

 SPERRY RAND

UNIVAC

1106
SYSTEM

SYSTEM
DESCRIPTION

UNIVAC

1106

ERRATA SHEET TO
SYSTEM DESCRIPTION,
UP-7685
Programming Information
Exchange
Bulletin 1, UP-7700.1

- Page 6, Figure 2-1 -- Minimum system requirement is 2 UNISERVO* VI C or 2 UNISERVO* VIII C Magnetic Tape Units.
- Page 30, paragraph 4.7.9. -- At end of paragraph, add: "*The j and a fields together serve to specify any of the 128 control registers."
- Page 51, Table of UNISERVO* VIII C CHARACTERISTICS - For TAPE SPEED, change "42.5" to read "120".
- Page 93, Appendix B -- For topmost word format, change "MULTIPLE" to read "MULTIPLY".
- Page 105, Appendix D -- In introductory sentence, change "STAT-PACK" to read "MATH-PACK".

Please mark your copies of UP-7685 according to this Errata Sheet. Bulk distribution being made is the same as distribution for UP-7685. For details refer to original bulk distribution sheet. For additional copies, this sheet and original manual order via a Sales Help Requisition through your local Univac Representative from Holyoke, Massachusetts. All Holyoke stock will contain this Errata Sheet.

MANAGER, Documentation and Library Services

*Trademark of Sperry Rand Corporation

LISTS: UP-7700.1 to Lists 10U, 217, 630, 692 and distribution on Special Bulk Distribution Sheet. DATE: April 25, 1969

DIVISION OF SPERRY RAND CORPORATION
S.P.L.S. P.O. BOX 8100 PHILADELPHIA, PA. 19101

UP-3860 REV. 2

CONTENTS

CONTENTS	<i>i</i>
1. INTRODUCTION	1
2. SYSTEM DESCRIPTION	2
2.1. GENERAL	2
2.2. SYSTEM COMPONENTS	3
2.2.1. Central Processor	3
2.2.2. Main Storage	3
2.2.3. Auxiliary Storage	3
2.2.4. Operator's Control Console	4
2.3. CONFIGURATIONS	5
3. MAIN STORAGE	8
3.1. GENERAL	8
3.2. STORAGE MODULE	9
3.3. PACKAGING	9
3.4. STORAGE CAPACITY	9
3.5. STORAGE PROTECTION	10
3.5.1. Storage Protection Modes	10
3.5.1.1. Privileged Mode	10
3.5.1.2. User Program Mode	10
3.6. RELATIVE ADDRESSING	11
4. CENTRAL PROCESSOR UNIT	12
4.1. GENERAL	12
4.2. PRINCIPAL SECTIONS	12
4.3. INSTRUCTION WORD FORMAT	13
4.3.1. Function Code	13
4.3.2. Partial-Word or Immediate Operand Designator	13
4.3.3. Control Register Designator	13
4.3.4. Index Register Designator	13
4.3.5. Index Modification Designator	13
4.3.6. Indirect Address Designator	13
4.3.7. Address Field	14

4.4. CONTROL REGISTERS	14
4.4.1. Index Registers	14
4.4.2. Arithmetic Accumulators	15
4.4.3. Access Control Registers	15
4.4.4. R Registers	15
4.4.4.1. R0 – Real Time Clock	15
4.4.4.2. R1 – Repeat Counter	17
4.4.4.3. R2 – Mask Register	17
4.5. ARITHMETIC SECTION	17
4.5.1. The Adder	18
4.5.2. Arithmetic Accumulators	18
4.5.3. Partial-Word Transfers	18
4.5.4. Split-Word Arithmetic	19
4.5.5. Shifting	19
4.5.6. Double-Precision Fixed-Point Arithmetic	20
4.5.7. Floating-Point Arithmetic	20
4.6. EXECUTIVE SYSTEM CONTROL FEATURES	22
4.6.1. Processor State Register	22
4.6.2. Interrupts	24
4.6.3. Guard Mode	24
4.7. INSTRUCTION REPERTOIRE	26
4.7.1. Data Transfer Instructions	26
4.7.2. Fixed-Point Arithmetic	27
4.7.3. Floating-Point Arithmetic	27
4.7.4. Index Register Instructions	28
4.7.5. Logical Instructions	28
4.7.6. Shift Instructions	29
4.7.7. Repeated-Search Instructions	29
4.7.8. Unconditional Jump Instructions	30
4.7.9. Conditional Jump Instructions	30
4.7.10. Test (or Skip) Instructions	30
4.7.11. Executive System Control Instructions	31
4.7.12. Input/Output Instructions	31
4.7.13. Other Instructions	31
5. PROCESSOR INPUT/OUTPUT CONTROL SECTION	32
5.1. GENERAL DESCRIPTION	32
5.2. PERIPHERAL CONTROL	32
5.3. INTERNALLY SPECIFIED INDEX MODE	32
5.4. EXTERNALLY SPECIFIED INDEX MODE	33
5.5. BUFFER MODE DATA TRANSFERS	35

5.6. INPUT/OUTPUT INFORMATION WORDS	35
5.6.1. Data Words	35
5.6.2. Function Words	36
5.6.3. Identifier Words	36
5.6.4. Status Words	36
5.7. PRIORITY CONTROL	36
5.8. INPUT/OUTPUT INSTRUCTIONS	37
5.8.1. Monitored Instructions	37
6. PERIPHERAL SUBSYSTEMS	38
6.1. AVAILABLE PERIPHERAL EQUIPMENT	38
6.2. THE FLYING HEAD DRUMS	39
6.2.1. UNIVAC FH-432/FH-1782 Magnetic Drum Subsystem	40
6.2.2. UNIVAC FH-432 Magnetic Drum	41
6.2.3. UNIVAC FH-1782 Magnetic Drum	42
6.2.4. UNIVAC FH-880 Magnetic Drum	43
6.3. FASTRAND MASS STORAGE SUBSYSTEM	44
6.4. UNISERVO MAGNETIC TAPE SUBSYSTEMS	47
6.4.1. UNISERVO VI C Magnetic Tape Subsystem	48
6.4.2. UNISERVO VIII C Magnetic Tape Subsystem	50
6.4.3. UNISERVO VIII C/VI C Subsystem	51
6.4.4. Fully Simultaneous UNISERVO VIII C Subsystem	52
6.5. UNIVAC HIGH SPEED PRINTER SUBSYSTEM	52
6.6. PUNCHED CARD SUBSYSTEM	54
6.6.1. UNIVAC Card Reader	54
6.6.2. UNIVAC Card Punch	55
6.7. UNIVAC 9300 SUBSYSTEM	55
7. DATA COMMUNICATIONS	57
7.1. GENERAL	57
7.2. UNIVAC 1106 COMMUNICATIONS SUBSYSTEM	57
7.3. UNIVAC SYNCHRONOUS DATA TERMINALS	58
7.3.1. Word Terminal Synchronous	58
7.3.2. UNIVAC Communication Terminal Synchronous	59
7.4. UNIVAC DATA COMMUNICATION TERMINAL (DCT) 2000	60

7.5. UNISCOPE 300 VISUAL COMMUNICATION TERMINAL SUBSYSTEM	63
7.5.1. The Keyboard	63
7.5.2. CRT Display	64
7.5.3. Subsystem Configurations	64
7.5.4. Reliability	64
7.5.5. Special Features	65
7.6. UNIVAC 9200/9300 REMOTE SUBSYSTEMS	66
7.6.1. The UNIVAC 9200 System	67
7.6.2. The UNIVAC 9300 System	67
7.6.3. The UNIVAC Data Communication Subsystem, DCS-1	69
8. PROGRAMMED SYSTEMS SUPPORT	70
8.1. AVAILABLE SOFTWARE	70
8.2. THE EXECUTIVE SYSTEM	70
8.2.1. Multiple Modes of Operation	71
8.2.1.1. Batch Processing	71
8.2.1.2. Demand Processing (Time-Sharing)	71
8.2.1.3. Real Time	71
8.2.1.4. Multiprogramming	71
8.2.2. Techniques for Utilization of Mass Storage	72
8.2.3. The Primary Functional Areas of the Executive System	72
8.2.3.1. Executive Control Language	72
8.2.3.2. The Supervisor	72
8.2.3.3. Time Slicing	73
8.2.3.4. Storage Compacting	73
8.2.3.5. Facilities Assignment	73
8.2.3.6. The File-Control System	74
8.2.3.7. Operator Communications	74
8.2.3.8. The Diagnostic System	74
8.2.3.9. Input/Output Device Handlers	74
8.2.3.10. Auxiliary Processors	75
8.2.3.11. Utilities	75
8.2.3.12. System Setup	75
8.3. THE ASSEMBLER	75
8.3.1. Symbolic Coding Format	75
8.3.2. Assembler Directives	76
8.3.3. Additional Features	76
8.4. FORTRAN V	76
8.4.1. Language Extensions and Enhancements	76
8.4.2. Compiler Organization	80
8.5. CONVERSATIONAL FORTRAN V	81
8.5.1. System Features	81
8.5.2. System Concepts	81
8.5.3. Conversational Processor and the Executive System	82
8.5.4. Conversational FORTRAN Language	82

8.6. LIFT, FORTRAN II TO FORTRAN V TRANSLATOR	82
8.7. COBOL	83
8.7.1. The Processor	84
8.7.2. Special Features	85
8.8. ALGOL	85
8.9. SORT/MERGE	85
8.10. MATHEMATICAL FUNCTION PROGRAMS	86
8.11. APPLICATION PROGRAMS	87
8.11.1. Linear Programming System	87
8.11.2. APT III	88
8.11.3. PERT	88
8.12. MATH-PACK	89
8.13. STAT-PACK	89
APPENDICES	
A. NOTATIONAL CONVENTIONS	91
B. SUMMARY OF WORD FORMATS	93
C. INSTRUCTION REPERTOIRE BY FUNCTION CODE	95
D. MATH-PACK ROUTINES	105
E. STAT-PACK ROUTINES	109
FIGURES	
1-1. The Central Processor and Operator's Console	1
2-1. Minimum Multi-Module Storage Configuration	6
2-2. 262K Multi-Module Storage Configuration	7
4-1. Control Register Address Assignments	16
4-2. J-Determined Partial Word Operation	19
4-3. Processor State Register Format	23
7-1. CTMC Subsystem	57
TABLES	
4-1. Fixed-Address Assignments	25
7-1. DCT 2000 Field Installable Options	61

1. INTRODUCTION

A medium-scale extension to the proven UNIVAC 1100 series of computers is provided by the UNIVAC 1106 System. Its modular design provides growth potential within the UNIVAC 1106 as well as possible upgrading to the larger UNIVAC 1108 System.

The UNIVAC 1106 permits the user to select an executive system most applicable to his needs. This selection ranges from EXEC II, which provides a powerful batch processing capability, to the UNIVAC 1106 Executive System with its real-time, multiprogramming, demand, and batch processing ability.

In addition to a versatile executive system, the UNIVAC 1106 utilizes the complete family of UNIVAC 1108 unit processor software, application packages, and peripheral subsystems.



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

2. SYSTEM DESCRIPTION

2.1. GENERAL

The UNIVAC 1106 System is a general purpose, high performance unit processor system, making good use of 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 of varying complexity.

As the workload increases, this modularity also enables the addition of input/output subsystems and main storage. Among the principal features of the 1106 System are:

- Large, modular, parity-checked, high speed main storage
- 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 1106 System is organized to allow a number of tasks to be performed simultaneously under the direction of a common Executive control system.

The system is composed of four types of components:

- Central Processor
- Main Storage
- Auxiliary Storage Subsystems
- Peripheral Subsystems

2.2.1. Central Processor

The processor can perform all functions required for the execution of instructions including arithmetic, input/output, and Executive control. Included in the processor is a set of 166-nanosecond integrated circuit control registers providing multiple accumulators, index registers, input/output access control registers, and special-use registers.

2.2.2. Main Storage

The main storage read/restore cycle time is 1.5 microseconds. The minimum multi-module storage of the UNIVAC 1106 System is 65,536 36-bit words and is expandable up to a maximum of 262,144 words.

2.2.3. Auxiliary Storage

The auxiliary magnetic drum storage subsystems are an integral part of each UNIVAC 1106 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. Both types of drum can transfer data at 1,440,000 characters per second (240,000 words per second). The average access time of the FH-432 is 4.3 milliseconds; that of the FH-1782 is 17 milliseconds.

2.2.4. Operator's Control Console

The UNIVAC 1106 Operator's Control Console subsystem is a free-standing input/output device for directing and monitoring the operation of the CPU. The console is always connected to input/output channel 15.

The basic Control Console includes the following components:

- Display Console

The Display Console contains a keyboard and a Cathode Ray Tube (CRT) display. The keyboard is a standard four-bank keyboard which can generate 63 Fielddata codes. A row of eight interrupt keys is located above the keyboard. The Cathode Ray Tube can display 16 lines of 64 characters each. One UNIVAC PAGEWRITER printer is required for the minimum configuration if this console is selected.

- Day Clock

The day clock on the console 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 console.

The day clock may be selected to be active either externally or by program.

