up to 80 characters each at a rate of 25 characters per second.

- Day Clock

  The day clock on the operator's control panel displays the time of day in hours, minutes, and hundredths of minutes. It furnishes the time of day to the CPU every 600 milliseconds and sends a day clock interrupt signal to the CPU every 6 seconds. The day clock may be manually disabled from the operator's control panel.

  On a multiprocessor system, any one day clock may be selected to be active either externally or by program.

- Operator's Control Panel

  The operator's control panel includes fault, disable, and mode indicators for its CPU and associated main storage modules. Available to the operator are displays and controls associated with selecting and releasing jumps and stops. Also displayed is the Program Address Counter, Memory Select Register, and the time display of the day clock. Accessible to the operator are system controls associated with the CPU and subsystems logically connected to the CPU.

- Additional Features

  An auxiliary right- or left-wing console to accommodate control/display panels is included when using the Communications Terminal Module Controller (CTMC) subsystems.

2.3. CONFIGURATIONS

The introduction of the UNIVAC 1108 System with its many components provides a most flexible system. The many configurations of individual components are almost limitless. Figures 2–5, 2–6, and 2–7 illustrate typical possibilities.

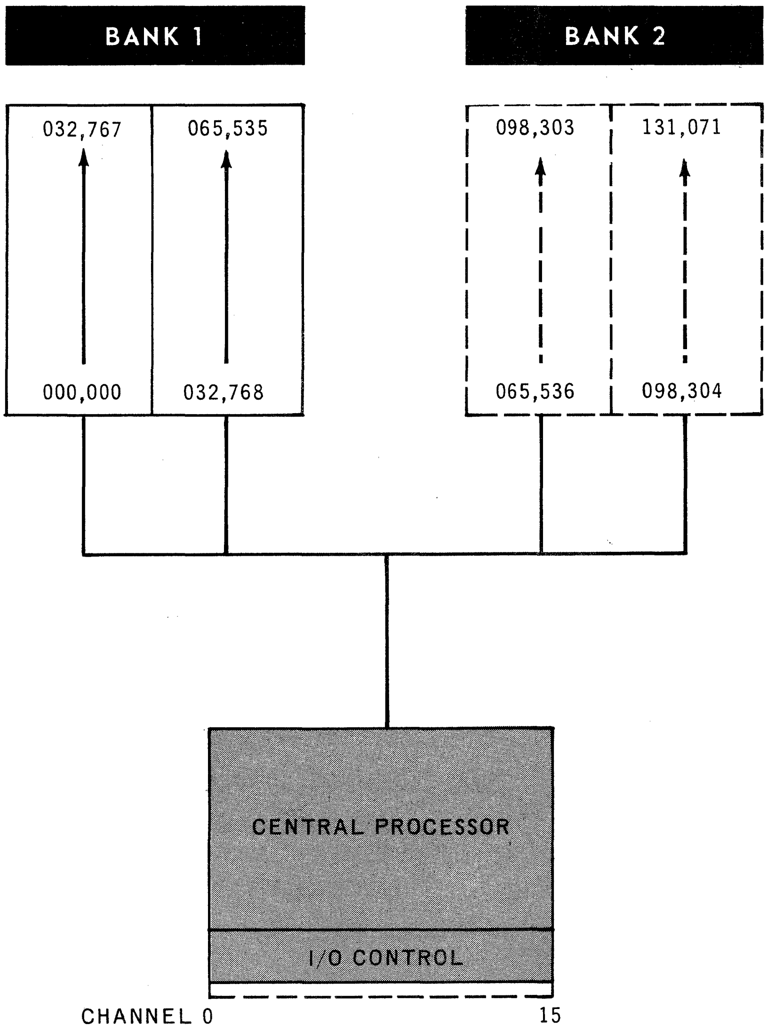


Figure 2–5. Unit Processor System with Noninterleaved Storage

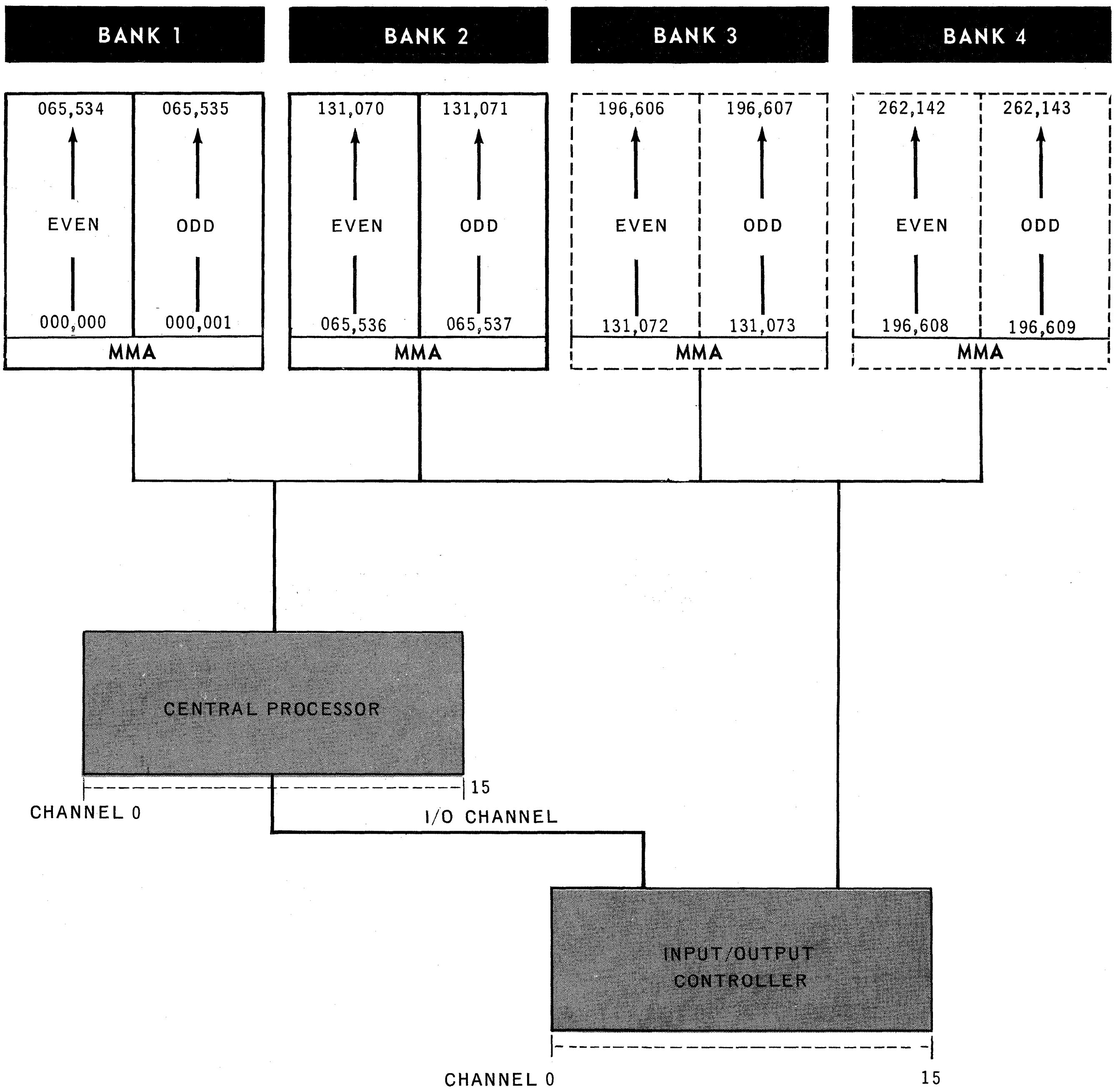


Figure 2–6. Single Processor with I/O Controller and Interleaved Storage

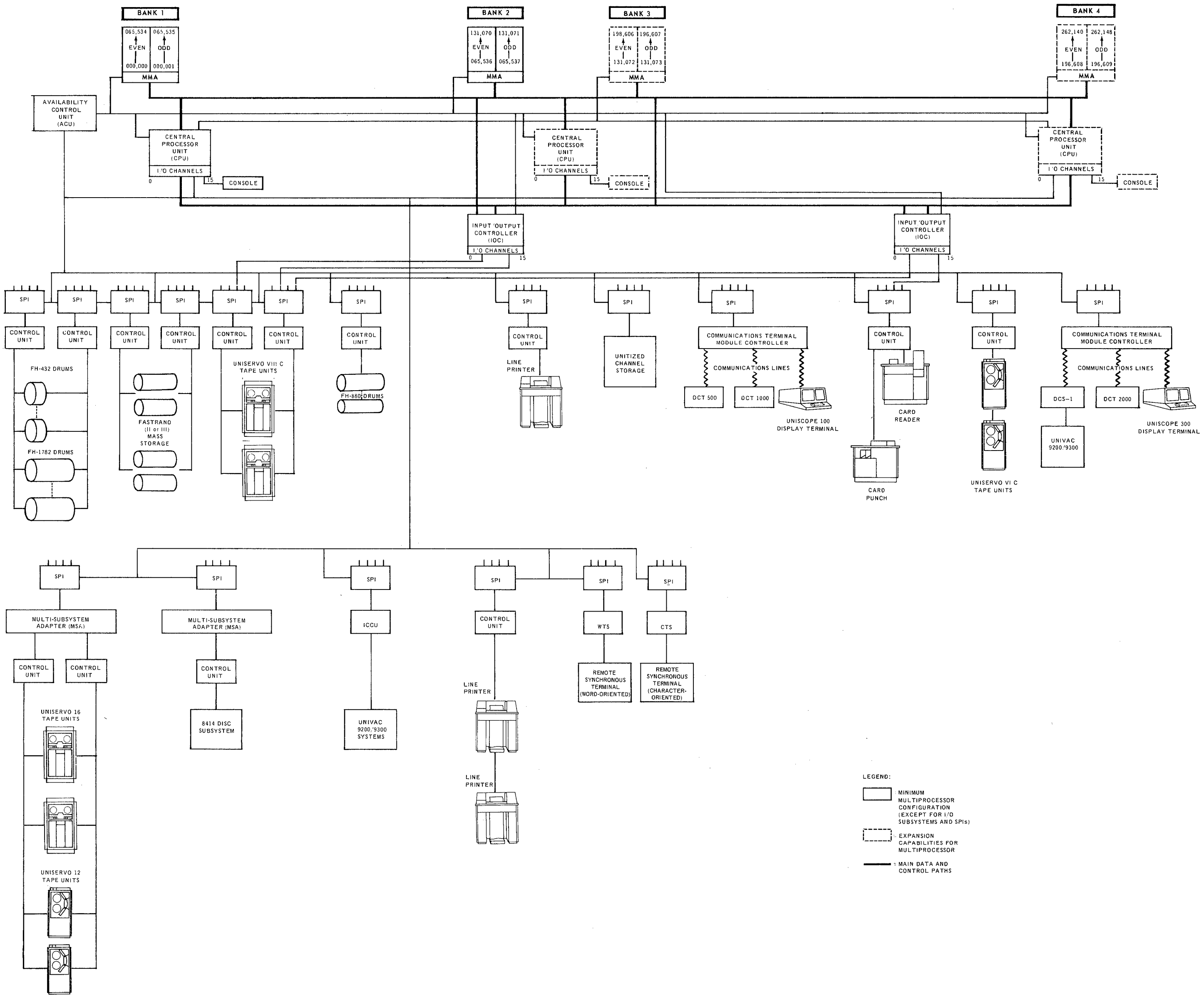

BANK 1
065,534
EVEN
000,000
065,535
ODD
000,001
MMA
BANK 2
131,070
EVEN
065,536
131,071
ODD
065,537
MMA
BANK 3
198,606
EVEN
131,072
196,607
ODD
131,073
MMA
BANK 4
262,140
EVEN
196,608
262,148
ODD
196,609
MMA
AVAILABILITY CONTROL UNIT (ACU)
CENTRAL PROCESSOR UNIT (CPU)
I/O CHANNELS
0
15
CONSOLE
CENTRAL PROCESSOR UNIT (CPU)
I/O CHANNELS
0
15
CONSOLE
CENTRAL PROCESSOR UNIT (CPU)
I/O CHANNELS
0
15
CONSOLE
INPUT/OUTPUT CONTROLLER (IOC)
I/O CHANNELS
0
15
INPUT/OUTPUT CONTROLLER (IOC)
I/O CHANNELS
0
15
SPI
CONTROL UNIT
FH-432 DRUMS
FH-1782 DRUMS
FASTRAND (II or III) MASS STORAGE
UNISERVO VIII C TAPE UNITS
FH-880 DRUMS
LINE PRINTER
UNITIZED CHANNEL STORAGE
COMMUNICATIONS TERMINAL MODULE CONTROLLER
COMMUNICATIONS LINES
DCT 500
DCT 1000
UNISCOPE 100 DISPLAY TERMINAL
CARD READER
CARD PUNCH
UNISERVO VI C TAPE UNITS
COMMUNICATIONS TERMINAL MODULE CONTROLLER
COMMUNICATIONS LINES
DCS–1
DCT 2000
UNIVAC 9200/9300
UNISCOPE 300 DISPLAY TERMINAL
MULTI-SUBSYSTEM ADAPTER (MSA)
UNISERVO 16 TAPE UNITS
UNISERVO 12 TAPE UNITS
MULTI-SUBSYSTEM ADAPTER (MSA)
8414 DISC SUBSYSTEM
ICCU
UNIVAC 9200/9300 SYSTEMS
LINE PRINTER
LINE PRINTER
WTS
REMOTE SYNCHRONOUS TERMINAL (WORD-ORIENTED)
CTS
REMOTE SYNCHRONOUS TERMINAL (CHARACTER-ORIENTED)
LEGEND:
= MINIMUM MULTIPROCESSOR CONFIGURATION (EXCEPT FOR I/O SUBSYSTEMS AND SPIs)
= EXPANSION CAPABILITIES FOR MULTIPROCESSOR
= MAIN DATA AND CONTROL PATHS

Figure 2–7. Multi-Processor System

2.4. UNIVAC ARRAY PROCESSOR

The UNIVAC Array Processor (UAP) is a special purpose processor used to manipulate large arrays of data such as seismic, geologic, geodetic, weather, medical, and so forth. The UAP is designed to operate as a subsystem with the 1108 Operating System, thus relieving the 1108 processor of the burden of the repetitive manipulation and summations of large arrays. UAP is interfaced by means of an I/O channel and accesses main storage by means of Multi-Module Access (MMA) Units or Shared Memory Interface (SPI).

The UAP is designed to provide a balance between hardware complexity and a maximum of efficiency and flexibility of use. Its design is specifically structured for the specialized tasks of convolution, spectrum analysis, and other array manipulations. The features of the UAP include a 36-bit floating-point capability; fixed-point capabilities include both 36- and 12-bit words.

The UAP has its own arithmetic and indexing registers. It includes 20 hardware instructions for the rapid processing of arithmetic operations associated with matrices or vector processing. The most important commands are convolution, Fast Fourier Transform (FFT), and interpolation.

The instruction repertoire can have many variations depending upon its function. For example, all add operations can be subtract operations. Control bits (stacking or summing bits) allow the contents of the corresponding destination addresses to be added to the operation results before storage back at the destination.

Several 1108 Processor/Array configurations are possible. The minimum configuration is an Array Processor and one 1108 processor with 131K main storage capacity. The largest configuration consists of an Array Processor and two 1108 processors with a total storage capacity of 262K main storage.

A feature kit consisting of printed cicuit cards is also available for field expansion of the Array Processor main storage capacity in increments of 65K up to the maximum capability of 262K.

3. MAIN STORAGE

3.1. GENERAL

The main storage of the UNIVAC 1108 Multi-Processor System is a high performance, fast access repository for instructions and data. Its design fully supports the concepts of multiprogramming, multiprocessing, modularity, and reliability around which the entire UNIVAC 1108 Multi-Processor System is constructed. Basic features of the 1108 main storage are:

- 750 nanosecond read/restore cycle time;
- 65,536, 131,072, 196,608, or 262,144, 36-bit words;
- Parity checking on all storage references;
- Access by up to three central processors and two IOC's;
- Modular expansion of two, four, six, or eight 32,768-word modules (one, two, three, or four 65,536-word module pairs or banks) in multi-module storage;
- Hardware storage protection – lockout boundaries establishable in 512-word increments;
- Relative addressing and dynamic program relocatability through program base registers;
- Online serviceability – module pairs may be removed for servicing without stopping the entire system;
- Overlapping/interleaved main storage access in multi-module storage to boost processor performance and to minimize access conflicts among processors.

While these features are all discussed generally as storage features, many of them such as relative addressing, storage protection, overlapping/interleaving are actually functions of each processor. With proper multiprocessor system organization, main storage becomes a set of components of the system which are allocatable in the same manner as peripheral devices. In realizing this objective, the design departs from the traditional close integration of the processor and the storage elements in the following ways:

- Main storage is composed of independently accessible modules; yet it presents a continuous addressing structure to the processors.
- In order to service more than one processor, a method of establishing priority among processors (CPU's and IOC's) at each module is provided in case two or more processors attempt to reference the same module simultaneously.
- To ensure that a processor will wait for access to storage, the processor and the module communicate on a request/acknowledge basis.

With these considerations in mind, the main storage modules (by means of the MMA) become passive components which perform the following functions:

- Grant storage access to a number of processors on a priority basis;
- Accept an address from any processor;
- Store or retrieve a word at that address;
- Issue an acknowledgement signifying that a storage reference has been completed;
- Generate or check parity on all data and deliver an interrupt signal to the processor requesting access should a parity error occur.

This processor/module relationship has significant advantages for the immediate as well as the future needs of the system. Addition of processors or banks of storage is simplified. It becomes a simple matter to add processors or storage elements, or to replace them with improved equipment, module by module, as technology advances.

3.2. STORAGE MODULE

The basic storage module of multi-module storage includes 32,768 words of ferrite core array. Each word is 36 bits long and carries two additional parity bits in nonaddressable levels, one bit for each half-word. The main components of each type of module are a 15-bit address register, a 36-bit read/restore register, parity checking circuits, and request/acknowledge circuits.

The 15-bit address register of each storage module provides addressing for 32,768 words. Since an 18-bit address is generated within the processor for each storage reference, three bits are available for selection of one of the eight possible storage modules.

Parity is checked on reading or calculated on writing for each storage access. If a parity error is detected, the storage module sends a parity error interrupt signal to the processor which it is currently serving, and rewrites the word in its incorrect form to ensure subsequent data errors when the word is again referenced. Preservation of the error in this way facilitates fault location, since the Executive can determine whether the failure is transient or is associated with a marginal or complete failure of the module.

3.3. MULTI-MODULE ACCESS UNIT (MMA)

In a multiprocessor system, an MMA unit is connected between each pair of main storage modules and the processors which may reference it. The MMA unit furnishes five priority-ordered processor connection paths (one for each CPU and one for each IOC) to each of the modules of the pair. Should an access conflict occur among processors, the MMA grants storage access to the processor having the highest priority, then the next, and so on. Communications between a processor and a single storage module can, therefore, be asynchronous; if the storage module is busy servicing one processor, a passive wait cycle is induced in others of lower priority that may be requesting access. Because a delay in honoring an input/output transfer can result in an undesirable "go-around" on drum, reread or rewrite on tape, or actual loss of data in the case of real-time input, Input/Output Controllers are ordinarily attached to the higher priority inputs of the MMA, followed by CPU's, which have built-in precedence of I/O over computational activities.

3.4. PACKAGING

Two 32,768-word storage modules (module pair) within a single cabinet constitute a bank. An adjacent cabinet contains dc power supplies for operation of the bank and the associated MMA.

3.5. STORAGE CAPACITY

Available storage capacity ranges from 65,536 words to the system maximum of 262,144 words, in steps of 65,536 words, according to the following:

- 65,536 words (two modules) – Minimum for unit processor system
- 131,072 words (four modules)– Minimum for multiprocessor system
- 196,608 words (six modules)
- 262,144 words (eight modules)

3.6. ADDRESSING

Two special techniques for referencing the main storage modules are used to increase processor performance and to reduce the occurrence of multiprocessor access conflicts. The first, called overlapping, enables the CPU to retrieve the current operand and the next instruction simultaneously; the second, called interleaving, enables two processors (CPU's, IOC's, or CPU and IOC) to access a pair of modules with minimum access conflicts.

3.6.1. Overlapping

The CPU can determine whether its current operand and next instruction lie in different storage modules, and if they do, the CPU retrieves the two words in parallel from main storage at an effective 100 percent performance increase.

The overlapping feature permits the separation of the instruction and data of a program into separate physical banks. Furthermore, the base register of the CPU allows either the instruction or data area of a program to be relocated independently—a significant advantage in storage compacting to overcome program fragmentation.

3.6.2. Interleaving

Interleaving is a method of addressing a main storage module pair (bank) in an even/odd fashion. It significantly reduces storage access conflicts in a multiprocessor system, and thereby increases overall system performance. With interleaving, one module of a pair contains all even numbered locations and the other contains all odd numbered locations. Thus, in a fully-expanded eight module system, modules 0, 2, 4, 6 are referenced for even addresses while modules 1,3, 5, 7 are referenced for odd. The even/odd module pairs consist of modules 0 and 1, 2 and 3, 4 and 5, and 6 and 7.

For a practical example, substitute the letters A, B, C, D for the modules contained in two banks, and assume data is being stored sequentially by a program. (The same assumption may be made for instructions being executed sequentially by the same program.) With the overlapping feature, assume processor number 1 starts executing instructions and retrieving data, with the instruction area in bank 1 and the data area in bank 2. For simplicity, assume the starting instruction and data addresses are at even numbered locations. The processor will then reference module A-B-A-B. . . for sequential instructions, and C-D-C-D. . . for sequential data locations. In any single storage interval, either modules A-C or B-D will be busy while their alternates will be idle. If another processor starts an identical process, but references an odd address to begin with, both processors may run concurrently without one impeding the operation of the other.

Assuming that both processors in the above example started at even addresses, the processor with lower priority passively waits one storage cycle after which the two are again in synchronization and may operate simultaneously.

3.7. STORAGE PROTECTION

To prevent inadvertent program reference to out-of-range storage addresses, the 1108 processor includes a hardware storage protection feature. The controlling element in this feature is the Storage Limits Register, the contents of which are as follows:

INSTRUCTION AREA		DATA AREA	
UPPER BOUNDARY	LOWER BOUNDARY	UPPER BOUNDARY	LOWER BOUNDARY
35 27	26 18	17 9	8 0

The Storage Limits Register (SLR) can be loaded by the Executive system to establish allowable operating areas for the program currently in execution. These areas are termed the program instruction (I) and data (D) areas. Before control is given to a particular program, the Executive loads the SLR with the appropriate I and D boundaries.

Before each main storage reference, the processor performs a limits check on the address, comparing it against the limits of either the I or D field of the SLR. An out-of-limits address generates a guard mode interrupt, thereby allowing the Executive to regain control and take appropriate action.

3.7.1. Storage Protection Modes

The Executive system can establish two different modes of storage protection by means of control fields in the Processor State Register (PSR) described in Section 4. Normally, the Executive itself operates in open mode; that is, the SLR may be loaded but the PSR is set to disregard this, and the Executive can reference any location in main storage.

3.7.1.1. Privileged Mode

Another mode can be established in the PSR for privileged programs. This privileged mode protects against out-of-bounds writes. Privileged programs (such as real-time programs or Executive-controlled subroutines) may enter nonalterable (re-entrant) subroutines, which are part of the Executive. Though these privileged programs are assumed to be thoroughly checked out, the system is still fully protected against unexpected occurrences since write protection is in effect.

3.7.1.2. User Program Mode

In the user program mode, read, write, and jump storage protection is in effect. Therefore, user programs are limited to those areas assigned by the Executive. If the user program reads, writes, or jumps to an out-of-limits address, an interrupt returns control to the Executive for remedial action.

Read/jump protection allows the Executive to stop the program at the point of error, terminate it, and provide diagnostic information to the programmer, thereby minimizing wasted time and smoothing the checkout process.

A particular advantage of read/jump protection is that classified (confidential) programs can be confidently run together; they are fully protected from audit (inadvertent or otherwise) by other programs.

3.8. RELATIVE ADDRESSING

Relative addressing is a feature of great significance in multiprogramming, time-sharing, and real-time operations, for it allows storage assignments for one program (the one going into execution) to be changed dynamically by the Executive to provide continuous storage for operation of another program, and it permits programs to dynamically request additional main storage according to processing needs. An additional advantage is that systems programs stored in auxiliary storage may be brought in for operation in any available area without complicated relocation algorithms.

Relative addressing is provided for through base registers contained within the CPU. Two separate registers control the basing of the program instruction and data bank, and a third register controls the selection of the appropriate base register.

4. 1108 PROCESSOR

4.1. GENERAL

The UNIVAC 1108 Central Processor Unit (CPU) is the principal component of the UNIVAC 1108 Multi-Processor System and, generally, the one by which the entire system is identified. It can operate under Executive or user program modes of control; it performs both arithmetic and logical operations; and it accommodates and supervises up to 16 input/output channels.

4.2. PRINCIPAL SECTIONS

The processor is logically divided into six interacting sections, each of which is identified and briefly described below.

- Control Registers – The CPU has 128 program-addressable control registers used for arithmetic operations, indexing, and input/output a·cess control.

- Arithmetic Section – This section contains the adder registers and control circuits necessary for performing fixed and floating-point arithmetic, partial-word selection, shifting, logical operations, and tests.

- Control Section – This section provides the basic control and logic for instruction decoding and execution. It includes the Program Address Counter used for the sequential accession of instructions; the Program Control Register in which instructions are staticized for execution; and the Processor State Register (PSR), which determines various processor operating modes. The Control Section also services interrupts.

- Input/Output Section – This section controls and multiplexes data flow between main storage and 16 input/output channels. It includes an interrupt priority network and paths to peripheral subsystems for both control signals and data.

- Indexing Section – This section contains parallel index adders and threshold test circuitry. It is used generally for processor control functions, operand address development, program relocation, and input/output transfer control.

- Storage Class Control Section – The Storage Class Control section receives the final operand address from the index adder and establishes address and data paths to one of eight possible storage modules. Storage Class Control also determines whether a final address refers to a control register.

4.3. INSTRUCTION WORD FORMAT

The format of the 1108 instruction word is illustrated below followed by an explanation of each field. Some fields have more than one meaning depending on the class of instruction.

f	j	a	x	h	i	u
35 30	29 26	25 22	21 18	17	16	15 0

4.3.1. Function Code – f Field

These six bits specify the operation to be performed. For function codes above 70_8, the f and j fields are combined to produce a 10-bit function code. An illegal function code generates an interrupt.

4.3.2. Partial-Word or Immediate-Operand Designator – j Field

For function codes less than 70_8, the j designator specifies partial-word or immediate-operand selection. (See Figure 4–2 for specific partial-word selections.)

4.3.3. Control Register Designator – a Field

The four-bit a field designates which control register, within a group selected by the function code, is involved in the operation. For some operations, the a field refers to an arithmetic register; for others, it refers to either an index register or some other control register. In input/output instructions, it specifies the channel and its associated input or output access control register. For function code 70_8, the a and j fields together address one of the 128 control registers.

4.3.4. Index Register Designator – x Field

The x field specifies one of the 15 index registers to be used in address modification. When the x field is set to 00_8, indexing is suppressed.

4.3.5. Index Modification Designator – h Field

The h field controls modification of the index value (Xm) by the increment field (Xi) after indexing (see 4.4.1). If h = 1, the right half of the index register is modified by the contents of its left half; if h = 0, modification is suppressed.

4.3.6. Indirect Address Designator – i Field

The i field controls the use of indirect addressing during instruction execution. If i = 0, the instruction functions normally. If i = 1, the 22 least significant bit positions of the instruction (x, h, i, and u fields) are replaced in the instruction register with the contents of the 22 least significant bit positions of (U). Indirect addressing continues as long as i = 1 with full indexing capability at each level.

4.3.7. Address Field – u Field

The u field normally specifies the operand address. However, for certain instructions it holds constants. For example, the shift instructions use the seven least significant bit positions to hold the shift count. In all instructions, the value in the u field may be modified by the contents of an index register.

4.4. CONTROL REGISTERS

The 128 program-addressable control registers are grouped to provide multiple index registers, accumulators, input/output access control registers, and special registers (see Figure 4–1).

The control registers are 36-bit integrated-circuit registers, with a basic cycle time of 125 nanoseconds. Two parity bits are included with each control register.

Effective use of multiple accumulators and index registers for the development and use of constants, index values, and operands substantially improves performance. UNIVAC 1108 compilers, for example, perform significantly better through multiple register usage and can produce highly efficient code.

In the following descriptions only programmable registers are discussed. The Executive, through modes established by the Processor State Register, has exclusive use of the duplicate set of control registers as well as the access control registers indicated by the shaded areas in Figure 4–1.

4.4.1. Index Registers

Control register locations $1_8 - 17_8$ are Index Registers and have the following format:

X_i	X_m
35 MODIFIER INCREMENT OR DECREMENT 18	17 INDEX MODIFIER 0

The Xm portion of the index register is an 18-bit modifier to be added to the base operand address of the instruction. The Xi portion of the index word updates the Xm portion, *after* base operand address modification.

Index register modification is specified by a 1 bit in the h field of the instruction, while indexing itself is specified by a nonzero value in the x field. Both functions take place within the basic instruction execution cycle.

When cascaded indirect addressing is used in a programmed operation, full indexing capabilities are provided at each level. Indirect addressing replaces the x, h, i, and u portions of the instruction register, beginning with a new indexing cycle for each cascaded sequence. This process continues until the i field is zero.

Index register 00, while program addressable, stores the contents of the Processor State Register (PSR) upon occurrence of an interrupt. Since its contents are overwritten at each interrupt, it is not generally useful for programming purposes.

4.4.2. Arithmetic Accumulators

Control register locations (14_8 – 33_8) are arithmetic accumulators for programmed storage of arithmetic operands and results. The computation is performed in other registers within the arithmetic section.

Depending upon the instruction, the accumulators are used to hold a variety of word formats. Double-precision instructions and a number of logical instructions reference two contiguous accumulators, that is, A and A + 1. In arithmetic operations, A + 1 always holds the least significant part of an operand or result. Some instructions, such as single precision floating-point operations, call on a one-word operand from main storage but produce a two-word result in the specified A and A + 1 registers.

4.4.3. Access Control Registers

Control register locations (40_8 – 77_8) are Input and Output Access Control Registers (ACR's). They are guard mode protected and may be referenced only by the Executive. Formats of the access control words are detailed in Section 5.

The word-by-word transmission of data over an I/O channel is governed by the contents of the ACR's. Two ACR's, one for input and one for output, are assigned to each of the sixteen channels. Input ACR's (locations 40_8 – 57_8) control input data transfers while output ACR's (locations 60_8 – 77_8) govern the transmission of output data and function words.

When an input/output operation is initiated, the programmed access control word (ACW) is loaded into the ACR corresponding to the channel associated with the specified peripheral unit.

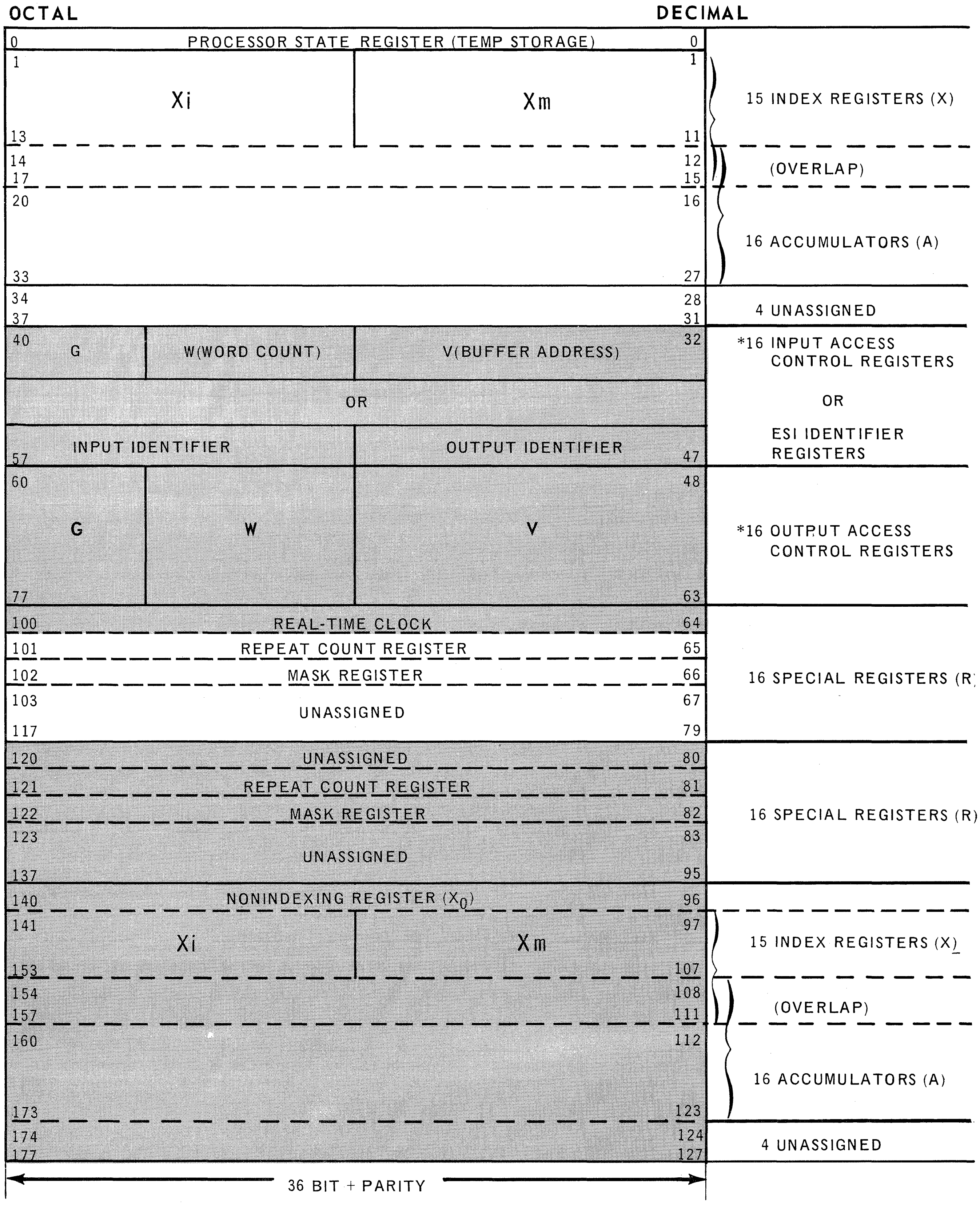
4.4.4. R Registers

The sixteen control register locations (100_8 – 117_8) are R registers. The first three of these (R0, R1, R2) have specified functions and formats as described below. The remaining R registers are not specifically assigned; typically they are used as loop counters, transient registers, or storage for intermediate values or constants.