- Operator's Control and Indicator Panel

The Operator's Control and Indicator Panel includes fault, disable, and mode indicators for its CPU and associated main storage modules; displays and controls associated with selecting and releasing jumps and stops, with the Program Address Counter and Memory Select Register, and with the time display of the Day Clock. It also includes system controls associated with the CPU and subsystems which are logically connected to the CPU.

■ **Additional Features**

– **Auxiliary Console**

An auxiliary right- or left-wing console can be included to accommodate control/display panels for Communication Terminal Module Controller subsystems.

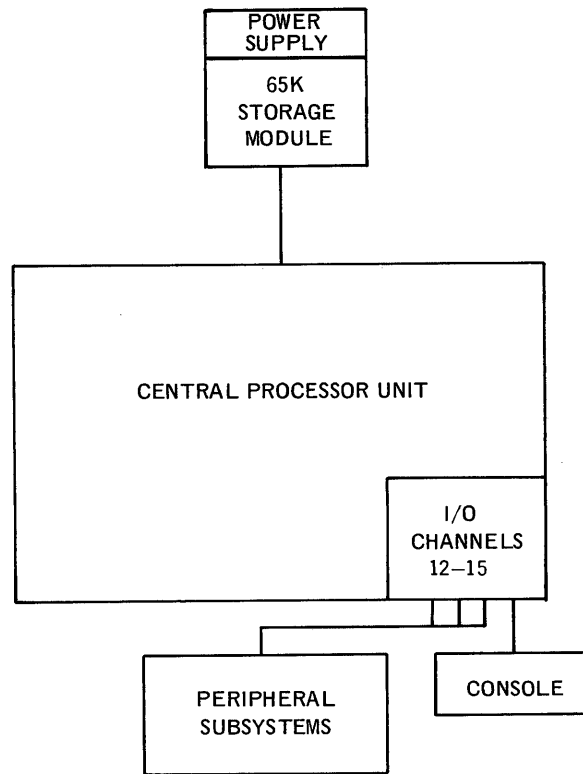
– **UNIVAC PAGEWRITER Printer**

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 prints lines of up to 80 characters per second.

2.3. CONFIGURATIONS

Figure 2-1 depicts the minimum multi-module storage configuration for the UNIVAC 1106 System. The CPU consists of the Command-Arithmetic Section and an Input/Output Section with four I/O channels. Main storage provides 65K 36-bit words. The minimum peripheral subsystems needed to support the software offered are: 1 FH-432/1782 Magnetic Drum Subsystem with 6 FH-432 drums or 1 FH-1782 drum, 1 UNISERVO Magnetic Tape Subsystem with either 4 UNISERVO VI C or 4 UNISERVO VIII C Magnetic Tape Units, an Operator's Console, and 1 online UNIVAC 9300 System.

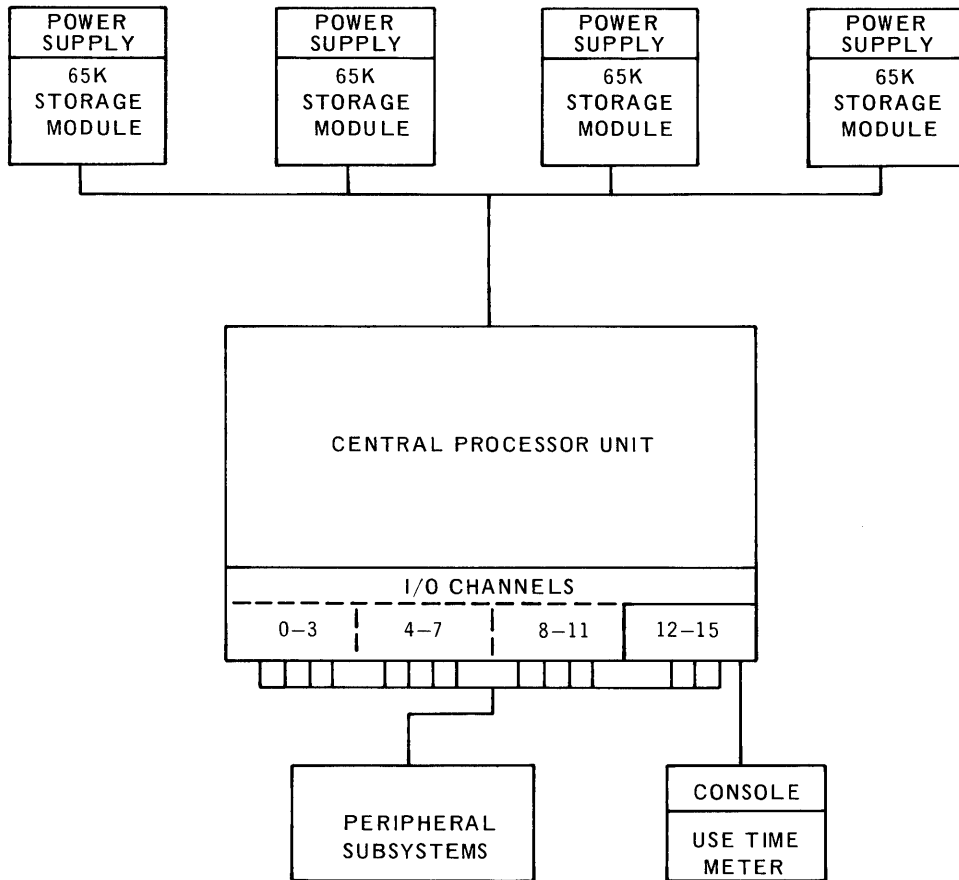
Figure 2-2 illustrates the 262K multi-module storage configuration for the UNIVAC 1106 System.



CONSOLE AND MINIMUM PERIPHERAL SUBSYSTEMS FOR 1106 OPERATING SYSTEM:

- 1 Operator's Console with:
Display Console and PAGERITER Printer
- 1 FH-432/1782 Magnetic Drum Subsystem with:
1 FH-1782 Drum or equivalent
- 1 UNISERVO Magnetic Tape Subsystem with:
4 UNISERVO VI C Magnetic Tape Units; or
4 UNISERVO VIII C Magnetic Tape Units
- 1 Online UNIVAC 9300 System with:
8K Main Storage
132 Print Position Printer
Multiplexer I/O Channel
600 cpm 80-Column Card Reader
Inter-Computer Control Unit

Figure 2-1. Minimum Multi-Module Storage Configuration



-----OPTIONAL

NOTE: EXEC II supports 131K main storage maximum.

Figure 2-2. 262K Multi-Module Storage Configuration

3. MAIN STORAGE

3.1. GENERAL

The main storage of the UNIVAC 1106 System is a high performance, fast access repository for instructions and data. Its design fully supports the concepts of multiprogramming, modularity, and reliability around which the entire UNIVAC 1106 System is constructed. Among its featured characteristics are:

- 1.5 microsecond read/restore cycle time
- 65,536, 98,304, 131,072, 196,608, or 262,144 36-bit words of multi-module storage
- Parity checking on all storage references
- Hardware storage protection – lockout boundaries establishable in 512-word increments
- Relative addressing and dynamic program relocatability through program base registers
- Online serviceability – multi-module storage pairs may be removed for servicing without stopping the entire system.

Main storage performs the following functions:

- Accepts an address from the processor.
- Stores or retrieves a word at that address
- Issues an acknowledgement signifying that a storage reference has been completed
- Checks parity on all data and delivers an interrupt signal to the processor should a parity error occur.

The modular storage concept has significant advantages for the immediate as well as the future needs of the system. Addition of banks of storage is simplified. It becomes a simple matter to add storage elements, or to replace them with improved equipment bank by bank, as technology advances.

3.2. STORAGE MODULE

Each 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 the 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 at each storage reference, three bits are available for selection of one of the four possible storage modules.

Parity is checked on reading or calculated on writing for each storage access. If a parity error is detected, the storage bank sends a parity error interrupt signal to the processor 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. PACKAGING

For multi-module storage, one or 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.

3.4. STORAGE CAPACITY

Available multi-module storage capacity ranges from 65,536 words to the system maximum of 262,144 words in accordance with the following:

- 65,536 words (two modules) – Minimum for system
- 98,304 words (three modules)
- 131,072 words (four modules)
- 196,608 words (six modules)
- 262,144 words (eight modules) – Maximum for system

3.5. STORAGE PROTECTION

To prevent inadvertent program reference to out-of-range storage addresses, the 1106 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 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.

3.5.1. 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 Storage Limits Register may be loaded but the PSR is set to disregard this, and the Executive can reference any location in main storage.

3.5.1.1. 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 non-alterable (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.

3.5.1.2. 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.

3.6. 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. CENTRAL PROCESSOR UNIT

4.1. GENERAL

The UNIVAC 1106 Central Processor Unit (CPU) is the principal component of the UNIVAC 1106 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.

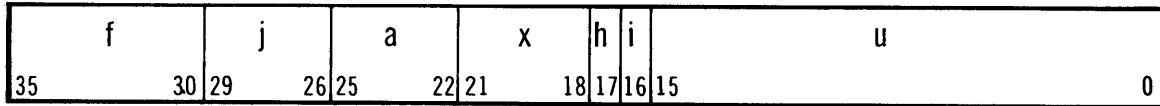
4.2. PRINCIPAL SECTIONS

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 buffer 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 function 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 1106 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.



4.3.1. Function Code

These six bits specify the operation to be performed. For function codes above 70_g, 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

For function codes less than 70_g, 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

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_g the a and j designators together address one of the 128 control registers.

4.3.4. Index Register Designator

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

4.3.5. Index Modification Designator

The h field controls modification of the index value (X_m) by the increment field (X_i) 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

The i designator 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 the word at the effective address. Indirect addressing continues as long as i = 1 with full indexing capability at each level.

4.3.7. Address 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 consist of 36-bit integrated-circuit registers, with a basic cycle time of 166 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 1106 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 00-15 are Index Registers and have the following format:



The X_m portion of the index register is an 18-bit modifier to be added to the base operand address of the instruction. The X_i portion of the index word updates the X_m 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.

4.4.2. Arithmetic Accumulators

Control register locations 12–27 are arithmetic accumulators, for programmed storage of arithmetic operands and results. The computation is performed in nonaddressable transient registers within the arithmetic section.

Depending upon the instruction, use of the accumulators results in a variety of word formats. Double precision instructions and a number of logical instructions reference two contiguous accumulators, i.e., 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 memory but produce a two-word result in the specified A and A + 1.

4.4.3. Access Control Registers

Control register locations 32–63 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.

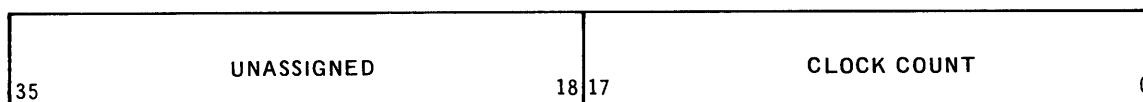
The word-by-word transmission of data over an I/O channel is governed by the contents of the ACR's. Two ACR's, one for input and one for output, are assigned to each of the sixteen channels. Input ACR's (locations 32–47) control input data transfers while output ACR's (locations 48–63) govern the transmission of output data and Function Words.

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

4.4.4. R Registers

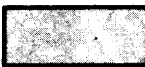
The sixteen control register locations 64–79 are R Registers. The first three of these (R0, R1, R2) have specified functions and formats as described below. The remaining R Registers are not specifically assigned; typically they are used as loop counters, transient registers, or storage for intermediate values or constants.

4.4.4.1. R0 – Real Time Clock



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.

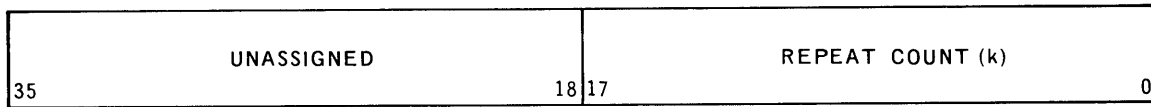
OCTAL		DECIMAL		
0	PROCESSOR STATE REGISTER (TEMP STORAGE)		0	
1	Xi	Xm	1	
13			11	
14			12	
17			15	
20			16	
33			27	
34			28	
37			31	
40	G	W(WORD COUNT)	V(BUFFER ADDRESS)	32
	OR			
57	INPUT IDENTIFIER		OUTPUT IDENTIFIER	47
60	G	W	V	48
77				63
100	REAL-TIME CLOCK		64	
101	REPEAT COUNT REGISTER		65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

 = EXECUTIVE (GUARD MODE PROTECTED)

*See 5.2.

Figure 4-1. Control Register Address Assignments

4.4.4.2. R1 – Repeat Counter



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. The contents of the effective address are compared to the contents of the Arithmetic Register 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 1106 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). Right shifts may be either logical or algebraic (sign bits are filled to the left).
- 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 by the double precision expansion kit.
- 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.
- Alphanumeric 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 1106 Processor is a one's 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 any of these instructions, both designators are cleared. The overflow designator is set when a significant bit is generated 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. This indicates the involvement of a negative quantity as one of the operands.

After the instruction has been performed, the settings of the designators remain unchanged 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 non-addressable transient registers that are used in actual computation and are contained within the arithmetic section itself.

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 1106 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 designator 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 designator.

J (OCTAL)	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 → 16-18
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 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 the most significant bit of the accumulator is unequal to the next most significant bit.

4.5.6. Double Precision Fixed Point Arithmetic

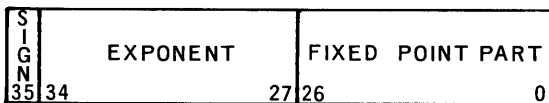
The System provides 72-bit, double precision fixed point addition and subtraction. Operands are processed as if they occupied a single 72-bit register. Bit 71, the high order bit, is the sign bit.

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.

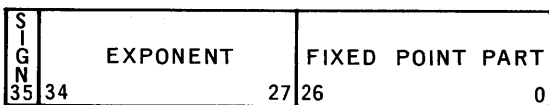
4.5.7. Floating Point Arithmetic

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^{-37} with eight-digit precision. The word formats are given below.

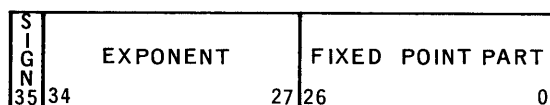
Source Operand Format



Result Format



WORD 1

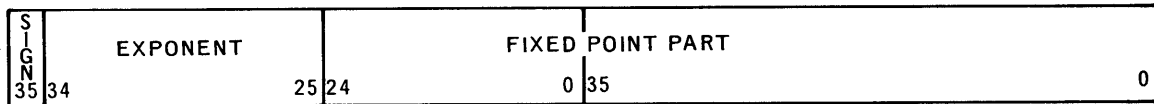


WORD 2

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 developed 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 (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 $\frac{1}{2}$ 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

The Executive has complete control of the entire 1106 system. Special hardware features are provided to permit this control.

The integrity of the system is maintained by Guard Mode operation. In this mode, certain instructions, registers, and storage locations are available for the exclusive use of the Executive System.

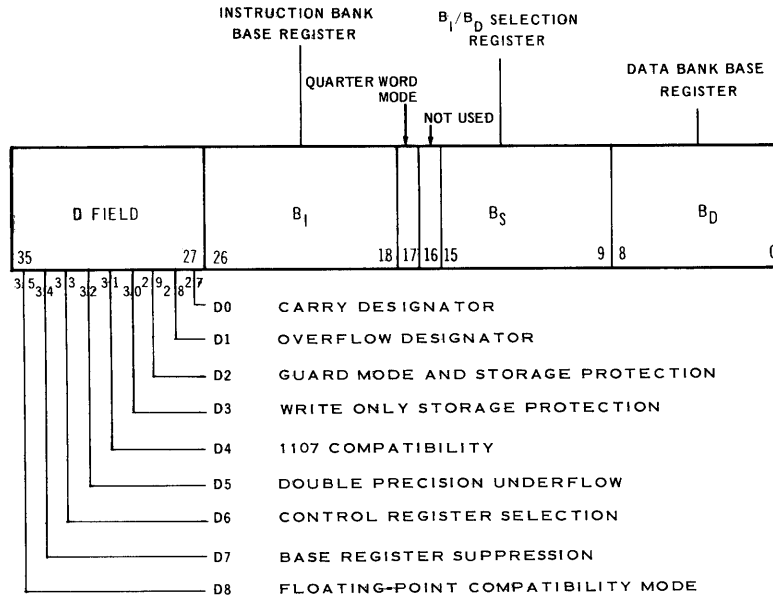
4.6.1. Processor State Register

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 worker programs, and registers status information concerning worker programs when it regains control in consequence of interrupts. Figure 4-3 explains in detail the significance of each bit of the PSR.

The Executive uses a special instruction, Load Processor State Register, to load the PSR and to govern 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
- Floating point underflow mode (double precision operations)
- 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 registered in the PSR and then its contents are transferred to Control Register location 00 (Index Register zero). The PSR is then cleared in preparation for Executive operations. The Executive saves the contents of Control Register 00 so that it can reinstate conditions as control is returned to the program that was interrupted.



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 00–37_g, 100–117_g)
- =1 Selects Executive Control Register Set (Locations 120–177_g)

The Executive passes control to User Programs with bit 33 = 0, which selects the worker set. An interrupt forces this bit to 1 after (PSR) has been transferred to X0, making the upper Control Registers available to the Executive.

D5 DOUBLE PRECISION UNDERFLOW – Double Precision Floating Point Operations

- =0 Interrupts on D.P. 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 1106 Mode – Full Range Addressing
- =1 The upper two bits of the effective address (U) are stripped, allowing 1107 program compatibility with the 1106. In this mode, only 65,536 words of storage are available.

D3 WRITE-ONLY STORAGE PROTECTION

- =0 Read, Write and Jump storage protection under Guard Mode
- =1 Write Protection Only under Guard Mode

D2 GUARD MODE

- =0 Guard Mode off. All instructions and storage references to Access Control Registers (40–77_g), the Real Time Clock (100_g) and Executive Control Registers (120–177_g) 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 Arithmetic Overflow at time of interrupt.

D0 CARRY DESIGNATOR

- =1 Arithmetic Carry at time of interrupt

OTHER FIELDS

0–8 } BASE REGISTERS. These registers
 9–15 } provide the absolute base address values
 18–26 } on which programs “float” in 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 1106 System is extensive. 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 sub-routines 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 signal the Executive to initiate remedial or terminating action when they are encountered. Table 4-1 lists the fixed-address assignments. Note that they are all interrupt locations except for 200g, which receives the status word, and 216g, 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 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
- Select Interrupt Location
- Load Channel Select Register
- Halt Instructions
- All I/O Instructions
- Disabling of I/O interrupts for more than 100 microseconds
- Attempting to write into any of the Executive control registers (32-64 or 80-127)

Guard Mode is disabled by the occurrence of any interrupt.

DECIMAL ADDRESS	OCTAL ADDRESS	FIXED ASSIGNMENT
128	200	Status Word for External Interrupt on CPU #0
129	201	Unassigned
130	202	Unassigned
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	214–5	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	Unassigned
155	233	Unassigned
156	234	Unassigned
157	235	Main Storage Parity Error Interrupt (Bank #1)
158	236	Main Storage Parity Error Interrupt (Bank #2)
159	237	Main Storage Parity Error Interrupt (Bank #3)
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
168–255	250–377	Unassigned
Last address	–1	Main Storage Parity Error Interrupt (Bank #0)

Table 4–1. Fixed-Address Assignments

4.7. INSTRUCTION REPERTOIRE

The UNIVAC 1106 Processor is provided with an unusually powerful and flexible instruction repertoire. Many of the instructions are effectively access and completed in the time of one storage cycle. Included in the complete set of instructions is a group which permits fast and simplified control by the Executive System.

In the following discussion, the instructions in the UNIVAC 1106 repertoire are grouped by functional class to illustrate the power of the repertoire. Appendix C lists them numerically by octal function code, stating exactly what each one does. The octal 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	27
Load X Modifier	26
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 X	06
Store R	04

To store two Arithmetic registers with one instruction:

Double Store A	71, 12
----------------	--------

Two special purpose transfers:

Store Zero	05
Block Transfer	22

All 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 to A	15
Add Magnitude to A	16
Add Negative Magnitude to A	17
Add Upper	20
Add Negative Upper	21
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
Add to X	24
Add Negative to X	25

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:

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

These instructions address the appropriate 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		Inclusive OR		Exclusive OR	
	0	1		0	1
0	0	0	0	0	1
1	0	1	1	1	0

The three simple logical instructions are:

Logical OR	40
Logical Exclusive OR	41
Logical AND	42

One special replace instruction is also logical:

Masked Load Upper	43
-------------------	----

Several other instructions such as the Repeated Masked Searches 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 address 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 the base address of the operand with that in the arithmetic register. They skip the next instruction when a specified condition is met or take the next instruction in sequence when the repeat count in R1 reaches 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)

Mask Search Equal	71,00
Mask Search Not Equal	71,01
Mask 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 the base address of the operand. If the specific condition is not met, 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 Nonzero	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

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 Nonzero	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 enables the Executive System to maintain proper control of the operating system.

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

These instructions are used for establishing Processor State, storage limits boundaries, interrupt locations, and identification of I/O channels.

4.7.12. Input/Output Instructions

This group of instructions allows the program (usually 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
No Operation	74,06
Test and Set	73,17

5. PROCESSOR INPUT/OUTPUT CONTROL SECTION

5.1. GENERAL DESCRIPTION

The input/output control section of the 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.

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.

5.2. PERIPHERAL CONTROL

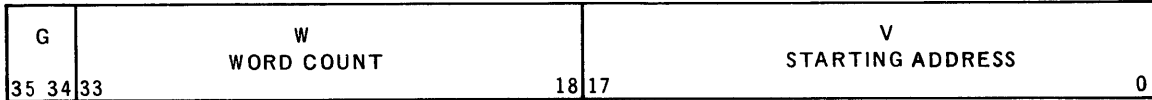
Input and output can be performed in either of two modes: Externally Specified Index (ESI) for multiplexed data communications devices, and Internally Specified Index (ISI) for other types of peripheral equipment. All I/O channels except channel 15, which is always assigned to the control console, can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

5.3. 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 40₈ to 77₈).

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



- V 18 bits, the starting 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 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.

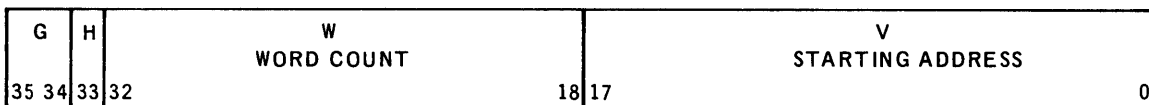
5.4. EXTERNALLY SPECIFIED INDEX MODE

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.

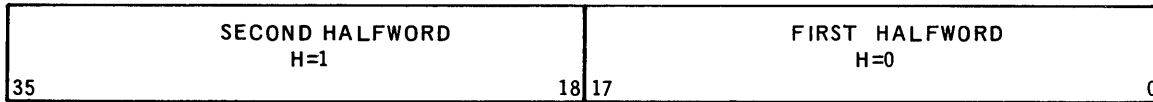
Any I/O channel can be set to the ESI mode by means of a switch. Furthermore, by means of a patch card, an ESI channel can be set to operate in either halfword (18-bit) or quarterword (9-bit) mode.

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 for ISI to enable control of half- and quarter-word transfers. The halfword Access Control Word is as follows:



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 half words. 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:

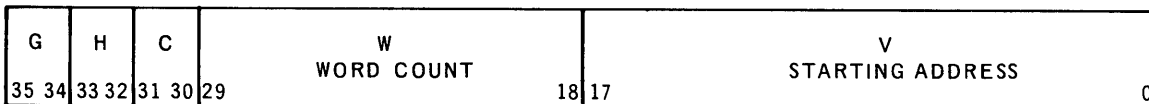


H = 0; use first halfword of location V and switch H to 1.

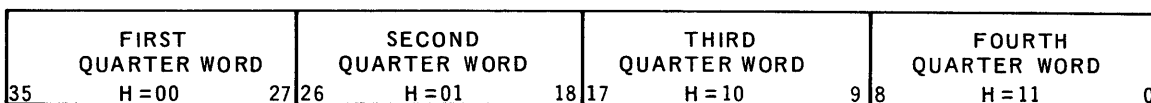
H = 1; use second halfword of location V, change V address as specified by G, and switch H to 0.

On input, the first halfword of an incoming message causes the associated ESI access control word to be transferred from main storage to the subsystem. 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 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 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, W, and V have the same meanings as for ISI, though 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:



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.

5.5. 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 Access Control Words into the locations specified by the ESI addresses.
- (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 the ESI mode this instruction loads the 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 word 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.

5.6. 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.

5.6.1. 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 acknowledgement. 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.

5.6.2. 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.

5.6.3. 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 accompanying external function signal identifies it as an identifier word. The subsystem stores it in a special register and compares it with each word read 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.

5.6.4. 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).

5.7. 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 in this sequence, 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. External Interrupt (ESI)
6. Input Monitor Interrupt (ESI)
7. Output Monitor Interrupt (ESI)
8. Real Time Clock Interrupt
9. External Interrupt (ISI)
10. Input Monitor Interrupt (ISI)
11. Output Monitor Interrupt (ISI)
12. Function Monitor Interrupt

5.8. 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 above, in 5.7.

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

5.8.1. 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.

This is not the case in ESI Mode. The Load Function in Channel and Monitor (LFCM) instruction never results in a Monitor Interrupt.