4.4.4.1. R0 – Real Time Clock

UNASSIGNED		CLOCK COUNT	
35	18	17	0

This register is initially loaded by the program. The contents are then decremented once each 200 microseconds. A real-time clock interrupt occurs when the clock count goes through zero. Thus, if the clock is initially loaded with the value 5000, an interrupt occurs in exactly one second.



OCTAL				DECIMAL	
0	PROCESSOR STATE REGISTER (TEMP STORAGE)			0	
1 13	Xi		Xm	1 11	15 INDEX REGISTERS (X)
14 17				12 15	(OVERLAP)
20 33				16 27	16 ACCUMULATORS (A)
34 37				28 31	4 UNASSIGNED
40	G	W(WORD COUNT)	V(BUFFER ADDRESS)	32	*16 INPUT ACCESS CONTROL REGISTERS
	OR				OR
57	INPUT IDENTIFIER		OUTPUT IDENTIFIER	47	ESI IDENTIFIER REGISTERS
60 77	G	W	V	48 63	*16 OUTPUT ACCESS CONTROL REGISTERS
100	REAL-TIME CLOCK			64	16 SPECIAL REGISTERS (R
101	REPEAT COUNT REGISTER			65	
102	MASK REGISTER			66	
103 117	UNASSIGNED			67 79	
120	UNASSIGNED			80	16 SPECIAL REGISTERS (R)
121	REPEAT COUNT REGISTER			81	
122	MASK REGISTER			82	
123 137	UNASSIGNED			83 95	
140	NONINDEXING REGISTER (X_0)			96	
141 153	Xi		Xm	97 107	15 INDEX REGISTERS (X)
154 157				108 111	(OVERLAP)
160 173				112 123	16 ACCUMULATORS (A)
174 177				124 127	4 UNASSIGNED

▨ = EXECUTIVE (GUARD MODE PROTECTED)

*See 5.2.

Figure 4–1. Control Register Address Assignments

4.4.4.2. R1 – Repeat Counter

UNASSIGNED	REPEAT COUNT (k)
35 18	17 0

The repeat counter controls repeated operations such as Block Transfer and search instructions. To execute a repeated instruction k times, the repeat counter is loaded with k prior to the execution of the instruction.

4.4.4.3. R2 – Mask Register

The mask register functions as a filter in determining which portions of words are to be tested in repeated masked search operations or in logical comparisons. (U) is compared to (A) only with respect to those positions which contain one's in the mask register. In repeated masked search operations, both the mask register and the repeat counter are loaded prior to executing the search command.

4.5. ARITHMETIC SECTION

In the UNIVAC 1108 System the manipulation of data (addition, subtraction, multiplication, division, shifting) takes place in the arithmetic section of the central processor. During the execution of an arithmetic instruction, storage registers within the arithmetic section itself are used for actual computation. The arithmetic section has the following characteristics:

- On fixed-point single precision instructions, the j designator selects all or a portion of one of the operands (half, third, quarter or sixth word) for use in the arithmetic operation.

- Special split-word arithmetic instructions provide for simultaneous addition or subtraction of corresponding half or third words of the two operands.

- When a shift matrix is used, a multiposition shift requires the same time as a one place shift. Right and left shifts of single or double length operands can be specified. Left shifting is logical (zeros are filled to the right) or circular (end-around). Right shifts may be either logical, algebraic (sign bits are filled to the left), or circular.

- Sixteen arithmetic registers in the control register section, acting as sixteen accumulators, allow parallel and cumulative computation. Full double-precision floating point arithmetic is provided.

- When the results of arithmetic operations are in double-length form, they are automatically stored in consecutive control registers and are available for retrieval as double-length results.

- Comparisons utilizing the mask register allow any selection of bits in one 36-bit word to be directly compared with corresponding bits of another word.

4.5.1. The Adder

The adder in the 1108 Processor is a ones complement subtractive adder for 36-bit or 72-bit operations. For purposes of analysis and debugging, the programmer may manually simulate the computer operation by simple binary or octal addition.

Two special internal designators associated with the arithmetic adder are the overflow designator and the carry designator. The fixed-point addition and subtraction instructions, single and double precision, are the only instructions which affect these two designators.

Before the execution of one of these instructions both designators are cleared. The overflow designator is set upon generation of a significant bit in the sign position. Thus a positive result from two negative quantities or a negative result from two positive quantities sets the overflow designator. The carry designator is set whenever an end-around carry is generated.

After the instruction has been performed, the designators remain either set or clear until another of the designated arithmetic instructions is initiated. Both designators are set in time to be tested immediately after the specified instruction has been executed.

When an interrupt occurs, the hardware stores the settings of the carry and overflow designators in the processor State Register (see 4.6.1) and control passes to the Executive system. This information is automatically returned to the designators when the Executive returns control to the interrupted program.

4.5.2. Arithmetic Accumulators

The sixteen arithmetic accumulators can be addressed directly by the programmer and are available for storing operands and results of arithmetic computations. These arithmetic accumulators should not be confused with the nonaddressable transient registers contained within the arithmetic section itself used in actual computation.

With the Add To X and Add Negative To X instructions, the index registers also act as accumulators in the same manner as the arithmetic registers.

4.5.3. Partial-Word Transfers

To minimize shifting and masking and to allow computation based on selected portions of words, the 1108 System permits the transfer of partial words into and out of the arithmetic section in a varying pattern (see Figure 4–2).

By selecting the coding of the j field in the instruction word and bit 17 of the Processor State Register, a programmer may transfer a chosen portion of an operand to or from a control register or the arithmetic section. The transfer to an arithmetic register may also be accompanied by sign extension for subsequent arithmetic operations, depending on the j field.

J	PSR BIT 17	BIT POSITIONS OF (U)→ A, X, or R	BIT POSITIONS OF (A), (X), or (R)→U
00	–	35–00 → 35–00	35–00 → 35–00
01	–	17–00 → 17–00	17–00 → 17–00
02	–	35–18 → 17–00	17–00 → 35–18
03	–	17–00 → S 17–00	17–00 → 17–00
04	0	35–18 → S 17–00	17–00 → 35–18
	1	26–18 → 08–00	08–00 → 26–18
05	0	11–00 → S 11–00	11–00 → 11–00
	1	08–00 → 08–00	08–00 → 08–00
06	0	23–12 → S 11–00	11–00 → 23–12
	1	17–09 → 08–00	08–00 → 17–09
07	0	35–24 → S 11–00	11–00 → 35–24
	1	35–27 → 08–00	08–00 → 35–27
10	–	05–00 → 05–00	05–00 → 05–00
11	–	11–06 → 05–00	05–00 → 11–06
12	–	17–12 → 05–00	05–00 → 17–12
13	–	23–18 → 05–00	05–00 → 23–18
14	–	29–24 → 05–00	05–00 → 29–24
15	–	35–30 → 05–00	05–00 → 35–30
16	–	18 bits* → 17–00	NO TRANSFER
17	–	18 bits* → S 17–00	NO TRANSFER

* If x = 0, h, i, and u are transferred
If $x \neq 0$, $u + (X_x)_m$ is transferred

S = Sign Extension, where the sign is that of the j-determined final contents of A.

Figure 4–2. j-Determined Partial-Word Operation

4.5.4. Split-Word Arithmetic

The System can perform addition and subtraction of half-words or third-words simultaneously. The right halves of two operands, for example, are added and the sum is stored in the right half of the selected accumulator. At the same time, the left halves of the same two operands are added and the result is stored in the left half of the same accumulator. There is no carry interaction between the halves. The same holds true for thirds of words. Each partial word operates as an independent arithmetic register with its own end-around carry.

4.5.5. Shifting

The System can perform both single-length shifting (36 bits) or double-length shifting (72 bits), treating the latter as if operating with a single 72-bit register. A high speed shift matrix makes execution time independent of the number of places involved in the shift, which means that an operand can be shifted from 0 to 72 positions in one main storage cycle time.

Six types of shift operations are provided:

- Right Circular – bits shifted out at the right reappear at the left.
- Left Circular – bits shifted out at the left reappear at the right.
- Right Logical – zeros replace bits shifted out of the most significant positions.
- Left Logical – zeros replace bits shifted out of the least significant positions.
- Right Algebraic – sign bits replace bits shifted out of the most significant positions.
- Scale-Factor Shift – a single or double accumulator left shift which positions the word and simultaneously counts the number of shifts required until $(A_{35}) \neq (A_{34})$.

4.5.6. Double-Precision Fixed Point Arithmetic

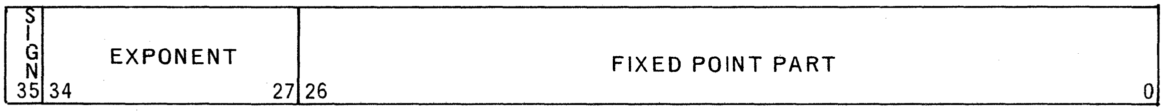
The System provides 72-bit, double-precision fixed point addition and subtraction. Operands are processed as if they occupied a single 72-bit register. Bit 71, the high order bit, is the sign bit.

In addition, several arithmetic instructions produce two-word results. With fixed point multiplication, a double-length product is stored in two arithmetic registers for integer and fractional operations. Integer and fractional division is performed upon a double-length dividend with the quotient retained in A and the remainder retained in A + 1.

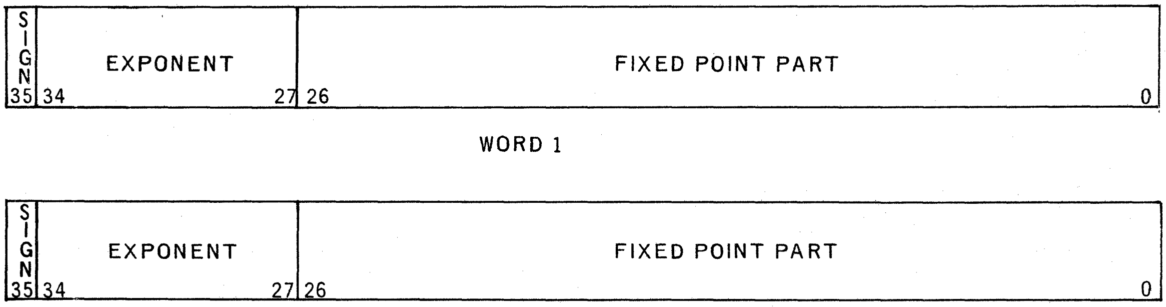
4.5.7. Floating-Point Arithmetic

The System is equipped with an extensive hardware repertoire of floating-point instructions. If the arithmetic is single precision, the range is from 10^{38} to 10^{-38} with eight-digit precision. The word formats are given below.

Source Operand Format



Result Format



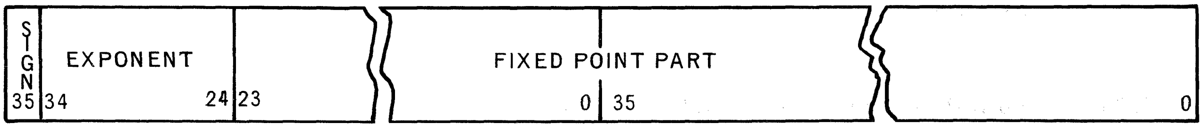
WORD 1

WORD 2

In a single-precision floating point operation word 1 is the more significant portion of the result. Word 2 contains the less significant portion. Mathematical error tracing can determine how much accuracy is being lost in calculations using this format. The least significant word is displaced 27 bits to the right of the binary point in the significant word. Hence, its exponent is always adjusted by −27. The two-word result of this single-precision operation is stored in two contiguous Arithmetic Registers.

If the arithmetic is double precision, the range is from 10^{307} to 10^{-308} with 18–digit precision. The values are expressed in two adjacent words, as shown in the following format.

Source and Result Format



Full double-precision operations do not require a repeated sign and exponent in the 36 least significant bits.

In any of the floating-point formats the exponent can assume a range of values as follows:

Single precision:	(8 bits)	000–255
Double precision:	(11 bits)	0000–2047

To express negative exponents, the hardware biases or floats the exponent on a midvalue. The sign bit of the floating-point word applies to the fixed-point part. The true and biased ranges of the exponent are as follows:

	True	Biased
Single precision:	-128_{10} to $+127_{10}$	$0 - 255_{10}$
Double precision:	-1024_{10} to $+1023_{10}$	$0 - 2047_{10}$

A positive fixed-point part is normally assumed to be in range ½ to 1. Such a value places a 1 bit in the most significant bit position. When this condition exists, the floating-point number is said to be normalized. A negative fixed-point part causes the entire floating-point word to be complemented, and a 0 appears in this position.

Floating-point instructions are also provided for the following operations:

- Determining differences in exponents;
- Packing and unpacking exponents and fixed-point parts (single and double precision);
- Conversion – Single to double precision
  Double to single precision.

4.6. EXECUTIVE SYSTEM CONTROL FEATURES

To maintain the multiprogramming and multiprocessing environment, the Executive must have complete control of the entire UNIVAC 1108 System. Special hardware features are provided to permit this control.

The multiprogramming and multiprocessing capabilities of the system are based upon guard mode operation. In this mode certain instructions, registers, and storage locations are available for the exclusive use of the Executive system. Under the guard mode, unrelated programs cannot interact.

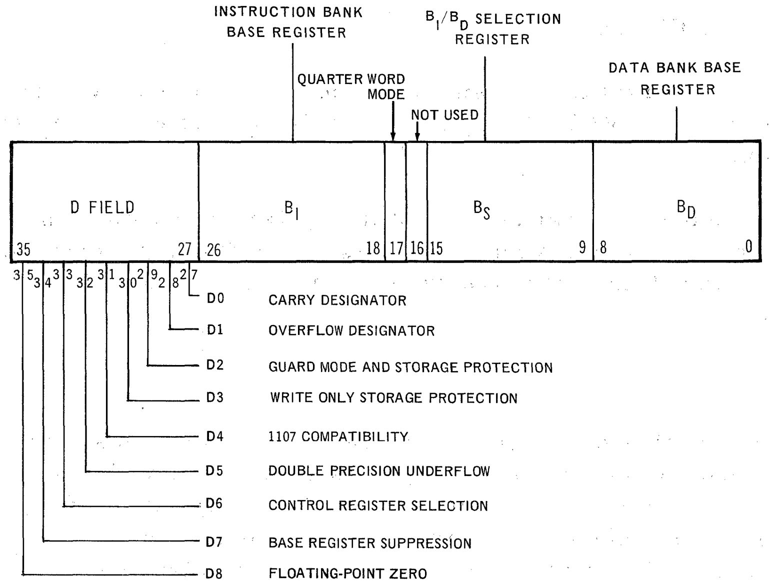
4.6.1. Processor State Register

The Processor State Register (PSR) stores a 36-bit representation of various states and conditions affecting the current operations of the processor. By means of this register, the Executive sets up control modes for itself, governs the operation of user programs, and registers status information concerning user programs when it regains control as a result of an interrupt. Figure 4–3 explains in detail the significance of each bit of the PSR.

The Executive uses a special instruction, Load Processor State Register, for loading the PSR, and governing the following functions and conditions:

- Program base addresses
- Quarter word operations in the processor
- Carry and overflow status
- Guard mode
- Storage protection mode
- 1107 compatibility mode
- Double-precision floating-point underflow mode interrupt suppression
- Base register suppression
- Control register process selection
- IBM 7090 floating-point compatibility mode

The contents of the PSR are stored automatically as soon as an interrupt occurs. Program carry and overflow status are first stored in PSR and then its contents are transferred to index register location 00. The PSR is then force-cleared in preparation for Executive operations. The Executive saves the contents of index register 00 so that it can reinstate conditions when control is returned to the program that was interrupted.


INSTRUCTION BANK BASE REGISTER
B$_I$/B$_D$ SELECTION REGISTER
QUARTER WORD MODE
NOT USED
DATA BANK BASE REGISTER
D FIELD
B$_I$
B$_S$
B$_D$
35
27
26
18
17
16
15
9
8
0
35
34
33
32
31
30
29
28
27
D0 CARRY DESIGNATOR
D1 OVERFLOW DESIGNATOR
D2 GUARD MODE AND STORAGE PROTECTION
D3 WRITE ONLY STORAGE PROTECTION
D4 1107 COMPATIBILITY
D5 DOUBLE PRECISION UNDERFLOW
D6 CONTROL REGISTER SELECTION
D7 BASE REGISTER SUPPRESSION
D8 FLOATING-POINT ZERO

D FIELD

D8 7090 FLOATING-POINT COMPATIBILITY MODE

=0 Clears exponent to zero when a fixed point part equal to zero is generated

=1 Produces relative floating point 0

D7 BASE REGISTER SUPPRESSION

=0 Allows contents of base registers to be added to every U address.

=1 Base register addition on storage reference is suppressed when instruction i-designator= 1.

D6 CONTROL REGISTER SECTION

=0 Selects user program control register set (locations 00_8–37_8, 101_8–117_8)

=1 Selects Executive control register set (locations 120_8–177_8)

The Executive passes control to user programs with D6 = 0, which selects the worker set. An interrupt forces this bit to 1 after (PSR) has been transferred to control register X0, making the upper control registers available to the Executive.

D5 DOUBLE-PRECISION UNDERFLOW – Double Precision Floating-Point Operations

=0 Interrupts on double-precision floating underflow

=1 Clears results to zero and continues

This is a program-requestable option which is set up for the program by the Executive.

D4 1107 COMPATIBILITY MODE

=0 1108 mode – full range addressing

=1 The upper two bits of the effective address (U) are stripped, allowing 1107 program compatibility with the 1108. In this mode, only 65,536 words of storage are available.

D3 MODIFIED STORAGE PROTECTION (WRITE ONLY)

=0 Read, write and jump storage protection under guard mode

=1 Write protection only (even if D2=0)

D2 GUARD MODE

=0 Guard mode off. All instructions and all storage references to access control registers (40_8–77_8), the real time clock (100_8) and executive control registers (120_8–177_8) are permitted.

=1 Guard mode on. Invalidates all instructions and control register references described above to enforce the integrity of the system. Only when guard mode is on are the contents of the storage limits register effective in storage protection. (If D3 = 1, there is no read or jump protection.)

D1 OVERFLOW DESIGNATOR

=1 Overflow (fixed-point addition or subtraction)

D0 CARRY DESIGNATOR

=1 Carry (fixed-point addition or subtraction)

OTHER FIELDS

26–18, 15– 9, 8– 0 } BASE REGISTERS. These registers provide the absolute base address values on which programs "float" in main storage during execution.

17 QUARTER-WORD MODE BIT

=1 Quarter-word mode effective

=0 Quarter-word mode not effective

Figure 4–3. Processor State Register Format

4.6.2. Interrupts

The interrupt network of the UNIVAC 1108 System is extensive. It is the means of effecting real-time, multiprogramming and time-sharing operations on the system. The interrupt is a control signal generated by either a peripheral subsystem (external interrupt) or the control section of the central processor. Specific interrupt locations are assigned within the lower regions of main storage for each condition. These interrupt locations are programmed to capture the interrupted address and enter interrupt response subroutines in the Executive system. The synchronization of input/output activities and response to real-time situations is accomplished through some of these interrupts.

Other interrupts are provided for certain error conditions within the central processor. These may result from a programming fault such as an illegal instruction, a main storage parity error, or a user program violation such as an attempt to write into a protected area of storage or a violation of guard mode. These fault interrupts are used by the Executive to initiate remedial or terminating action when they are encountered. Table 4–1 lists the fixed-address assignments. Note that all assigned locations are all interrupt locations except for 200_8 through 202_8, which receive status words and 216_8, which stores the day clock count.

4.6.3. Guard Mode

The guard mode prevents user programs from executing any of the instructions listed below. These instructions are reserved for the Executive. It also protects certain locations in main storage reserved for Executive operations.

Guard mode is established by the Load Processor State Register instruction. Execution of this instruction with the appropriate PSR bit pattern is the only way that guard mode can be made operative and provides the only direct access to the PSR. Under guard mode, an attempt to perform any of the privileged instructions or functions listed below results in a processor interrupt.

- Load Processor State Register
- Load Storage Limits Register
- Initiate Interprocessor Interrupt
- Select Interrupt Location
- Load Channel Select Register
- All I/O Instructions
- Prevent All I/O Interrupts and Jump
- Cascading indirect addressing for more than 100 microseconds
- Cascading the Executive instruction for more than 100 microseconds
- Attempting to write into any of the Executive control registers (40_8–100_8 or 120_8 –177_8)
- Alarm
- Disable Day Clock
- Enable Day Clock
- Load Last Address Register

Guard mode is disabled by the occurrence of any interrupt. This stores the contents of PSR in user index register 0, clears certain bit positions of the PSR, sets D6=1, and establishes Executive mode operation.

DECIMAL ADDRESS	OCTAL ADDRESS	FIXED ASSIGNMENT
128	200	Status Word for External Interrupt on CPU #0
129	201	Status Word for External Interrupt on CPU #1
130	202	Status Word for External Interrupt on CPU #2
131–135	203–207	Unassigned
136	210	Power Loss Interrupt
137	211	ESI Access Control Word Parity Error Interrupt
138	212	ISI Access Control Word Parity Error Interrupt
139	213	I/O Data Parity Error Interrupt
140–141	214–215	Unassigned
142	216	Day Clock Input
143	217	Day Clock Interrupt
144	220	ISI Input Monitor Interrupt
145	221	ISI Output Monitor Interrupt
146	222	ISI Function Monitor Interrupt
147	223	ISI External Interrupt
148	224	ESI Input Monitor Interrupt
149	225	ESI Output Monitor Interrupt
150	226	Unassigned
151	227	ESI External Interrupt
152	230	Unassigned
153	231	Real Time Clock Interrupt
154	232	Interprocessor Interrupt #0
155	233	Interprocessor Interrupt #1
156	234	Unassigned
157	235	Main Storage Parity Error Interrupt (MEM 2)
158	236	Main Storage Parity Error Interrupt (MEM 3)
159	237	Main Storage Parity Error Interrupt (MEM 4)
160	240	Control Register Parity Error Interrupt
161	241	Illegal Instruction Interrupt
162	242	Executive Return Interrupt
163	243	Guard Mode/Storage Limits Protection Fault Interrupt
164	244	Test and Set Interrupt
165	245	Floating-Point Characteristic Underflow Interrupt
166	246	Floating-Point Characteristic Overflow Interrupt
167	247	Divide Fault Interrupt
168–175	250–257	Unassigned
176–255	260–377	Status Words for External Interrupts from IOC's
Last Address	–1	Main Storage Parity Error Interrupt (MEM 1)

Table 4–1. Fixed-Address Assignments

4.7. INSTRUCTION REPERTOIRE

The UNIVAC 1108 central processor is provided with an unusually powerful and flexible instruction repertoire. Many 1108 instructions are effectively accessed and completed in one main storage cycle. In addition to a complete set of instructions, including an extremely fast set of single- and double-precision floating-point instructions, the repertoire includes a group which permits fast and simplified control by the Executive system operating in a multiprogramming or multiprocessing environment.

In the following discussion, the instructions in the 1108 repertoire are grouped by functional class to illustrate the power of the repertoire. Appendix C lists them numerically by function code (in octal) stating exactly what each one does. The function codes are listed here to facilitate reference to Appendix C.

4.7.1. Data Transfer Instructions

To load the Arithmetic registers:

Load A	10
Load Negative A	11
Load Magnitude A	12
Load Negative Magnitude A	13

To load the Index and R registers:

Load R	23
Load X Modifier	26
Load X	27
Load X Increment	46

To load two Arithmetic registers with one instruction:

Double Load A	71,13
Double Load Negative A	71,14
Double Load Magnitude A	71,15

To store the Arithmetic registers:

Store A	01
Store Negative A	02
Store Magnitude A	03

To store other control registers:

Store R	04
Store X	06

To store two Arithmetic registers with one instruction:

Double Store A	71,12

Two special purpose transfers:

Store Zero	05
Block Transfer	22

Any transfer instructions, except double-length transfers, move selected parts of words. That is, the partial-word feature allows any sixth, quarter, third, or half word to be loaded into the lower portion of an arithmetic register when using Load instructions. Similarly, when using a Store instruction any sixth, quarter, third or half word can be transmitted from the lower portion of an arithmetic register, index register, or R register to main storage.

4.7.2. Fixed-Point Arithmetic

Single-word operations on arithmetic registers:

Add to A	14
Add Negative A	15
Add Magnitude to A	16
Add Negative Magnitude to A	17
Add Upper	20
Add Negative Upper	21
Add to X	24
Add Negative to X	25
Multiply Integer	30
Multiply Single Integer	31
Multiply Fractional	32
Divide Integer	34
Divide Single Fractional	35
Divide Fractional	36

Double-length operations on two arithmetic registers:

Double Precision Fixed Point Add	71,10
Double Precision Fixed Point Add Negative	71,11

Special format operations:

Add Halves	72,04
Add Negative Halves	72,05
Add Thirds	72,06
Add Negative Thirds	72,07

4.7.3. Floating-Point Arithmetic

The repertoire includes both single- and double-precision floating point operations, using one-word and two-word operands, respectively. Ones complement arithmetic is used.

Single Precision:

Floating Add	76,00
Floating Add Negative	76,01
Floating Multiply	76,02
Floating Divide	76,03
Load and Unpack Floating	76,04
Load and Convert to Floating	76,05

Double Precision:

Double Precision Floating Add	76,10
Double Precision Floating Add Negative	76,11
Double Precision Floating Multiply	76,12
Double Precision Floating Divide	76,14
Double Load and Convert to Floating	76,15

Miscellaneous:

Magnitude of Characteristic Difference to Upper	76,06
Characteristic Difference to Upper	76,07
Floating Expand and Load	76,16
Floating Compress and Load	76,17

4.7.4. Index Register Instructions

These instructions can be used when modifying, loading, or storing the contents of index registers:

Store X	06
Add to X	24
Add Negative to X	25
Load X Modifier	26
Load X	27
Load X Increment	46
Test Less Than or Equal to Modifier	47
Jump Modifier Greater and Increment	74,12
Load Modifier and Jump	74,13

The a field of these instructions address the index register. Four of the index registers are overlapped with the arithmetic registers; thus all arithmetic instructions, such as multiply or shift, can operate directly on these four index registers.

4.7.5. Logical Instructions

The logical or Boolean operations are defined by the following truth tables.

Logical AND

	0	1
0	0	0
1	0	1

Inclusive OR

	0	1
0	0	1
1	1	1

Exclusive OR

	0	1
0	0	1
1	1	0

The three simple logical instructions are:

Logical OR	40
Logical Exclusive OR	41
Logical AND	42

One special instruction is also logical:

Masked Load Upper	43

Several other instructions such as the repeated masked search instructions employ logical operations in combination with other functions.

4.7.6. Shift Instructions

The twelve shift functions include circular, logical, and algebraic shifts. Circular shifts are end-around. Logical shifts fill in zeros on the end opposite the shift direction, whereas algebraic shifts fill in sign bits. The shift count (from 0 through 72 places) is taken from the u field (indexed when specified) of the shift instruction.

Right shift instructions:

Single Shift Circular	73,00
Double Shift Circular	73,01
Single Shift Logical	73,02
Double Shift Logical	73,03
Single Shift Algebraic	73,04
Double Shift Algebraic	73,05

Left shift instructions:

Load Shift and Count	73,06
Double Load Shift and Count	73,07
Left Single Shift Circular	73,10
Left Double Shift Circular	73,11
Left Single Shift Logical	73,12
Left Double Shift Logical	73,13

4.7.7. Repeated Search Instructions

Search instructions operate as repeated comparison operations, comparing the value at u with that in A. They skip the next instruction when a specified condition is met or take the next instruction in sequence when the repeat count in R1 has been decremented to zero.

Algebraic (Sign considered):

Search Equal	62
Search Not Equal	63
Search Less Than or Equal	64
Search Greater	65
Search Within Range	66
Search Not Within Range	67

Masked Algebraic (Sign Considered):

Masked Search Equal	71,00
Masked Search Not Equal	71,01
Masked Search Less Than or Equal	71,02
Masked Search Greater	71,03
Masked Search Within Range	71,04
Masked Search Not Within Range	71,05

Masked Alphanumeric (Unsigned):

Masked Alphanumeric Search Less Than or Equal	71,06
Masked Alphanumeric Search Greater	71,07

4.7.8. Unconditional Jump Instructions

These instructions transfer control to the location specified by the indexed u address:

Store Location and Jump	72,01
Load Modifier and Jump	74,13

4.7.9. Conditional Jump Instructions

These instructions make a comparison and if a specific condition is met, they transfer program control to the instruction location specified by u. If not, the next instruction in sequence is executed.

Jump Greater and Decrement	70 *
Double Precision Zero Jump	71,16
Jump Positive and Shift	72,02
Jump Negative and Shift	72,03
Jump Zero	74,00
Jump Non Zero	74,01
Jump Positive	74,02
Jump Negative	74,03
Jump Keys	74,04
Halt Keys and Jump	74,05
Jump No Low Bit	74,10
Jump Low Bit	74,11
Jump Modifier Greater and Increment	74,12
Jump Overflow	74,14
Jump No Overflow	74,15
Jump Carry	74,16
Jump No Carry	74,17
Jump Input Channel Busy	75,02
Jump Output Channel Busy	75,06
Jump Function in Channel	75,12

* The j and a designators specify one of 128 control registers.

4.7.10. Test (or Skip) Instructions

These instructions make a comparison and if the specified condition is met, the next instruction is skipped. If not, the next instruction is executed.

Test Even Parity	44
Test Odd Parity	45
Test Less Than or Equal to Modifier	47
Test Zero	50
Test Non Zero	51
Test Equal	52
Test Not Equal	53
Test Less Than or Equal	54
Test Greater	55
Test Within Range	56
Test Not Within Range	57
Test Positive	60
Test Negative	61
Double Precision Test Equal	71,17

4.7.11. Executive System Control Instructions

This group of instructions allows proper Executive system control of programs operating in a multiprogramming and multiprocessing environment.

Executive Return	72,11
Prevent All I/O Interrupts and Jump	72,13
Store Channel Number	72,14
Load Processor State Register	72,15
Load Storage Limits Register	72,16
Initiate Interprocessor Interrupt	73,14
Select Interrupt Locations	73,15
Load Channel Select Register/Load Last Address Register	73,16
Allow All I/O Interrupts and Jump	74,07

These instructions are used for establishing processor state, storage limits boundaries, interrupt locations, identification of I/O channels, and interprocessor communication and task assignment.

4.7.12. Input/Output Instructions

This group of instructions allows the Executive system to initiate, test, and control input/output operations. Monitored instructions interrupt the program when the indicated transfer is completed.

Load Input Channel	75,00
Load Input Channel and Monitor	75,01
Jump Input Channel Busy	75,02
Disconnect Input Channel	75,03
Load Output Channel	75,04
Load Output Channel and Monitor	75,05
Jump Output Channel Busy	75,06
Disconnect Output Channel	75,07
Load Function in Channel	75,10
Load Function in Channel and Monitor	75,11
Jump Function in Channel	75,12
Allow All Channel External Interrupts	75,14
Prevent All Channel External Interrupts	75,15

4.7.13. Other Instructions

Execute	72,10
Test and Set	73,17
No Operation	74,06

5. SHARED PROCESSING SYSTEM

5.1. GENERAL

The UNIVAC 1108 Shared Processing System is a general purpose, high performance multiple processor system providing increased input/output capabilities as well as enhanced computational performance. The Shared Processing System is basically a two processor configuration, one of which incorporates all the input/output of the system, while the other is dedicated to processing only. The computational processor, employing only control and arithmetic sections, is not effected by any I/O transfers and therefore is dedicated to processing regardless of type. The input/output processor, consisting of an input/output section as well as an arithmetic and control section, handles all data transfer within the system.