6. PERIPHERAL SUBSYSTEMS

6.1. AVAILABLE PERIPHERAL EQUIPMENT

Peripheral subsystems are attached to the 1106 processor 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 1106 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 1106 peripheral subsystems are:

High Performance Drums

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

Mass Storage

- FASTRAND II Mass Storage subsystem
- FASTRAND III Mass Storage subsystem

Magnetic Tape

- UNISERVO VI C Magnetic Tape subsystem
- UNISERVO VIII C Magnetic Tape subsystem

Auxiliary Systems

- Punched card subsystem (Reader/Punch)
- High speed printer subsystem

Data Communication Systems

- Communication Terminal Module Controller subsystem
- Word Terminal Synchronous
- Communication Terminal Synchronous
- Data Communication Terminal – DCT 2000
- Data Communication Subsystem – DCS-1
- UNISCOPE Visual Communication Terminal
- UNIVAC 9300 subsystem

In addition to the standard UNIVAC 1106 subsystems, the following subsystems function with the 1106 system as well:

- UNISERVO IIA tape subsystem
- UNISERVO IIIA tape subsystem
- UNISERVO IIIC tape subsystem
- UNISERVO IVC tape subsystem

6.2 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 UNIVAC 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 entire size of the subsystem, with a designated identifier word. Upon finding a match it supplies the address of the match or commences reading 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 data request that can be serviced fastest can then be selected from the system queue by the Input/Output handler. 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.

Any FH drum subsystem 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 1106 Operating System is designed to use quxiliary 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 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.

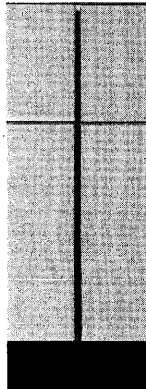
6.2.1. UNIVAC FH-432/FH-1782 Magnetic Drum Subsystem

A valuable characteristic of UNIVAC 1106 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 1106 storage configuration. An efficient blend can be made of high speed storage (for rapidly required software, program segments, tables, and indices) with 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 the CPU. 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.

6.2.2. UNIVAC FH-432 Magnetic Drum



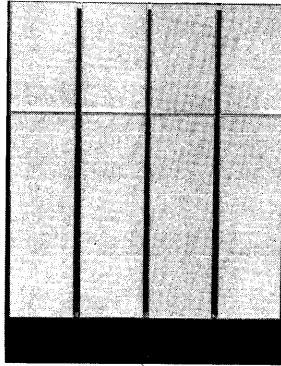
The FH-432 Magnetic Drum is a high speed, high performance drum. A minimum FH-432 Magnetic Drum Subsystem includes one drum (262,144 36-bit words of storage), a control unit, and power supplies, contained in two cabinets. To augment the systems, cabinets may be added, each containing one drum with a storage capacity of 262,144 36-bit words. 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 2,048 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 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 6.2.1.

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 or 2 per subsystem
NUMBER OF DRUMS PER SUBSYSTEM	1 to 8 (12,582,912 characters maximum)

6.2.3. UNIVAC FH-1782 Magnetic Drum



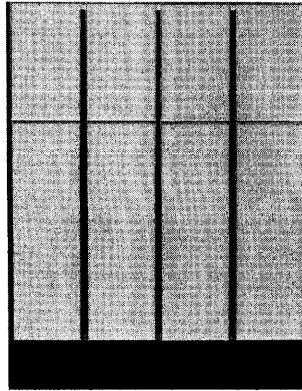
The FH-1782 magnetic drum is similar to the FH-432 drum except that average access time is four times greater and the storage capacity is eight times greater; this increase is achieved partly by an increase in the number of data tracks to 1,536 and partly by an increase in the recording density. Each track has its own read/write head, and average access time is 17 milliseconds.

A single FH-1782 drum stores 2,097,152 words, equivalent to 12,582,912 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 6.2.1.

CHARACTERISTICS	
STORAGE CAPACITY	2,097,152 computer words of 36 data bits plus parity bits, or 12,582,912 alpha-numeric 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)

6.2.4. UNIVAC FH-880 Magnetic Drum



The FH-880 magnetic drum 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 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 6,144 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 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)

6.3. FASTRAND MASS STORAGE SUBSYSTEM

The FASTRAND Mass Storage subsystem provides 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 II 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 pre-positioning the heads in each drum unit. Pre-positioning 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 pre-positioned unit, the only time delay is for the latency and address circuit activation. These are the mechanical design features which contribute to the 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 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.

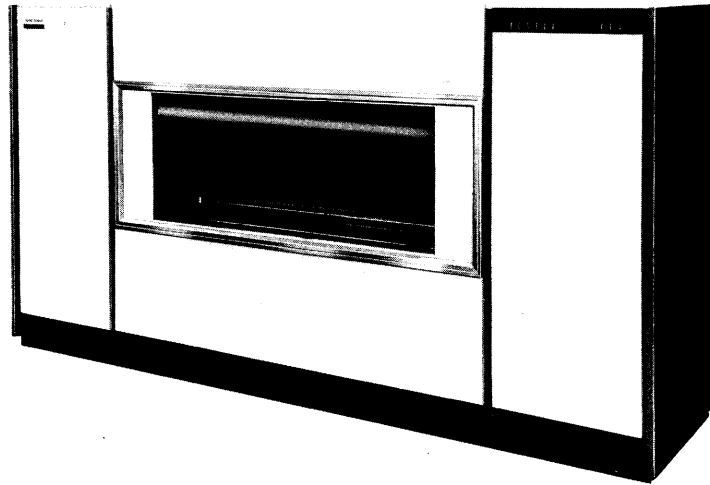
A single FASTRAND subsystem can accommodate up to eight FASTRAND units. Any of the Input/Output channels of a UNIVAC 1106 Processor can accommodate a FASTRAND subsystem. Like the FH-432/1782 drum subsystem, either single or dual channel operation is possible, providing full-scale two-drum-unit simultaneity.

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. Two control units which share some components are housed in one cabinet. In this system the FASTRAND units are organized into two banks and two simultaneous operations are possible so long as they call for drum units in different banks.

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) and write lockout feature for data protection. By means of this lockout feature, the operator can manually prohibit writing on 1, 2, 4, 8, 16, 32, or all 192 tracks starting with the first track of each head.

FASTRAND II AND III MASS STORAGE UNITS



CHARACTERISTICS

STORAGE CAPACITY (PER UNIT)	FASTRAND II: 22,020,096 36-bit words (132,120,576 characters) FASTRAND III: 33,030,144 36-bit words (198,180,864 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	870 revolutions per minute
MOVEABLE READ/WRITE HEADS	64
CHARACTER TRANSFER RATE	FASTRAND II: 157,698 characters per second FASTRAND III: 236,547 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).

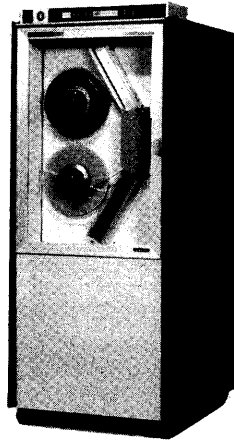
6.4. UNISERVO MAGNETIC TAPE SUBSYSTEMS

Magnetic tape subsystems for the 1106 system can be composed of UNISERVO VI C Magnetic Tape Units, UNISERVO VIII C Magnetic Tape Units, or any combination thereof, up to a total of 16 units. If only the UNISERVO VIII C Units are used, it is possible to double the transfer rate by using two UNISERVO VIII C Control Units, thus establishing simultaneous input and output capabilities.

Such a mixed magnetic tape subsystem affords a useful flexibility in combining high speed and economic medium speed tape units. Moreover, the seven- and nine-track unit options permit data recorded in traditional industry-compatible form to be handled, and yet at the same time allow the upgrading of these records in line with the USASCII code and packed-decimal data.

The UNISERVO VIII C/VI C subsystem is available in single- and dual-channel configurations. A dual-channel configuration includes a UNISERVO VIII C control unit and a UNISERVO VIII C auxiliary control unit, each of which requires a separate I/O channel of the CPU. With this configuration, two read operations or a read and a write operation can take place simultaneously, while any of the tape units not being read or written on can be rewinding.

6.4.1. 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 processes 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 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 this density more than 11,520,000 characters may be stored on a single reel in 600-character blocks.

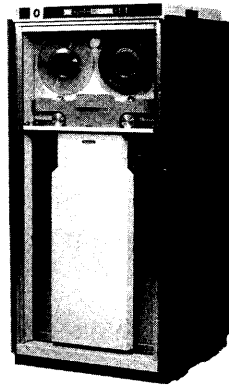
Rewind takes place at 160 inches per second, enabling a full reel of 2,400 feet to be rewound in 180 seconds.

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 Magnetic Tape units are fully compatible with IBM 727, 729 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 Fielddata code, thus ensuring tape compatibility among installations.

CHARACTERISTICS	
TRANSFER RATE	8,500, 23,700, and 34,200 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

6.4.2. 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 6-bit alphanumeric character, or a 6-bit binary value may be stored per frame. In nine-track mode one parity and eight data bits are recorded in each frame across the width of the tape.

Data packing density 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 characters per second (in seven-track mode), or bytes per second (in nine-track mode).

Even higher transfer rates result if 6-bit characters are read or written in nine-track mode. This method of operation yields transfer rates of 32,000, 88,960, and 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 will normally be used, the 200 and 556 densities being used only for 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 Magnetic Tape Units are fully compatible with IBM 727, 729 Mod 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 Fielddata code, thus ensuring tape compatibility among installations.

Two different UNISERVO VIII C tape subsystems are available: the UNISERVO VIII C/VI C subsystem and the fully simultaneous UNISERVO VIII C subsystem.

CHARACTERISTICS	
TRANSFER RATE	24,000, 66,720 and 96,000 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

6.4.3. UNISERVO VIII C/VI C Subsystem

The UNISERVO VIII C/VI C Subsystem permits any combination of UNISERVO VI C and UNISERVO VIII C Magnetic Tape Units, up to 16, to be intermixed on a single or dual channel subsystem. A Shared Peripheral Interface may be attached to permit access from one to four processors.

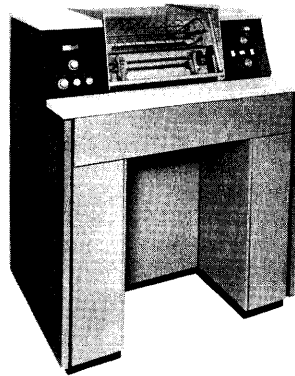
The tape units in a mixed subsystem may have seven- and nine-track capability in nearly any combination of units. The only restriction is that a slave UNISERVO VI C unit may have the nine-track capability only if the associated master unit has that capability.

6.4.4. Fully Simultaneous UNISERVO VIII C Subsystem

The fully simultaneous subsystem includes two UNISERVO VIII C control units, each connected to a separate I/O channel of the CPU. With this dual channel configuration, any two operations, including write/write, can be performed in addition to rewinds. This effectively doubles the maximum subsystem transfer rate provided by a single-channel subsystem.

The tape units in this subsystem require fully simultaneous features to provide the write/write capability. They may be either seven- or nine-track models. UNISERVO VI C units are not permitted.

6.5. UNIVAC HIGH SPEED PRINTER SUBSYSTEM



The UNIVAC High Speed Printer Subsystem provides the UNIVAC 1106 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 comprised of a High Speed Printer Control Unit connected to a High Speed Printer capable of printing from 1200 to 1600 lines per minute. The printer contains 63 printable characters: the 26 letters of the alphabet, the ten arabic numerals, and 27 special characters. Different symbols may be factory supplied upon order.

PARAMETER	SPECIFICATION
Printing Speed (with single-line spacing)	1200/1600 lines per minute maximum, depending upon program and character contingencies
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	Bi directional, 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 for its Polyester Film

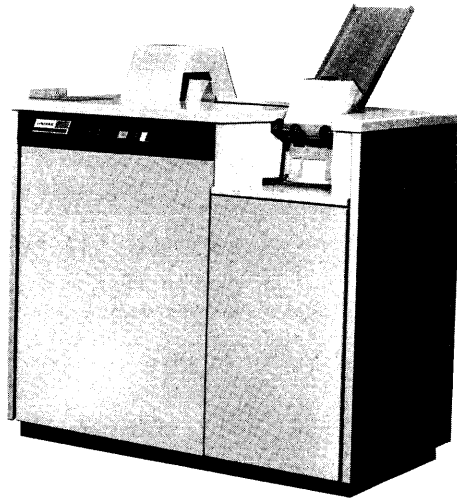
6.6. PUNCHED CARD SUBSYSTEM

The Punched Card subsystem comprises a UNIVAC 900 cpm Card Reader and a UNIVAC 300 cpm Card Punch which are attached to the same Control Unit on a single input/output channel of the CPU.

The Card Reader uses column-parallel photodiode sensing, with automatic redundant read checking, error cards being ejected into a separate stacker. A file feed device is standard. Cards may be read in Fielddata code or in row or column binary.