Among the principal features of the UNIVAC 1108 Shared Processing System are:

- 128 integrated circuit registers in each processor
- Shared, large, modular, parity checked, high-speed main storage
- Partial redundancy among systems components
- Program address relocation access shared memory
- Storage protection
- Partial word addressability in 6, 9, 12, and 18-bit portions as well as full-word (36-bits) and double-word (72-bits) addressing available to both processors
- High speed, random access, auxiliary storage
- Privileged mode for the Executive system in both processors
- Guard mode employed in both processors for user programs
- Fully software supported and available with the complete Executive library.
- Complete compatability with the 1108 Unit Processor, 1108 Multi-Processor, and 1106 Unit Processor Systems.

5.2. SYSTEM COMPONENTS

The UNIVAC 1108 Shared Processing System is organized to allow a number of tasks to be perfomed simultaneously under the direction of a common Executive control system. The system is composed of five types of components:

- Input/Output Processor
- Supporting Processor
- Main Storage
- Auxiliary Storage Subsystems
- Peripheral Subsystems

5.2.1. Input/Output Processor

The input/output processor can perform all arithmetic, input/output functions, and executive control. Included in this processor is a set of 128, 125-nanosecond integrated circuit control registers providing multiple accumulators, index registers, input/output access control registers, and special use registers. In addition, a duplicate set of the user registers is provided for the Executive.

The input/output processor services up to sixteen high-speed I/O channels and is responsible for the total input/output activity of the system. The Externally Specified Index (ESI) feature provides a basis for efficient communication processing.

The arithmetic section contains the adder registers and control registers necessary for performing fixed or floating point, single or double precision, arithmetic; partial-word selection; shifting; logical operations; and tests. A split-word arithmetic feature provides for simultaneous addition and subtraction of corresponding half or third words of two operands. A powerful shift matrix permits single or double word, left or right, logical or circular shifts in one storage cycle.

5.2.2. Computational Processor

The computational processor has all of the features of the I/O Processor with the exception of the input/output capability. The control and arithmetic sections of the input/output processor and the computational processor are identical. In essence, the computational processor ‘‘backends’’ the input/output Processor and by performing the bulk of computational activity frees the input/output processor to maximize input/output operations.

5.2.3. Main Storage

The main storage read/restore cycle time is 750 nanoseconds. The minimum storage of the Shared Processor System is 131,071, 36-bit words. It is expandable up to a maximum of 262,144, 36-bit words. Parity checking is provided on all storage references. Relative addressing and dynamic program relocatability is achieved through program base registers.

Main storage is shared between processors, and its performance is greatly enhanced through the interleaving and overlapping features. Conflicting access requests are resolved by the Multi-Module Access Unit.

5.2.4. Auxiliary Storage

The auxiliary magnetic drum storage subsystems are an integral part of each UNIVAC 1108 Shared Processor System. Up to eight FH-432 or FH-1782 magnetic drums, or any combination of the two types, may be attached to one or two control units. A minimum of three FH-432 drums are required by the UNIVAC 1108 Shared Processor System. The low access time and high transfer rate offered by these drum subsystems aid significantly in achieving the balanced system.

5.2.5. Peripheral Subsystems

Peripheral subsystems are attached to the input/output Processor through general purpose input/output channels. With its adaptable input/output arrangement, the UNIVAC 1108 Shared Processor System can communicate with many real-time devices such as analog/digital converters, key sets, communication terminals, tracking and radar systems, display systems, and other information processing systems.

The minimum UNIVAC 1108 Shared Processor System requires 131K words of main storage, one FH-432/1782 Magnetic Drum Subsystem with three FH-432 drums or one FH-1782 drum, one FASTRAND subsystem with one FASTRAND II or FASTRAND III drum, one UNISERVO Magnetic Tape Subsystem with either two UNISERVO VI–C or two UNISERVO VIII–C tape units, a Display Console and one online UNIVAC 9300 System or equivalent card and printer capabilities.

Figure 5–1 depicts a 131K storage configuration for the UNIVAC 1108 Shared Processor System. The input/output processor consists of the command-arithmetic section and input/output section with eight I/O channels. The computational processor consists only of a command-arithmetic section that is identical to the input/output processor.

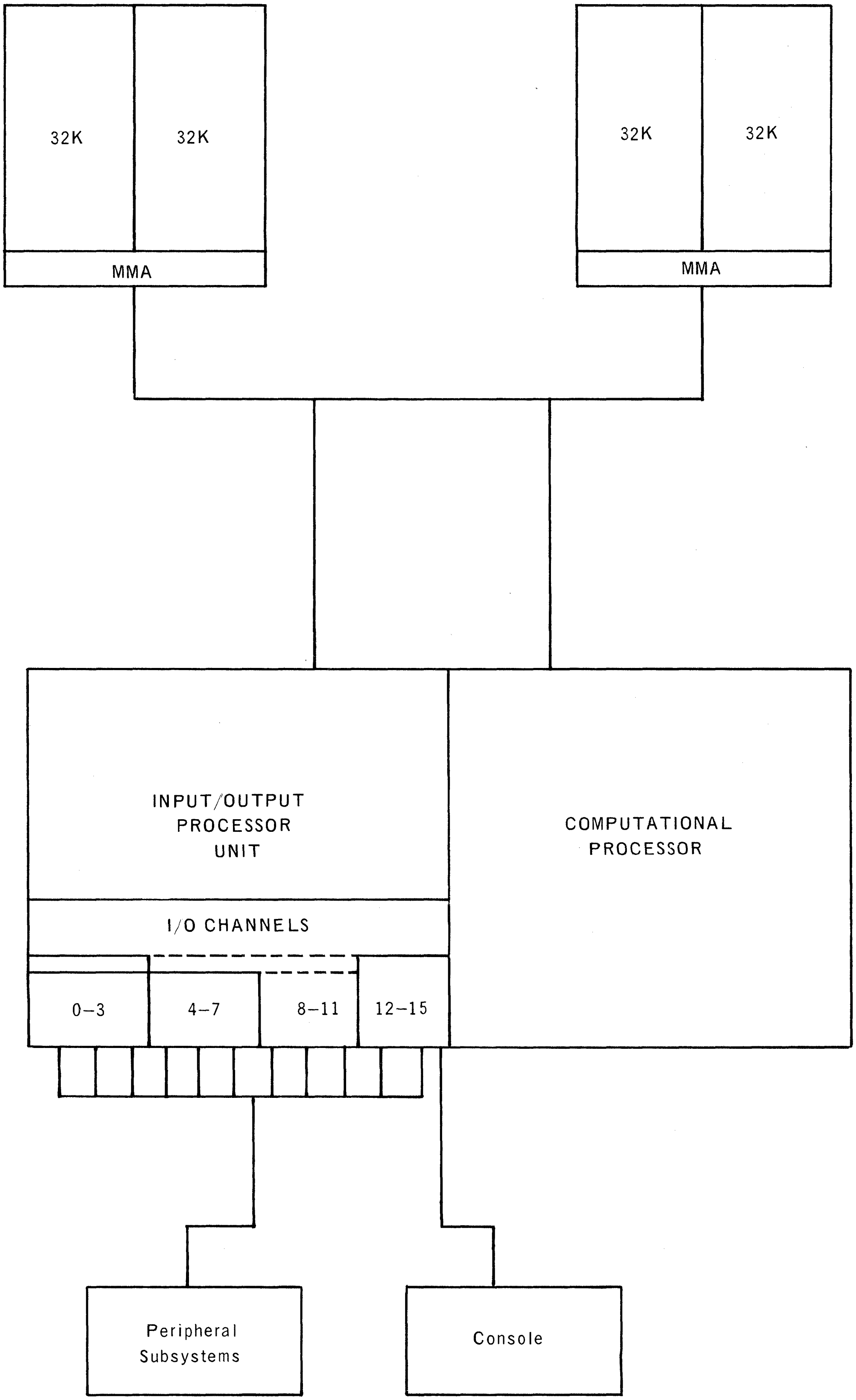


Figure 5–1. UNIVAC 1108 Shared Processing System

6. PROCESSOR INPUT/OUTPUT CONTROL SECTION

6.1. GENERAL DESCRIPTION

The input/output control section of the UNIVAC 1108 Central Processor Unit (CPU) controls transmission of data between main storage and the peripheral subsystems. It communicates with a peripheral subsystem over one of 16 bidirectional input/output channels. Data is transmitted with all 36 bits of a word in parallel; thus each channel has 72 data lines (36 input and 36 output) plus control signal lines. Although most peripheral subsystems use both input and output lines, data flows in only one direction on a channel for a specific I/O instruction.

In the shared processor system, the input/output processor, consisting of an input/output section as well as arithmetic and control sections, handles all data transfer within the system.

The I/O control section acts as a small processor to operate many peripheral subsystems concurrently. A programmed I/O instruction selects a specified channel and I/O device on the selected channel, and sets up the required conditions for a given activity. From that point on, the I/O control section automatically controls transmission of data to or from the subsystem at its natural speed. When a subsystem requests a word, the I/O control section refers to an access control word which specifies the location in main storage to or from which data is to be transferred.

6.2. PERIPHERAL CONTROL

Input and output can be performed in either of two modes: Externally Specified Index (ESI) for multiplexed data communications devices and Internally Specified Index (ISI) for other types of peripheral equipment. All I/O channels except channel 15, which is always assigned to the display console, can operate in either mode depending on the setting of a mode switch associated with each channel.

6.3. INTERNALLY SPECIFIED INDEX MODE

Each channel operates in one of three states: input, output, and function. Input and output are the data transmission states. The function state is actually an output state during which the processor sends one or more function words to the subsystem. Each function word specifies an operation to be performed by the subsystem.

The actual word-by-word transmission (regardless of transfer state) is governed by an access control word stored in an access control register. Two of these registers, one for input and one for output, are assigned to each I/O channel (see Figure 4–1, registers 40_8 to 77_8).

The format of the ISI access control word is as follows:

G	W WORD COUNT	V STARTING ADDRESS
35 34	33 18	17 0

V —18 bits, the starting or next address in main storage for data transfer.
W —16 bits, the number of words still to be transferred. It decreases by 1 each time a word is transferred.
G —2 bits, the incrementation control for V.
= 00, increment V by 1 after each word is transferred.
= 10, decrement V by 1 after each word is transferred.
= 01 or 11, do not change V.

In initiating an input/output operation, the processor stores an access control word in the input or output access control register associated with a given channel. Depending on the contents of G, the I/O control section transfers subsequent words to or from successive locations in main storage (increasing or decreasing addresses) or to or from a single location. After each transfer, the word count W is decreased by one and tested for zero. A nonzero calls for transfer of the next word in the block; a zero terminates the transfer; and if the instruction calls for monitoring, the input/output monitor interrupt is set.

6.4. EXTERNALLY SPECIFIED INDEX MODE

The Externally Specified Index (ESI), in conjunction with data communications equipment, allows multiplexed remote communication devices to communicate with main storage over a single I/O channel on a self-controlled basis without disturbing the main program. Each such remote device communicates with its own area of main storage.

Any I/O channel can be set to ESI mode by means of a switch. Furthermore, by means of a patch card, an ESI channel can be set to operate in either half-word (18-bit) or quarter-word (9-bit) mode.

Because any channel can be used by many devices in ESI mode, data flow must be governed by an access control word unique to the device currently in operation rather than to the channel as in ISI. These access control words are stored in main storage at addresses assigned to the devices. As a device transfers data, it presents the address of its own access control word; thus, no complicated program monitoring is necessary to control data flow.

The format for the ESI access control word differs somewhat from that of ISI to enable control of half- and quarter-word transfers. The half-word access control word is as follows:

G	H	W WORD COUNT	V STARTING ADDRESS
35 34	33	32 18	17 0

The G, W, and V fields have the same meaning as in the ISI access control word except that W is reduced to 15 bits and counts the characters to be transferred. There is also an H field of one bit; this field is used to indicate which half of location V is to be used, as follows:

SECOND HALFWORD H=1	FIRST HALFWORD H=0
35 18	17 0

where:

H = 0; use first halfword of location V and switch H to 1.

H = 1; use second halfword of location V; change V address as specified by G, and switch H to 0.

On input the first halfword of an incoming message causes the associated ESI access control word to be transferred from main storage to the I/O control section. Since the H bit is zero the data goes to the lower half of location V. V is not altered but H is set to 1 and w is decremented. The access control word is then returned to main storage until the next data from the same source arrives. At that time the access control word is again transferred from main storage. Since H now equals 1, the data goes to the upper half of location V, address V is changed as specified by G, W is decremented, H is set to 0, and the access control word is stored. A similar sequence applies for output transfers.

Quarter-word operations are similar except that additional programmed control is provided in terminating transmission. For this purpose the access control word includes two extra control bits.

G	H	C	W WORD COUNT	V STARTING ADDRESS
35 34	33 32	31 30	29 18	17 0

G, W, and V have the same meanings as for ISI although W is now only 12 bits long and now counts quarter words. H is the quarter-word designator designating the portion of V that is being addressed as follows:

FIRST QUARTER WORD	SECOND QUARTER WORD	THIRD QUARTER WORD	FOURTH QUARTER WORD
35 H=00 27	26 H=01 18	17 H=10 9	8 H=11 0

Notice that the data is stored in reverse of the order used in half-word operation.

C is a two-bit control field that prevents loss of data by generating an extra programmed monitor interrupt if required and a programmed end-of-transmission signal. The normal monitor interrupt occurs when W goes from 1 to 0. However, when bit 30 is set to 1, the subsystem sends a monitor interrupt to the processor as W decreases from 2 to 1. Similarly, if bit 31 is set to 1 the subsystem generates the end-of-transmission signal as W goes from 2 to 1.

6.5. BUFFER MODE DATA TRANSFERS

A buffer mode data transfer which occurs independently of main program control is used to transfer data between main storage and the communication subsystem. Before the transfer, the program performs the following steps:

(1) Loads the locations specified by the ESI addresses with access control words.

(2) Activates the channel to be used.

(3) Sends a function word to the communication subsystem. This step is not required to effect transfers from low or medium speed Communication Terminal Modules.

Step 2 is accomplished by one of the following four instructions. The access control word should specify a one-word dummy buffer since such a buffer is not normally used in the ESI mode.

LIC	Load Input Channel
LICM	Load Input Channel and Monitor
LOC	Load Output Channel
LOCM	Load Output Channel and Monitor

Step 3 is performed by a Load Function In Channel instruction. In ESI mode this instruction loads the access control word for the function into the output access control register for the channel and forces one external function transfer.

Data is then transferred in quarterwords between main storage and the subsystem without main program intervention. Each time a partial word is transferred to or from storage, 1 is automatically subtracted from the W count of the access control word. When this count becomes zero, the transfer is complete. If monitor is specified, an internal I/O monitor interrupt is set.

6.6. INPUT/OUTPUT INFORMATION WORDS

Four types of information words are transmitted between the processor and the peripheral subsystems. Each is accompanied by a control signal which identifies it for the receiving unit.

- Data words which go in either direction;
- Function words which go from processor to subsystem;
- Identifier words which go from processor to subsystem;
- Status words which go from subsystem to processor.

6.6.1. Data Words

Data is transmitted all bits in parallel, the number of bits depending on the mode of operation. That is, for ISI, 36-bit parallel; for ESI, 18-bit or 9-bit parallel.

To identify the word as data, the subsystem accompanies the word with an input data request signal. The I/O section acknowledges receipt by returning an input acknowledge signal. Similarly, on output the system requests data by means of an output data request. As soon as data is available, the I/O control section sends it and supplies an output acknowledgement to the subsystem.

6.6.2. Function Words

The 36-bit function word contains operating instructions for the peripheral subsystem. This includes a function code specifying what is to be done and a unit select code if the subsystem controls more than one peripheral device.

The I/O control section identifies the information as a function word by sending an External function signal after placing the word on the data lines.

6.6.3. Identifier Words

The identifier word is used as a search key for any of the search functions. When such a word is sent to a subsystem that is set to perform a search operation, an external function signal accompanying it identifies it as an identifier word. The subsystem stores it in a special register and compares it with each word read at the peripheral device until it finds an identical word. It then terminates the search and stores the location of the matching word in the status word for further use by the program. On a search/read function the subsystem starts reading as soon as the matching word is found.

6.6.4. Status Words

The 36-bit status word, generated by the subsystem, indicates whether an I/O instruction has been completed normally or not. Indicators within the word indicate any abnormal or error conditions. The CPU stores the word in its external interrupt status location for analysis and further action (see 4.6.2).

6.7. PRIORITY CONTROL

Input/output operations are arranged in sequence by a priority control network within the input/output section of the CPU. Although all sixteen I/O channels may be available for data transmissions between the processor and peripheral units at the same time, only one channel can communicate with the central processor at any given instant. Priority control circuits resolve situations in which two or more I/O channels simultaneously seek to communicate with the CPU. The following lists the priorities in descending sequence. If two or more requests have the same priority, priority is based on the I/O channel number (lower numbered channel first).

1. Output Data Request (ODR)
2. Input Data Request (IDR)
3. Real Time Clock Decrement
4. Power Loss Interrupt
5. I/O Parity Error Interrupt
6. External Interrupt (ESI)
7. Input Monitor Interrupt (ESI)
8. Output Monitor Interrupt (ESI)
9. Real Time Clock Interrupt
10. External Interrupt (ISI)
11. Input Monitor Interrupt (ISI)
12. Output Monitor Interrupt (ISI)
13. Function Monitor Interrupt
14. Interprocessor Interrupt #0
15. Interprocessor Interrupt #1

6.8. INPUT/OUTPUT INSTRUCTIONS

The I/O instructions allow the program to activate, test, deactivate, and control I/O operations.

The following six instructions prepare the I/O section of the processor to perform I/O operations on the specified channel. Their use and operation has been explained in 6.7.

Load Input Channel	75,00
Load Input Channel and Monitor	75,01
Load Output Channel	75,04
Load Output Channel and Monitor	75,05
Load Function in Channel	75,10
Load Function in Channel and Monitor	75,11

The following two instructions have no effect upon the operation of the I/O channels but test the specified channel to determine if it is active in the specified mode:

Jump Input Channel Busy	75,02
Jump Output Channel Busy	75,06

Similarly, the following instruction does not have any effect upon the operation of the specified channel but tests the specified output channel to determine if the first function word has been sent to the subsystem after the channel has been activated in the function mode:

Jump Function in Channel	75,12

The following two instructions enable and disable the servicing of external interrupts by the I/O Section:

Allow All Channel External Interrupts	75,14
Prevent All Channel External Interrupts	75,15

The following two instructions enable the program to terminate operations of an I/O channel. They are used principally to initiate operations on a channel that has not terminated properly.

Disconnect Input Channel	75,03
Disconnect Output Channel	75,07

6.8.1. Monitored Instructions

Input mode, output mode, or function mode can be activated on a channel either with or without monitor.

When ISI is in effect, at the end of a monitored instruction after the data has been transferred, the channel is deactivated and the I/O section sends a monitor interrupt to the CPU; this informs the Executive system that the transfer is complete.

The Load Function In Channel And Monitor (LFCM) instruction never results in a monitor interrupt when it specifies a channel conditioned for ESI operation.

7. THE INPUT/OUTPUT CONTROLLER

7.1. GENERAL DESCRIPTION

The Input/Output Controller (IOC) is an independent device that controls the operation of up to 16 peripheral subsystems under the direction of as many as three different CPU's. It is functionally similar to the I/O section of the CPU discussed in Section 6. Once an I/O request has been issued to it, the controller is in complete control of the operation, transferring data between main storage and the peripherals without further attention from the CPU that originated the request.

It provides the following enhancements to the multiprocessor system:

- Independent data paths between its peripheral subsystems and main storage
- Efficient, high-speed data communication capabilities
- Chained-buffer operation (scatter-read/gather-write)
- Additional I/O channels

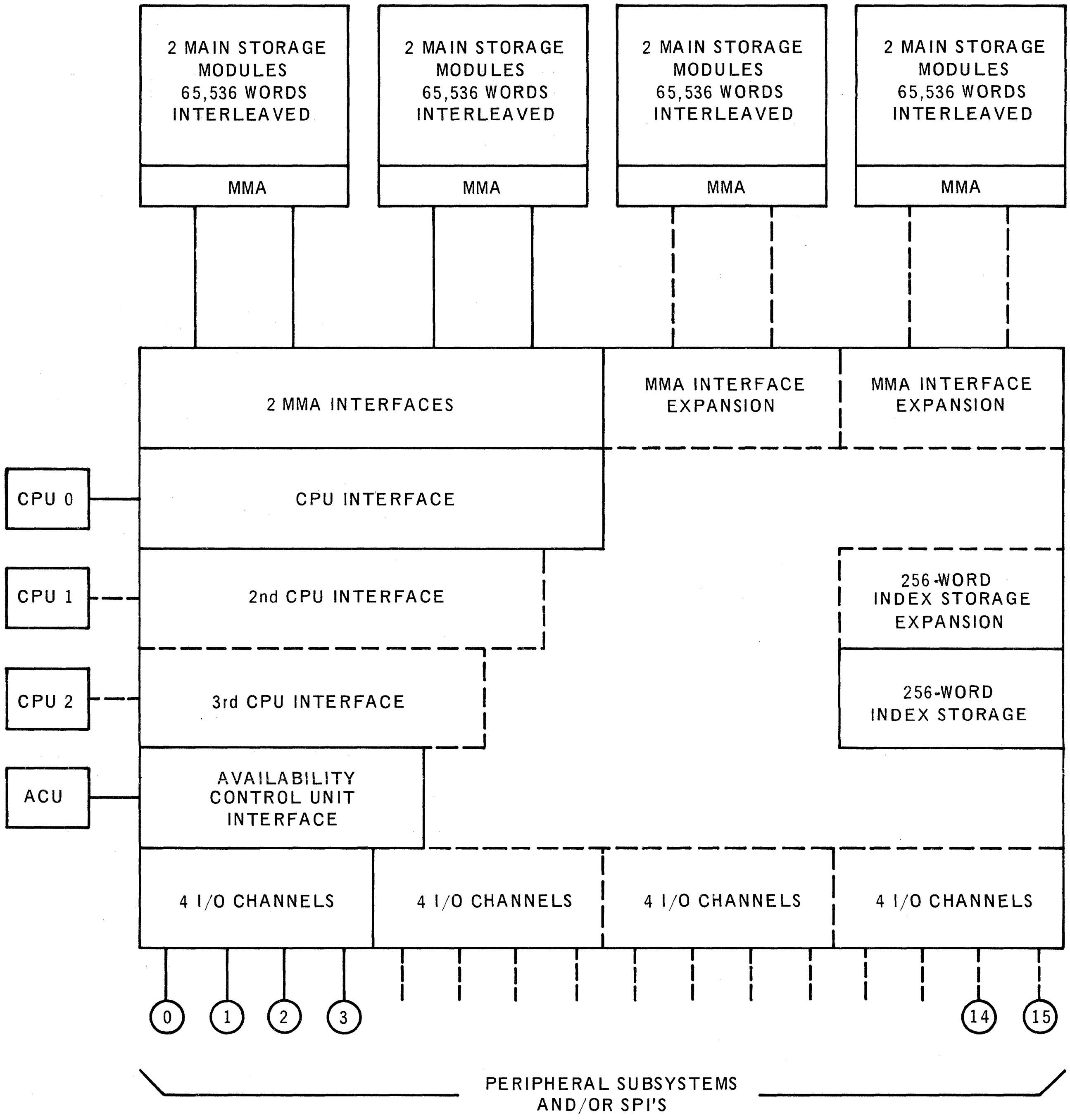
The multiprocessor system includes one or two Input/Output Controllers each of which controls its own group of peripheral subsystems (see Figure 6–1). Each provides a direct path between main storage and 4, 8, 12, or 16 high-speed bidirectional data channels. Each channel can transfer data at speeds up to 440,000 words per second. The IOC has an aggregate transfer rate of 1,333,000 words per second for all channels.

An I/O channel of the IOC performs all the functions that a processor I/O channel performs. Data transfers proceed by way of an IOC between a peripheral device and main storage, independent of the cyclic operation of the processor, thus allowing the processor more time for processing. No direct peripheral-to-peripheral (drum-to-printer) transfers are possible. All transfers must be buffered in main storage.

The IOC can interface with 1, 2, or 3 processors and has access to all of main storage by way of the Multi-Module Access units. However, rather than using the I/O access control registers of the processor, it has its own high speed index storage for access control.

7.2. POINTER REGISTERS

The addressing of the access control words in index storage is controlled by pointer registers. These 16 six-bit pointer registers are assigned one to each I/O channel of the IOC. The program loads a given pointer register with the index storage address of an access control word. Thus the fixed relationship is between the peripheral subsystem and the pointer register, rather than between the subsystem and certain access control registers in the index storage.


2 MAIN STORAGE MODULES 65,536 WORDS INTERLEAVED
2 MAIN STORAGE MODULES 65,536 WORDS INTERLEAVED
2 MAIN STORAGE MODULES 65,536 WORDS INTERLEAVED
2 MAIN STORAGE MODULES 65,536 WORDS INTERLEAVED
MMA
MMA
MMA
MMA
2 MMA INTERFACES
MMA INTERFACE EXPANSION
MMA INTERFACE EXPANSION
CPU 0
CPU INTERFACE
CPU 1
2nd CPU INTERFACE
256-WORD INDEX STORAGE EXPANSION
CPU 2
3rd CPU INTERFACE
256-WORD INDEX STORAGE
ACU
AVAILABILITY CONTROL UNIT INTERFACE
4 I/O CHANNELS
4 I/O CHANNELS
4 I/O CHANNELS
4 I/O CHANNELS
0
1
2
3
14
15
PERIPHERAL SUBSYSTEMS AND/OR SPI'S

Solid lines indicate basic unit.

Dotted lines indicate optional expansion units.

Figure 7–1. The Input/Output Controller

7.3. ESI AND ISI

Like the input/output section of the CPU, the IOC operates in ESI (quarterword or halfword) or ISI modes, the mode being a field-installed option for each I/O channel.

The ISI mode is used for any I/O channels that are connected to such peripheral subsystems as magnetic tapes, magnetic drums, printers, and punched card equipment. Function word and data transfers for such subsystems make use of pointer registers which reference the appropriate ISI access control word in the index storage.

The ESI mode is used for data transfers to and from communications subsystems by the Communication Terminal Module Controller (CTMC) (see Section 9). The CTMC multiplexes data transfers to and from many communication lines through a single I/O channel on a controlled basis without disturbing the program sequence of the processor. When operating in ESI mode, the data flow for each communication line terminal is governed by its own data access control word in index memory.

The index storage holds both ESI and ISI access control words. It permits increased throughput by providing data chaining capabilities (scatter-read/gather-write) for ISI subsystems. The first 64 words of index storage are reserved for ISI data access control words (both input and output) and external function access control words. These registers can serve the same purpose as the 16 input access control registers and the 16 output access control registers in the CPU.

The remaining 192 words in the basic IOC store the ESI data access control words for operation with CTMC subsystems. Expanding the index memory to 512 words provides an additional 256 ESI data access control words. These control words are the counterpart of the ESI data access control words held in main storage for use by the processor I/O section. Their use reduces the number of main storage references from three to one for each ESI data transfer.

7.4. COMMAND AND CONTROL WORDS

The Executive system schedules the program 3 to be run on the system and allocates main storage, processor, IOC, peripherals, and running time for each program depending on its needs and its priority. Therefore, if a program requires a particular peripheral, the Executive system specifies the IOC and channel requirement. It also specifies a CPU to transfer a command packet from the main storage to the IOC. Once the CPU has transferred this command packet, the IOC performs the transfer independently.

The IOC accepts the following command and control words:

IOC Command Words

External Function Access Control Words

Data Access Control Words

Data Chaining Control Words

Each of these is described briefly in the following paragraphs.

7.4.1. IOC Command Word

The command word specifies the type of transfer (processor to IOC, IOC to processor, or no transfer) and the function to be performed (activate input, output, or external function buffer). It also designates the IOC channel number and whether or not chaining is required.

ZEROS	F	N	P	M	C	D	K	A
35 30	29 27	26 23	22	21	20	19 18	17 9	8 0

where:

F specifies one of eight functions.

N specifies one of 16 I/O channels.

P specifies whether a pointer register is required.

M specifies whether monitor interrupt is required, for ISI channels.

C specifies whether chaining of access control words is required.

D specifies direction of data transfers or no transfer.

K specifies the number of control words to be transferred.

A specifies the address in index memory of the first access control word.

7.4.2. Data Chaining

Data chaining is the linking of a series of access control words to provide the IOC with the capabilities for scatter-read/gather-write operations. An ISI channel is placed in the chain mode when the F field of the IOC command word contains a function code of 1, 2, or 3, and the C field contains a 1. For ESI channels, since chaining is not required, the C field must always be 0. When the channel is in the chain mode, the pointer-register count is incremented by one each time the word count field of an access control word changes from one to zero. The IOC immediately reads the next index storage location specified by the pointer register. If the word count is not zero, it is the normal access control word for the next buffer area. A word count of zero indicates that this is an end-of-string word signalling the IOC that the end of that packet has been reached. This word directs the IOC either to send a terminate function code to the peripheral subsystem or to jump to a new packet of access control words in the index storage. In either case, a monitor interrupt may be specified. If the chain is to be broken, sending a terminate function must be specified. The end-of-string word has the following format:

ZEROS	WC	M	J	T	NOT USED	A
35	33 18	17	16	15	14 6	5 0

where:

WC Word Count; always zero to indicate end-of-string

M Monitor interrupt

J Initiates a jump to address A in index memory

T Terminates output operations (for magnetic drum subsystems)

A Address in index memory of the first word of a new packet of data access control words

7.4.3. External Function Access Control Word

This word is stored in the index memory. It specifies the location in main storage of the first function word. When the IOC receives a command packet, it transmits the function word to the peripheral subsystem on the I/O channel designated by the IOC command word. One or more function words may be transmitted to the subsystem. After completing the function word transfer, the IOC places the channel in input or output mode as specified by the external function access control word.

ZEROS	MON	OUT	IN	WC	A
35 24	23	22	21	20 18	17 0

where:

MON Places channel in monitor mode

OUT Places channel in output mode For ISI only

IN Places channel in input mode

WC Specifies the number of external function words to be sent to the subsystem

A Address of the first function word to be sent to the subsystem

7.4.4. Data Access Control Word

This word specifies a data buffer in main storage. The ESI and ISI formats are the same as those used in the processor input/output section described in Section 6.

When the last function word has been transferred to a peripheral subsystem, the first data access control word is read from index storage. If two or more data access control words are to be used following the external function access control word, it is necessary to specify chaining. If chaining has been specified and data transfers for the access control word are completed, the next sequential word is read from index storage. The chain is broken when an end-of-string word specifying the packet end has been reached.

8. PERIPHERAL SUBSYSTEMS

8.1. AVAILABLE PERIPHERAL EQUIPMENT

Peripheral subsystems are attached to the UNIVAC 1108 Central Processor Unit (CPU) or to the independent Input/Output Controller (IOC) through general purpose input/output channels, which have no restriction as to the manner in which peripheral subsystems may be attached. The governing factor for peripheral attachment is the transfer rate of the devices in the subsystem. Since the channels are numbered in order of priority, real-time equipment or equipment with very high transfer rates should be attached to the lower numbered channels which have the higher priority.

With this adaptable input/output arrangement, the UNIVAC 1108 System can communicate with many real time devices such as analog/digital converters, key sets, communication terminals, tracking and radar systems, display systems, and other information processing systems.

The UNIVAC 1108 peripheral subsystems are:

High Performance Drums

FH-432 Magnetic Drum Subsystem
FH-1782 Magnetic Drum Subsystem
FH-432/FH-1782 Magnetic Drum Subsystem
FH-880 Magnetic Drum Subsystem

Mass Storage

FASTRAND II Mass Storage Subsystem
FASTRAND III Mass Storage Subsystem
8414 Disc Subsystem

Magnetic Tape

UNISERVO VI-C Magnetic Tape Subsystem
UNISERVO VIII-C Magnetic Tape Subsystem
UNISERVO 12 Magnetic Tape Subsystem
UNISERVO 16 Magnetic Tape Subsystem
UNISERVO 12/16 Magnetic Tape Subsystem

Auxiliary Systems

Punch Card Subsystem (Reader/Punch)
High Speed Printer Subsystem
Unitized Channel Storage

Data Communication Systems (See Section 9)

Communications Terminal Module Controller Subsystem
Word Terminal Synchronous
Communications Terminal Synchronous
Data Communication Terminal – DCT 2000
Data Communication Terminal – DCT 1000
Data Communication Terminal – DCT 500
Data Communication Subsystem – DCS-1
UNISCOPE 300 Visual Communication Terminal
UNISCOPE 100 Visual Communication Terminal
UNIVAC 9300 System

In addition to the standard UNIVAC 1108 subsystems, UNIVAC 1107 subsystems function with the UNIVAC 1108 System as well. Included in this category are:

UNISERVO II-A Magnetic Tape Subsystem
UNISERVO III-A Magnetic Tape Subsystem
UNISERVO III-C Magnetic Tape Subsystem
UNISERVO IV-C Magnetic Tape Subsystem
UNIVAC 1004 Card Processor

8.2. THE FLYING HEAD DRUMS

The UNIVAC Flying Head (FH) series of high speed large-capacity magnetic drum storage units provide modular auxiliary storage essential for the operation of large and complex systems. These units vary from the ultra fast FH-432 (with an average access time of 4.3 milliseconds) to the large capacity (12.5 million alphanumeric characters) FH-1782 which provides extensive fast access storage that can be used for large data files that have to be referenced frequently.