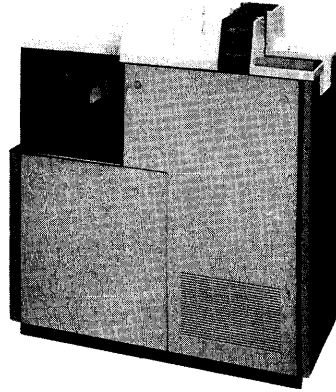
The Card Punch operates on a row-by-row basis and has an automatic check-read station. Incorrectly punched cards are recognized by the Input/Output Handler examining the status word upon the termination of the function, and such cards are passed to the error stacker. Under program control additional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

6.6.1. UNIVAC Card Reader



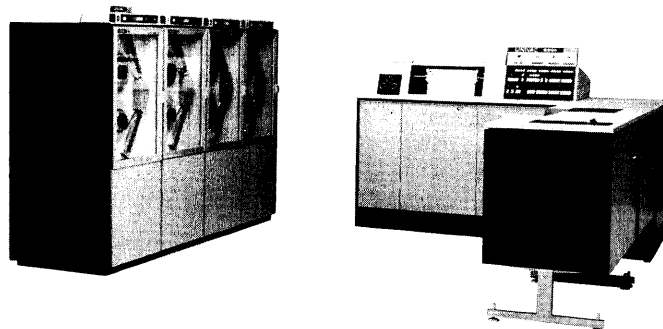
CHARACTERISTICS	
CARD READING SPEED	900 cards/minute
INPUT HOPPER CAPACITY	3,000 cards
OUTPUT STACKER CAPACITY	2,400 cards
REJECT STACKER CAPACITY	100 cards
READ MODES	Fielddata, column binary, row binary
I/O CHANNELS	1 shared with card punch

6.6.2. UNIVAC Card Punch



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

6.7. UNIVAC 9300 SUBSYSTEM



The UNIVAC 9300 System is a card/tape oriented system incorporating a very high speed (600 nanosecond) plated-wire memory. The basic system includes a versatile central processor unit with 8,192 to 16,384 byte memory, a 600 cpm card reader, a 75–200 cpm card punch, and a 600 lpm printer.

All card I/O operations, printing, and one read or write tape operation can be performed simultaneously with processing. With a dual-channel tape subsystem using two control units, the system can perform simultaneous read, write, and compute operations.

Expanded tape systems enable an even wider range of processing capabilities. Two or three concurrent tape operations such as card-to-tape, tape-to-print, and tape sorting are possible with the 9300 tape systems.

CHARACTERISTICS

SYSTEM ORIENTATION	Card/Magnetic Tape
BASIC MEMORY	8192 bytes
MAXIMUM MEMORY	32,768 bytes
MEMORY CYCLE TIME	600 nanoseconds
ADD (DECIMAL) INSTRUCTION TIME (TWO 5-DIGIT FIELDS)	52 microseconds
CARD READER - BASIC READER 1001 CARD CONTROLLER	600 cpm 1000/2000 cpm
CARD PUNCH - COLUMN ROW	75-200 cpm 200 cpm
READ/PUNCH	optional
PRINTING SPEED	600 lpm
NUMERIC PRINTING (optional)	1200 lpm
OVERLAPPED I/O UNITS	Standard
MULTIPLEXER CHANNEL RATE	85,000 bytes/sec
REGISTERS	8 for processor functions 8 for I/O functions
MAGNETIC TAPE RATE	34,160 bytes/sec
SIMULTANEOUS TAPE READ, WRITE, AND PROCESS	optional

7. DATA COMMUNICATIONS

7.1. GENERAL

There is a wide variety of methods for communicating with the UNIVAC 1106 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. The following list typifies the wide range of equipment available for data communication with the UNIVAC 1106 System:

- UNIVAC 1106 Data Communications Subsystem
- Synchronous Data Terminals
- UNIVAC Data Communication Terminal 2000
- UNISCOPE 300 Visual Communication Subsystem
- UNIVAC 9200/9300 Remote Subsystems

7.2. UNIVAC 1106 COMMUNICATIONS SUBSYSTEM

The UNIVAC 1106 Communications Subsystem enables the UNIVAC 1106 System to receive and transmit data by way of any common carrier at any of the standard rates of transmission up to 4800 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 7-1, the subsystem consists of two principal elements. The UNIVAC Communication Terminal Module (CTM) makes direct connection with the communication facilities. The UNIVAC Communication Terminal Module Controller (CTMC) transmits data between the modules and the Central Processor. A Communication Terminal Module Controller may be connected to any processor I/O channel, multiplexing up to 16 CTM's to that channel.

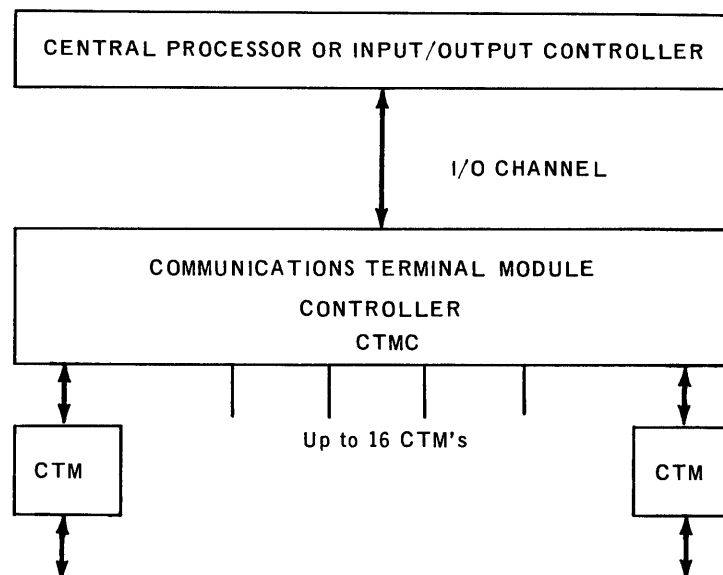


Figure 7-1. CTMC Subsystem

There are three basic types of input and output CTM's: low speed (up to 300 bits per second), medium speed (up to 1600 bits per second), and high speed (2000 to 4800 bits per second). Each is easily adjusted to the speed and other characteristics of the type of line with which it is to operate. Each CTM accommodates 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 8-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 1600 BPS		
High	To 4800 BPS		
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.

7.3. UNIVAC SYNCHRONOUS DATA TERMINALS

Each of the two types of synchronous data terminals discussed in this section links high speed (up to 40,800 bits per second) data circuits directly to an input/output channel without a control unit.

7.3.1. Word Terminal Synchronous

The UNIVAC 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 1106 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 UNIVAC WTS can handle data at speeds of 2000, 2400, and 40,800 bits per second.

The WTS allows the UNIVAC 1106 System to exchange data with a remote UNIVAC 1004 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.

7.3.2. UNIVAC Communication Terminal Synchronous

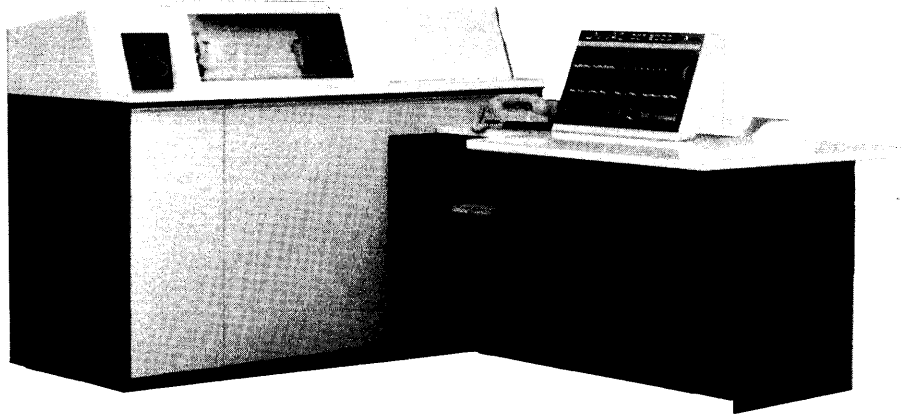
Like the UNIVAC WTS, the UNIVAC Communication Terminal Synchronous (CTS) is a central site device that links a synchronous data communication line to a processor input/output channel, at rates of 2000, 2400, or 40,800 bits per second. (The synchronous rate is set according to the characteristics of the 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 UNIVAC 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 System 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		
	UNIVAC WTS	UNIVAC CTS
SPEED	2000, 2400, 40,800 bits per second	
COMMUNICATION FACILITIES	2400 bits per second, or less: private voice lines. 40,800 bits per second or less: broad band transmission lines	
CONTROL CODING	Field for 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

7.4. 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 Systems, 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 UNIVAC 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 comprises a bar printer, a card reader/punch (not needed when used only as a printer terminal), a control unit, and an operator's console. Its design emphasizes:

- 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 7-1).

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 7-1. 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 USASCII 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	USASCII XS-3 (DLT compatible)

7.5. UNISCOPE 300 VISUAL COMMUNICATION TERMINAL SUBSYSTEM



The UNIVAC UNISCOPE 300 Terminal 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 Terminal itself includes a keyboard and CRT display, a core 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. He 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.

7.5.1. The Keyboard

The keyboard includes alphanumeric keys, cursor controls and 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 indicates 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.

7.5.2. 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 core storage unit and is regenerated on the screen 60 times per second.

7.5.3. 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 rate of the line 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 is 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-digit capacity or up to 48 terminals of 512-digit 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 programming. 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.

7.5.4. 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 automatically 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.

7.5.5. 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.

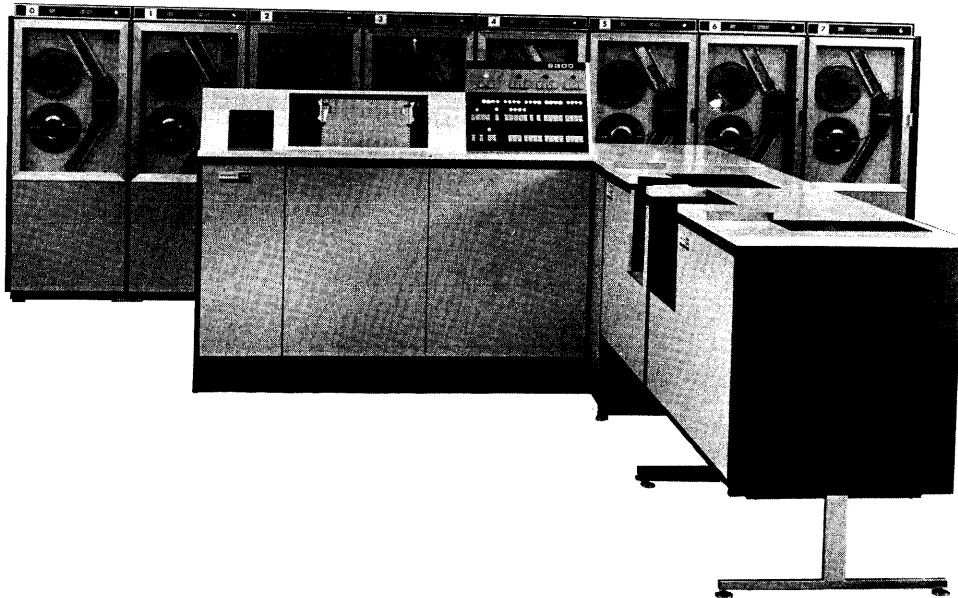
Of particular interest is the "roll and scroll" technique implemented by use of the insert and delete functions. In this case, text on the screen appears to roll up or down with the screen remaining filled. Typically, this is useful for scanning through large tables or other lengthy text that is too 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 output, 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 Half or full duplex operation Party line available through polling Non-significant spaces suppressed

7.6. UNIVAC 9200/9300 REMOTE SUBSYSTEMS



The user of a 9200/9300 system can, without impairing to the slightest degree its capabilities as a self-contained data processor, demand the full and powerful facilities of the 1106 system. This is accomplished by use of the DCS-1 (Data Communications System) to link the 9200/9300 system to communications lines. The DCS-1 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 UNIVAC 9200/9300 equipment has memory large enough and internal speeds fast enough to utilize communications lines to the fullest. Furthermore, the DCS-1 subsystem has provisions for error detection which are flexible enough to be adaptable to any reasonable requirements of the user.

When the 9200/9300 system is being used as a remote subsystem of a UNIVAC 1106 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.

7.6.1. The UNIVAC 9200 System

The basic 9200 System is designed and priced for use in conjunction with modest sized punched card installations. But, because it uses advanced hardware innovations such as plated-wire memory and integrated circuits and because it includes such software benefits as internal programming and a full set of software packages, it compares favorably with systems two or three times larger.

The basic system includes a versatile processing unit with 8,192 to 16,384 byte memory, a 400 cpm card reader, a 75 to 200 cpm card punch and a 250 to 500 lpm printer.

Up to eight additional I/O subsystems can be connected to the system by means of an I/O Multiplexer. These subsystems include such peripheral equipment as the UNIVAC 1001 card controller, the UNISCOPE 300 Visual Communications Terminal, and the Data Communications Subsystem, DCS-1.

The UNIVAC 9200 system is modular and can be easily expanded to the 9300 configuration. Because of programming compatibility, this expansion from the smaller system to the larger requires no reprogramming.

7.6.2. The UNIVAC 9300 System

The 9300 System is compatible with the 9200 system; it offers faster computing, larger and faster memory, faster punched card handling and printing, and the UNISERVO VI C Magnetic Tape Subsystem. Operating speeds are summarized in the Table of characteristics below.