FH magnetic drum subsystems have an individual read/write head for each track. Thus any word on an FH series drum is available to the system in an average half-revolution access time of 4.3 milliseconds (FH-432) or 17.0 milliseconds (FH-1782 and FH-880).

Each word in all FH subsystems is individually addressable so that the fullest use can be made of premium storage. This enables offline search operations in which the control unit matches the contents of any drum area, up to the capacity of the subsystem, with a designated identifier word. Upon finding a match it supplies the address of the match or commences reading and transferring data to main storage. This entire process is carried out offline without any processor 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.

A new function has been incorporated in the control logic of the FH-432/1782 subsystem to predict and reduce storage access time. This function enables the program to request the current angular position of the drum under the read/write heads on a particular drum unit. The input/output handler can then select from the subsystem queue the data request that can be serviced fastest. It is possible to sequence drum requests and to make multiple accesses to a drum unit during a single revolution, instead of having to wait an average of half a revolution for each request.

The transfer rate of data to and from the FH drum subsystem 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 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.

Through the addition of Shared Peripheral Interfaces (SPI), a single- or dual-channel FH drum subsystem may be accessed by multiple processors. This not only provides a safeguard in case of failure but also enables all processors to access all drum units as common storage, so that all CPU's in a multiprocessor system can share a single major task, or tasks can be allocated to individual CPU's but with data storage being shared.

The FH-432/1782 drum subsystems may operate with either one or two control units, using one or two input/output channels. Availability of two channels 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; therefore, in case of a failure of one of the power supplies, the subsystem can still operate on a single-channel basis.

The UNIVAC 1108 Operating System is planned to use auxiliary drum storage instead of magnetic tapes as much as possible. This reduces manual handling and access and transfer time when compiling and assembling, and during basic batch input/output operations.

These FH drum subsystems have many advantages in standard data processing as well as real-time operation. This is especially true in applications where rapid file processing and sort/merge routines are more prevalent.

Large capacity with rapid access affords convenient intermediate storage. Instead of multiple 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 short access time of the FH-432 drum permits lesser-used control segments to be stored outside of main storage. They can then be read into a common overlay area only when required. This arrangement greatly reduces the amount of main storage required for the Executive system.

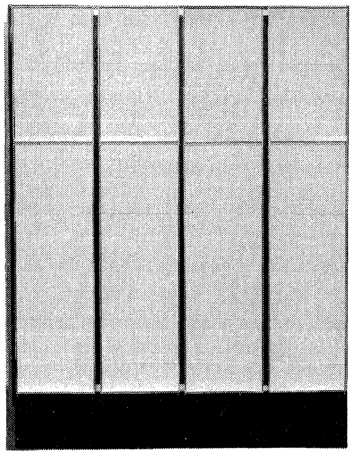
8.2.1. FH-432/FH-1782 Magnetic Drum Subsystems

A valuable characteristic of UNIVAC 1108 drum subsystems is the ability to associate, in the same subsystem, the ultrahigh 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 1108 storage configuration. An efficient blend can be made of high speed storage for rapidly required software, program segments, tables, and indices and greater access time but large capacity storage for less frequently used program segments, data files, and message assembly/disassembly areas. A judicious mix of speed, capacity, and economy can be planned and the mix can readily be altered as requirements change. Character transfer rates are identical for the FH-432 and FH-1782 drum units. The only functional difference in a data transfer is the variation in access time.

This subsystem is available in both single- and dual-channel versions to provide a hierarchy of auxiliary storage for both unit- and multi-processors. The dual-channel version includes two electrically and logically independent control units each on a different I/O channel. This enables simultaneous operation of any two drums in the subsystem and provides the hardware redundancy required for multi-processing.

8.2.2. FH-432 Magnetic Drum Subsystem



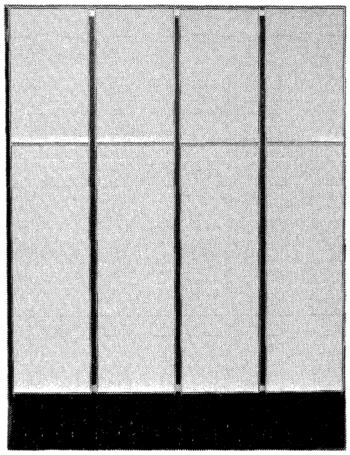
The FH-432 Magnetic Drum Subsystem is designed for single-channel operation and is suited primarily to unit processor configurations. A minimum Flying Head 432 (FH-432) Magnetic Drum Subsystem includes three drums (786,432 36-bit words of storage), a control unit, and power supplies, contained in two cabinets. To augment the systems, cabinets may be added, each containing one or two drums with a storage capacity of 262,144 36-bit words per drum. Of the 432 tracks on each drum, 384 are used for data; the remaining tracks are used for spares, parity, and timing functions. There are 2048 words of data per 3 tracks. Reading and writing are 3-bit parallel operations on all three tracks of a band simultaneously. Thus the maximum transfer rate is 240,000 words or 1,440,000 alphanumeric characters per second.

Up to eight FH-432 Magnetic Drums may be accommodated in a single subsystem, affording a maximum subsystem capacity of 2,097,152 words or 12,582,912 alphanumeric characters.

FH-432 units may be intermixed with FH-1782 units in the same subsystem to provide a powerful blend of ultrahigh speed and large capacity storage. This mixed subsystem is described in 7.2.3.

CHARACTERISTICS	
STORAGE CAPACITY	262,144 computer words of 36 data bits plus parity bits, or 1,572,864 alphanumeric characters per drum
AVERAGE ACCESS TIME	4.3 milliseconds
DRUM SPEED	7,200 revolutions per minute
NUMBER OF READ/WRITE HEADS	432 – one per track
CHARACTER TRANSFER RATES	1,440,000, 720,000, 360,000, 180,000, 90,000
WORD TRANSFER RATES	240,000, 120,000, 60,000, 30,000, 15,000
I/O CHANNELS REQUIRED	1 per subsystem
NUMBER OF DRUMS PER SUBSYSTEM	3 to 8 (12,582,912 characters maximum)

8.2.3. FH-1782 Magnetic Drum Subsystem



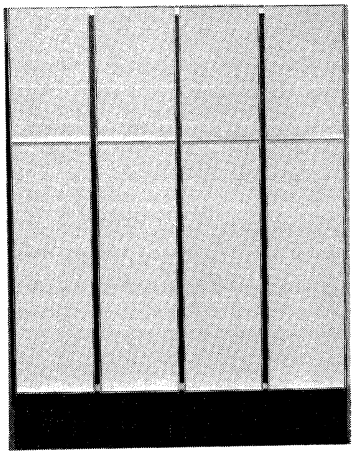
The Flying Head 1782 (FH-1782) Magnetic Drum is identical to the FH-880 drum except that the storage capacity is 2-2/3 times greater; this increase is achieved partly by an increase in the number of data tracks to 1536 and partly by an increase in the recording density. Each track has its own read/write head, and average access time is unchanged at 17 milliseconds.

A single FH-1782 drum stores 2,097,152 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.

The increased recording density results in a character transfer rate equal to that of the FH-432 drum; this arrangement enables FH-1782 drums to be associated with FH-432 drums in the same subsystem as described in 8.2.3.

CHARACTERISTICS	
STORAGE CAPACITY	2,097,152 computer words of 36 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 RATES	1,440,000; 720,000; 360,000; 180,000; 90,000
WORD TRANSFER RATES	240,000; 120,000; 60,000; 30,000; 15,000
I/O CHANNELS REQUIRED	1 or 2 per subsystem
NUMBER OF DRUMS PER SUBSYSTEM (MAX.)	8 (total of 100,663,296 characters.)

8.2.4. FH-880 Magnetic Drum Subsystem



The FH-880 Magnetic Drum Subsystem has three times the capacity of the FH-432 drum; however, its access time is four times greater than the FH-432 drum.

A single FH-880 drum stores 786,432 words, equivalent to 4,718,592 alphanumeric characters. Up to eight FH-880 drums can be accommodated in a single subsystem giving a subsystem capacity of 37,748,736 characters.

Of the 880 tracks on the drum, 768 are used for storing data, 32 for parity, and the remainder for spares and timing purposes. There is one read/write head per track. The 768 tracks of data storage are organized into 128 bands of six tracks each, with each bank having a capacity of 6144 words. Reading and writing are performed in 6-bit parallel mode on all six tracks of a band simultaneously at a maximum transfer rate of 60,000 words or 360,000 alphanumeric characters per second.

CHARACTERISTICS	
STORAGE CAPACITY	786,432 36-bit words (4,718,592 alphanumeric characters) per drum
AVERAGE ACCESS TIME	17 milliseconds
DRUM SPEED	1770 revolutions per second
NUMBER OF READ/WRITE HEADS	880 (one per track)
WORD TRANSFER RATE	60,000 words per second maximum
CHARACTER TRANSFER RATE	360,000 characters per second maximum
I/O CHANNELS REQUIRED	1
NUMBER OF DRUMS PER SUBSYSTEM	1 to 8 (total of 37,748,736 characters maximum)

8.3. MASS STORAGE SUBSYSTEMS

Two types of mass storage are available; drum (see 8.3.1) and disc (see 8.3.2).

8.3.1. FASTRAND Mass Storage Subsystems

The FASTRAND Mass Storage Subsystems (FASTRAND II and III) provide very large capacity random access storage. Great flexibility is provided by the availability of both a single- and a dual-channel subsystem and by the Fastband option, available on a unit basis. Each Fastband track has a permanently assigned read/write head, as distinguished from the ordinary track, which shares a head with a number of other tracks. Access to Fastband data is faster because there is no need for head positioning. FASTRAND units include two large magnetic drums, which, like those used in the FH-432 and FH-1782 subsystems, employ flying heads. However, to reduce cost, only a limited number of read/write heads are used. These move laterally over 192 recording tracks.

There are 64 read/write heads per unit, gang-mounted on 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 over all tracks is 86 milliseconds, and the minimum is 30 milliseconds. Average latency is half a drum revolution time (35 milliseconds).

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

An independent position control feature allows greater flexibility and decreases average access time. This is done in a multi-unit subsystem by concurrently prepositioning the heads in a number of drum units. Prepositioning the heads saves time, because once the position instructions have been transmitted to one FASTRAND unit the Executive system can immediately initiate another operation on a different FASTRAND unit without waiting for completion of the positioning operation. Whenever the system reads from or writes on such a prepositioned unit, the only time delay is for the address circuit activation and latency. These are the mechanical design features which contribute to the FASTRAND's operating speed.

In any effort to reduce processing time, offline search is an important advantage. In this operation the computer instructs the FASTRAND subsystem to locate a specific piece of data, and then goes on with other processing while the storage search takes place. When the subsystem finds the data, it notifies the computer and sends it the data. Also, all other functions of the FASTRAND unit permit the computer to continue its work while records are being read from or written on the drums.

All data on a FASTRAND subsystem is recorded in 28-word groups known as sectors. Parity is recorded by sector and the parity bits are automatically checked.

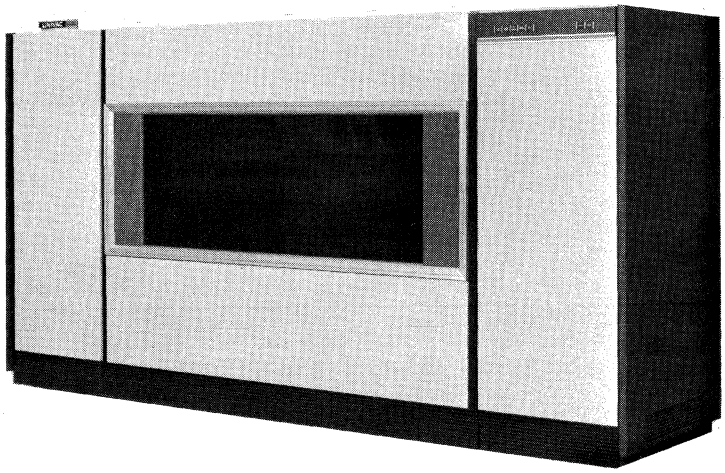
A single FASTRAND subsystem can accommodate up to eight FASTRAND units. Any of the input/output channels of a CPU or IOC can accommodate a FASTRAND subsystem. Like the FH-432/1782 drum subsystem, either single- or dual-channel operation is possible, providing full-scale two-drum-unit simultaneity.

Two different dual-channel FASTRAND subsystems are available. They differ in the amount of hardware redundancy in the control units and therefore in their capability to perform simultaneous operations. The dual-access FASTRAND subsystem includes two complete control units and therefore does not require the two-bank organization of FASTRAND storage units. The control units, being logically and electrically independent, provide the parallel data paths that permit simultaneous operations on any two FASTRAND units in the subsystem.

An optional feature called Fastband includes 24 additional tracks with fixed read/write heads. This provides rapid access (35 ms. average). The write lockout feature is also available 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 starting with the first track of each head.

Physically, the FASTRAND III mass storage units are similar to the FASTRAND II mass storage units. The basic difference is in their recording densities (over 1500 bits per inch for FASTRAND III mass storage units versus 1000 bits per inch for FASTRAND II mass storage units). Since the recording density is 50 percent greater, the storage capacity and word and character transfer rates of the FASTRAND III mass storage units are 50 percent greater than those of the FASTRAND II mass storage units.

FASTRAND II AND III MASS STORAGE UNITS



CHARACTERISTICS

STORAGE CAPACITY (PER UNIT)	FASTRAND II: 22,020,096 36-bit words (132,120,576 alphanumeric characters) FASTRAND III: 33,030,144 36-bit words (198,180,864 alphanumeric characters)
AVERAGE ACCESS TIME	92 milliseconds
RECORDING DENSITY	FASTRAND II: 1,000 bits per inch FASTRAND III: 1,500 bits per inch
TRACKS PER INCH	106
DRUM SPEED	880 revolutions per minute
MOVEABLE READ/WRITE HEADS	64
CHARACTER TRANSFER RATE	FASTRAND II: 157,696 characters per second FASTRAND III: 236,544 characters per second
WORD TRANSFER RATE	FASTRAND II: 26,283 words per second FASTRAND III: 39,424 words per second
**FASTBANDS (FIXED READ/ WRITE HEADS)	24
FASTBAND AVERAGE ACCESS TIME	35 milliseconds
FASTBAND STORAGE CAPACITY (PER UNIT)	FASTRAND II: 43,008 36-bit words (258,048 characters) FASTRAND III: 64,512 36-bit words (387,072 characters)
**WRITE LOCKOUT PROTECTION	Yes
I/O CHANNELS	1 or 2 per subsystem
NO. OF UNITS PER SUBSYSTEM	8

***Optional*

NOTE: Addition of Fastband increases capacity by 258,048 characters per unit (2,064,384 if all 8 units include Fastbands).

8.3.2. UNIVAC 8414 Disc Subsystem

The 8414 Disc Subsystem offers the vast storage capacity of eight disc drives, expandable in disc drive modules. The advantages of the UNIVAC 8414 Disc Subsystem are as follows:

- substantially increases throughout performance;
- provides for incremental growth;
- has expanded potential for online processing; and
- enhances capabilities for real time and multiprogramming.

The 8414 Disc Subsystem provides the 1108 System with an expandable, random access, external storage medium. The basic subsystem consists of one control unit, one Multi-Subsystem Adapter (MSA), and from two to eight 8414 disc drives.

Data is transferred between the processor and subsystem a word at a time. Data is stored on and retrieved from the disc drive serially. The MSA translates the byte-oriented code of the 8414 disc into the 36-bit word format of the 1108 System.

The UNIVAC 8414 Disc Subsystem offers many processing advantages, especially in applications where rapid file processing and sort/merge routines are prevalent. Information is stored on widely used removable disc packs which can hold over 6.4 million 36-bit words of information. A maximum 8414 subsystem can store up to 51.8 million, 36-bit words on data online. The removability characteristics of the disc pack permits virtually unlimited offline storage and easy interchange of information without conversion to other media.

Each disc pack contains 11 discs with the data records on the inside surfaces. Twenty read/write heads are mounted on a single accessor mechanism which moves the 20 heads in unison between the periphery and the central area of the disc. The accessor mechanism can assume one of 203 tracks across the disc surface. This simultaneous head movement creates 200 addressable data recording cylinders in the disc pack, with three cylinders reserved as alternate tracks. Each cylinder contains twenty tracks, number 0 through 19. The addressing of an individual track in the pack is by track number (000-202) and by read/write head number (0-19).

Access to different tracks within a cylinder is faster than access to tracks in different cylinders since changing tracks requires only electronic switching, whereas accessing a different cylinder requires physical movement of the accessor mechanism. There are 4060 (203 x 20) tracks in a disc pack assembly. Data capacity figures are based on 4000 tracks, thus allowing for 60 spare tracks in a disc pack assembly.

The 8414 Disc Subsystem provides for dual access and prepositioning of access arms. This includes:

- Simultaneous read/read, read/write, write/write concurrent with positioning functions.
- Alternate data and command paths available to any component of the system.
- Optimized selection of I/O service request for a queue.

CHARACTERISTICS

Number of drives per subsystem	2–8
Number of disc packs per drive	1
Number of R/W head accessor mechanism	1
Number of R/W heads per disc pack	20
Number of tracks per disc surface	203
Number of recording surfaces per disc pack	20
Number of addressable tracks per surface	203
Number of addressable tracks per disc pack	4060
Number of sectors per track	44*
Capacity 36-bit words per disc pack	4.9 million*
Minimum access time	20 ms
Average	60 ms
Maximum access time	130 ms
Disc pack speed	2400 rpm
Data transfer rate	2.5M bits/sec.

*Using simulated FASTRAND

8.4. UNISERVO MAGNETIC TAPE SUBSYSTEMS

Four magnetic tape subsystems are available with the UNIVAC 1108 System. They are:

UNISERVO VI-C Magnetic Tape Subsystem
UNISERVO VIII-C Magnetic Tape Subsystem
UNISERVO 12 Magnetic Tape Subsystem
UNISERVO 16 Magnetic Tape Subsystem

Three basic methods of operation are available:

- Single-Channel Operation – In this method of operation, one or more tape units are connected to a single I/O channel through the appropriate control units. Only one function, on any one of the tape units, may be active at any single instant.
- Dual-Channel Operation – In this method of operation, two or more tape units are connected to two I/O channels through the appropriate control unit. Since two I/O channels are being used, simultaneous read/read or read/write on any two tape units is possible.
- Simultaneous Operation – In this method of operation, two or more tape units are connected to two I/O channels through the appropriate control unit just as in dual-channel operation. Additional circuits in the control unit and the tape units, however, permit simultaneous read/read, read/write, write/read, and write/write on any two tape units.

The following table indicates the availability of the various methods of operation for the tape subsystems:

	Single Channel	Dual Channel	Simultaneous
UNISERVO VI-C Subsystem	X	X	
UNISERVO VIII-C Subsystem	X		X
UNISERVO 12 Subsystem	X	X	X
UNISERVO 16 Subsystem	X	X	X

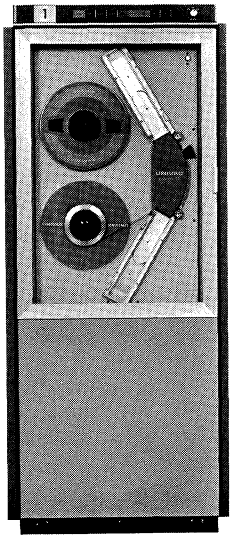
Magnetic tape subsystems may consist of up to 16 tape units with the appropriate control units. Systems are available for both 7- and 9-track operation. The 7- and 9-track options permit data recorded in the traditional industry compatible form to be handled, and yet at the same time allows the upgrading of these records in line with the ASCII code and packed-decimal formats.

Those systems employing either UNISERVO VIII-C or 16 control units may be converted to mixed systems; that is, systems using a mixture of UNISERVO VI-C/VIII-C tape units or a mixture of UNISERVO 12/16 tape units. This feature is available only for 7-track operation. Such a mixed system provides a useful flexibility by combining high speed units with economic medium speed units.

Since the UNISERVO 12 and 16 subsystems are byte oriented, these tape units must be connected to the UNIVAC 1108 System through the Multi-Subsystem Adapter (MSA) The MSA translates the byte output of the UNISERVO 12 and 16 tape units into a 36-bit format suitable for use in the 1108 System. On output, the MSA performs the word-to-byte translation.

A Shared Peripheral Interface (SPI) may be attached to the UNISERVO VI-C/VIII-C subsystem to permit access to one to four processors (CPU's or IOC's). While the SPI feature is incorporated in the MSA, the MSA may be connected only to CPU I/O channels.

8.4.1. UNISERVO VI-C Magnetic Tape Subsystem



The UNISERVO VI-C Magnetic Tape Unit is a low cost unit having moderate speed and transfer rates for applications involving massive file passing, extensive sorting, or other applications for which high speed magnetic tape subsystems would be desirable.

A UNISERVO VI-C subsystem can have up to 16 magnetic tape units connected to one or two input/output channels. Dual-channel operations permit a reading operation and a reading or writing operation to be performed simultaneously on any two magnetic tape units while all other tape units are rewinding.

Backward read capability is standard on all units.

The master/slave concept is employed in the logic of the UNISERVO VI-C subsystem; circuitry has been built into one of the UNISERVO VI-C units which will allow 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 either 200, 556, or 800 characters per inch, as selected. The tape speed is 42.7 inches per second, giving maximum transfer rates of 8,540, 23,741 and 34,160 alphanumeric characters per second, respectively.

The 800 characters per inch density is normally used, the 200 and 556 densities being used only for compatibility purposes. At 800 characters per inch more than 11,520,000 characters may be stored on a single reel in 600-character blocks.

Rewind takes place at 160 inches per second, enabling a full reel of 2400 feet to be rewound in 180 seconds.

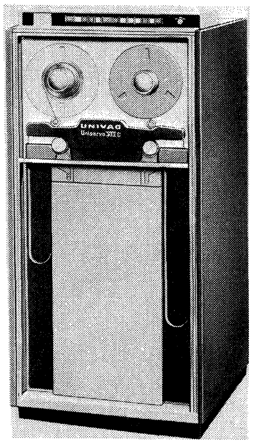
Data may be recorded in variable-length blocks under program control with character and block (horizontal and vertical) parity. A read-after-write head allows immediate verification of all data written, and under the control of the Executive input/output handler, repeated read and write operations are undertaken whenever read or write errors occur.

UNISERVO VI-C tape units are fully compatible with IBM 727,729 Models 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 furnished with a hardware translator to convert between tape code and Fieldata code, thus ensuring tape compatibility among installations.

CHARACTERISTICS

TRANSFER RATE	8,500, 23,700, and 34,200 alphanumeric characters per second.
RECORDING DENSITY	200, 556 and 800 6-bit characters 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 BLOCK	0.75 inch (7 track) 0.60 inch (9 track)
TRACKS ON TAPE	7 tracks: 6 data, 1 parity; optional, 9 tracks: 8 data, 1 parity
MAXIMUM NUMBER OF UNITS IN SUBSYSTEM	16
STANDARD FEATURE	Backward read
PROCESSOR INPUT/OUTPUT CHANNELS	1 or 2

8.4.2. UNISERVO VIII-C Magnetic Tape Subsystem



A UNISERVO VIII-C Subsystem can have up to 16 magnetic tape units, and can incorporate either one or two control units attached to one or two input/output channels for single or dual channel operator.

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 six-bit alphanumeric character or a six-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.

Data packing density in seven-track mode is set either by the program or by a manual switch on each unit to either 200, 556, or 800 frames per inch. Physical tape speed is 120 inches per second giving maximum transfer rate of 24,000, 66,720, and 96,000 alphanumeric characters per second.

An even higher transfer rate results if six-bit characters are read or written in nine-track mode. This method of operation yields a transfer rate of 128,000 characters per second.

The rewinding rate is 240 inches per second; a full reel of 2,400 feet can be rewound in 120 seconds. The 800-frame-per-inch packing density is normally used; the 200 and 556 densities are being used only for seven-track mode compatibility purposes.

Reading may take place with the tape moving either forward or backward, an ability valuable for saving rewind time especially during sort/merge operations. Writing takes place when the tape is moving forward only.

Data may be recorded in variable-length blocks under program control with character and block (horizontal and vertical) parity. A read-after-write head allows immediate verification of all data written. Under the control of the software Input/Output Handler, repeated read and write operations are undertaken in an attempt to recover from an error.

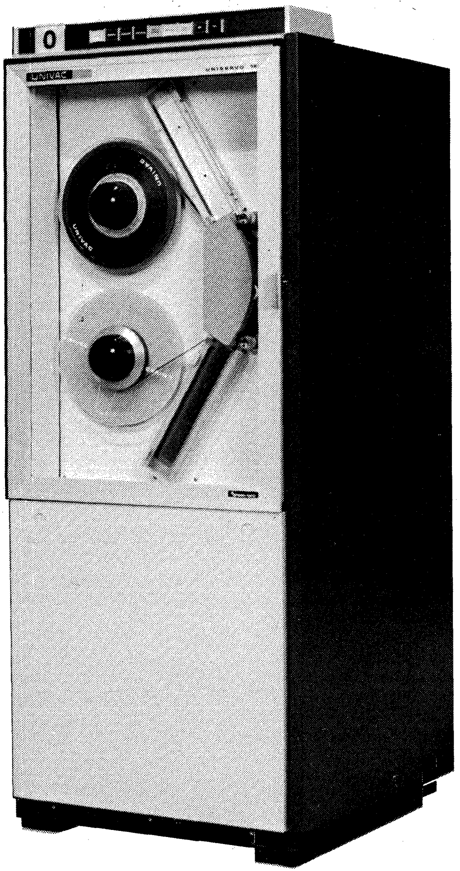
Programming problems with this tape subsystem are insignificant since the control unit, combined with the Executive input/output handler, deals with all operations except the system response to a nonrecoverable error.

UNISERVO VIII-C tape units are fully compatible with IBM 727, 729 Models 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 furnished with a hardware translator to convert between tape code and Fieldata code, thus ensuring tape compatibility among installations.

CHARACTERISTICS

TRANSFER RATE	24,000, 66,720, and 96,000 alphanumeric characters per second
RECORDING DENSITY	200, 556, and 800 6-bit characters per inch
TAPE SPEED	120 inches per second
TAPE WIDTH	0.5 inch
TAPE LENGTH	2,400 feet
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
STANDARD FEATURE	Backward Read
PROCESSOR INPUT/OUTPUT CHANNELS	1 or 2

8.4.3. UNISERVO 12 Magnetic Tape Subsystem



The UNISERVO 12 Magnetic Tape Subsystem is a low-cost medium performance subsystem with 7- or 9-track, 1600 frames per inch, and phase encoding as the available tape formats. One master tape unit, with a power supply and control circuits, controls up to three slave units. Up to 12 tape units are synchronously controlled by a UNISERVO 12/16 control unit.

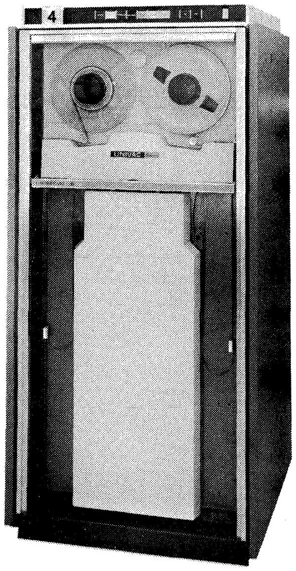
This subsystem offers a peak transfer rate of 68,000 frames per second in the primary tape format. The 7- or 9-track NRZ formats with a peak transfer rate of 34,000 frames per second also can be incorporated. Tape data validity checking facilities include read check while writing, longitudinal redundancy check, vertical parity check (9-track phase tapes) and cyclic redundancy check (diagonal, 9-track NRZ tapes).

"On the fly" single track read error correction is standard for phase tapes. On 9-track NRZ tapes, single track read error correction is provided by a second attempt at an operation after error detection and repositioning. This provides the ability to correct tape errors in either the forward or backward direction. This simplifies the error correction programming routines and assists in the recovery of unusual error conditions which otherwise would result in a nonrecoverable error. A programmable low gain read assists in the reading of tape records containing high noise levels.

A powerful simultaneity feature allows the usage of read/read, read/write, write/read, and write/write functions.

CHARACTERISTICS	
TRANSFER RATE	34 kiloframes/second
TAPE SPEED	42.7 inches/second
TAPE DIRECTION READING WRITING	 Forward or backward Forward
TAPE WIDTH	0.5 inch
TAPE LENGTH (MAX.)	2,400 feet (plastic)
THICKNESS	1.5 mils
BLOCK LENGTH	Variable
INTERBLOCK GAP	0.75 inch (7-track) 0.6 inch (9-track)
INTERBLOCK GAP TIME (7-TRACK)	17.6 milliseconds (nonstop) 23.6 milliseconds (start/stop)
INTERBLOCK GAP TIME (9-TRACK)	14.1 milliseconds (nonstop) 20.1 milliseconds (start/stop)
REVERSAL TIME	25 milliseconds
REWIND TIME	3 minutes (2,400 feet)
DUAL DENSITY	Feature available
SIMULTANEOUS OPERATION	Feature available

8.4.4. UNISERVO 16 Magnetic Tape Subsystem



The UNISERVO 16 Magnetic Tape Subsystem is similar to the UNISERVO 12 Magnetic Tape Subsystem except that it has higher performance capabilities. The peak transfer rate is 192,000 frames per second in the primary tape format (9-track, 1600 frames per inch, phase encoding) and 96,000 frames per second in the NRZ format.

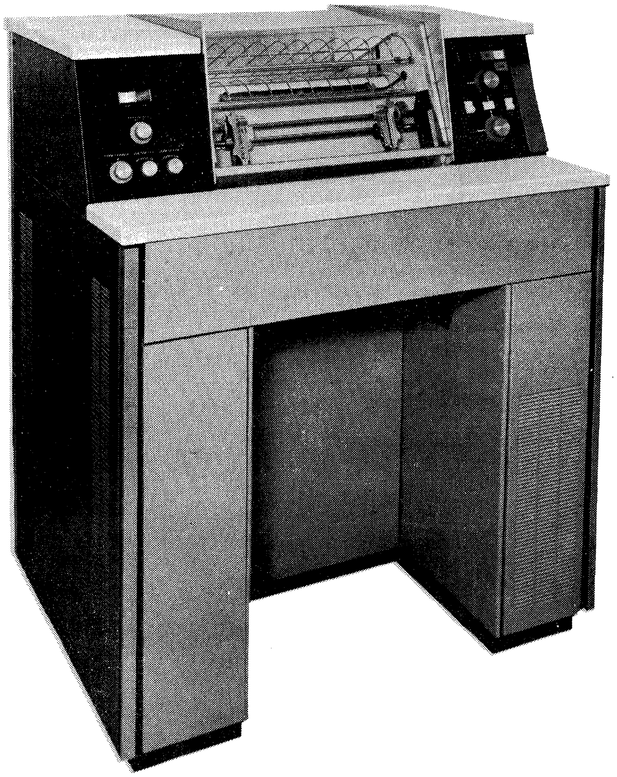
The UNISERVO 16 dual access, simultaneous feature is more powerful in that it provides write/write operation on an individual handler basis in addition to read/read and read/write. Furthermore, it provides complete subsystem redundancy by the addition of individual power supplies for each control unit and independent access paths to each tape unit. Data validity checking facilities include read check while writing, longitudinal redundancy check and a cyclic redundancy check (diagonal) on 9-track NRZ tapes and vertical parity check on 9-track phase tapes. The UNISERVO 16 is a free-standing tape handler, that is, the a master/slave approach is not used.

The basic tape control unit may be extended for the utilization of UNISERVO 16 tape units by the addition of a UNISERVO 16 feature in the control unit. Intermixing of UNISERVO 12 and UNISERVO 16 magnetic tape handlers on the same subsystem is possible with the addition of this feature.

CHARACTERISTICS

TRANSFER RATE	96 kiloframes/second
TAPE SPEED	120 inches/second
TAPE DIRECTION READING WRITING	 Forward or backward Forward
TAPE WIDTH	0.5 inch
TAPE LENGTH (MAX.)	2,400 feet (plastic)
THICKNESS	1.5 mils
BLOCK LENGTH	Variable
INTERBLOCK GAP	0.75 inch (7-track) 0.6 inch (9-track)
INTERBLOCK GAP TIME (7-TRACK)	6.25 milliseconds (nonstop) 9.25 milliseconds (start/stop)
INTERBLOCK GAP TIME (9-TRACK)	5.0 milliseconds (nonstop) 8.0 milliseconds (start/stop)
REVERSAL TIME	10 milliseconds
REWIND TIME	2 minutes (2,400 feet)
DUAL DENSITY	Feature available
DUAL ACCESS, SIMULTANEOUS OPERATION	Feature available

8.5. UNIVAC HIGH SPEED PRINTER SUBSYSTEM



The UNIVAC High Speed Printer Subsystem provides the UNIVAC 1108 System with an output printing unit that is capable of printing single or multiple copies of data. Each line of output data may contain up to 132 printed characters. Printing operations occur on a request acknowledge basis allowing the processor to perform other processing functions while the printer is printing data.

This subsystem is a High Speed Printer Control Unit connected to a High Speed Printer capable of printing from 1200 lines per minute for a full 63-character set to 1600 lines per minute for a 43 sequential character set. The printer contains 63 printable characters: the 26 letters of the alphabet, the 10 arabic numerals, and 27 special characters. Different symbols may be factory supplied upon order.

CHARACTERISTICS

Printing Speed (with single-line spacing)	1200/1600 lines per minute maximum, depending upon the number of sequential characters to be printed on each line
Line Spacing Speed	11.5 ms for spacing first line and for spacing each subsequent line as follows: 5.06 ms at 6 lines per inch 5.7 ms at 8 lines per inch
Characters Per Line	132 characters (including spaces) per line
Spacing of Characters	0.1 inch along print line
Ribbon Feed	Bidirectional, self-reversing, self-correcting
Type of Ribbon	Fabric ribbon interchangeable with carbon MYLAR* ribbon (optional) for "one-time" operation
Vertical Line Spacing	Manually selected. Either 6 lines per inch or 8 lines per inch
Number of Characters	Up to 63 different characters: standard font consists of alphabetic characters A–Z, numeric characters 0–9, 27 punctuation marks and symbols. Modified fonts available upon request
Print Format	Full print width of 132 characters can be placed anywhere on 16.5 inch form. With 22 inch width form, only central 13.2 inch portion can be used. Format variation under full control of programming
Paper Forms	Continuous forms with standard edge sprocket holes from 4 to 22 inches in width. Carbons may be attached or unattached with multicopy forms up to a maximum of six parts. Recommended pack thickness up to .0155 inch for optimum print quality
Paper Container	Maximum dimensions accommodated entirely within base of machine: 16 inches high, 16 inches long, and 22.5 inches deep

*DuPont trademark

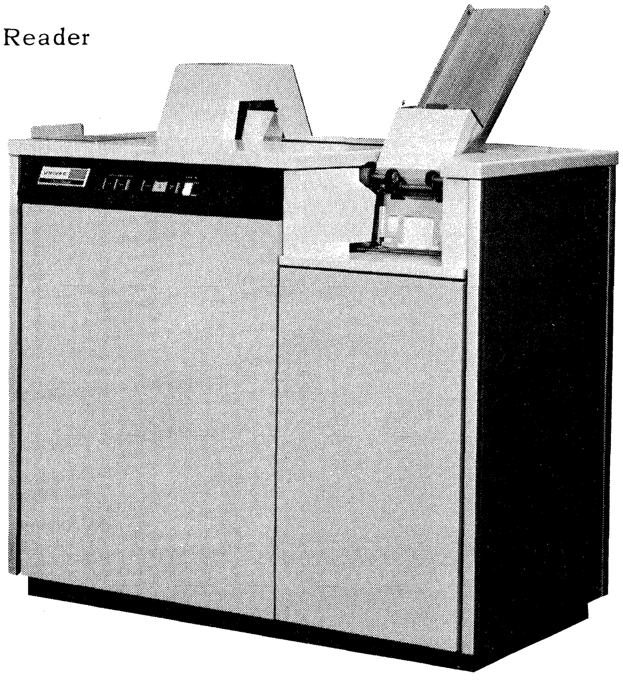
8.6. PUNCHED CARD SUBSYSTEM

The Punched Card Subsystem consists of a UNIVAC 900 cpm Card Reader and a UNIVAC 300 cpm Card Punch which are attached to a control unit on a single input/output channel of a CPU or IOC.

The card reader uses column-parallel photodiode sensing with automatic photodiode checking, error cards being ejected into a separate stacker. A file feed device is standard. Data from the reader may be translated into Fieldata code before transfer to main storage or transferred directly in row or column binary.

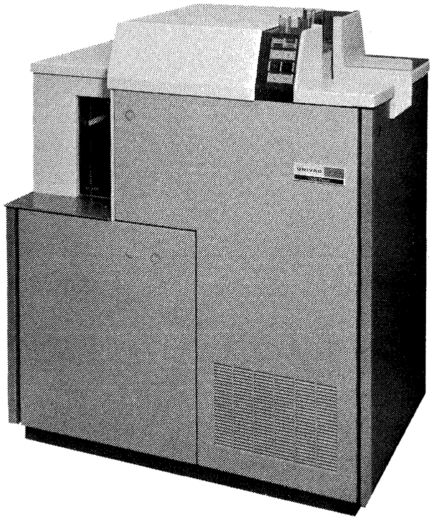
The card punch operates on a row-by-row basis and has an automatic check-read station. Incorrectly punched cards are recognized by the input/output handler examining the status word upon the termination of the function, and such cards are passed to the error stacker. Under program control additional attempts to punch the data can be made, and if an unacceptable error rate is achieved, the job may be suspended for maintenance action. Data may be punched in Fieldata code or in row and column binary.

8.6.1. UNIVAC Card Reader



CHARACTERISTICS	
CARD READING SPEED	900 cards/minute
INPUT HOPPER CAPACITY	3,000 cards
OUTPUT STACKER CAPACITY	2,100 cards
REJECT STACKER CAPACITY	100 cards
READ MODES	Fieldata, column binary, row binary
I/O CHANNELS	1 shared with card punch

8.6.2. UNIVAC Card Punch



CHARACTERISTICS	
CARD PUNCHING SPEED	300 cards/minute
INPUT HOPPER CAPACITY	1000 cards
OUTPUT STACKER CAPACITY	2 stackers of 850 cards each
PUNCH MODES	Fieldata, column binary, row binary
I/O CHANNELS	1 shared with card reader

8.7. UNITIZED CHANNEL STORAGE SUBSYSTEM

Photo to be supplied
when available.

The Unitized Channel Storage Subsystem is a self-contained subsystem utilized in the hierarchy of a drum subsystem and fully software supported. The purpose of this subsystem is to provide large capacity, random access mass storage for a multiprogramming environment in which the drum subsystems receive a high degree of usage. Utilization of this static storage subsystem significantly improves and enhances the performance of the 1108 mass storage subsystem by greatly reducing the access times (due to drum latency) inherent in drum storage devices. This latency time reduction results in a substantial increase in system throughput. The Unitized Channel Storage Subsystem is also beneficial to an environment which requires a great deal of swapping information in and out of storage.

Another advantage of the Unitized Channel Storage Subsystem is the elimination of a problem inherent to rotary type secondary storage devices. This problem concerns the time loss that occurs whenever a data transfer is interrupted and then resumed. If this interrupt is of such length and time that the next word of information has passed by the read/write heads, then an entire revolution must occur before data transmission from drum to main storage can be reinstituted. Unitized Channel Storage data transfer operations can be interrupted, then resumed without loss of time or data.

CHARACTERISTICS

STORAGE CAPACITY PER UNITIZED CHANNEL STORAGE MODULE	262K computer words of 36 data bits plus parity bit (subsystem expandable to four 262K modules providing a maximum storage capacity of 1048K words per subsystem)
INSTRUCTION ACCESS TIME	2.25 microseconds
INSTRUCTION EXECUTION TIME	0.75 microseconds
WORD TRANSFER RATE	2.25, 4.0 or 8.0 microseconds
NUMBER OF CONTROLLERS REQUIRED FOR EACH SUBSYSTEM	1 (capable of interfacing with an I/O channel and a maximum of four 262K unitized channel storage modules)
NUMBER OF I/O CHANNELS REQUIRED	1 per subsystem
CONTROLLING OPERATING SYSTEM	Exec II or Exec 8

9. DATA COMMUNICATIONS

9.1. GENERAL

There is a wide variety of methods for communicating with the UNIVAC 1108 System. Data transfer rates can vary widely, and many communication terminals can be multiplexed to one remote terminal which has direct high speed access to the processor.

9.2. UNIVAC 1108 COMMUNICATIONS SUBSYSTEM

The UNIVAC 1108 Communications Subsystem enables the UNIVAC 1108 System to receive and transmit data by way of any common carrier at any of the standard rates of transmission up to 50,000 bits per second. It can receive data from or transmit data to low speed, medium speed, or high speed lines in any combination.

As illustrated in Figure 9–1, the subsystem consists of two principal elements. The UNIVAC Communications Terminal Module (CTM) makes direct connection with the communication facilities. The UNIVAC Communications Terminal Module Controller (CTMC) transmits data between the modules and the Central Processor. A Communications Terminal Module Controller may be connected to any processor I/O channel, multiplexing up to 16 CTM's to that channel.

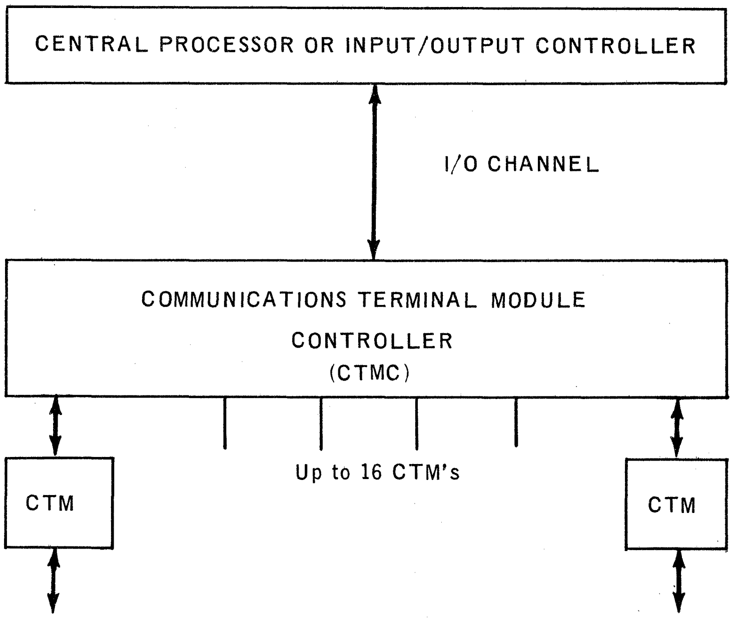

CENTRAL PROCESSOR OR INPUT/OUTPUT CONTROLLER
I/O CHANNEL
COMMUNICATIONS TERMINAL MODULE
CONTROLLER
(CTMC)
Up to 16 CTM's
CTM
CTM

Figure 9–1. CTMC Subsystem

There are three basic types of input and output CTM's: low speed (up to 300 bits per second), medium speed (up to 1800 bits per second), and high speed (2000 to 50,000 bits per second). Each is easily adjusted to the speed and other characteristics of the type of line with which it is to operate. Most CTM's accommodate two full-duplex or two half-duplex communication lines.

In addition to the serial modules, there are also input and output parallel modules and a dialing module available. The parallel modules operate at speeds up to 75 eight-bit characters per second. The automatic dialing module enables the processor to establish communication with remote points through the common carrier's switching network.

Characteristics of the six types of modules are summarized in the following table:

TYPE	SPEED	MODE	LEVEL
Low	To 300 BPS	Asynchronous Bit Serial	5, 6, 7 or 8
Medium	To 1800 BPS		
High	To 50,000 BPS	Synchronous Bit Serial	
Dialing	Variable	Bit Parallel	4
Parallel Out	To 75 CPS	Timing Signal Bit Parallel	8
Parallel In			

BPS = Bits per second; CPS = characters per second.

9.3. SYNCHRONOUS DATA TERMINALS

Each of the two types of synchronous data terminals discussed in this section links high speed (up to 230,400 bits per second) data circuits directly to an input/output channel without a control unit.

9.3.1. Word Terminal Synchronous

The Word Terminal Synchronous (WTS) is a central-site device that links a single high speed synchronous data communication line to a single I/O channel of the Processor. Though data on the line is bit serial, the WTS communicates with main storage a word at a time (six 6-bit characters per word) reducing the transfer time and the size of the buffer in main storage. However, the most significant advantage is that it reduces manipulation of data by the processor since it adds character and message parity to outgoing data and checks parity of incoming data. Upon detecting a parity error, it generates an external interrupt. The WTS can handle data at speeds of 2000, 2400, 40,800 and up to 230,400 bits per second.

The WTS allows the UNIVAC 1108 System to exchange data with a remote UNIVAC 1004 Card Processor. Such a configuration could be used for order entry, inventory control, and a wide variety of remote processing operations. Optional automatic dialing gives the WTS access to any station on the public network.

9.3.2. Communication Terminal Synchronous

Like the WTS, the UNIVAC Communication Terminal Synchronous (CTS) is a central site device that links a synchronous data communication line to a processor input/output channel at rates of 2000, 2400, or 40,800 bits per second. (The synchronous rate is set according to the characteristics of the line with which it is used; see the table of Synchronous Data Terminal characteristics.) But unlike the WTS, the CTS communicates one character at a time (five, six, seven, or eight bits per character) instead of a word at a time.

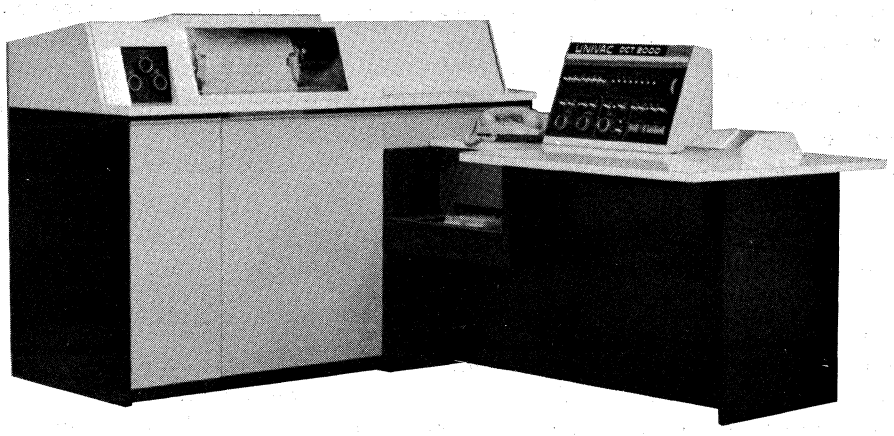
The CTS may optionally have the ability to dial remote locations automatically, and it may generate and check parity according to patchboard wiring.

A remote installation with equipment such as a UNIVAC 1004 Card Processor can, over the communication lines and through the CTS, have access to a UNIVAC 1108 System, thereby in effect affording the remote user the advantages of large-scale computer equipment. Conversely, the remote equipment can be used as an output terminal for the central facilities.

CHARACTERISTICS

	WTS	CTS
SPEED	2000, 24000, and 40,800 and up to 230,400 bits per second	
COMMUNICATION FACILITIES	2000 bits per second, public switched voice lines. 2400 bits per second, private voice lines. 40,800 to 230,400 bits per second broad band transmission lines.	
CONTROL CODING	Control characters and codes denoting Synch, Start of Message, End of Message, and the like, selected by plugboard.	
I/O TRANSFER MODE	One word (six 6-bit characters) per transfer	One character (5, 6, 7 or 8 bits) per transfer
DATA CODING	6 bits per character	5, 6, 7 or 8 bits per character
I/O CHANNELS REQUIRED	One per WTS	One per CTS

9.4. UNIVAC DATA COMMUNICATION TERMINAL (DCT) 2000



The UNIVAC Data Communication Terminal (DCT) 2000 is a combination printer and card reader/punch designed to transfer large quantities of data efficiently over voice-grade facilities. This terminal, tied into a network with computers (the UNIVAC 1004 Card Processors or other DCT 2000 systems), can handle up to 250 blocks per minute. The DCT 2000 is also available without the combination card reader/punch for use as a printer terminal.

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

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

- Reliability – through the use of the latest monolithic integrated circuits.

  Monolithic integrated circuits far exceed the reliability of ordinary transistorized circuits; furthermore, they produce less heat, use less power, and are less affected by fluctuating environmental conditions.

- Expandability – through the use of an input/output channel.

  The input/output channel permits the use of four additional input or output devices; for example, a paper tape punch might be added.

- Flexibility – through the use of eleven field-installable features (see Table 9–1).

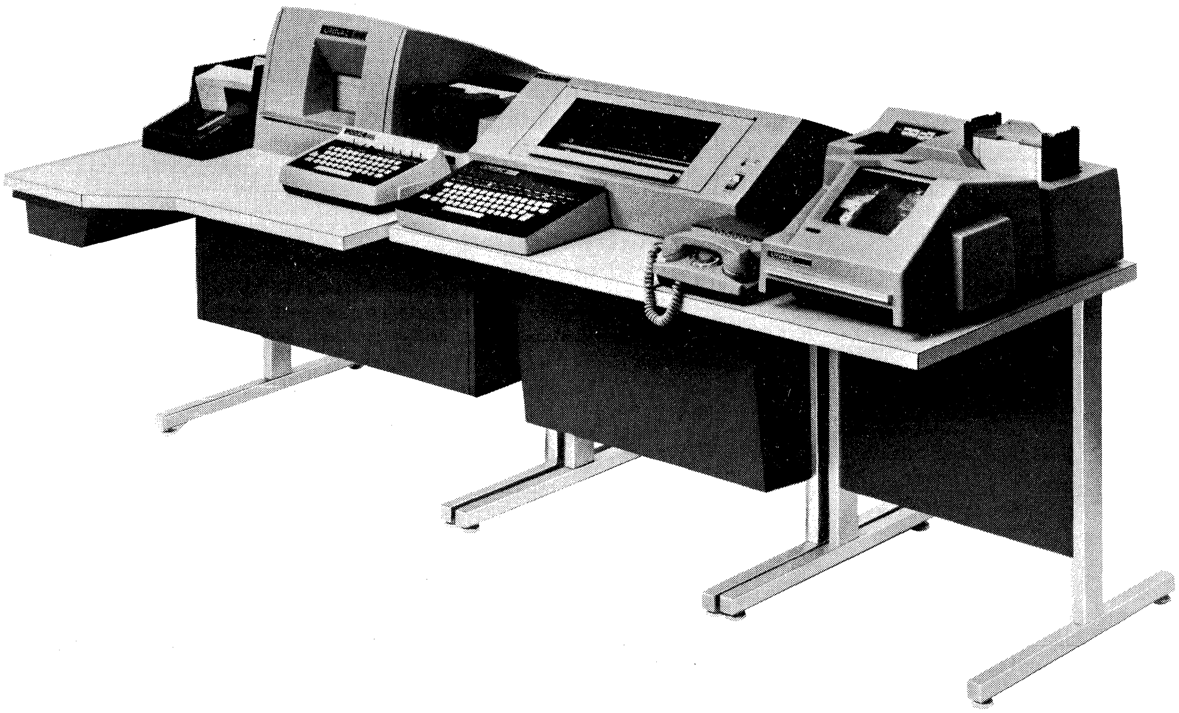
OPTION NAME	DESCRIPTION
Punch Check and Alternate Stacker	Allows a check of the actual punch die movement and diverts incorrectly punched cards to an error stacker while automatically repunching the data into another card.
128 Print Positions	Allows the basic 80 print-position line to be expanded to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

Table 9–1. DCT 2000 Field-Installable Options

CHARACTERISTICS

CARD READING SPEED	200 cards per minute
CARD PUNCHING SPEED	75-200 cards per minute
PRINTING SPEED	250 lines per minute
PRINTING POSITIONS PER LINE	80 or 128
PRINTABLE CHARACTERS	63 plus space
BUFFER STORAGE	256 character capacity in two buffers, 128 characters each.
TRANSLATION CAPABILITIES	Hollerith to ASCII Hollerith to XS-3 (DLT compatible)
TRANSMISSION METHOD	Block by block
TRANSMISSION MODE	Half duplex; 2 or 4 wire (nonsimultaneous; two-way transmission)
TRANSMISSION FACILITIES	Voice-grade telephone toll exchange, or private line
TRANSMISSION RATE	2400 bits per second (private line) 2000 bits per second (switched telephone network)
TRANSMISSION CODE	ASCII XS-3 (DLT compatible)

9.5. UNIVAC DATA COMMUNICATION TERMINAL (DCT) 1000



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

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

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

- Automatic Error Correction. This eliminates manual correction protection procedures such as reloading cards and retyping input data.

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

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

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

The DCT 1000 can be tailored to complement the transmission facility which fits the application best. The following options are available:

Line Type	– Switched or Private
Private Line	– 2 Wire or 4 Wire
Modulation	– Synchronous or Asynchronous
Transmission Speed	– Asynchronous 300, 1200, or 1800 Baud Synchronous up to 4800 Baud
Interface	– EIA RS-232 (Synchronous or Asynchronous) MIL STD 188B (Synchronous)
Direct Connection	– To CTM or DCS without modems
I/O Channel	– Direct to 1100, 400, or 9000 Series I/O channel by the terminal multiplexer.

The DCT 1000 transmission control procedures are completely compatible with the UNISCOPE 100. Therefore, DCT 1000's and UNISCOPE 100's can be intermixed on the same transmission line or on the same UNIVAC multiplexer. This mix and match capability yields an almost limitless number of configurations. Control can be achieved at the central computer with a single common handler.

DCT 1000 (printer only) stations can be used to furnish hard copy for the UNISCOPE 100. The printing operation is not dependent on the display hardware and does not delay any operator functions at the display stations.

When the DCT 1000 is not transmitting or receiving data, it need not be idle. The DCT 1000 can be used offline to generate paper tapes, list cards, or for media conversion. Additionally, while the DCT 1000 is receiving or transmitting data online, the punch can be used offline.

CHARACTERISTICS

CARD READING SPEED	40 cards per minute
CARD PUNCHING SPEED	35 cards per minute
PRINTING SPEED	30 characters per second
PRINTING POSITIONS PER LINE	132 (adjustable tractor)
PRINTABLE CHARACTERS	63 plus space
PAPER TAPE SPEEDS	50 characters per second
BUFFER STORAGE	320 character capacity in two buffers, 160 characters each
TRANSLATOR SELECTIONS	ASCII Code H (Scientific) Code A (Business) Code Binary with additional feature
TRANSMISSION METHOD	Block by block
TRANSMISSION MODE	Half duplex; 2 or 4 wire (nonsimultaneous; two-way transmission)
TRANSMISSION FACILITIES	Voice-grade telephone toll exchange or private line
TRANSMISSION RATE	A synchronous 300, 1200, or 1800 bits per second; synchronous 4800 bits per second.

9.6. UNIVAC DATA COMMUNICATION TERMINAL (DCT) 500

The UNIVAC Data Communication Terminal (DCT) 500 is a low cost, unbuffered, asynchronous keyboard/printer terminal similar in operation to a teletypewriter. Tht DCT 500 is, however, two to three times faster than a teletypewriter and provides up to a 132-column format and five carbons. The DCT 500 can replace existing teletypewriters with little or no changes in the software handlers for point-to-point communications networks over voice-grade telephone toll lines or private lines. In a multi-party polled environment, the DCT 500 operates in accordance with ASCII procedures.

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

Optional Features:

- Automatic Answering
- Master/Slave Operation
- Print Monitor
- Internal Modern
- Paper Tape

CHARACTERISTICS

TRANSMISSION CODE	8-level ASCII
INTERFACES	EIA Standard RS-232B/CCITT Internal Modem
TRANSMISSION MODE	Half-Duplex or Full-Duplex (2 or 4 wire)
TRANSMISSION RATE	110, 150, or 300 bits per second (selectable)
PRINTING RATE	30 characters per second
FONT SELECTIONS	ASCII, EBCDIC A (Business)/H(Scientific)
PRINTABLE CHARACTERS	63 plus space
PRINT POSITIONS PER LINE	132 (adjustable tractor)
PAPER TAPE READER/PUNCH RATE	50 characters per second

9.7. UNISCOPE 300 VISUAL COMMUNICATION TERMINAL SUBSYSTEM

The UNIVAC UNISCOPE 300 Visual Communication Terminal Subsystem is designed for applications requiring direct user interaction with the central system. These terminals can be located at or near the site, or they may be operated as remote stations over standard data communication facilities.

The UNISCOPE 300 itself includes a keyboard and CRT display, a main storage unit to store data as it is typed or received and displayed on the CRT, and control circuits. As the operator types a message on the keyboard, the data is accumulated in the storage unit and displayed on the CRT with a cursor marking the location of the next character to be typed. The operator can then make changes before releasing the message for transmission to the computer. Messages from the computer are similarly displayed and can be altered and returned to the computer by the operator.

9.7.1. The Keyboard

The keyboard includes alphanumeric keys, cursor controls, editing keys, and function keys. The alphanumeric section is similar to the standard electric typewriter in layout and operation. The character set has 56 symbols ordinarily, but up to 96 characters are available. The cursor controls can set the cursor to any point on the screen, so that deletions and changes can be made while composing or editing messages. Up to 40 different function keys can be added to the keyboard. The meanings of 35 of these function keys can be varied by means of overlays that fit over the group. The overlays are cards each of which has an edge formed according to an identifying code; when an overlay is in position, the coded edge causes the operation of some combination of seven switches controlling the significance of the function keys. On the face of the overlay, and appearing adjacent to the function keys when the overlay is in place, are markings to indicate the corresponding use of each key. One hundred and twenty-two different overlays can be used, enabling representation of as many as 4000 different functions. Typically, the different overlays can be used to identify stations, operators, applications, or security requirements. In addition to initiating a function, each function key displays a unique symbol on the CRT.

9.7.2. CRT Display

Sixteen lines of 64 characters each can be displayed on the 5-inch by 10-inch screen. The data is actually stored in a 1024-character main storage unit and is regenerated on the screen 60 times per second.

9.7.3. Subsystem Configurations

The subsystem can be used in single- and multi-station configurations.

The single-station terminal is completely self-contained. It interfaces directly with the data communications equipment through a standard EIA interface. Transmission line rate must be greater than 2000 characters per second. It can be used individually in a private line circuit or a number of them can be connected to a multipoint party line. In this latter arrangement the unit responds to a poll code from the central system.

The multi-station configuration provides a more economical arrangement where a large number of terminals are required. This configuration requires a multi-station control unit. The control unit is modular; it provides I/O message buffering, character generation, and control logic for up to 24 terminals of 1024 character capacity or up to 48 terminals of 512 character capacity. In this configuration each terminal is completely independent of all others. The terminals can be located as much as 1500 feet from the control unit.

Optionally the control unit can be equipped to communicate over two separate lines. This makes possible independent, concurrent communication (input and/or output) over the two lines; or, if desired, one line can be reserved as backup.

In another configuration, two control units can be connected to the same set of terminals with one unit switched on and the other off to provide a backup control unit. The units can be switched on or off manually or by the program. By this means, one extra control unit can serve as a spare for as many as 48 sets of displays.

Polling techniques allow single-station and multi-station units to be mixed freely on one multipoint party line. This capability increases line utilization and significantly decreases line costs.

9.7.4. Reliability

In both single- and multi-station configurations the subsystem checks the character parity and message parity of incoming messages, and generates them for outgoing messages. Erroneous messages are retransmitted on request. A simple message acknowledgment system ensures that no messages are lost or duplicated.

To further improve reliability, the central system can perform marginal tests on the storage unit.

9.7.5. Special Features

Several special features of the subsystem contribute especially to its use as an online device.

A pair of special function codes enable the program to insert or delete a line of copy. The insert function moves the line marked by the cursor and all lines below it down one line, the bottom line moving off the screen. It then inserts the new text. The delete function erases the line marked by the cursor and all lines move up one line, leaving the bottom line blank. By means of these functions, the program can rearrange data on the screen with a minimum of new transmission.

Of particular interest in the ‘‘roll and scroll’’ technique implemented by use of the insert and delete functions. In this case text on the screen appears to roll up or down with the screen remaining filled. Typically, this is useful for scanning through large tables or other lengthy text that is too large for the screen. As soon as the required part of the table is located, the operator can stop the rolling by means of a function key.

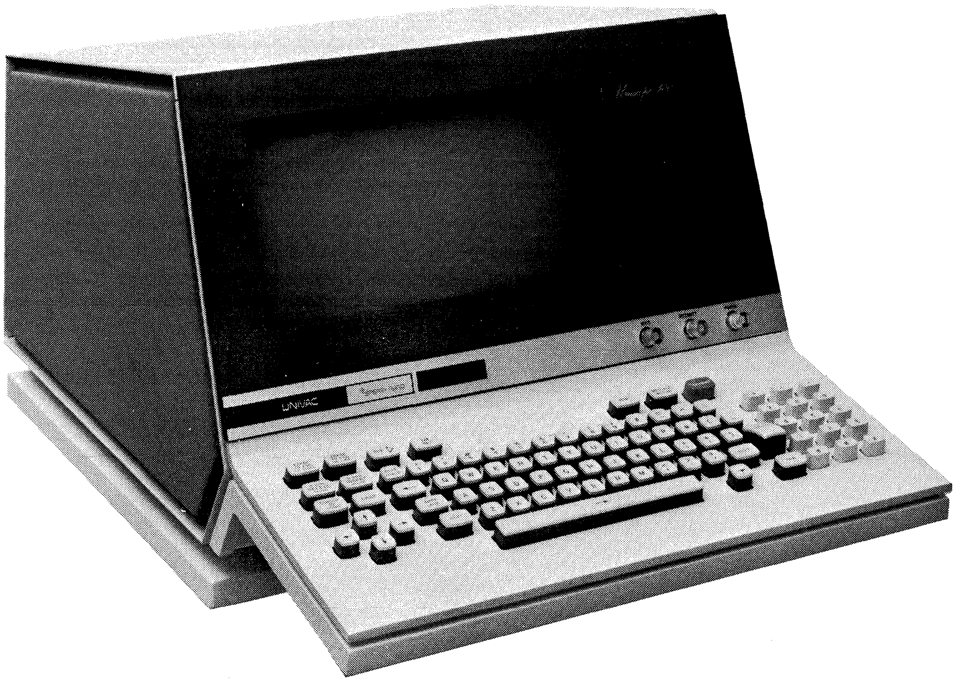
The capability of split-screen operation also increases the versatility of the terminal. This makes possible simultaneous, independent display of several messages. In this case an end-of-field symbol designates the end of each separate message. Line insert and delete functions can then be used on the messages separately and each one can be rolled and scrolled without interfering with the others.

For emphasis, specific messages or parts of messages from the system can be programmed to blink. In addition, unsolicited messages from the central system are accompanied by an audible signal. Such a message can be programmed for immediate display, overriding other outputs, or it may be withheld pending operator response.

CHARACTERISTICS

DISPLAY	1024 characters total (16 lines of 64 characters) 10-inch by 5-inch screen
KEYBOARD	56 to 96 different symbols Editing keys Up to 40 function keys which, with overlays, produce more than 4000 functions
ERROR CONTROL	Character and message parity All messages acknowledged
DATA TRANSMISSION FACILITIES	More than 2000 characters per second Voice grade lines Half- or full-duplex operation Parity line by polling available Nonsignificant spaces suppressed

9.8. UNISCOPE 100 VISUAL COMMUNICATION TERMINAL SUBSYSTEM



The UNISCOPE 100 Visual Communication Terminal Subsystem is a low-cost, alphanumeric display designed for a broad range of applications which require direct operator interaction with a central computer system. Due to its modular construction, the UNISCOPE 100 can operate either as a data entry or as a display device. It can be conveniently located at the central computer site or at a remote station where it is connected to the system by way of telephone lines.

The UNISCOPE 100 is a self-contained unit consisting of a cathode-ray tube display screen, refresh storage, character generator, control logic, operator keyboard, and communication interfaces. Special interfaces for direct computer connection and hard copy output are also available. A variety of presentation formats are offered which provide a total display capacity of 480, 512, 960, or 1024 USA Standard Code for Information Interchange (ASCII) characters. Each of these units is capable of displaying the complete ASCII set of 96 characters which include upper and lower case alphabetics. Hardware editing features enable the operator to completely edit any message prior to transmitting it to the computer.

Up to 31 UNISCOPE 100 terminals may be connected to a single communication line or to a computer input/output channel by means of a multiplexer. This general-purpose multiplexer is available with all the communication line interfaces available on the UNISCOPE 100 terminal, thus permitting a mixture of single units and multiple units on one communication system. The multiplexer also provides broadcasting of output messages to multiple devices.

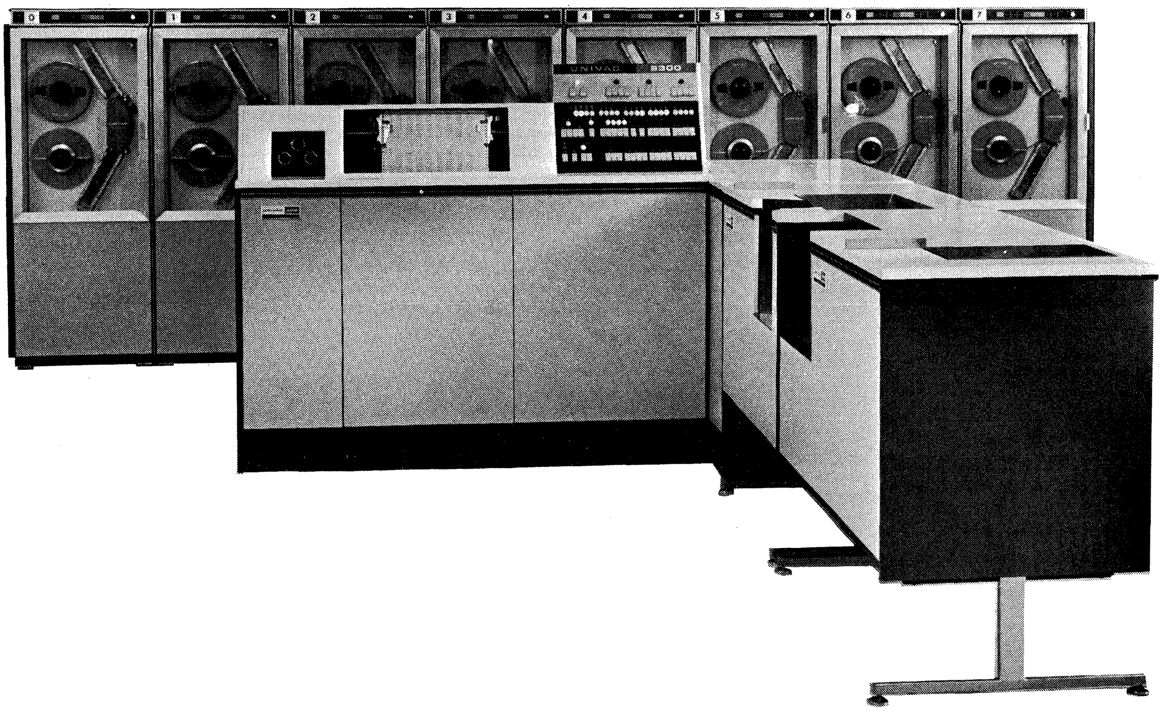
The keyboard has been functionally designed to approximate the conventional electric typewriter with its keyboard appearance, touch pressure, key travel, and slope characteristics. Typewriting speeds in excess of eighty words per minute can be accommodated by the keyboard. Because of its similarity to the standard typewriter, little additional training is required to operate it.