All card I/O operations, printing, and one read or write tape operation can be performed simultaneously with processing. With a dual-channel tape subsystem using two control units, the system can perform simultaneous read, write, and compute operations.

Expanded tape systems enable an even wider range of processing capabilities. Two or three concurrent tape operations such as card-to-tape, tape-to-print, and tape sorting are possible with the 9300 tape systems.

CHARACTERISTICS

	UNIVAC 9200	UNIVAC 9300
SYSTEM ORIENTATION	Card	Card/Magnetic Tape
BASIC MEMORY MAXIMUM MEMORY	8192 bytes 16,384 bytes	8192 bytes 32,768 bytes
MEMORY CYCLE TIME	1200 nanoseconds	600 nanoseconds
ADD (DECIMAL) INSTRUCTION TIME (TWO 5-DIGIT FIELDS)	104 microseconds	52 microseconds
CARD READER - BASIC READER 1001 CARD CONTROLLER	400 cpm 1000/2000 cpm	600 cpm 1000/2000 cpm
CARD PUNCH - COLUMN ROW	75-200 cpm Not used	75-200 cpm 200 cpm
READ/PUNCH	Not used	optional
PRINTING SPEED	250 lpm	600 lpm
VARIABLE PRINTING SPEED (optional)	250/500 lpm alphanumeric 500 lpm numeric	Not used
NUMERIC PRINTING (optional)	Not used	1200 lpm
OVERLAPPED I/O UNITS	Standard	Standard
MULTIPLEXER CHANNEL RATE	85,000 bytes/sec	85,000 bytes/sec
REGISTERS	8 for processor functions 8 for I/O functions	8 for processor functions 8 for I/O functions
MAGNETIC TAPE RATE	Not used	34,160 bytes/sec
SIMULTANEOUS TAPE READ, WRITE, AND PROCESS	Not used	optional
DATA TRANSMISSION RATE THROUGH DCS-1	2000, 2400 or 50,000 bits per second	2000, 2400 or 50,000 bits per second

7.6.3. The UNIVAC Data Communication Subsystem, DCS-1

The DCS-1 subsystem provides communication capability for the UNIVAC 9000 series computers. 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 UNIVAC 1106 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 UNIVAC DCS-1 under program control by connecting the output line to the input line to verify transmission and reception 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 memory 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. Longitudinal redundancy check is optional.
MULTIPLEXER CHANNELS REQUIRED	One per subsystem.

8. PROGRAMMED SYSTEMS SUPPORT

8.1. AVAILABLE SOFTWARE

The programmed systems support (software) provided with the UNIVAC 1106 system has been designed to meet the computing requirements of the advanced users of today. The requirements and demands of the medium-scale user have grown considerably in the past, and it is in response to these changing requirements that Univac offers with the 1106 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 1106 system, Univac has combined many years of experience in multi-programming 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 1106 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 1106 software system, based upon the UNIVAC FH-432 and FH-1782 magnetic drums with their rapid access and high transfer rate, and backed up by mass storage, can give optimum performance.

In addition to the operating system described in this section, a less complex operating system is also available. This operating system, called EXEC II, is a functional subset of the operating system herein described.

8.2. THE EXECUTIVE SYSTEM

To take full advantage of the speed and hardware capabilities of the UNIVAC 1106 System and to make effective use of a given hardware configuration, a comprehensive internal operating environment has been created.

Only through central control of all activities of the 1106 System can this environment of the combined hardware and software systems be fully established and maintained to satisfy the requirements of a variety of 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 1106 System easily, while relieving him of concern for the internal housekeeping requirements.

8.2.1. Multiple Modes of Operation

The technical capabilities of the UNIVAC 1106 Executive System cover a great variety of data processing activities. Its design is such that minimal penalties are imposed upon any one of these activities by the support provided for the others, and an installation not interested in making use of the full range of activities may specify capabilities to be eliminated at system generation time.

8.2.1.1. 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.

8.2.1.2. Demand Processing (Time-Sharing)

Complementing the batch processing capabilities of the 1106 Executive System are its time-sharing capabilities, the simultaneous accommodation by the Executive System of requests and demands from users at 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.

8.2.1.3. Real Time

The Executive System is also designed to be 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 real time programs.

8.2.1.4. 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. Batch and demand operations are handled concurrently by the Executive System, whenever sufficient storage facilities are available.

8.2.2. 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.

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.

8.2.3. The Primary Functional Areas of the Executive System

The UNIVAC 1106 Executive System is composed of many different routines which perform many different functions. These functions and routines are summarized in the following paragraphs.

8.2.3.1. 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 heading character (V), 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.

8.2.3.2. The Supervisor

The Supervisor is the 1106 Executive System component that controls the sequencing, setup, and execution of all runs entering the 1106 System. It is designed to control the execution of a main program and several symbionts.

The Supervisor contains three levels of scheduling; coarse scheduling, dynamic allocation of storage space, and CPU dispatching. Runs entering the 1106 System are sorted 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 central 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.

8.2.3.3. Time Slicing

In order to accommodate demand processing, periodic time slices must be assigned to particular routines. 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.

8.2.3.4. Storage Compacting

Certain 1106 hardware features make feasible the dynamic relocation of programs residing in central storage – a necessity for effective multiprogramming. At program termination, the assigned storage is returned to the pool of available central 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.

8.2.3.5. 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 UNIVAC 1106 System. The Executive System maintains current inventory tables that indicate the facilities available for assignment and the runs that are using the currently unavailable facilities.

8.2.3.6. The File-Control System

The 1106 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 to manipulate files and to inform the user of the current status and structure of his files. Provisions are made for random storage and retrieval of data, under the direction of the user. User program files and data files are maintained and processed in the same environment.

8.2.3.7. 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.

The 1106 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 up, the Executive defers lower priority displays.

8.2.3.8. Diagnostic System

A comprehensive diagnostic system within the UNIVAC 1106 Executive System aids in checking out user programs. Both allocation-time and assembly-time commands trigger snapshot dumps. Postmortem dumps are available through an Executive Control Statement.

8.2.3.9. Input/Output Device Handlers

The Input/Output device handlers control the activities of all input/output channels and peripheral equipment attached to the 1106, including UNISERVO magnetic tape units, UNIVAC FH-432 and FH-1782 drums, FASTRAND Mass Storage, card readers, printers, and communication line terminals.

8.2.3.10. 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.

8.2.3.11. 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.

8.2.3.12. 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.

8.3. THE ASSEMBLER

The Assembler translates a symbolic language composed of brief expressions to machine-language relocatable object coding for the UNIVAC 1106 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 prepare for program segmentation and 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 loading mechanism of the system. If requested, it supplies a listing of the original symbolic coding and an edited octal representation of each word generated. Flags indicate errors detected by the assembler in the symbolic coding.

8.3.1. Symbolic Coding Format

In 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. Appendix C lists the instructions recognized by the assembler.

8.3.2. Assembler Directives

The symbolic assembler directives control or direct the assembly processor just as operation codes control or direct the central 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 expressions, to adjust the location counter value, and to afford the programmer special control over the generation of object coding.

8.3.3. Additional Features

Facilities for interprogram communication permit separately assembled programs (or subprograms) to be linked together at load 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 subprograms (or 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, at or before the time of execution.

8.4. 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 USASI 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).

8.4.1. Language Extensions and Enhancements

The following extensions and enhancements are included 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 sub-expressions occur within a program unit, it is obviously desirable to evaluate each sub-expression only once. Where the common sub-expressions 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 sub-expressions 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 re-evaluated at each occurrence, as in most other FORTRAN Systems.
- Non-standard 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 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 may not be mixed with other types.
 Double precision and complex variables cannot be mixed.
- Extended subscript expressions are permissible, having the form $\pm M_1 \pm M_2 \dots \pm M_i$ where M is of the form $K_1 * K_2 * K_3 \dots K_j$. 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 . 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 a LIST on an INPUT/OUTPUT Statement and the associated FORMAT Statement. A NAMELIST name (1–6 alphanumeric characters) is defined by its appearance in a NAMELIST statement, and thereafter may appear only in an input or output statement that requires a list.
- The DEFINE statement is of the form `DEFINE R(a1, ..., an) = e` or `DEFINE R = e`, where R and a_i are variable names and e is any expression not involving any undefined R's. The DEFINE Statement generates in-line code when a procedure is involved, 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 (e.g. specification statements or a set of internal subprograms) may be added to the source code from an internally available element.
- The IMPLICIT Type statement is of the form `IMPLICIT type (a1, a2...), type (a1, a2, ...)`, 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 beginning with certain designated letters (the a_i) are of a certain type.
- The ENTRY statement is of the form `ENTRY name (a1, a2, ..., an)` 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 via 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 1106 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 (i.e. SQRT, DSQRT or CSQRT for REAL, DOUBLE PRECISION or COMPLEX arguments respectively.)
- The 1106 FORMAT Control has been augmented by the addition of new FORMAT control specifications of the form:
 - Gw.d Used for input and output of any of the five types of variables. If 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 wth 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 'c₁,c₂,..., 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 (e₁, e₂), OR(e₁, e₂), BOOL(e₁), COMPL(e₁), and XOR(e₁, e₂).
- 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 a COMMON statement.
 - 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) on call to a subroutine or function sub-program to indicate the transmission of a statement number.
- Data will be accepted in the explicit type statement or DIMENSION statements; for example, REAL A/1.51/,B(2,3)/6*1.01/.

8.4.2. Compiler Organization

The 1106 FORTRAN V source language processor accepts FORTRAN statements and produces a highly efficient relocatable object code element. FORTRAN, like all other UNIVAC 1106 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 intention was to create a compiler which, while quite rapid as a processor, would produce 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.

8.5. CONVERSATIONAL FORTRAN V

An important element of the programmed systems support provided with a UNIVAC 1106 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 non-sequential trial-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.

8.5.1. 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.

8.5.2. System Concepts

A session is 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 is the session the user is currently engaged in and 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.

8.5.3. Conversational Processor and the Executive System

The conversational 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.

8.5.4. Conversational FORTRAN Language

Conversational FORTRAN consists of procedural statements defined by the UNIVAC 1106 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. Any programs constructed by the 1106 Conversational FORTRAN compiler may also be compiled by the batch compiler.

8.6. 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. LIFT itself, written in FORTRAN V, is fully integrated with the Executive System.

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. These nine areas are:

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.

8.7. COBOL

1106 COBOL is based on the Department of Defense publication ‘‘COBOL edition 1965’’. The 1106 COBOL accepts statements written in the COBOL language as adapted for the 1106 system and produces a program ready for execution.

The major features implemented on the 1106 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

The following features, implemented in UNIVAC 1106 COBOL, are special UNIVAC extensions to the COBOL language:

- COBOL sub-program communication
- MONITOR
- Dynamic date
- Common Storage
- Page Control

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

8.7.1. The Processor

The COBOL processor is a six-phase compiler. It is completely modular in relocatable elements and it is handled as any program in the system for easy expandability and maintenance. Likewise, the COBOL processor produces as its output relocatable binary elements stored on the drum or mass storage, which are indistinguishable from other elements in the system. Other output from the compiler includes extensive diagnostic messages, source language listings, machine language listings, and special cross reference listings of name definitions and their references. The machine language listing consists of side-by-side Procedure Division statements and the corresponding generated symbolic machine code.

The compiler diagnostics 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 so that it cannot easily be executed. This type of diagnostic is identified by the word **ERROR*** preceding the actual message.

8.7.2. Special Features

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 1106 processor such as FORTRAN V.

Overpunched Sign Convention – Tapes and cards prepared in the overpunched sign convention can be processed.

8.8. ALGOL

The ALGOL language allows the mathematician or engineer to prepare programs for the UNIVAC 1106 without the necessity of becoming familiar with the details of the internal machine operation. From this pseudo-mathematical source language, the ALGOL Compiler generates 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 1106 ALGOL is the “Revised Report on the Algorithmic Language, ALGOL 60” (Communications of the ACM, Vol. 6, January 1963, 1–17).

UNIVAC 1106 ALGOL is an extended hardware representation of ALGOL 60 designed to employ the UNIVAC 1106 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.

8.9. SORT/MERGE

The UNIVAC 1106 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 is used to take advantage of any inherent sequence in the original data. Strings may be written upon magnetic tape or drum. The UNIVAC 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 advantage can be gained thereby.

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 and 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.

8.10. MATHEMATICAL FUNCTION PROGRAMS

The UNIVAC Software System includes an extensive collection of basic mathematical subroutines and functions. This collection includes all of the standard FORTRAN functions and has been expanded by over 50% 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), are 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:

1. Library Functions
2. The FORTRAN Built-in Function
3. 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:**

a LMJ B11,SQRT
a + 1 + Address of X
a + 2 + normal return
The result is left in A0.

■ **COBOL Calling Sequence:**

ENTER SQRT REFERENCING X

8.11 APPLICATION PROGRAMS

The UNIVAC 1106 System has an extensive library of application programs and sub-routines. 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 1106 System.