The keyboard includes cursor controls and editing keys, and the layout is right-left assignment balanced to efficiently distribute the work load. The keys are arranged for convenient function discrimination.

CHARACTERISTICS

DISPLAY	480, 512, 960 or 1024 characters (80 per line, 6 lines; 32 per line, 16 lines; 80 per line, 12 lines; 64 per line, 16 lines) on 10-inch wide and 5-inch high screen. Characters are ASCII, and also include both upper and lower case alphabetics. Split screen.
KEYBOARD	Alphanumeric and symbolic with 8 cursor control keys and 5 editing keys.
STORAGE	Magnetic drum, 7.2-microsecond cycle time, 7-bit ASCII code with parity bit.
DATA TRANSMISSION	Direct - up to 50,000 characters per second Lines - up to 9600 bits per second Half or full duplex Party line polling Nonsignificant space suppression Block transmission Message segmentation
POWER	Standard office receptacles

9.9. UNIVAC 9200/9300 ONSITE/REMOTE SUBSYSTEMS



The user of a UNIVAC 9200/9300 or 9200 II/9300 II Systems can, without impairing to the slightest degree its capabilities as a self-contained data processor, demand the full and powerful facilities of the 1108 System. This is accomplished by use of the Intercomputer Control Unit (ICCU) to connect the 9200/9300 to the 1108 System and the DCS (Data Communications Subsystem) to link a 9200/9300 System to communication lines. The DCS permits selection of communication speed for the most efficient use of the means of transmission; dial-switched voice line, private line, or broad-band line.

The 9200/9300 equipment has storage large enough and internal speeds fast enough to utilize communications lines to the fullest. Further more, the DCS subsystem has provisions for error detection which are flexible enough to be adaptable to any reasonable requirements of the user.

When a 9200/9300 System is being used as a remote subsystem of an 1108 installation, the Executive software system will enable the remote user to send his program and data over a communication line and receive the complete output later either at the point of origin or at some other designated location.

The minimum 9200/9300 System consists of a central processor unit, 8K of main storage, and a printer. All other peripherals are options and must be specially ordered.

CHARACTERISTICS

	9200	9200 II	9300	9300 II
SYSTEM ORIENTATION	Card/Disc	Card/Tape/Disc	Card/Tape/Disc	Card/Tape/Disc
BASIC MAIN STORAGE	8192 bytes	8192 bytes	8192 bytes	16,384 bytes
MAXIMUM MAIN STORAGE	16,384 bytes	32,768 bytes	32,768 bytes	32,768 bytes
MAIN STORAGE CYCLE TIME	1200 nanoseconds per byte	1200 nanoseconds per byte	600 nanoseconds per byte	600 nanoseconds per byte
ADD (DECIMAL) INSTRUCTION TIME (TWO 5-DIGIT NUMBERS)	104 microseconds	104 microseconds	52 microseconds	52 microseconds
MULTIPLY, DIVIDE, AND EDIT CAPABILITY	Optional	Optional	Standard	Standard
CARD READER BASIC READER	400 cpm	400/600 cpm	600 cpm	600 cpm
UNIVAC 1001 CARD CONTROLLER	1000/2000 cpm	1000/2000 cpm	1000/2000 cpm	1000/2000 cpm
CARD PUNCH COLUMN	75 - 200 cpm	75 - 200 cpm	75 - 200 cpm	75 - 200 cpm
ROW	Not available	200 cpm 250 cpm	200 cpm 250 cpm	200 cpm 250 cpm
	Read/punch optional	Read/punch optional	Read/punch optional	Read/punch optional
PRINT SPEED	250, 300, or 500 lpm	250, 300, 500, 600, 900, 1100, 1200, or 1600 lpm	600, 900, 1100, 1200, or 1600 lpm	600, 900, 1100, 1200, or 1600 lpm
OVERLAPPED I/O UNITS	Standard	Standard	Standard	Standard
MAGNETIC TAPE TRANSFER RATE	Not available	34,160 bytes per second	34,160 bytes per second Simultaneous read, write, and process is optional	34,160 bytes per second Simultaneous read, write, and process is optional
PAPER TAPE READ	300 cps	300 cps	300 cps	300 cps
PUNCH	110 cps	110 cps	110 cps	110 cps
DISC SUBSYSTEMS STORAGE UNIVAC 8410	3.2 to 12.8 million bytes	3.2 to 12.8 million bytes	3.2 to 12.8 million bytes	3.2 to 12.8 million bytes
UNIVAC 8411	Not available	7.25 to 58 million bytes	Not available	7.25 to 58 million bytes
UNIVAC 8414	Not available	58.352 to 233.36 million bytes	Not available	58.352 to 233.36 million bytes
MULTIPLEXER CHANNEL TRANSFER RATE	85,000 bytes per second	85,000 bytes per second	85,000 bytes per second	85,000 bytes per second
SELECTOR CHANNEL TRANSFER RATE	Not available	350,000 bytes per second	Not available	350,000 bytes per second
DATA COMMUNICATION SUBSYSTEMS	Up to 8 duplex lines	Up to 8 duplex lines	Up to 8 duplex lines	Up to 8 duplex lines
REGISTERS	8 for processor functions 8 for I/O functions	8 for processor functions 8 for I/O functions	8 for processor functions 8 for I/O functions	8 for processor functions 8 for I/O functions

9.10. THE UNIVAC DATA COMMUNICATION SUBSYSTEM (DCS-1)

The DCS-1 subsystem provides communication capability for the 9000 Series computer. Connected to a multiplexer channel, it enables synchronous data transmission at speeds up to 50,000 bits per second between the 9200/9300 computers and the 1108 System over standard communication circuits. The unit is physically small so that two of them can be mounted in space available in the 9200/9300 main frame.

The subsystem is modular, permitting field modifications as demands for various options arise. As communications needs grow, different interfaces can be substituted to upgrade capabilities.

Its many features include the following:

- Automatic Error Checking

  The subsystem checks character and message parity by sending either odd or even parity bits. Longitudinal redundancy can be checked by hardware or by user software.

- Self Testing

  The hardware tests the DCS-1 under program control connecting the output line to the input line to verify transmission and receipt of data.

- Unattended Answering

  The subsystem responds to incoming calls from dialed lines without operator intervention.

- Variable Message Length

  A message may be of any length, from one character up to available storage size.

- Compatibility

  The DCS-1 is compatible with a number of UNIVAC systems including 1108, 494, DCT-2000, and the 1004.

CHARACTERISTICS

SPEED AND FACILITIES	Dial voice lines - 2000 bits per second Private voice lines - 2400 bits per second Broad-band lines - 50,000 bits per second
DATA CODING	Five to eight levels, plus parity.
CHECKING	Odd or even message and character parity. Longitudinual redundancy check is optional.
MULTIPLEXER CHANNELS REQUIRED	One per subsystem.

10. PROGRAMMED SYSTEMS SUPPORT

10.1. AVAILABLE SOFTWARE

The programmed systems support (software) provided with the UNIVAC 1108 System has been designed to meet the total computing requirements of the advanced users of today. The requirements and demands of the large-scale user have grown considerably in the past, and it is in response to these changing requirements that UNIVAC offers with the 1108 computer a software system designed to meet today's requirements and to allow the change and growth required to meet tomorrow's challenge. The degree of effective utilization of any computing system is in direct proportion to the scope and versatility of the software. With the 1108 System, UNIVAC has combined many years of experience in multiprogramming and communication oriented systems to provide a system that is easy to operate and easy to use, yet one which ensures user program integrity in a demand-response environment.

The UNIVAC 1108 computer system includes a full, complete set of software, ranging from high level language compilers to basic mathematical functions. The major software items are:

The Executive System

The Assembler

FORTRAN V

Conversational FORTRAN

LIFT – FORTRAN II to FORTRAN V Translator

COBOL (Extended and Revised)

ALGOL

SORT/MERGE

Application Programs

The entire 1108 software system, based upon the FH-432 and FH-1782 magnetic drums with their rapid access and high transfer rate, and backed up by mass storage, can give unsurpassed performance.

10.2. THE EXECUTIVE SYSTEM

To take full advantage of the speed and hardware capabilities of the UNIVAC 1108 System and to make effective use of a given hardware configuration, a comprehensive internal operating environment has been created.

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

Only through central control of all activities of the 1108 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 system, which controls and coordinates the functions of the complex internal environment. By presenting a relatively simple interface to the programmer it allows him to use the 1108 System easily, while relieving him of concern for the internal interaction between his program and other co-existent programs.

10.2.1. Multiple Modes of Operation

The technical capabilities of the UNIVAC 1108 Executive System cover a great variety of data processing activities. It is designed to maximize batch processing and acceptable amounts of real time and demand processing. An installation not interested in making use of the full range of activities may specify capabilities to be eliminated at system generation time.

10.2.1.1. Batch Processing

Foremost among these capabilities is the support provided for batch processing. The system is designed to ease run preparation and submission, to shorten job turn-around time, and reduce the need for operator intervention and decisions.

10.2.1.2. Demand Processing (Time-Sharing)

Complementing the batch processing capabilities of the 1108 Executive System are its time-sharing capabilities, the simultaneous accommodation by the Executive system of requests and demands from users at numerous remote inquiry terminals, operating in a demand (or conversational) mode. All facilities available to the batch processing user are also available in the demand mode, the primary difference being that the Executive system permits the user additional flexibility in the statement and control of individual runs. The demand user may communicate directly with either the Executive or a user program or he may communicate with a conversational processor, such as Conversational FORTRAN.

10.2.1.3. Real Time

The Executive system is equally applicable to programs which have real time requirements. The Communications Terminal Module Controller, together with efficient scheduling and interrupt processing features of the Executive system, provide an environment satisfactory for any real time program.

10.2.1.4. Multiprogramming

Runs may come from many sources, remote and central. The various runs, through the Executive system's use and control of efficient multiprogramming techniques may, at any given moment, be in different stages of activity; input, processing, and output may all be occurring simultaneously within the hardware, thus ensuring efficiency of operation.

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

10.2.2. Techniques for Utilization of Mass Storage

The Executive system is designed to ensure effective and efficient utilization of the mass storage devices. The consequence is an unprecedented ability to relieve operators and programmers of the responsibility of maintaining and handling cards and magnetic tapes, thus eliminating many of the errors which heretofore have accompanied the use of large-scale software systems. At the same time, the overall operating efficiency is considerably improved.

Permanent data files and program files are maintained on the mass storage devices, with full facilities for modification and manipulation of these files. Security measures are established by the Executive system to ensure that files are not subject to unauthorized use. Provisions are also made within the Executive system for automatic relocation of infrequently used files to magnetic tape, as unused mass storage space approaches exhaustion. When the use of files relocated in such a manner is requested, they are retrieved and restored under control of the Executive system with no inconvenience to the user.

10.2.3. The Primary Functional Areas of the Executive System

The UNIVAC 1108 Executive System is composed of many different routines which perform many different functions. These functions and routines are summarized in the following paragraphs.

10.2.3.1. Executive Control Language

In the Executive system the user has a simple means of directing the execution of the individual activities of a run and of relaying operational information concerning the run to the Executive. This is accomplished through a set of control commands, capable of performing all of the functions desirable or necessary in a modern Executive system. The command language is open ended and easily expanded, so that features and functions may be added as the need arises.

The basic format of an Executive Control Statement is quite simple and is adaptable to a large number of input devices. Statements are not restricted to card-image format, and may be of variable lengths. Each statement consists of a master space (@), for recognition purposes, followed by a command and a variable number of expressions. The end of a statement is indicated by the end of a card, a carriage return, or an equivalent signal, depending on the type of input device.

10.2.3.2. The Supervisor

The Supervisor is the 1108 Executive System component that controls the sequencing, setup, and execution of all runs entering the 1108 System. It is designed to control the execution of a large number of programs without any interaction among them.

The Supervisor contains three levels of scheduling: coarse scheduling, dynamic allocation of storage space, and CPU dispatching. Runs entering the 1108 System are classified into information files and these files are used by the Supervisor for run scheduling and processing. Control statements for each run are retrieved and scanned by the control command interpreter in the Supervisor to facilitate the selection of runs for setup by the Coarse Scheduler. The coarse scheduling of each run primarily depends on two factors – the priority of the run and its facility requirements.

The Dynamic Allocator takes runs set up by the Coarse Scheduler and allots storage space according to the needs of the individual tasks of a run. Each run may be thought of as being made up of tasks, where a task is a single operation of a system processor or the execution of a user program. All tasks for a given run are processed serially but not necessarily consecutively; if there are several runs, the tasks of separate runs are interleaved.

When time-sharing of main storage is appropriate, the Dynamic Allocator initiates "storage swaps". This involves writing one program on drum storage and replacing it temporarily in main storage with another program. Such action is taken only to provide reasonable response time to remote demand-processing terminals.

The CPU Dispatching routine is a third level of scheduling; it selects among the various tasks currently occupying main storage whenever it is appropriate to switch the commitment of the CPU from one task to another. Under normal circumstances, a batch program is allowed to use the CPU either until it becomes interlocked against some event or until some higher priority program is freed of all of its interlocks.

10.2.3.3. Time Slicing

In order to accommodate demand processing, periodic time slices are assigned to all tasks except when in the real time mode. This is done by the timing routine, which interlocks the currently running program and, at specified intervals, examines a separate queue of periodically scheduled routines. Based on the required duty cycle of the demand routines, their priorities, and the priorities of other ready routines, the demand routine is moved to the ready queue with an adjusted priority. In this manner, even low-priority demand routines are given at least occasional use of the CPU.

10.2. 3.4. Storage Compacting

Certain 1108 hardware features make feasible the dynamic relocation of programs residing in main storage – a necessity for effective multiprogramming. At program termination, the assigned storage is returned to the pool of available main storage.

Storage is compacted only if a requirement exists for contiguous storage and if compacting can meet this requirement. Compacting is never performed unnecessarily. Instead the storage contents control routine always fits programs into gaps in the in-use store, if possible.

10.2.3.5. Facilities Assignment

Available facilities and their disposition are indicated to the system at system generation time; thereafter, the Executive system assigns these facilities, as needed and as available, to fulfill the facilities requirements of all runs entering the 1108 System. The Executive system maintains current inventory tables that indicate what facilities are available for assignment and which runs are using the currently unavailable facilities.

10.2.3.6. The File-Control System

The 1108 File-Control System affords the highest degree of operational flexibility in storing and retrieving data without concern for the physical characteristics of the recording devices. Thus, most files are made insensitive to input/output media characteristics as the system adjusts the interface between the file and the device. Security measures ensure that files are not subject to unauthorized use or destruction. Facilities are provided to roll out files from mass storage devices to magnetic tape, as well as reconstruct such files on the mass storage devices when the user calls for them.

Comprehensive utility routines are available for manipulation of files and to inform the user of the current status and structure of his files. Provisions are made for random storage and retrieval-access of data under the direction of the user. User program files and data files are maintained and processed in the same environment.

10.2.3.7. Operator Communications

The Executive system has been designed for operation with a minimum of operator intervention. However, some functions frequently in use are beyond the scope of the Executive system while others demand operator concurrence. In addition, certain information must be presented automatically to the operator while other information must be available to answer operator requests.

Operator functions are required for a large variety of activities. The 1108 Executive system groups them into four classes, thus equally dividing operator duties in a multi-operator installation. These four functional classes are System Control, Input/Output Activity, Communications Activity, and Hardware Confidence Activity. These functions may be associated with as many as three operator consoles or as few as one, depending on the complexity and layout of the installation.

The 1108 hardware system has as standard equipment a display console to enhance operator-system communications. The advantages of a visual display to the operator are obvious and the possible display functions endless. The Executive system displays information such as current system load and operator requests associated with I/O setup and I/O interlocks. The operator can request other information such as backlog status. If the display area becomes filled, the Executive defers lower priority displays.

10.2.3.8. Diagnostic System

A comprehensive diagnostic system within the 1108 Executive System aids in checking out user programs. Both allocation-time and assembly-time commands trigger snapshot dumps. Postmortem dumps of user tasks are available through an Executive Control Statement.

10.2.3.9. Input/Output Device Handlers

The Input/Output device handlers control the activities of all input/output channels and peripheral equipment attached to the 1108, including UNISERVO magnetic tape units FH-432, FH-1782 and FH-880 drums, 8414 Disc, Unitized Channel Storage, FASTRAND Mass Storage, card readers, printers, and communication line terminals.

10.2.3.10. Auxiliary Processors

The system includes a set of auxiliary processors which perform functions that complement those of the source language processors such as FORTRAN, COBOL and Assembler. This set of processors includes the Collector for linking relocatable subprograms, the Procedure Definition Processor for inserting and modifying procedure definitions in a library, the ELT Processor used to insert elements, and the DATA Processor to introduce data descriptions. A comprehensive set of library elements complements these processors.

10.2.3.11. Utilities

Included within the Utilities section of the Executive system are diagnostic routines, program file manipulation routines, file utility routines, and cooperative routines which aid the user by performing such functions as reading cards, printing line images, transferring files from device to device, and carrying out housekeeping functions required for file-residence on mass storage devices.

10.2.3.12. Processor Interface Routines

The Processor Interface Routines provide a simple standard interface for all processors within the system. Complete facilities are provided for the input of source-language statements and the output of the resulting relocatable binary code.

10.2.3.13. System Setup

The System Setup section of the Executive system provides an installation with a means of generating a system tailored to its particular needs and for subsequently maintaining this system with a minimum of effort.

10.3. THE ASSEMBLER

The Assembler translates a symbolic language composed of brief expressions to machine-language relocatable object coding for the UNIVAC 1108 System.

The symbolic language includes a wide variety of sophisticated operators which allow the development of a desired object code based on information generated at assembly time. The instruction operation codes are assigned mnemonic codes which describe the hardware function of each instruction. By the use of Assembler directives, the programmer can generate data words, values, or instructions based on specific conditions at assembly time. Multiple location counters make it possible to control address generation during assembly of a source code program. The Assembler produces a relocatable binary output in a form suitable for processing by the collector. If requested, it supplies a listing of the original symbolic coding and an edited octal representation of each word generated. Flags indicate errors in the symbolic coding detected by the assembler.

10.3.1. Symbolic Coding Format

When writing instructions using Assembly language, the programmer is primarily concerned with three fields: a label field, an operation field, and an operand field. It is possible to relate the symbolic coding to its associated flowchart, if desired, by appending comments to each instruction line or program segment.

All fields except the label field, which begins in column 1, are in free form providing the greatest convenience possible for the programmer. Consequently, the programmer is not hampered by the necessity of considering fixed form boundaries in the design of his symbolic coding. The order of the fields, however, is fixed. Appendix C lists the instructions recognized by the assembler.

10.3.2. Assembler Directives

The symbolic assembler directives control or direct the assembly processor. They are represented by mnemonics written in the operation field of a symbolic line of code. Their flexibility is the key to power of the assembler. The directives are used to equate labels to expressions, to change the location counter value, and to give the programmer full control over the generation of object coding.

10.3.3. Additional Features

Facilities are provided which permit separately assembled elements (or subprograms) to be linked together at collection time. A label followed by an asterisk is defined as an external label which can be referenced by other programs, as well as by the program in which it is defined. A job to be executed may be composed of many elements. The recompilation of any element does not necessitate the recompilation of the remaining elements which compose the job. The program is constructed, using the technique above, before the time of execution.

10.4. FORTRAN V

FORTRAN V is an algebraic language designed primarily for scientific and engineering computations. It closely resembles the language of mathematics. It is the logical outgrowth of the earlier FORTRAN languages and is generally compatible with them (although the earlier languages are not a proper subset of FORTRAN V). The FORTRAN V language has been extended to provide more flexibility in data handling and to make programming easier. FORTRAN V, being an outgrowth of the earlier FORTRAN languages (in particular, UNIVAC 1107 FORTRAN IV and IBM FORTRAN IV as announced in IBM form C-28-6274-1), accepts these languages as compatible although the reverse is not necessarily true.

FORTRAN V has all the features of the proposed ANSI FORTRAN IV language plus many valuable extensions which significantly increase the power and flexibility of the language, particularly in the areas of data handling. For further information, consult the *FORTRAN V Programmers Reference Manual, UP-4060* (current revision).

10.4.1. Language Extensions and Enhancements

The following extensions and enhancements are currently available in UNIVAC 1108 FORTRAN V.

- The PARAMETER statement assigns specified integer values to specified variables at the time of compiling; PARAMETER I = 2 replaces I with the integer 2 whenever it occurs in the source program. This facilitates the assignment of different values to frequently used parameters in different compilations of the same program.

- The ABNORMAL statement permits increased optimization of object programs. Where common subexpressions occur within a program unit, it is obviously desirable to evaluate each subexpression only once. Where the common subexpressions contain function references, however, there is a possibility that the function will produce a different result upon successive references with the same arguments. Because of this possibility, most FORTRAN systems are forced to reevaluate subexpressions containing function references at each occurrence. UNIVAC FORTRAN V permits all functions that can produce different results from identical sets of arguments to be designated ABNORMAL. All common expressions except those that reference ABNORMAL functions are evaluated only once. When the ABNORMAL statement does not appear at all in a program, all function references except library functions are considered ABNORMAL and are reevaluated at each occurrence, as in most other FORTRAN systems.

- Nonstandard subroutine returns (of the form RETURN k) are permitted where k specifies the subroutine argument to which a return is made.

- In conjunction with RETURN statements, statement labels may be used as subprogram arguments.

- A variable may have up to seven subscripts.

- Internal subprograms are permitted; that is, main program or subprogram and internal subprograms are part of the same program unit, which requires only one compilation.

- Variables of different types may occur in the same expression with two exceptions:

  Logical variables cannot be mixed with other types.

  Double precision and complex variables cannot be mixed.

- Extended subscript expressions are permissible having the form $\pm M_1 \pm M_2 \ldots \pm M_i$ where M is of the form $K_1{*}K_2{*}K_3 \ldots K_j$. The K's represent either an integer constant, an unsubscripted integer variable, or a parameter variable. No more than one K may be a DO index.

- Forward and Backward DO Loops (that is, increasing and decreasing index variable) are permitted.

- A generalized assigned GO TO may be used; the assigned GO TO need not have a list of possible assignments.

The following language extensions and enhancements, not available with FORTRAN IV or earlier versions, are now available with FORTRAN V:

- A string of consecutive bits, called a field, may be defined and operated on by making use of the FLD (e_1,e_2,e_3) intrinsic function, where e_1 and e_2 determine a field of the expression e_3. In this instance, e_1 and e_2 are integer expressions which give the starting position (e_1) and the length (e_2) of the field being defined. The FLD function may be used for extraction and insertion of bit fields.

- The NAMELIST statement which is nonexecutable provides data-characteristic information at object time, and may be used instead of specifying LIST on an INPUT/OUTPUT statement and the associated FORMAT. A NAMELIST name (1–6 alphanumeric characters) is defined by its appearance in a NAMELIST statement, and thereafter may appear only in formatted READ and WRITE statements.

- The DEFINE statement is of the form DEFINE R(a_1, ..., a_n) = e or DEFINE R = e, where R and a_i are variable names and e is any expression not involving any undefined R's. Inline code is generated when a DEFINE is referenced, thereby eliminating the overhead of a subroutine and enabling the optimizing capabilities to apply. Such a statement provides the following benefits:

  – All statement functions of FORTRAN IV operate more efficiently at object time.

  – Mathematical equivalence between arrays and variables can be attained.

  – Subscripting of subscripts is in effect, permitted to any level.

  – Any legal FORTRAN V expression can be treated as a subscript expression.

  – Dynamic storage allocation can be achieved.

- The INCLUDE statement is of the form INCLUDE n, LIST where n is the name assigned to a set of FORTRAN statements previously filed with the procedure definition processor which are to be included in the program at this point. The word LIST is optional and if entered, the ''included'' statements will be listed whenever the source program is listed. Thus, a frequently used number of statements (for example, specification statements or a set of internal subprograms) may be added to the source code from an internally available element.

- The IMPLICIT type statement of the form IMPLICIT type (a_1,a_2,a_n), type (b_1, b_2, ...b_n) where type is INTEGER, REAL, LOGICAL, DOUBLE PRECISION or COMPLEX and the a_i represent alphabetic characters or a range of alphabetic characters. The IMPLICIT type statement allows the user to declare the type of variables by specifying that variables beginning with certain designated letters (the a_i) are of a certain type.

- The ENTRY statement is of the form ENTRY name (a_1,a_2, ..., a_n) where name is the name of an entry point and where the a_i are dummy arguments. The entry statement permits an entry to an internal or external subroutine or function by a CALL statement or a function reference to an ENTRY statement. Entry is made at the first executable statement following the ENTRY statement.

- The compile time-interpretive DELETE statement provides the programmer with a simple facility to prevent compilation of a section of source code. It is of the form DELETE n or DELETE n, V where n is a statement label and V is the integer 0 or 1 (or is a name assigned the value 0 or 1 by the PARAMETER statement). V=0 implies that the DELETE statement is not effective while V=1 implies that DELETE is effective.

- FORTRAN V processes double precision quantities in 1108 double precision format, within:

  – The FORTRAN V Compiler

  – Mathematical function routines (where appropriate)

  – The I/O Conversion routines

  – Compiled FORTRAN V programs

- The arithmetic type of the argument to library and intrinsic functions is used by the compiler to determine the correct function routine to be called (namely, SQRT, DSQRT or CSQRT for REAL, DOUBLE PRECISION or COMPLEX arguments, respectively.)

- The 1108 FORMAT Control has been augmented by the addition of new FORMAT control specifications in the following forms:

  Gw.d used for input and output of any of the five types of variables. If the output item is REAL, E or F editing code is used depending on magnitude.

  Tw causes the pointer in an input or output record to point to the w*th* character in the record.

  Lw is logical field specification.

- A ''Master Space'' character (7–8 keypunch) will cause the compiler to ignore all subsequent information on the line. The space thus ignored may be used for comments.

- Hollerith Strings may have the form 'c_1,c_2,..., c_i' where c_i is any Hollerith character, including blank.

- In a typeless expression the computer word (36 bits) is considered as a bit string. Permissible typeless expressions are alphanumeric constants and Boolean functions. The Boolean functions are AND (e_1, e_2), OR(e_1, e_2), BOOL(e_1), COMPL(e_1), and XOR(e_1, e_2).

- An additional input statement of the form READ (unit, format, ERR=n, END=m) is included. The END and ERR clauses can only be indicated in the third or fourth argument positions. Control changes to statement number n if an input error is encountered. Control goes to statement number m if an end-of-file is encountered.

- Miscellaneous Extension

  – Free-field input is specified by an empty FORMAT statement:

    ```
        READ (5,100) A,B,C
    100 FORMAT ( )
    ```

  – In a subroutine or function subprogram the maximum dimension for an array may be transferred as an argument. In FORTRAN V, the information may be provided by COMMON.

  – An array may be dimensioned in an explicit type statement by including the dimension parameters in parentheses.

  – The EDIT statement provides the user the option of suppressing and restoring compiler listings for any part of the program, overriding control card listing options. Valid forms of the statement are:

    ```
    START EDIT SOURCE

    START EDIT CODE

    STOP EDIT SOURCE

    STOP EDIT CODE
    ```

  – FORTRAN V accepts & (2–8 punch) preceding a label as an argument for use in a subroutine or function subprogram to indicate the transmission of a statement label.

  – Data will be accepted in the explicit type statement or DIMENSION statements; for example, REAL A/1. 51/,B(2,3)/6*1.01/.

10.4.2. Compiler Organization

The 1108 FORTRAN V source language processor accepts FORTRAN statements and produces a highly efficient relocatable object code element. FORTRAN, like all other UNIVAC 1108 processors, generates its own code and does not require an assembler pass.

The FORTRAN V compiler is modular and consists of six phases. Although the phases have been separated on the basis of general operations performed on the source program, not every phase processes the entire program.

The compiler, while quite as rapid as a processor, produces an object program optimized with respect to both storage requirements and execution time.

The compilation process involves the successive execution of the six phases summarized in the following list:

Phase 1 transforms the FORTRAN V program source code into an internal format. Files and tables of relational information, implicit in the source program but not easily accessed, are constructed.

Phase 2 deals with storage assignments for variables and performs an analysis of loops.

Phase 3 deals with arithmetic optimization and index register optimization.

Phase 4 deals with loop optimization.

Phase 5 deals with code generation and storage assignment for those quantities not assigned storage by Phase 2.

Phase 6 the final phase, completes the generated instructions in a relocatable binary format and optionally edits all output, including error messages.

The compiler performs several types of optimization on a source program:

- Local Optimization

  This involves the reduction of expressions involving nothing but constants to a single constant.

- Inter-statement Arithmetic Optimization

  This optimization has three forms: (a) the elimination of redundancies in loading of index and arithmetic registers, (b) the recognition of common subexpressions from previous statements, and (c) the removal from a loop of those computations within a loop structure which are constant relative to the loop.

- Inter-statement Indexing Optimization

  This involves a study of the DO-loop structure, entries and exits from loops, the form of subscripts and the loop parameters.

10.5. CONVERSATIONAL FORTRAN V

An important element of the programmed systems support provided with a UNIVAC 1108 System is the Conversational FORTRAN Processor which provides a dynamic and efficient means for constructing, debugging, and modifying a program.

The prominent characteristic of the system is that it enables the user to program from a remote device with a minimum amount of preplanning. He can think freely on line by constructing and testing routines in a nonsequential trail-and-error manner. The time in which the user engages the conversational system is considered a "session". During a session, the system responds on a statement-by-statement basis. Each statement is translated, verified and if desired, executed immediately. Once this is done, the system sends a reply to the device. The user then reacts to the system's message. If he is using the system as a desk calculator, the message will most likely be the result of a requested computation. If he is constructing a program, it may be diagnostic information indicating that the statement contained an error. In either case, the user now converses with the system as to the next step to be taken.

10.5.1. System Features

Some of the features provided by the conversational system are:

- The user has immediate and sustained access to the machine.
- The user has the ability to construct, execute, and alter statements or complete routines; to change values of variables; to rename variables; and to request information selectively.
- The user can store complete routines or portions of routines, take checkpoints during execution of a complex of routines, and load source-statements from optional devices.
- The user may continue his session at the device after an extensive time lapse.
- The user is provided with diagnostic messages and logical analysis to allow modification and debugging to take place at the same level as routine construction.

10.5.2. System Concepts

A session is defined as the time in which the user engages the services of the conversational processor. Normally a session is identified by a user-selected name. During a session, the user is free to use the processor in the manner that is most conducive to the accomplishment of his objective. He may construct, execute, and save single statements, groups of statements, or a complete program.

The environment at the device during the session the user is currently engaged in consists of all those routines needed for solution of a particular problem. These routines can be assembled from either the user's library or they can be system routines provided by the processor.

All conversational operations are performed during the current session at the device. When the user is constructing, executing, testing, or modifying statements, he is considered to be performing operations on an active image. The user has the ability to obtain from storage source statements which then become part of the active image. Only one image may be active at any time at a device.

10.5.3. Conversational FORTRAN Processor and the Executive System

The conversational FORTRAN processor (CFOR) is but one of a number of source-language processors made available to the user by the Executive system. The action it performs is integrated with the multitude of activities controlled by the Executive. For this reason, specific attention was given to the choice of services it renders so as to eliminate duplication of facilities already offered by the system. This is particularly applicable to the service language portion of the programming language. It is assumed that the user will employ the normal job-control language statements of the Executive to perform housekeeping services.

10.5.4. Conversational FORTRAN Language

Conversational FORTRAN consists of procedural statements defined by the UNIVAC 1108 FORTRAN V language and service statements which allow the user to regulate the system during construction, execution, and modification of a program.

The Conversational FORTRAN Compiler analyzes the FORTRAN statements introduced by the user at his terminal and immediately checks these for errors. If there are any, it identifies the error for the user so that it can be corrected. If the FORTRAN statement is without error, it is stored in an intermediate form for interpretive execution at a future time; or the statement may be executed immediately and its result stored for later use.

The Conversational FORTRAN language encompasses the U.S.A. Standards Institute (USASI) FORTRAN. It is also defined by FORTRAN V and any programs constructed by the 1108 Conversational FORTRAN compiler may also be compiled by the batch compiler.

10.6. LIFT, FORTRAN II TO FORTRAN V TRANSLATOR

LIFT is a source language translator which accepts a FORTRAN II source language program as input, performs a translation, and prepares a source language program acceptable to the FORTRAN V Compiler. There is a need for translation since FORTRAN II is not a proper subset of FORTRAN V; that is, there are statement types in FORTRAN II that are not acceptable to FORTRAN V.

There are nine areas of incompatibility between FORTRAN II and FORTRAN V, and the basic purpose of LIFT is to generate FORTRAN V Source Statements which replace the unacceptable FORTRAN II statements.

1. The "F" Card

2. Functions

3. Boolean Statements

4. Double-Precision and Complex Statements

5. COMMON Statements

6. Arithmetic Statement Functions

7. Dimension Statements

8. Hollerith Literals

9. Implicit Multiplication

There are also five types of FORTRAN II statements that, although acceptable to the FORTRAN V processor, are converted to their FORTRAN V equivalents. LIFT offers two features that ease the transfer between computers: the ASSIGN and REPLACE card options. The ASSIGN card allows a temporary change to be made to the I/O Assignment Table, and the REPLACE card allows the user to have every occurrence of a variable name replaced with another variable. The standard output produced by LIFT consists of a listing of the FORTRAN II program, an annotated list of the translated program, and a symbolic program element suitable for use as input to any FORTRAN V Compiler.

10.7. COBOL

The 1108 system provides two COBOL capabilities: an FD ANSI COBOL which complies with the requirements of the ANSI Standards for COBOL document X3.23–1968, and a DOD COBOL.

10.7.1. FD ANSI COBOL

The FD ANSI COBOL compiler provided for the 1108 System contains all the features of the ANSI standard, with the exception of the report writer, plus several useful extensions to the standard. These features are listed in the paragraphs that follow.

10.7.1.1. ANSI Standard COBOL Features

The FD ANSI COBOL compiler contains the following features of the ANSI standard:

■ Nucleus	level 2
■ Table handling	level 3
■ Sequential access	level 2
■ Random access	level 2
■ Sort	level 2
■ Segmentation	level 2
■ Library	level 2

10.7.1.2. Extensions to FD ANSI Standard COBOL

The FD ANSI COBOL compiler of the 1108 System contains the following extensions to the ANSI standard:

- CODASYL – oriented subprogramming feature – uses the CALL and ENTRY verbs with LINKAGE SECTION.
- UNIVAC subprogramming features – provides USE FOR ENTRY POINTS, GO TO, and PERFORM.
- UNIVAC page control features.
- COMMON-STORAGE SECTION feature – for using blank common. The feature may be used with subprogramming to provide communications between COBOL/COBOL and COBOL/FORTRAN programs.
- Indexed sequential file handling capability feature – COBOL verbs using the ISFMS indexed sequential handler.
- A new COBOL file handler featuring superior input/output capabilities.
- Special printer forms handling – feature which provides change forms messages for operator action.
- READY and RESET debugging verbs.
- MONITOR debugging verb feature with the ability to suspend and reactivate its function.
- Multiple receiving fields feature for ADD and SUBTRACT verbs.

10.7.2. DOD COBOL

The DOD COBOL compiler of the 1108 System is based on the Department of Defense publication "COBOL Edition 1965". It accepts statements written in the COBOL language as adapted for the 1108 System and produces an executable program. For further information concerning 1108 COBOL, refer to the "UNIVAC *1106/1108 COBOL Supplementary Reference Manual*" *UP–7626* (current revision).

10.7.2.1. DOD COBOL Features

The major DOD COBOL features implemented on the 1108 System are:

- COMPUTE verb and arithmetic expression in conditional statement
- Table handling (SEARCH)
- Segmentation
- Mass storage
- COPY
- Characters used in arithmetic expression (+, –, *, /, **, =), and in relational expressions(=, >, <)
- Literals up to 132 characters
- The ENTER verb
- The LOCK option on the CLOSE verb
- The ADVANCING option on the WRITE verb
- The REVERSED option of the OPEN verb
- Operands used in arithmetic can be up to 18 digits long
- AND and OR connectors in compound conditions
- Parentheses in compound conditions
- All abbreviations of conditional statements
- The OBJECT-COMPUTER paragraph
- The APPLY clause
- RERUN
- The DATA-COMPILED clause
- Library provisions
- Multiple results from arithmetic verbs

10.7.2.2. Extensions and Special Features of DOD COBOL

The following features, implemented in UNIVAC 1108 COBOL, are special UNIVAC extensions to the COBOL language:

- COBOL subprogram communications
- MONITOR
- Dynamic date
- Common storage
- Page control

A Report Generator, not part of the COBOL processor, is available for COBOL programs.

The following are special features implemented in the UNIVAC 1108 COBOL:

- Segmentation – COBOL programs can be segmented by use of priority numbers on procedural sections.
- Monitor – Provides dynamic program checkout facilities.
- Library – The procedure definition processor is available to store environment, data, and procedure division descriptions so they can be retrieved by the COPY and INCLUDE verb.
- Rerun – The programmer can specify rerun after any number of records have been processed or when an end of reel is encountered.
- Common Storage – Since COBOL programs can be chained (an Executive function), intermediate data results can be maintained between programs using the common storage provision of UNIVAC COBOL. The elements sharing common storage may be from another 1108 processor such as FORTRAN V.
- Overpunched Sign Convention – Tapes and cards prepared in the overpunched sign convention can be processed.

10.7.3. Operational Characteristics

Both COBOL compilers of the 1108 System operate similiarly to any processor under the EXEC 8 System. (A version of the DOD COBOL compiler also operates under the EXEC II operating system.) The compilers accept COBOL source input from a user-specified file and produce relocatable binary output and source output in user-specified files. In addition, several listings may also be specified by the user. These include:

- Source language listings
- Diagnostic messages
- Cross-reference listings
- Object code listings

The compiler diagnostics messages are of two categories:

Warning – A minor source language error has been detected which does not affect the program being produced. This type of diagnostic is identified by the word ERROR preceding the actual message.

Fatal – A major source language error has been detected which very likely will adversely affect the program being produced. The compiler will continue to process the source language but will flag the program produced as being in error. This type of diagnostic is identified by the word ERROR* preceding the actual message.

10.8. ALGOL

The ALGOL language allows the mathematician or engineer to prepare programs for the UNIVAC 1108 without the necessity of becoming familiar with the details of the internal machine operation. The ALGOL compiler then generates, from this pseudo-mathematical source language, efficient coding in a relocatable binary format acceptable to the Executive for execution, the filing system for cataloging and filing, or both.

The basis for the UNIVAC 1108 ALGOL is the "Revised Report on the Algorithmic Language, ALGOL 60" (Communications of the ACM, Vol. 6, January 1963, 1–17).

UNIVAC 1108 ALGOL is an extended hardware representation of ALGOL 60 designed to employ the UNIVAC 1108 processor and associated peripheral equipment efficiently. Certain extensions to basic ALGOL have been made. It can handle the powerful input/output logic; it has the ability to name strings; and it is capable of performing complex and double precision arithmetic.

10.9. SORT/MERGE

The UNIVAC 1108 Sort/Merge package is fully modular with every functional unit completely self-contained. This permits the various units to be individually adapted to their own particular tasks, enabling them to be associated in the most effective form and allowing updating and augmentation.

The package is not a generator of specialized sort/merge routines; rather, the user calls and adapts the independent modules for all his specific sorting needs by presenting his parameter values on control cards at load time.

In the internal sort the replacement selection method, which takes advantage of any inherent sequence in the original data, is used. Strings may be written upon magnetic tape or drum. The FH-432 and FH-1782 drums, because of their high transfer rates and rapid access, minimize processor waiting time and thus greatly speed efficient sort operations. Any random access unit areas may be defined by the supervisor and these are automatically used by the subsystem if an advantage can be gained.

The input data to be sorted may be stored on magnetic tape, punched cards, or magnetic drum. User own coding may be inserted on the first and final passes of the sort/merge operation and may also replace the standard comparison routines. Sorting generally requires the use of two magnetic tape units although additional units can be employed to give faster times.

Keys may be in multiple form and can be recorded, modified, and packed. Standard collating sequences are intrinsically provided for, but the user may define any collating sequences he requires, up to a maximum of seven, and any combination of these may be utilized in the same run. Fixed or variable length items can be handled.

The sort/merge package normally uses 20,000 words of main storage, 262,000 words of magnetic drum storage, and magnetic tape units of any kind as required; but the user may specify more main and drum storage, and additional magnetic tape units to increase efficiency and speed.

10.10. MATHEMATICAL FUNCTION PROGRAMS

The UNIVAC Software System includes an extensive collection of basic mathematical functions and subroutines. This collection includes all of the standard FORTRAN functions and has been expanded to give the programmer a more complete coverage of the often used mathematical routines. Each of these mathematical routines has been carefully developed to offer the programmer maximum accuracy and range with a minimum routine size and executive time. These routines are available to each of the program languages, FORTRAN V, Assembler, COBOL (through the use of the ENTER verb option), and ALGOL. One group of routines, the series of exponentiation routines (NEXPi), is automatically referenced when the FORTRAN V source program indicates exponentiation with the operator, **, and inline exponentiation is not feasible.

The various mathematical function programs are:

- Library Functions
- The FORTRAN Built-in Function
- The Exponentiation Functions

These routines are available to the processors in different manners. For example, the routine SQRT provides a single precision square root of the argument (x). To utilize this routine, the processors employ the following calling sequences:

- FORTRAN V Calling Sequence:
  ROOT = SQRT (X), when X is a FORTRAN V real variable.

- ASSEMBLY Calling Sequence of a FORTRAN V Math Function:

```
a       LMJ    X11,SQRT
a + 1    +     Address of X
a + 2    +     SEQNUM, PRONAM     . WALK BACK WD AND NORMAL RETURN
```

  REAL (single precision, floating point) function values are left in A0 on normal return.

- COBOL Calling Sequence:

  ENTER SQRT REFERENCING X

10.11. APPLICATION PROGRAMS

The UNIVAC 1108 System has an extensive library of application programs and subroutines. The major application programs such as linear programming (LP), automatically programmed tools (APT III), and PERT are briefly described in the following paragraphs while others are simply mentioned by titles. This is by no means an exhaustive list of programs or subroutines. It is meant only to point out the many types of application programs available for UNIVAC 1108 System.

10.11.1. Linear Programming System

Linear programming (LP) has become one of the most useful and frequently used operations research techniques in manufacturing and transportation industries. In the production and distribution of products, LP provides a solution to minimize costs or maximize profits. The LP system developed for the UNIVAC 1108 System embodies a powerful algorithm which employs the "product form of the inverse" method. Further, the algorithm is improved with an advanced path selection technique. The package is coded in FORTRAN V and assembly language.

The more prominent features of the 1108 LP System are as follows:

- The system can accommodate up to 4094 rows and 99,000 columns, with its maximum density a function of the mass storage assigned.

- The use of random access, high speed magnetic drums places this system at a distinct advantage over other LP systems which depend upon slower and nonrandom access devices such as disc and tapes for their operation.

- Both single-precision and double-precision computations are available.

- The control language of this system is a powerful and flexible interpretive control language, containing a large number of commands. The sophisticated user may use macros in constructing his command string. In addition, the solution can be interrupted and parameters changed to optimize the solution path. On the other hand, the average user may still execute his LP problem using only a few basic LP commands.

- SHARE standard format is used for the LP data.
- System parameters may be easily reset.
- LP data can be solved, revised, and resolved in one run. Nonlinear programming can be accomplished by approximating nonlinear functions with linear steps and reoptimizing the problem at each step.

In addition to the above, long LP runs can be split with dump and restart procedures. Also, postoptimal parametric programming or a complete tableau can be obtained. The final output includes the objective function value, optimal basis, vector levels, shadow prices, and reduced costs.

10.11.2. APT III

Automatically programmed tools (APT) is a system for the computer-assisted programming of numerically controlled machine tools: flame cutters, drafting machines, and similar equipment. It is production-oriented, written to take full advantage of numerically controlled techniques in engineering and manufactering with the least expenditure of effort, time, and money.

APT enhances most of the usual advantages found in numerical control: reduced lead time, greater design freedom and flexibility, lower direct costs, greater accuracy, improved production forecasting, lower tooling costs, better engineering control of the manufacturing process, and simplified instruction of changes.

The APT III program represents over 100 man-years of development and testing. After extensive experience with an earlier program, APT II, the Aerospace Industries Association made a new start and wrote APT III from the beginning, during the calendar year 1961. At the completion of this package, APT III was turned over to the Illinois Institute of Technology Research Institute for further development under the APT Long-Range Program (ALRP). Standard versions of APT are available to UNIVAC 1108 installations without payment of any royalty or fee. Participation in IITRI activities and access to their experimental versions of APT require membership in the APT long-range program.

Univac participated in the original writing of APT III and has been a member of the APT long-range program from the beginning. Numerical control specialists are continually working to keep the UNIVAC 1108 APT program in the forefront of the art. As implemented on the UNIVAC 1108 System APT III will continue to conform to the latest APT long-range program specifications.

10.11.3. PERT

The UNIVAC 1108 PERT system is a generalized applications program for project/program planning and control. It consists of both TIME and COST modules. The modular design of this program allows separate processing of the time networks and the cost structure while simultaneously providing for integrated time and cost reporting.

The PERT/TIME module is both activity and event oriented. The input to this module is provided by a deck of cards which describes the network to the system.

These cards are then processed to perform network computations and, if needed, update a TIME master file created by a previous run. As a final step, the PERT/TIME module generates various useful reports, such as an activity report, an event report and a milestone report.

The PERT/COST module is based upon the framework provided by the "DOD/NASA Guide to PERT/COST System Design". The DOD/NASA design is based upon the concept of costing work packages rather than individual network activities. A work package is a discrete unit of work required to complete a specific job or process. The work packages of a research and development project are directly related to activities or groups of activities on the project network. The work package is the basic unit of the PERT/COST system for which actual project costs are collected and compared with estimates for purposes of cost control.

The input to the COST module is a deck of cards describing the work breakdown structure, the actual budgeted and estimated costs for the work packages, and a table of labor and overhead rates. This information is processed to accumulate costs up through a cost breakdown tree and, if needed, is integrated with the TIME output. Required cost reports at desired summarizing levels are then generated.

The PERT system is oriented towards alphanumeric characters. Necessary information such as event codes, charge and summary numbers, responsible organizations, performing organizations, and resource codes can be designated in any arbitrary fashion as a string of alphanumeric characters. Event codes, for example, do not have to be numeric, nor do they have to be in any particular sequence. The program module performs all necessary ordering internally. The system converts all computed times to calendar dates and makes necessary adjustments for vacations and holidays, if any.

10.12. MATH-PACK

MATH-PACK provides the UNIVAC 1108 System with a comprehensive library of 78 fundamental mathematical subprograms coded in FORTRAN V. The purpose of this library is to present to the mathematician, the scientist, and the engineer many of the more frequently used tools of numerical analysis. These subroutines and function subprograms are designed to speed up and simplify solutions to problems encountered in many areas of scientific research.

The subprograms are grouped into fourteen categories:

- Interpolation
- Numerical Integration
- Solution of Equations
- Differentiation
- Polynomial Manipulation
- Matrix Manipulation: Real Matrices
- Matrix Manipulation: Complex Matrices
- Matrix Manipulation: Eigenvalues and Eigenvectors
- Matrix Manipulation: Miscellaneous
- Ordinary Differential Equations
- Systems of Equations
- Curve Fitting
- Pseudo-Random Number Generators
- Specific Functions

Each of these classes contains subroutines and function subprograms that are generally useful for problems commonly encountered by mathematicians, scientists, and engineers.

Appendix D lists all of the MATH-PACK subprograms.

10.13. STAT-PACK

STAT-PACK provides the UNIVAC 1108 System with a comprehensive library of 91 fundamental statistical subprograms coded in FORTRAN V. The purpose of this library is to present to the statistician, the scientist, the operations research specialist, and the engineer many of the more frequently used tools of statistical analysis. These subroutines and function subprograms are designed to speed up the preparation of solutions to statistical problems encountered within many areas of scientific research.

The subprograms are grouped into thirteen categories:

- Descriptive Statistics
- Elementary Population Statistics
- Distribution Fitting and Plotting
- Chi-Square Tests
- Significance Tests
- Confidence Intervals
- Analysis of Variance
- Regression Analysis
- Time Series Analysis
- Multivariate Analysis
- Distribution Functions
- Inverse Distribution Functions
- Miscellaneous

Each of these classes contains subroutines and function subprograms that are generally useful for problems commonly encountered by statisticians, scientists, and engineers.

Appendix E lists all of the STAT-PACK subprograms.

10.14. GENERAL PURPOSE SYSTEM SIMULATOR

The general purpose system simulator (GPSS) is designed for testing and evaluating a system by simulating operations on a symbolic system representation. GPSS is an interpretive program which accepts parametric statements in which the user describes the system to be simulated. These statements specify the actions which are to be taken upon the model.

There are three entity types in GPSS: dynamic, equipment, and statistical. The dynamic entities (transactions) are the units of traffic. Dynamic entities compete for equipment entities within a scheduling model. This model is based on a transaction priority and simulation of intervals relative to scheduled event time. GPSS is an event-oriented simulation.

Equipment entities are of three types: facilities, storages, and logic switches. Facilities process only one transaction during any increment of simulation time. Storages have the property of capacity and are capable of simultaneously processing as many transactions as can be accommodated by their defined capacity. A logic switch is a binary indicator which can be used to record some system condition (physical or logical) that is instrumental in deciding when or how tasks are to be executed.

Statistical entities of GPSS allow the input of numerical information, establish numerical relationships among system variables, and facilitate output of the effects of equipment competitions.

10.15. SIMULA

SIMULA, developed at the Norwegian Computer Center at Oslo under contract with Univac, satisfies the need for studying complex systems stemming from modern technological development. It facilitates the numerous methods for planning, predicting, and decision making in all forms of research and practice.

SIMULA, a simulator based on ALGOL as a subset (they even share a common compiler), provides a language to describe a wide class of phenomena. It provides a programming language for generating (through a compiler) simulation programs which are used to analyze a series of instantaneous events in a complex system (known as a discrete event system) associated with the phenomena. With proper formulation, this definition encompasses most physical/logical systems dependent on time. Control is achieved by the Executive system.

The language is built around a few basic concepts selected to provide the research worker with a standardized approach to a wide class of problems. These concepts are integrated into the language in a way that makes it readable; hence, the language also serves as a useful tool for communication.

A SIMULA program describes a sequence of events rather than a set of permanent relationships. For example, the permanent relationship viewpoint of a store counter for serving a line of customers is that the customers are passive entities acted upon by the store clerk (a series of service completions). The drawback to this viewpoint is that interactions between passive entities cannot be studied (for example, impatient customer steps ahead in line in reaction to slow customers ahead of him). By subordinating the formation of permanent relationships, SIMULA permits the modeling of a great variety of decision rules and interactions between system entities.

The range of variation in decision rules and interactions between system components is so great that it is necessary to let the language contain a general algorithmic language. This is an important reason why ALGOL, which has a compatible block structure, was chosen.

In contrast with analytical methods, which offer more complete and reliable information than the statistical inference from a simulated sample of system runs, simulation enables the study of problems of greater complexity, and both transient and stationary states of systems may be analyzed whereas the analytical approach often is limited to stationary states.

SIMULA's statistical gathering statements have some semiautomatic output reporting routines, and the programmer has all the features of ALGOL available for gathering particular types of statistical simulation data and generating other types of reports.

10.16. FUNCTIONAL MATHEMATICAL PROGRAMMING SYSTEM (FMPS)

The functional mathematical programming system (FMPS) is an advanced methematical programming system designed to operate on the UNIVAC 1108 System. FMPS is designed as a series of procedures which may be called upon by the Executive system. It is particularly suited to:

- Investment planning
- Production scheduling
- Dynamic capital budgeting
- Inventory policy
- Advertising media selection
- Fleet compostion
- Warehouse location
- Allocation of research funds
- Communication network analysis
- Economic modeling
- Corporation modeling

The sequence of operations executed in an FMPS run is controlled by the user through statements written in a user-oriented control language closely resembling FORTRAN. All procedures can be initiated by this control language, and all procedures are available for execution when the FMPS program is loaded. The control language serves the following function:

- initialization and, if desired, modification of tolerances during the execution;
- assignment of input/output devices at the FMPS level;
- preprogramming of the action to be taken in case of exception or error conditions.

FMPS may be used as a self-contained package or as a segment of a user-designed optimization package. The major features of FMPS are described in the following paragraphs.

- Matrix Generator and Report Writer – GAMMA 3

GAMMA 3 is a sophisticated and powerful matrix generator and report writer designed to provide speed and accuracy in handling specialized matrix formats, extraction of data from analyzed solution, and producing useful and understandable reports. With GAMMA 3, the user also has the capability to prepare reports comparing multiple solutions. All details of the interface between GAMMA 3 and the FMPS are handled internally and automatically. The operation of GAMMA 3 is in three phases.

The data definition phase provides for the collection, manipulation, and maintenance of data in the form of lists and data tables. Multifile maintenance capability and display facilities can be used to handle user-prepared data files. Extensive computation capability and conditional logic are included. Built-in error checking is available through such techniques as the use of the empty condition of data cells which have never been loaded.

Problem formulation phase provides for automatic interfacing of data files and generation of problem structures. Extended capabilities such as bounding and ranging, and separable programming are supported as are multiple RHS, cost rows, bounds, and ranges. This phase can prepare either a full matrix or revise set input.

- Linear Programming Algorithm (LP)

The linear programming optimization procedure utilizes the most recent and efficient product form of inverse algorithm. The algorithm includes multiple pricing, upper and lower bounding, and range constraint capabilities. The inversion technique uses a matrix triangularization scheme which is one of the most advanced in the industry.

- Generalized Upper Bounded Algorithm (GUB)

The generalized upper bounded algorithm is designed to solve large linear programming/transportation/weighted distribution models without requiring a separate row in the matrix for each demand.

- Mixed Integer Programming (MIP)

Mixed integer programming is designed to solve mixed integer continous variable problems. MIP allows a stipulation to be entered that certain of the variables take on integer values. This procedure involves a dual decomposition method, linked with a direct search pure integer algorithm. The linear programming run is used to create constraints for the pure integer problem which, in turn, yields output to cause a right-hand side change in the linear program. The process is recursed until either the global optimum is found or a tolerance is met.

- Separable Programming Operating Algorithm (SEP)

  Separable programming algorithm provides the FMPS user with the capability of handling certain types of nonlinear functions. The nonlinearities must comply with the following important restrictions:

  – A nonlinear function in n variables must be separable into the sum of *n* functions, each in terms of only one of these variables, that is:

  $y = f(x_n) = f_1(x_1) + f_2(x_2) + --- + f_n(x_n)$

  – Each of the *n* functions must be representable by a piecewise linear approximation of that function.

- Nonlinear Recursive Procedures

  FMPS is designed to permit the incorporation of nonlinear procedures through recursive matrix modification. In general, a recursive solution method is applied to a problem with nonlinear constraints, nonlinear (concave) objective functions, or both. The problem is assumed to be expressible in linear terms, that is, with linear approximations substituted for designated nonlinear elements. Under these circumstances, the recursion procedure consists of:

  – Solution to the linear problem as an LP problem in the normal fashion.

  – Accessing a user-supplied file of information identifying the matrix elements of concern (that is, those involved in nonlinearity), and designating values to be used in calculating new values of these elements.

  – Determining, according to a procedure specified by the user, the appropriate new values of designated elements using solution activities of pertinent variables and the data supplied by the user.

  – Testing for convergence of the solution through use of user-specified tolerances of the difference between calculated and current element values, or other appropriate criteria.

  – Returning to start if the convergence criteria are not met.

  The cyclic process described will continue until convergence is attained or the number of recursion steps performed exceeds a limit specified by the user.

- Solution Reporting

  Solution reporting permits fully automatic response to changes in problem formulation. All tables and lists from the data definition phase are available as well as all output which the linear program is capable of transmitting. Full computational and logic capabilities exist, permitting the development of calculated results for reporting and the control of reporting procedures based on analysis of data and results. Comprehensive formatting and line control features are included.

The following procedures can be called upon when using the various FMPS modes:

INPUT used to read matrix data from cards or tape

OUTPUT used to display the input or current matrix in various formats

REVISE used to read correction data for modifying the matrix

CRASH used to create an initial basis structure for the current matrix and performing various preliminary validity checks on the matrix

OPTIMIZE and INVERT perform the actual linear programming solution

SOLUTION used for displaying the solution values in various formats

ERRORS displays the computation errors incurred during the solution process for the primal and dual problems

CONDITION prints out the communications region contents

GET allows the user to obtain information about a row or column and to alter his strategy in the control language

BASISOUT provides the facility to punch for file the current basis structure and bounds status

SAVE saves the contents of the communications region, the various internal work areas, and all internal files

BASISIN provides the facility to input a new basis or to modify the existing basis

RESTORE restores the data areas and internal files saved by the procedure SAVE

SCALE provides the facility to perform scaling on the elements of the matrix as well as the right-hand side

PARARHS and PARAOBJ perform post-optimal parametric analysis of the solution with respect to the right-hand side and objective function

RANGE performs post-optimal range analysis

PUBLISH activates a report writer routine to display the solution or input matrix in an application-oriented format

LOADLIST loads a list of ROW labels and/or column labels as selection lists or masks when selective output is desired

REBOUND provides the facility to set bounds on all finite rows contained in the model

LOGCST a general purpose procedure for treatment of nonlinear cost coefficients

APPENDIX A. NOTATIONAL CONVENTIONS

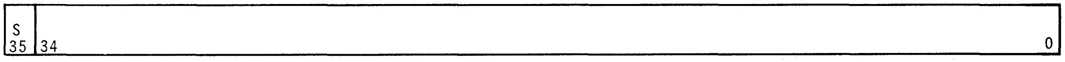
Abbreviations and symbols frequently used in the description of the instruction repertoire are given below:

$(\ )$	Contents of register or address within parentheses.
$(\ )'$	Complement of contents of register or address.
$\lvert(\ )\rvert$	Absolute value or magnitude.
$(\ )_{17-00}$	Subscripts indicate the bit positions involved. A full word is normally not subscripted. Subscripts are also used to designate octal or decimal notation.
( )c	Floating point biased exponent.
( )f	Final contents.
( )i	Initial contents.
( ) m	Floating point fixed point part.
( )j	j-designated portion.
f	Function code.
j	Partial word designator or function code extension.
a	Arithmetic register designator. In input/output instructions, "a" designates an I/O channel.
A	Arithmetic Register.
x	Index register designator.
x_a	Index register designator in a-field.
X	Index Register.
X_a	Index Register specified by coding x_a.
Xm	Modifier portion of an index register.
Xi	Increment portion of an index register.
r	Same as r_a.
r_a	Designator specifying an R Register. It is coded in the a-designator position of an instruction word.
R	R Register.
R_a	R Register specified by coding r_a.

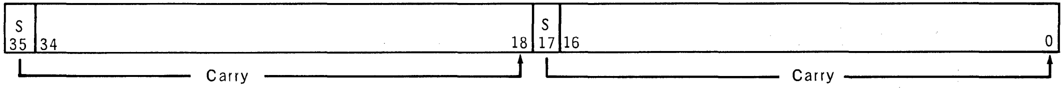
u	The base address of the operand (or the actual operand) as coded in u-field of an instruction.	
U	The effective address or value of the operand after application of indexing and indirect addressing.	
U_d	Destination address.	
U_s	Source address.	
h	h-designator of the instruction word. A value of 1 specifies incrementation of an index register.	
i	i-designator of the instruction word. A value of 1 specifies indirect addressing.	
PSR	Processor State Register.	
BI	I-Base Storage Block Number.	PSR Base Relative Addressing Fields
BS	Program Effective Switch point.	
BD	D-Base Storage Block Number.	
SLR	Storage Limits Register.	
CSR	Channel Select Register.	
P	Program Address Register.	
AND	Symbol denoting logical product, or logical AND.	
OR	Symbol denoting logical sum, or inclusive OR.	
XOR	Symbol denoting logical difference, or exclusive OR.	
→	Direction of data flow.	

APPENDIX B. SUMMARY OF WORD FORMATS

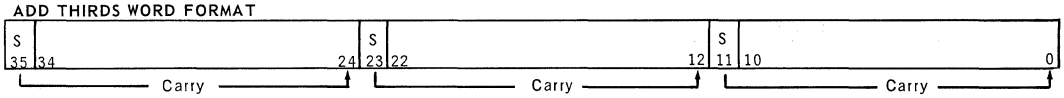
FIXED-POINT MULTIPLY SINGLE INTEGER RESULT



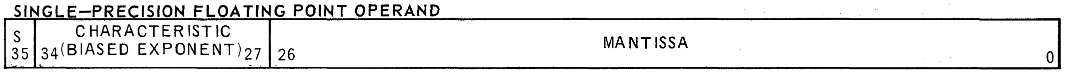
ADD HALVES WORD FORMAT



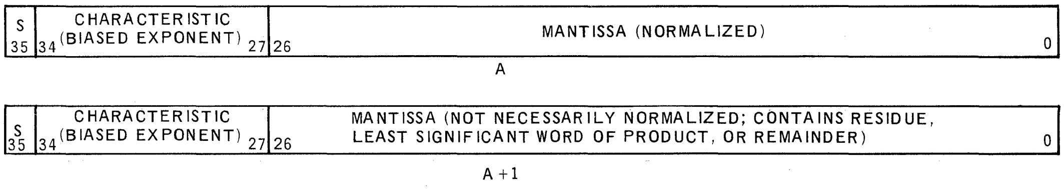
ADD THIRDS WORD FORMAT



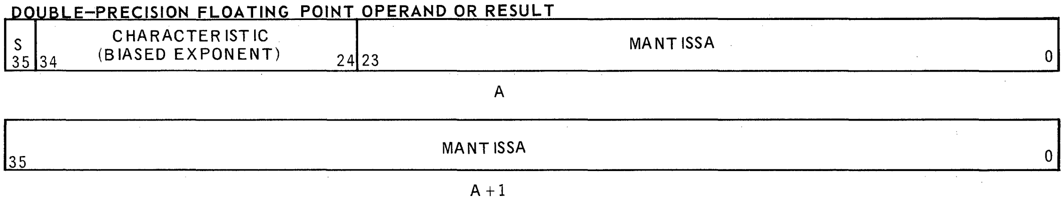
SINGLE–PRECISION FLOATING POINT OPERAND



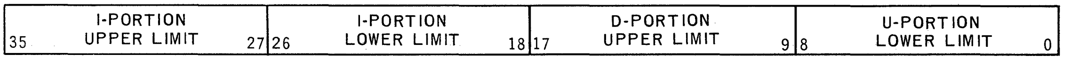
SINGLE–PRECISION FLOATING POINT RESULT



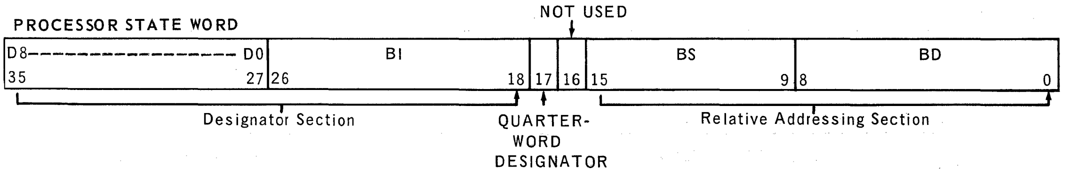
DOUBLE–PRECISION FLOATING POINT OPERAND OR RESULT



STORAGE LIMITS WORD



PROCESSOR STATE WORD



INSTRUCTION WORD

f	j	a	x	h	i	u
35 30	29 26	25 22	21 18	17	16	15 0

INDEX REGISTER WORD

X_i	X_m
35 8	17 0

ISI ACCESS CONTROL WORD

G	W	V
35 34	33 18	17 0

ESI ACCESS CONTROL WORD (half word)

G	H	W	V
35 34	33	32 18	17 0

ESI ACCESS CONTROL WORD (quarter word)