8.11.1. 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 1106 System embodies the latest advances in computer technology with the most powerful algorithm to date. The algorithm employs the "product form of the inverse" method and is improved with an advanced path selection technique. The package is coded in FORTRAN V and Assémbly Language.

The more prominent feature of the 1106 LP System are as follows:

2. The speed and random access properties of magnetic drums place the system at a distinct advantage over tape and disc handling procedures.
3. Both single precision and double precision computations are available. The selection may be manual or automatic.
4. The control language of this System is far superior to any other existing LP System. It is an interpretive control language. The sophisticated user may use macros in constructing his command string to implement the System. On the other hand, the average user may still execute his basic LP problem without detail knowledge of the control language.
5. The LP System is imbedded in the Executive System. This enhances the System as a powerful model builder. Matrix builders and output analyzers may be attached to the LP System to form corporate models. The entire model may be optimized and re-optimized in cycles in one computer run. Nonlinear programming may be accomplished by solving approximated linear functions in each cycle.

In addition to the above features, the 1106 LP System contains the flexibilities existing in other LP Systems. Vector levels can be specified and coefficients may be modified. Long LP runs can be split with restart procedures. Post-optimal parametric programming or a complete tableau can be obtained. The final output includes the objective function value, optimal basis, vector levels, and reduced costs.

8.11.2. APT III

APT (Automatically Programmed Tools) 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 manufacturing 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 introduction of changes.

The APT III program represents over one hundred man-years of development and testing. After extensive experience with our 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. The use of certain parts of APT requires membership in this 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 1106 APT program in the forefront of the art. As implemented on the UNIVAC 1106 System APT III will continue to conform to the latest APT long range program specifications.

8.11.3. PERT

The UNIVAC 1106 PERT System is a generalized applications program 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 System for which actual project costs are collected and compared with estimates for purposes of cost control.

The UNIVAC 1106 PERT System adheres rigorously to the DOD/NASA design and satisfies the purposes of PERT/COST concept.

Many government agencies and contractors are currently processing PERT/TIME data on existing systems. PERT is relatively new and will be subject to the inevitable modifications that will result from its initial pilot tests by DOD. In order to lessen the impact of these changes upon existing PERT programs and to provide for efficient integration of time and cost data, a modular design was adopted.

The modular structure of the program permits separate processing of the time networks and of the work package costing structure while simultaneously providing for integrated time and cost reporting. The PERT/TIME Module of the UNIVAC 1106 PERT System accepts as input a deck of cards describing the PERT network. The cards are processed and used to update the PERT/TIME Master File. Network computations are performed and the time reports are generated. The PERT/COST Module accepts the cost breakdown structure, actual and estimated costs for the project work packages, and a table of labor rates and applicable overhead percentages. Cost data is accumulated up through the cost breakdown tree, integrated with the time information, and the required cost reports are generated.

8.12. MATH-PACK

MATH-PACK provides the UNIVAC 1106 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:

- (1) Interpolation
- (2) Numerical Integration
- (3) Solution Of Equations
- (4) Differentiation
- (5) Polynomial Manipulation
- (6) Matrix Manipulation: Real Matrices
- (7) Matrix Manipulation: Complex Matrices
- (8) Matrix Manipulation: Eigenvalues and Eigenvectors
- (9) Matrix Manipulation: Miscellaneous
- (10) Ordinary Differential Equations
- (11) Systems Of Equations
- (12) Curve Fitting
- (13) Pseudo-Random Number Generators
- (14) 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.

8.13. STAT-PACK

STAT-PACK provides the UNIVAC 1106 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:

- (1) Descriptive Statistics
- (2) Elementary Population Statistics
- (3) Distribution Fitting and Plotting
- (4) Chi-Square Tests
- (5) Significance Tests
- (6) Confidence Intervals
- (7) Analysis of Variance
- (8) Regression Analysis
- (9) Time Series Analysis
- (10) Multivariate Analysis
- (11) Distribution Functions
- (12) Inverse Distribution Functions
- (13) 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.

APPENDIX A. NOTATIONAL CONVENTIONS

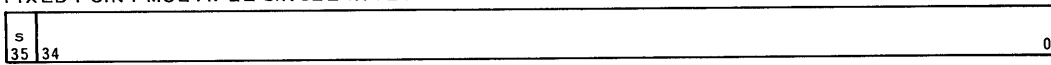
Abbreviations and symbols frequently used in the description of the instruction repertoire are given below:

()	Contents of register or address within parentheses.
()'	Complement of contents of register or address.
()	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 .
X _m	Modifier portion of an index register.
X _i	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 .

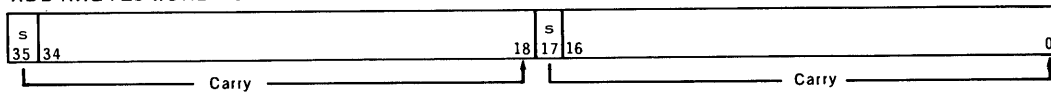
u	The base address of the operand (or the actual operand) as coded in u-field of an instruction.	
U	The effective address or value of the operand after application of indexing and indirect addressing.	
U _d	Destination address.	
U _s	Source address.	
h	h-designator of the instruction word. A value of 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

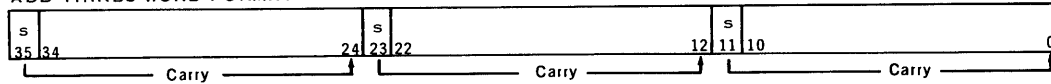
FIXED-POINT MULTIPLE SINGLE INTEGER RESULT



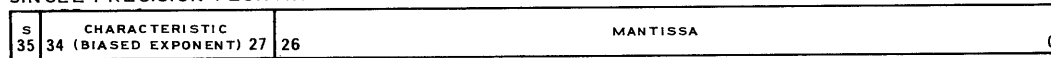
ADD HALVES WORD FORMAT



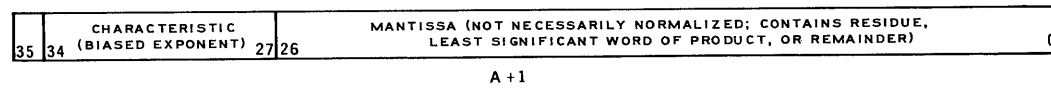
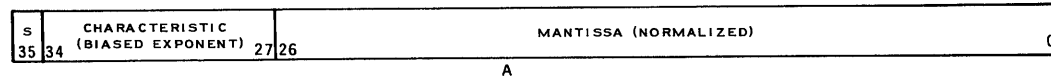
ADD THIRDS WORD FORMAT



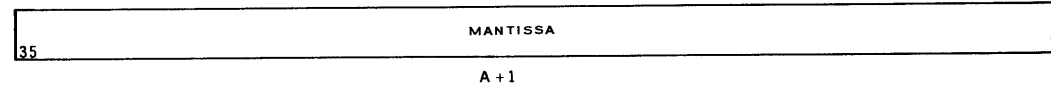
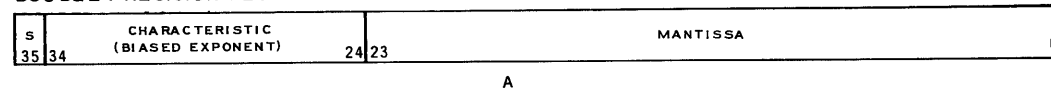
SINGLE-PRECISION FLOATING-POINT OPERAND



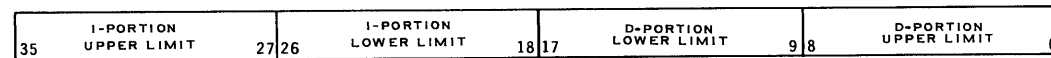
SINGLE-PRECISION FLOATING-POINT RESULT



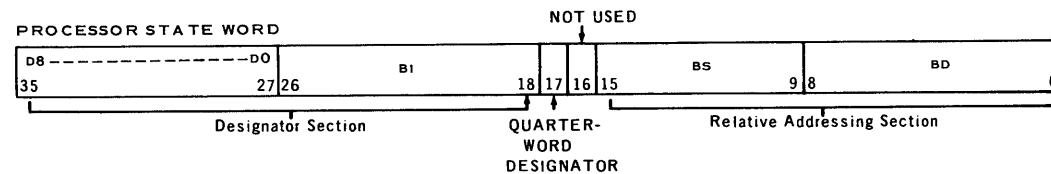
DOUBLE-PRECISION FLOATING-POINT OPERAND OR RESULT



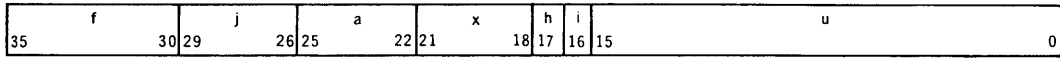
STORAGE LIMITS WORD



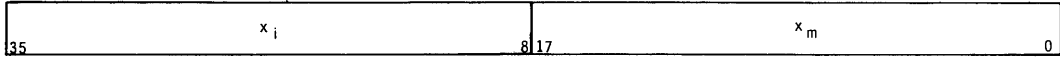
PROCESSOR STATE WORD



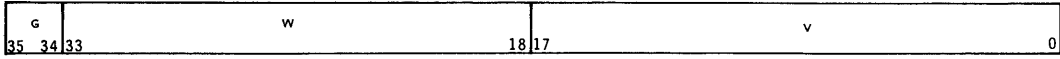
INSTRUCTION WORD



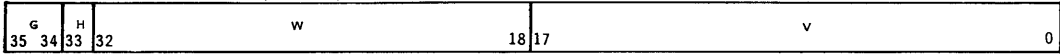
INDEX REGISTER WORD



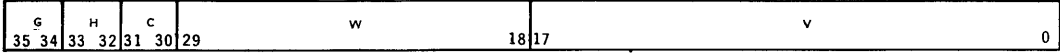
ISI ACCESS CONTROL WORD



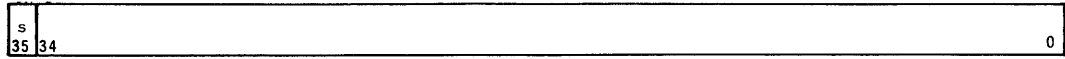
ESI ACCESS CONTROL WORD (HALF WORD)



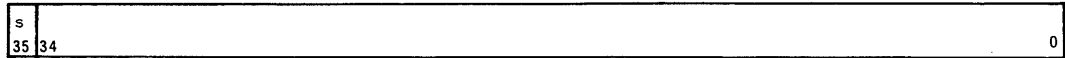
ESI ACCESS CONTROL WORD (QUARTER WORD)