G	H	C	W	V
35 34	33 32	31 30	29 18	17 0

SINGLE-PRECISION FIXED-POINT WORD

S	
35	34 0

DOUBLE-PRECISION FIXED-POINT WORD

S	
35	34 0

A

35 0

A + 1

FIXED-POINT INTEGER MULTIPLY RESULT

S	S	
35	34	33 0

A

35 0

A + 1

FIXED-POINT FRACTIONAL MULTIPLY RESULT

S	
35	34 0

A

	S
35 1	0

A + 1

BIASED ESI VALUES IN IACR'S

BIASED INPUT ESI VALUE	BIASED OUTPUT ESI VALUE
35 18	17 0

APPENDIX C. INSTRUCTION REPERTOIRE BY FUNCTION CODE

Instruction Code (Octal)		Mnemonic	Instruction	Description ②	Execution Time ① in μ sec.
f	j				
00	–	–	Illegal Code	Causes illegal instruction interrupt to address 241_8	–
01	0–15	S,SA	Store A	$(A) \rightarrow U$	.75
02	0–15	SN,SNA	Store Negative A	$-(A) \rightarrow U$	.75
03	0–15	SM,SMA	Store Magnitude A	$\|(A)\| \rightarrow U$	.75
04	0–15	S,SR	Store R	$(R_a) \rightarrow U$	.75
05	0–15	SZ	Store Zero	ZEROS → U	.75
06	0–15	S,SX	Store X	$(X_a) \rightarrow U$	.75
07	–	–	Illegal Code	Causes illegal instruction interrupt to address 241_8	–
10	0–17	L,LA	Load A	$(U) \rightarrow A$	.75
11	0–17	LN,LNA	Load Negative A	$-(U) \rightarrow A$	.75
12	0–17	LM,LMA	Load Magnitude A	$\|(U)\| \rightarrow A$	.75
13	0–17	LNMA	Load Negative Magnitude A	$-\|(U)\| \rightarrow A$	.75
14	0–17	A,AA	Add to A	$(A) + (U) \rightarrow A$	.75
15	0–17	AN,ANA	Add Negative to A	$(A) - (U) \rightarrow A$	.75
16	0–17	AM,AMA	Add Magnitude to A	$(A) + \|(U)\| \rightarrow A$	.75
17	0–17	ANM,ANMA	Add Negative Magnitude to A	$(A) - \|(U)\| \rightarrow A$	.75
20	0–17	AU	Add Upper	$(A) + (U) \rightarrow A + 1$	.75
21	0–17	ANU	Add Negative Upper	$(A) - (U) \rightarrow A + 1$	.75
22	0–17	BT	Block Transfer	$(X_x+u) \rightarrow X_a+u$, repeat K times	2.25 + 1.5K always
23	0–17	L,LR	Load R	$(U) \rightarrow R_a$	.75
24	0–17	A,AX	Add to X	$(X_a) + (U) \rightarrow X_a$	.75
25	0–17	AN,ANX	Add Negative to X	$(X_a) - (U) \rightarrow X_a$	.75
26	0–17	LXM	Load X Modifier	$(U) \rightarrow X_{a_{17-0}}$; $X_{a_{35-18}}$ unchanged	.875

Instruction Code (Octal)		Mnemonic	Instruction	Description[2]	Execution Time[1] in μ sec.
f	i				
27	0–17	L,LX	Load X	$(U) \rightarrow X_a$	.75
30	0–17	MI	Multiply Integer	$(A) \cdot (U) \rightarrow A, A+1$	2.375
31	0–17	MSI	Multiply Single Integer	$(A) \cdot (U) \rightarrow A$	2.375
32	0–17	MF	Multiply Fractional	$(A) \cdot (U) \rightarrow A, A+1$	2.375
33	–	–	Illegal Code	Causes illegal instruction interrupt to address 241_8	–
34	0–17	DI	Divide Integer	$(A, A+1) \div (U) \rightarrow A$; REMAINDER → A + 1	10.125
35	0–17	DSF	Divide Single Fractional	$(A) \div (U) \rightarrow A+1$	10.125
36	0–17	DF	Divide Fractional	$(A, A+1) \div (U) \rightarrow A$; REMAINDER → A + 1	10.125
37	–	–	Illegal Code	Causes illegal instruction interrupt to address 241_8	–
40	0–17	OR	Logical OR	(A) OR (U) → A + 1	.75
41	0–17	XOR	Logical Exclusive OR	(A) XOR (U) → A + 1	.75
42	0–17	AND	Logical AND	(A) AND (U) → A + 1	.75
43	0–17	MLU	Masked Load Upper	[(U) AND (R2)] OR [(A) AND (R2)] → A + 1	.75
44	0–17	TEP	Test Even Parity	Skip NI if (U) AND (A) has even parity	2.00/1.25
45	0–17	TOP	Test Odd Parity	Skip NI if (U) AND (A) has odd parity	2.00/1.25
46	0–17	LXI	Load X Increment	$(U) \rightarrow X_{a_{35\text{-}18}}$; $X_{a_{17\text{-}0}}$ unchanged	1.00
47	0–17	TLEM TNGM	Test Less Than or Equal to Modifier Test Not Greater Than Modifier	Skip NI if $(U)_{17\text{-}0} < (X_a)_{17\text{-}0}$; always $(X_a)_{17\text{-}0} + (X_a)_{35\text{-}18} \rightarrow (X_a)_{17\text{-}0}$	1.75/1.00
50	0–17	TZ	Test Zero	Skip NI if $(U) = \pm 0$	1.625/ .875
51	0–17	TNZ	Test Nonzero	Skip NI if $(U) \neq \pm 0$	1.625/ .875
52	0–17	TE	Test Equal	Skip NI if $(U) = (A)$	1.625/ .875
53	0–17	TNE	Test Not Equal	Skip NI if $(U) \neq (A)$	1.625/ .875
54	0–17	TLE TNG	Test Less Than or Equal Test Not Greater	Skip NI if $(U) \leq (A)$	1.625/ .875
55	0–17	TG	Test Greater	Skip NI if $(U) > (A)$	1.625/ .875
56	0–17	TW	Test Within Range	Skip NI if $(A) < (U) \leq (A+1)$	1.75 /1.00

Instruction Code (Octal)		Mnemonic	Instruction	Description[2]	Execution Time[1] in μ sec.
f	j				
57	0–17	TNW	Test Not Within Range	Skip NI if $(U) < (A)$ or $(U) > (A+1)$	1.75 /1.00
60	0–17	TP	Test Positive	Skip NI if $(U)_{35} = 0$	1.50 / .75
61	0–17	TN	Test Negative	Skip NI if $(U)_{35} = 1$	1.50 / .75
62	0–17	SE	Search Equal	Skip NI if $(U) = (A)$, else repeat	2.25 + .75 always
63	0–17	SNE	Search Not Equal	Skip NI if $(U) \neq (A)$, else repeat	2.25 + .75 always
64	0–17	SLE SNG	Search Less Than or Equal Search Not Greater	Skip NI if $(U) \leq (A)$, else repeat	2.25 + .75K always
65	0–17	SG	Search Greater	Skip NI if $(U) > (A)$, else repeat	2.25 + .75K always
66	0–17	SW	Search Within Range	Skip NI if $(A) < (U) \leq (A+1)$, else repeat	2.25 + .75K always
67	0–17	SNW	Search Not Within Range	Skip NI if $(U) \leq (A)$ or $(U) \geq (A+1)$, else repeat	2.25 + .75K always
70	[3]	JGD	Jump Greater and Decrement	Jump to U if (Control Register) ja > 0;go to NI if (Control Register)ja ≤ 0; always (Control Register)ja - 1 → Control Register ja	1.50/ .75 always
71	00	MSE	Mask Search Equal	Skip NI if (U) AND (R2) = (A) AND (R2), else repeat	2.25 + .75K always
71	01	MSNE	Mask Search Not Equal	Skip NI if (U) AND (R2) ≠ (A) AND (R2), else repeat	2.25 + .75K always
71	02	MSLE MSNG	Mask Search Less Than or Equal Mask Search Not Greater	Skip NI if (U) AND (R2) ≤ (A) AND (R2), else repeat	2.25 + .75K always
71	03	MSG	Mask Search Greater	Skip NI if (U) AND (R2) > (A) AND (R2), else repeat	2.25 + .75K always
71	04	MSW	Masked Search Within Range	Skip NI if (A) AND (R2) < (U) AND (R2) ≤ (A+1) AND (R2), else repeat	2.25 + .75K always
71	05	MSNW	Masked Search Not Within Range	Skip NI if (U) AND (R2) ≤ (A) AND (R2) or (U) AND (R2) > (A+1) AND (R2), else repeat	2.25 + .75K always

Instruction Code (Octal)		Mnemonic	Instruction	Description[2]	Execution Time[1] in μ sec.
f	i				
71	06	MASL	Masked Alphanumeric Search Less Than or Equal	Skip NI if (U) AND (R2) $\leq$ (A) AND (R2), else repeat	2.25 + .75K always
71	07	MASG	Masked Alphanumeric Search Greater	Skip NI if (U) AND (R2) > (A) AND (R2), else repeat	2.25 + .75 K always
71	10	DA	Double Precision Fixed Point Add	(A,A+1) + (U,U+1) → A,A+1	1.625
71	11	DAN	Double Precision Fixed Point Add Negative	(A,A+1) – (U,U+1) → A,A+1	1.625
71	12	DS	Double Store A	(A,A+1) → U,U+1	1.50
71	13	DL	Double Load A	(U,U+1) → A,A+1	1.50
71	14	DLN	Double Load Negative A	–(U,U+1) → A,A+1	1.50
71	15	DLM	Double Load Magnitude A	$\lvert(U,U+1)\rvert \rightarrow A,A+1$	1.50
71	16	DJZ	Double Precision Zero Jump	Jump to U if (A,A+1) = ±0; go to NI if (A,A+1) $\neq\pm$ 0	1.625/ .875 always
71	17	DTE	Double Precision Test	Skip NI if (U,U+1) = (A,A+1)	2.375/1.625
72	00	–	Illegal Code	Causes illegal instruction interrupt to address 241_8	–
72	01	SLJ	Store Location and Jump	(P) – Base Address Modifier [BI or BD] → U_{17-0}; jump to U+1	2.125 always
72	02	JPS	Jump Positive and Shift	Jump to U of $(A)_{35}$= 0; go to NI if $(A)_{35}$= 1; always shift (A) left circularly one bit position	1.50 / .75 always
72	03	JNS	Jump Negative and Shift	Jump to U if $(A)_{35}$=1, go to NI if $(A)_{35}$=0; always shift (A) left circularly one bit position	1.50 / .75 always
72	04	AH	Add Halves	$(A)_{35-18} + (U)_{35-18} \rightarrow A_{35-18}$ $(A)_{17-0} + (U)_{17-0} \rightarrow (A)_{17-0}$;	.75
72	05	ANH	Add Negative Halves	$(A)_{35-18} - (U)_{35-18} \rightarrow A_{35-18}$ $(A)_{17-0} - (U)_{17-0} \rightarrow A_{17-0}$;	.75
72	06	AT	Add Thirds	$(A)_{35-24} + (U)_{35-24} \rightarrow A_{35-24}$; $(A)_{23-12} + (U)_{23-12} \rightarrow A_{23-12}$; $(A)_{11-0} + (U)_{11-0} \rightarrow A_{11-0}$	.75
72	07	ANT	Add Negative Thirds	$(A)_{35-24} - (U)_{35-24} \rightarrow A_{35-24}$; $(A)_{23-12} - (U)_{23-12} \rightarrow A_{23-12}$; $(A)_{11-0} - (U)_{11-0} \rightarrow A_{11-0}$	.75
72	10	EX	Execute	Execute the instruction at U	.75 always
72	11	ER	Executive Return	Causes executive return interrupt to address 242_8	1.375 always

Instruction Code (Octal)		Mnemonic	Instruction	Description[2]	Execution Time[1] in μ sec.
f	j				
72	12	–	Illegal Code	Causes illegal instruction interrupt to address 241_8	–
72	13	PAIJ	Prevent All I/O Interrupts and Jump	Prevent all I/O interrupts and jump to U	.75 always
72	14	SCN	Store Channel Number	If a = 0: CHANNEL NUMBER → U_{3-0}; If a = 1: CHANNEL NUMBER → U_{3-0} and CPU NUMBER → U_{5-4}	.75
72	15	LPS	Load Processor State Register	(U) → PSR	.75
72	16	LSL	Load Storage Limits Register	(U) → SLR	.75
72	17	–	Illegal Code	Causes illegal instruction interrupt to address 241_8	–
73	00	SSC	Single Shift Circular	Shift (A) right circularly U places	.75 always
73	01	DSC	Double Shift Circular	Shift (A,A+1) right circularly U places	.875 always
73	02	SSL	Single Shift Logical	Shift (A) right U places; zerofill	.75 always
73	03	DSL	Double Shift Logical	Shift (A,A+1) right U places; zerofill	.875 always
73	04	SSA	Single Shift Algebraic	Shift (A) right U places; signfill	.75 always
73	05	DSA	Double Shift Algebraic	Shift (A,A+1) right U places; signfill	.875 always
73	06	LSC	Load Shift and Count	(U) → A; shift (A) left circularly until $(A)_{35} \neq (A)_{34}$; NUMBER OF SHIFTS → A + 1	1.125
73	07	DLSC	Double Load Shift and Count	(U,U+1) → A,A+1; shift (A,A+1) left circularly until $(A,A+1)_{71} \neq (A,A+1)_{70}$; NUMBER OF SHIFTS → A+2	2.125
73	10	LSSC	Left Single Shift Circular	Shift (A) left circularly U places	.75 always
73	11	LDSC	Left Double Shift Circular	Shift (A,A+1) left circularly U places	.875 always
73	12	LSSL	Left Single Shift Logical	Shift (A) left U places; zerofill	.75 always
73	13	LDSL	Left Double Shift Logical	Shift (A,A+1) left U places; zerofill	.875 always

Instruction Code (Octal)		Mnemonic	Instruction	Description[2]	Execution Time[1] in μ sec.
f	i				
73	14	III (a=0 or 1)	Initiate Interprocessor Interrupt	Initiate Interprocessor Interrupt	.75 always
		ALRM ($a=10_8$)	Alarm	Turn on alarm	.75 always
		EDC ($a=11_8$)	Enable Day Clock	Enable day clock	.75 always
		DDC ($a=12_8$)	Disable Day Clock	Disable day clock	.75 always
73	15	SIL	Select Interrupt Location	$(A)_{2-0} \rightarrow$ MSR	.75 always
73	16	LCR(a=0)	Load Channel Select Register	$(U)_{3-0} \rightarrow$ CSR	.875
		LLA(a=1)	Load Last Address Register	$(U)_{2-0} \rightarrow$ LAR	.875
73	17	TS	Test and Set	If $(U)_{30} = 1$, interrupt to address 244_8; if $(U)_{30} = 0$, go to NI; always $01_8 \rightarrow U_{35-30}$; $(U)_{29-0}$ unchanged	Alternate bank: 1.625 interrupt .875 NI Same bank: 2.0 interrupt 2.0 NI
74	00	JZ	Jump Zero	Jump to U if $(A) = \pm 0$; go to NI if $(A) \neq \pm 0$	1.50 / .75 always
74	01	JNZ	Jump Nonzero	Jump to U if $(A) \neq \pm 0$; go to NI if $(A) = \pm 0$	1.50 / .75 always
74	02	JP	Jump Positive	Jump to U if $(A)_{35} = 0$; go to NI if $(A)_{35} = 1$	1.50 / .75 always
74	03	JN	Jump Negative	Jump to U if $(A)_{35} = 1$; go to NI if $(A)_{35} = 0$	1.50 / .75 always
74	04	JK J	Jump Keys Jump	Jump to U if a = 0 or if a = lit SELECT JUMPS indicator; go to NI if neither is true	.75 always
74	05	HKJ HJ	Halt Keys and Jump Halt and Jump	Stop if a = 0 or if [a AND lit SELECT STOPS indicators] $\neq 0$; on restart or continuation, jump to U	.75 always
74	06	NOP	No Operation	Proceed to next instruction	.75 always
74	07	AAIJ	Allow All I/O Interrupts and Jump	Allow all I/O interrupts and jump to U	.75 always
74	10	JNB	Jump No Low Bit	Jump to U if $(A)_0 = 0$; go to NI if $(A)_0 = 1$	1.50 / .75 always
74	11	JB	Jump Low Bit	Jump to U if $(A)_0 = 1$; go to NI if $(A)_0 = 0$	1.50 / .75 always
74	12	JMGI	Jump Modifier Greater and Increment	Jump to U if $(X_a)_{17-0} > 0$; go to NI if $(X_a)_{17-0} \leq 0$; always $(X_a)_{17-0} + (X_a)_{35-18} \rightarrow X_{a\,17-0}$	1.625/ .75 always

Instruction Code (Octal)		Mnemonic	Instruction	Description[2]	Execution Time[1] in μ sec.
f	i				
74	13	LMJ	Load Modifier and Jump	(P) – Base Address Modifier [BI or BD] → $X_{a_{17-0}}$; jump to U	.875 always
74	14	JO	Jump Overflow	Jump to U if D1 of PSR = 1; go to NI if D1 = 0	1.50 / .75 always
74	15	JNO	Jump No Overflow	Jump to U if D1 of PSR = 0; go to NI if D1 = 1	1.50 / .75 always
74	16	JC	Jump Carry	Jump to U if D0 of PSR = 1; go to NI if D0 = 0	1.50 / .75 always
74	17	JNC	Jump No Carry	Jump to U if D0 of PSR = 0; go to NI if D0 = 1	1.50 / .75 always
75	00	LIC	Load Input Channel	For channel [a OR CSR]: (U) → IACR; set input active; clear input monitor	.75
75	01	LICM	Load Input Channel and Monitor	For channel [a OR CSR]: (U) → IACR; set input active; set input monitor	.75
75	02	JIC	Jump Input Channel Busy	Jump to U if input active is set for channel [a OR CSR]; go to NI if input active is clear	.75 always
75	03	DIC	Disconnect Input Channel	For channel [a OR CSR]: clear input active; clear input monitor	.75 always
75	04	LOC	Load Output Channel	For channel [a OR CSR]: (U) → OACR; set output active clear output monitor; clear external function (ISI only)	.75
75	05	LOCM	Load Output Channel and Monitor	For channel [a OR CSR]: (U) → OACR; set output active; set output monitor; clear external function (ISI only)	.75
75	06	JOC	Jump Output Channel Busy	Jump to U if output active is set for channel [a OR CSR]; go to NI if output active is clear	.75 always
75	07	DOC	Disconnect Output Channel	For channel [a OR CSR]: clear output active; clear output monitor; clear external function	.75 always
75	10	LFC	Load Function in Channel	For channel [a OR CSR]: (U) → OACR; set output active (ISI only), external function, and force external function; clear output monitor (ISI only)	.75
75	11	LFCM	Load Function in Channel and Monitor	For channel [a OR CSR]: (U) → OACR; set output active (ISI only), external function, force external function, and output monitor (ISI only)	.75

Instruction Code (Octal)		Mnemonic	Instruction	Description[2]	Execution Time[1] in μ sec.
f	i				
75	12	JFC	Jump Function in Channel	Jump to U if force external function is set for channel [a OR CSR]; go to NI if force external function is clear	.75 always
75	13	–	Illegal Code	If guard mode is set, causes guard mode interrupt to address 243_8. If guard mode is not set, same as NOP.	.75 always
75	14	AACI	Allow All Channel External Interrupts	Allow all external interrupts	.75 always
75	15	PACI	Prevent All Channel External Interrupts	Prevent all external interrupts	.75 always
75	16	–	Illegal Code	If guard mode is set causes guard mode interrupt to address 243_8. If guard mode is not set, same as NOP	.75 always
75	17	–	Illegal Code		
76	00	FA	Floating Add	$(A) + (U) \rightarrow A$; RESIDUE $\rightarrow A + 1$	1.875
76	01	FAN	Floating Add Negative	$(A) - (U) \rightarrow A$; RESIDUE $\rightarrow A + 1$	1.875
76	02	FM	Floating Multiply	$(A) \cdot (U) \rightarrow A, A + 1$	2.625
76	03	FD	Floating Divide	$(A) \div (U) \rightarrow A$; REMAINDER $\rightarrow A + 1$	8.25 [4]
76	04	LUF	Load and Unpack Floating	$\lvert(U)\rvert_{34-27} \rightarrow A_{7-0}$, zerofill; $(U)_{26-0} \rightarrow A + 1_{26-0}$, signfill	.75 always
76	05	LCF	Load and Convert to Floating	$(U)_{35} \rightarrow A+1_{35}$; [NORMALIZED $(U)]_{26-0} \rightarrow A+1_{26-0}$; if $(U)_{35}=0$, $(A)_{7-0} \pm$ NORMALIZING COUNT $\rightarrow A+1_{34-27}$; if $(U)_{35}=1$, ones complement of $[(A)_{7-0} \pm$ NORMALIZING COUNT$] \rightarrow A+1_{34-27}$	1.125
76	06	MCDU	Magnitude of Characteristic Difference to Upper	$\lvert\lvert(A)_{35-27}\rvert - \lvert(U)\rvert_{35-27}\rvert \rightarrow A+1_{8-0}$; ZEROS $\rightarrow A+1_{35-9}$	.75
76	07	CDU	Characteristic Difference to Upper	$\lvert(A)\rvert_{35-27} - \lvert(U)\rvert_{35-27} \rightarrow A+1_{8-0}$; SIGN BITS $\rightarrow A+1_{35-9}$	.75
76	10	DFA	Double Precision Floating Add	$(A,A+1) + (U,U+1) \rightarrow A,A+1$	2.625
76	11	DFAN	Double Precision Floating Add Negative	$(A,A+1) - (U,U+1) \rightarrow A,A+1$	2.625
76	12	DFM	Double Precision Floating Multiply	$(A,A+1) \cdot (U,U+1) \rightarrow A,A+1$	4.25
76	13	DFD	Double Precision Floating Divide	$(A,A+1) \div (U,U+1) \rightarrow A,A+1$	17.25 [5]

Instruction Code (Octal)		Mnemonic	Instruction	Description ②	Execution Time① in μ sec.
f	i				
76	14	DFU	Double Load and Unpack Floating	$\lvert(U)\rvert_{34\text{-}24} \rightarrow A_{10\text{-}0}$, zerofill; $(U)_{23\text{-}0} \rightarrow A{+}1_{23\text{-}0}$, signfill; $(U{+}1) \rightarrow A{+}2$	1.50
76	15	DFP	Double Load and Convert To Floating	$(U)_{35} \rightarrow A{+}1_{35}$; [NORMALIZED $(U,U{+}1)]_{59\text{-}0} \rightarrow A{+}1_{23\text{-}0}$ and A+2. If $(U)_{35} = 0$, $(A)_{10\text{-}0}$ ±NORMALIZING COUNT → $A{+}1_{34\text{-}29}$. If $(U)_{35} = 1$, ones complement of $[(A)_{10\text{-}0}$ ±NORMALIZING COUNT] → $A{+}1_{34\text{-}24}$	2.125
76	16	FEL	Floating Expand and Load	If $(U)_{35} = 0$, $(U)_{35\text{-}27} + 1600_8 \rightarrow A_{35\text{-}24}$; if $(U)_{35} = 1$, $(U)_{35\text{-}27} - 1600_8 \rightarrow A_{35\text{-}24}$; $(U)_{26\text{-}3} \rightarrow A_{23\text{-}0}$; $(U)_{2\text{-}0} \rightarrow A{+}1_{35\text{-}33}$; $(U)_{35} \rightarrow A{+}1_{32\text{-}0}$	1.00
76	17	FCL	Floating Compress and Load	If $(U)_{35} = 0$, $(U)_{35\text{-}24} - 1600_8 \rightarrow A_{35\text{-}27}$; if $(U)_{35} = 1$, $(U)_{35\text{-}24} + 1600_8 \rightarrow A_{35\text{-}27}$; $(U)_{23\text{-}0} \rightarrow A_{26\text{-}3}$; $(U{+}1)_{35\text{-}33} \rightarrow A_{2\text{-}0}$	1.625
77	0–17	–	Illegal Code	Causes illegal instruction interrupt to address 241_8	–

NOTES:

① The execution times given are for alternate bank memory access; for same bank memory access, execution time is .75 microseconds greater. Exceptions to this either show the execution times for both types of memory access or include the word "always" to indicate that the execution time is the same regardless of the type of memory access.

For function codes 01 through 06 and 22, add .375 microseconds to the execution times for 6-bit and 12-bit writes.

The execution time for Block Transfer and search instructions depend on the number of repetitions (K) required, that is, the number of words in the block being transferred or the number of words searched before a find is made.

For function codes 10 through 21, 23 through 32, 34 through 36, and 40 through 61:
if j = 16 or 17, the address of U, instead of the contents of U, is used. No operand search is made to main storage. The execution time without overlap is the same as with overlap.

② NI stands for next instruction.

③ The a and j fields together serve to specify any of the 128 control registers.

④ If 28 rather than 27 subtractions are performed, add .25 microseconds to the execution time.

⑤ If 61 rather than 60 subtractions are performed, add .25 microseconds to the execution time.

APPENDIX D. MATH-PACK ROUTINES

The following is a complete listing of MATH-PACK routines:

1. INTERPOLATION

   GNINT – Gregory-Newton Interpolation

   GNEXT – Gregory-Newton Extrapolation

   GNPOL – Gregory-Newton Polynomial Evaluation

   BESINT – Bessel Interpolation

   STINT – Stirling Interpolation

   CDINT – Gauss Central-Difference Interpolation

   AITINT – Aitken Interpolation

   YLGINT – Lagrange Interpolation

   SPLN1, SPLN2 – Spline Interpolation

2. NUMERICAL INTEGRATION

   TRAPNI – Trapezoidal Rule

   SIM1NI – Simpson 1/3 Rule

   SIM3NI – Simpson 3/8 Rule

   STEPNI – Variable Step Integration

   GENNI – Generalized Numerical Quadrature

   DOUBNI – Double Integration

   LGAUSS – Gauss Quadrature Abscissas and Weights

   SIMPTS – Simpson 1/3 Rule Abscissas and Weights

3. SOLUTION OF EQUATIONS

   NEWTIT – Newton-Raphson Iteration

   WEGIT – Wegstein Iteration

   AITIT – Aitken Iteration

   ROOTCP – Real and Complex Roots of Real or Complex Polynomial

4. DIFFERENTIATION

DERIV1 – First Derivative Approximation

DERIV2 – Second Derivative Approximation

NTHDER – Nth Derivative of a Polynomial

5. POLYNOMIAL MANIPULATION

GIVZRS – Polynomial Coefficients Given Its Zeros

CVALUE – Complex Polynomial Evaluation

POLYX – Real Polynomial Multiplication

CPOLYX – Complex Polynomial Multiplication

6. MATRIX MANIPULATION: REAL MATRICES

MXADD – Matrix Addition

MXSUB – Matrix Subtraction

MXTRN – Matrix Transposition

MXSCA – Matrix Multiplication by Scalar

MXMLT – Matrix Multiplication

MXMDIG – Matrix Multiplication by Diagonal Matrix Stored as a Vector

GJR – Determinant; Inverse; Solution of Simultaneous Equations

MXHOI – Inverse Accuracy Improvement

7. MATRIX MANIPULATION: COMPLEX MATRICES

CMXADD – Matrix Addition

CMXSUB – Matrix Subtraction

CMXTRN – Matrix Transposition

CMXSCA – Matrix Multiplication by Scalar

CMXMLT – Matrix Multiplication

CGJR – Determinant; Inverse; Solution of Simultaneous Equations

8. MATRIX MANIPULATION EIGENVALUES AND EIGENVECTORS

TRIDMX – Tridiagonalization of Real Symmetric Matrix

EIGVAL – Eigenvalues of Tridiagonal Matrix by Sturm Sequences

EIGVEC – Eigenvectors of Tridiagonal Matrix by Wilkinson's Method

9. MATRIX MANIPULATION: MISCELLANEOUS

DGJR – Double-Precision Determinant; Inverse; Solution of Simultaneous Equations

PMXTRI – Polynomial Matrix Triangularization

SCALE – Polynomial Matrix Scaling

MXROT – Matrix Rotation

10. ORDINARY DIFFERENTIAL EQUATIONS

EULDE – Euler's Method

HAMDE – Hamming's Method

INVAL – Initial Values for Differential Equation Solution

RKDE – Runge-Kutta Method

SODE – Second-Order Equations

MRKDE – Reduction of Mth-Order System to System of m First-Order Equations

11. SYSTEMS OF EQUATIONS

JACMX – Jacobi Iteration to Determine Eigenvalues and Eigenvectors of Symmetric Matrix

HJACMX – Jacobi Iteration to Determine Eigenvalues and Eigenvectors of Hermitian Matrix

LSIMEQ – Solution to a Set of Linear Simultaneous Equations

NSIMEQ – Functional Iteration to Determine Solution to Set of Non-Linear Equations

12. CURVE FITTING

CFSRIE – Coefficients of Fourier Series on a Continuous Range

FTRANS – Fourier Transform

DFSRIE – Coefficients of Fourier Series on Discrete Range

FITD – Fitted Value and Derivative Values for a Least-Squares Polynomial

ORTHLS – Orthogonal Polynomial Least-Squares Curve-Fitting

FITY – Fitted Values for a Least-Squares Polynomial

COEFS – Coefficients of a Least-Squares Polynomial

13. PSUEDO-RANDOM NUMBER GENERATORS

NRAND – Interval $(0,2^{27})$ Generator

RANDU – Uniform Distribution

RANDN – Normal Distribution

RANDEX – Exponential Distribution

14. SPECIFIC FUNCTIONS

BSSL – Zero- and First-Order Bessel Functions

BESJ – Regular Bessel Functions of Real Argument

BESY – Irregular Bessel Functions of Real Argument

BESI – Regular Bessel Functions of Imaginary Argument

BESK – Irregular Bessel Functions of Imaginary Argument

GAMMA – Gamma Function Evaluation

LEGEN – Legendre Polynomial Evaluation

ARCTNQ – Arctangent of a Quotient

APPENDIX E. STAT-PACK ROUTINES

The following is a complete listing of STAT-PACK routines:

1. DESCRIPTIVE STATISTICS

   FREQP – Frequency Polygon

   HIST – Histogram

   MHIST – Multivariate Histogram

   GROUP – Grouping of Data

2. ELEMENTARY POPULATION STATISTICS

   AMEAN – Arithmetic Mean

   GMEAN – Geometric Mean

   HMEAN – Harmonic Mean

   MEDIAN – Median

   MODE – Mode

   QUANT – Quantiles

   OGIVE – Distribution Curve

   IQRNG – Interpercentile Range

   RANGE – Range

   MNDEV – Mean Deviation

   STDEV – Standard Deviation

   CVAR – Coefficient of Variation

   ORDER – Order and Rank Statistics

   CMONT – Central Moments

   AMONT – Absolute Moments

   CUMLT – Cumulants

   SHPCOR – Sheppard's Corrections

   KURSK – Skewness and Kurtosis

3. DISTRIBUTION, FITTING, AND PLOTTING

BINOM – Binomial Distribution

POISON – Poisson Distribution

HYPER – Hypergeometric Distribution

PNORM – Normal Distribution

AFSER – Arne Fisher Series

4. CHI-SQUARE TESTS

CHI21S – CHI-Square Test of Sample Proportion - One Sample

CHI2JS – CHI-Square Test of Sample Proportion - J Samples

CHI2P – CHI-Square Test of Fit to Poisson Distribution

CHI2N – CHI-Square Test of Normality

CHISAM – CHI-Square Test of Homogeniety

CHICNT – CHI-Square Test for Independence

GENGOF – CHI-Square Test of General Goodness of Fit

5. SIGNIFICANCE TESTS

SIGPRP – Test of Significance of Proportion of Successes

SIGMN – Test of Significance of a Mean

SIGDMN – Test of Significance of the Difference Between Two Means

SIGDVR – Test of Significance of the Ratio Between Two Variances

6. CONFIDENCE INTERVALS

CFDMKV – Confidence Interval for the Mean; Known Variance

CFDMUV – Confidence Interval for the Mean; Unknown Variance

CFDMSU – Confidence Interval for the Difference Between Two Means

CFDVAR – Confidence Interval for Variance

TOLINT – Tolerance Intervals

7. ANALYSIS OF VARIANCE

ANOV1 – One-Way Cross Classification

ANOV2 – Two-Way Cross Classification

ANOV3 – Three-Way Cross Classification

MISDAT – Missing Data

VTRANS – Variable Transformations

ANOVRB – Randomized Blocks

ANOVLS – Latin Squares

ANOVSP – Split Plot Design

ANOSSP – Split-Split Plot Design

ANOVN2 – Two-Way Nested Design

ANOVN3 – Three-Way Nested Design

ANOCO – Analysis of Covariance

GLH – General Linear Hypotheses

8. REGRESSION ANALYSIS

RESTEM – Stepwise Multiple Regression

REBSOM – Back Solution Multiple Regression

CORAN – Correlation Analysis

9. TIME SERIES ANALYSIS

MOVAVG – Moving Averages

SEASHI – Shiskin's Seasonality Factors

WEMAV – Weighted Moving Averages

TRELS – Trend Analysis by Least Squares

VADIME – Variate Difference Method

TSFARG – Autoregressive Model

GEXSMO – Generalized Exponential Smoothing

AUXCOR – Auto-Correlation and Cross-Correlation Analysis

POWDEN – Power Density Functions

RCPROB – Residual Probabilities

10. MULTI-VARIATE ANALYSIS

GENVAR – Generalized Variance

DISHOT – Hotelling's Distribution

DSQ – Mahalanobois' Distribution

SIGTMN – Significance of a Set of Means

DISCRA – Discriminant Analysis

FACTAN – Factor and Principal Component Analysis

11. DISTRIBUTION FUNCTIONS

RNORM – Normal Distribution

CHI – CHI-Square Distribution

STUD – Student's Distribution

FISH – Fisher's Distribution

POIS – Poisson Distribution

BIN – Binominal Distribution

HYGEO – Hypergeometric Distribution

GAMIN – Incomplete Gamma Distribution

BETINC – Incomplete BETA Distribution

12. INVERSE DISTRIBUTION FUNCTIONS

TINORM – Inverse Normal Distribution

STUDIN – Inverse Student's Distribution

FISHIN – Inverse Fisher's Distribution

CHIN – Inverse CHI-Square Distribution

13. MISCELLANEOUS

PLOT1 – Plot of One Line

JIM – Matrix Inversion

MXTMLT – Left Multiplication of a Matrix
by its Transpose

UNIVAC

SPERRY RAND