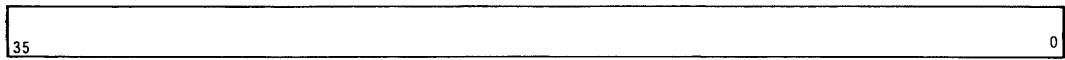
SINGLE-PRECISION FIXED-POINT WORD



DOUBLE-PRECISION FIXED-POINT WORD

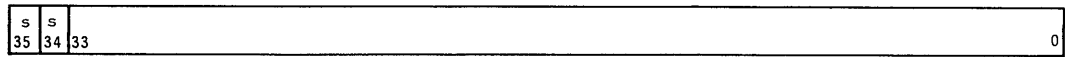


A

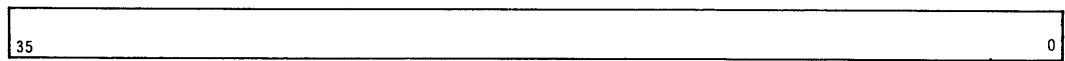


A + 1

FIXED-POINT INTEGER MULTIPLY RESULT

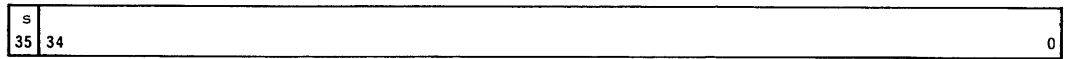


A

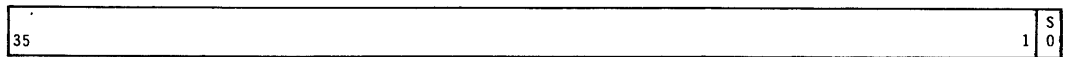


A + 1

FIXED-POINT FRACTIONAL MULTIPLY RESULT

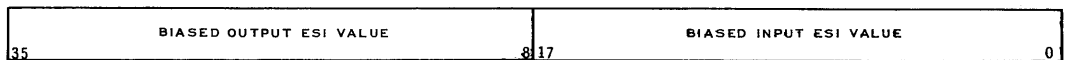


A



A + 1

BIASED ESI VALUES IN IACR'S



APPENDIX C. INSTRUCTION REPERTOIRE BY FUNCTION CODE

Function Code (Octal)		Mnemonic	Instruction	Description	Execution Time* in μ sec. With Overlap
f	j				
00	—	—	Illegal Code	Causes illegal instruction interrupt	—
01	0-15	SA	Store A	$(A) \rightarrow U$	1.5
02	0-15	SN,SNA	Store Negative A	$-(A) \rightarrow U$	1.5
03	0-15	SM,SMA	Store Magnitude A	$ (A) \rightarrow U$	1.5
04	0-15	S,SR	Store R	$(R_a) \rightarrow U$	1.5
05	0-15	SZ	Store Zero	ZEROS $\rightarrow U$	1.5
06	0-15	S,SX	Store X	$(X_a) \rightarrow U$	1.5
07	—	—	Illegal Code	Causes illegal instruction interrupt to address 241_8	1.5
10	0-17	L,LA	Load A	$(U) \rightarrow A$	1.5
11	0-17	LN,LNA	Load Negative A	$-(U) \rightarrow A$	1.5
12	0-17	LM,LMA	Load Magnitude A	$ (U) \rightarrow A$	1.5
13	0-17	LNMA	Load Negative Magnitude A	$- (U) \rightarrow A$	1.5
14	0-17	A,AA	Add to A	$(A) + (U) \rightarrow A$	1.5
15	0-17	AN,ANA	Add Negative To A	$(A) - (U) \rightarrow A$	1.5
16	0-17	AM,AMA	Add Magnitude To A	$(A) + (U) \rightarrow A$	1.5
17	0-17	ANM,ANMA	Add Negative Magnitude to A	$(A) - (U) \rightarrow A$	1.5
20	0-17	AU	Add Upper	$(A) + (U) \rightarrow A + 1$	1.5
21	0-17	ANU	Add Negative Upper	$(A) - (U) \rightarrow A + 1$	1.5
22	0-15	BT	Block Transfer	$(X_x + u) \rightarrow X_a + u$, repeat K times	$3.5 + 3.0K$
23	0-17	L,LR	Load R	$(U) \rightarrow R_a$	1.5
24	0-17	A,AX	Add To X	$(X_a) + (U) \rightarrow X_a$	1.5
25	0-17	AN,ANX	Add Negative To X	$(X_a) - (U) \rightarrow X_a$	1.5
26	0-17	LXM	Load X Modifier	$(U) \rightarrow X_{a_{17-0}}$; $X_{a_{35-18}}$ unchanged	1.666
27	0-17	L,LX	Load X	$(U) \rightarrow X_a$	1.5
30	0-17	MI	Multiply Integer	$(A) \cdot (U) \rightarrow A, A+1$	3.666

*See notes at end of table.

Function Code (Octal)		Mnemonic	Instruction	Description	Execution Time* in μ sec. With Overlap
f	j				
31	0-17	MSI	Multiply Single Integer	$(A) \cdot (U) \rightarrow A$	3.666
32	0-17	MF	Multiply Fractional	$(A) \cdot (U) \rightarrow A, A+1$	3.666
33	-	-	Illegal Code	Causes illegal instruction interrupt to address 241 ₈	-
34	0-17	DI	Divide Integer	$(A, A+1) \div (U) \rightarrow A; \text{REMAINDER} \rightarrow A+1$	13.950
35	0-17	DSF	Divide Single Fractional	$(A) \div (U) \rightarrow A+1$	13.950
36	0-17	DF	Divide Fractional	$(A, A+1) \div (U) \rightarrow A; \text{REMAINDER} \rightarrow A+1$	13.950
37	-	-	Illegal Code	Causes illegal instruction interrupt to address 241 ₈	-
40	0-17	OR	Logical OR	$(A) \text{ OR } (U) \rightarrow A+1$	1.5
41	0-17	XOR	Logical Exclusive OR	$(A) \text{ XOR } (U) \rightarrow A+1$	1.5
42	0-17	AND	Logical AND	$(A) \text{ AND } (U) \rightarrow A+1$	1.5
43	0-17	MLU	Masked Upper Load	$[(U) \text{ AND } (R2)] \text{ OR } [(A) \text{ AND } (R2)'] \rightarrow A+1$	1.5
44	0-17	TEP	Test Even Parity	Skip NI if $(U) \text{ AND } (A)$ have even parity	3.00/2.166
45	0-17	TOP	Test Odd Parity	Skip NI if $(U) \text{ AND } (A)$ have odd parity	3.00/2.166
46	0-17	LXI	Load X Increment	$(U) \rightarrow X_{a35-18}; X_{a17-0}$ unchanged	1.833
47	0-17	TLEM	Test Less Than or Equal to Modifier	Skip NI if $(U) \leq (X_a)_{17-0}$;	3.333/1.833
		TNGM	Test Not Greater Than Modifier	always $(X_a)_{17-0} + (X_a)_{35-18} \rightarrow (X_a)_{17-0}$	
50	0-17	TZ	Test Zero	Skip NI if $(U) = \pm 0$	3.166/1.666
51	0-17	TNZ	Test Nonzero	Skip NI if $(U) \neq \pm 0$	3.166/1.666
52	0-17	TE	Test Equal	Skip NI if $(U) = (A)$	3.166/1.666
53	0-17	TNE	Test Not Equal	Skip NI if $(U) \neq (A)$	3.166/1.666
54	0-17	TLE	Test Less Than or Equal	Skip NI if $(U) \leq (A)$	3.166/1.66
		TNG	Test Not Greater		
55	0-17	TG	Test Greater	Skip NI if $(U) > (A)$	3.166/1.666
56	0-17	TW	Test Within Range	Skip NI if $(A) < (U) \leq (A+1)$	3.33/1.66

*See notes at end of table.

Function Code (Octal)		Mnemonic	Instruction	Description	Execution Time* in μ sec. With Overlap
f	j				
57	0-17	TNW	Test Not Within Range	Skip NI if $(U) \leq (A)$ or $(U) > (A+1)$	3.33/1.66
60	0-17	TP	Test Positive	Skip NI if $(U)_{35} = 0$	3.0/1.5
61	0-17	TN	Test Negative	Skip NI if $(U)_{35} = 1$	3.0/1.5
62	0-17	SE	Search Equal	Skip NI if $(U) = (A)$, else repeat	3.5+1.5K
63	0-17	SNE	Search Not Equal	Skip NI if $(U) \neq (A)$, else repeat	3.5+1.5K
64	0-17	SLE SNG	Search Less Than or Equal Search Not Greater	Skip NI if $(U) \leq (A)$, else repeat	3.5+1.5K
65	0-17	SG	Search Greater	Skip NI if $(U) > (A)$, else repeat	3.5+1.5K
66	0-17	SW	Search Within Range	Skip NI if $(A) < (U) \leq (A+1)$, else repeat	3.5+1.5K
67	0-17	SNW	Search Not Within Range	Skip NI if $(U) \leq (A)$ or $(U) > (A+1)$, else repeat	3.5+1.5K
70	†	JGD	Jump Greater and Decrement	Jump to U if $(\text{Control Register})_{ja} > 0$; go to NI if $(\text{Control Register})_{ja} \leq 0$; always $(\text{Control Register})_{ja} - 1 \rightarrow \text{Control Register}_{ja}$	3.0/1.5
71	00	MSE	Mask Search Equal	Skip NI if $(U) \text{ AND } (R2) = (A) \text{ AND } (R2)$, else repeat	3.5+1.5K
71	01	MSNE	Mask Search Not Equal	Skip NI if $(U) \text{ AND } (R2) \neq (A) \text{ AND } (R2)$, else repeat	3.5+1.5K
71	02	MSLE	Mask Search Less Than or Equal	Skip NI if $(U) \text{ AND } (R2) \leq (A) \text{ AND } ((R2))$, else repeat	3.5+1.5K
		MSNG	Mask Search Not Greater		
71	03	MSG	Mask Search Greater	Skip NI if $(U) \text{ AND } (R2) > (A) \text{ AND } (R2)$, else repeat	3.5+1.5K
71	04	MSW	Masked Search Within Range	Skip NI if $(A) \text{ AND } (R2) < (U) \text{ AND } (R2) \leq (A+1) \text{ AND } (R2)$, else repeat	3.5+1.5K
71	05	MSNW	Masked Search Not Within Range	Skip NI if $(U) \text{ AND } (R2) \leq (A) \text{ AND } (R2)$ or $(U) \text{ AND } (R2) > (A+1) \text{ AND } (R2)$, else repeat	3.5+1.5K
71	06	MASL	Masked Alphanumeric Search Less Than or Equal	Skip NI if $(U) \text{ AND } (R2) \leq (A) \text{ AND } (R2)$, else repeat	3.5+1.5K

*See notes at end of table.

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

Function Code (Octal)		Mnemonic	Instruction	Description	Execution Time* in μ sec. With Overlap
f	j				
71	07	MASG	Masked Alphanumeric Search Greater	Skip NI if (U) AND (R2) > (A) AND (R2), else repeat	3.5+1.5K
71	10	DA	Double Precision Fixed Point Add	(A,A+1) + (U,U+1) \rightarrow A,A+1	3.167
71	11	DAN	Double Precision Fixed Point Add Negative	(A,A+1) - (U,U+1) \rightarrow A,A+1	3.167
71	12	DS	Double Store A	(A,A+1) \rightarrow U,U+1	3.0
71	13	DL	Double Load A	(U,U+1) \rightarrow A,A+1	3.0
71	14	DLN	Double Load Negative A	-(U,U+1) \rightarrow A,A+1	3.0
71	15	DLM	Double Load Magnitude A	(U,U+1) \rightarrow A,A+1	3.0
71	16	DJZ	Double Precision Zero Jump	Jump to U if (A,A+1) = \pm 0; go to NI if (A,A+1) \neq \pm 0	3.167/1.667
71	17	DTE	Double Precision Test Equal	Skip NI if (U,U+1) = (A,A+1)	4.667/3.167
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] \rightarrow U ₁₇₋₀ ; jump to U+1	3.83
72	02	JPS	Jump Positive and Shift	Jump to U if (A) ₃₅ =0; go to NI if (A) ₃₅ =1; always shift (A) left circularly one bit position	3.0/1.5
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	3.0/1.5
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	1.5
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	1.5
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	1.5
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	1.5

*See notes at end of table.

Function Code (Octal)		Mnemonic	Instruction	Description	Execution Time* in μ sec. With Overlap
f	j				
72	10	EX	Execute	Execute the instruction at U	1.5
72	11	ER	Executive Return	Interrupt to address 242_8	2.33
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	1.5
72	14	SCN	Store Channel Number	If a=0: CHANNEL NUMBER \rightarrow U_{3-0} ; If a=1: CHANNEL NUMBER \rightarrow U_{3-0} and CPU NUMBER \rightarrow U_{5-4}	1.5
72	15	LPS	Load Processor State Register	(U) \rightarrow PSR	1.5
72	16	LSL	Load Storage Limits Register	(U) \rightarrow SLR	1.5
72	17	—	Illegal Code	Causes illegal instruction interrupt to address 241_8	—
73	00	SSC	Single Shift Circular	Shift (A) right circularly U places	1.5
73	01	DSC	Double Shift Circular	Shift (A,A+1) right circularly U places	1.5
73	02	SSL	Single Shift Logical	Shift (A) right U places; zerofill	1.5
73	03	DSL	Double Shift Logical	Shift (A,A+1) right U places; zerofill	1.5
73	04	SSA	Single Shift Algebraic	Shift (A) right U places; signfill	1.5
73	05	DSA	Double Shift Algebraic	Shift (A,A+1) right U places; signfill	1.666
73	06	LSC	Load Shift and Count	(U) \rightarrow A; shift (A) left circularly until $(A)_{35} \neq (A)_{34}$, NUMBER OF SHIFTS \rightarrow A+1	2.0
73	07	DLSC	Double Load Shift and Count	(U,U+1) \rightarrow A,A+1; Shift (A,A+1) left circularly until $(A,A+1)_{71} \neq (A,A+1)_{70}$; NUMBER OF SHIFTS \rightarrow A+2	3.830
73	10	LSSC	Left Single Shift Circular	Shift (A) left circularly U places	1.5
73	11	LDSC	Left Double Shift Circular	Shift (A,A+1) left circularly U places	1.666

*See notes at end of table.

Function Code (Octal)		Mnemonic	Instruction	Description	Execution Time* in μ sec. With Overlap
f	j				
73	12	LSSL	Left Single Shift Logical	Shift (A) left U places; zerofill	1.5
73	13	LDSL	Left Double Shift Logical	Shift (A,A+1) left U places; zerofill	1.666
73	14	III (a=0 or 1)	Initiate Interprocessor Interrupt	Initiate Interprocessor Interrupt	1.5
		ALRM (a=10 ₈)	Alarm	Turn on alarm	1.5
		EDC (a=11 ₈)	Enable Day Clock	Enable day clock	1.5
		DDC (a=12 ₈)	Disable Day Clock	Disable day clock	1.5
73	15	SIL	Select Interrupt Location	(A) ₂₋₀ → MSR	1.5
73	16	LCR (a=0)	Load Channel Select Register	(U) ₃₋₀ → CSR	1.666
		LLA (a=1)	Load Last Address Register	(U) ₂₋₀ → LAR	1.666
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 → U ₃₅₋₃₀ ; U ₂₉₋₀ unchanged	3.166/1.666
74	00	JZ	Jump Zero	Jump to U if (A)=±0; go to NI if (A) ≠ ±0	3.0/1.5
74	01	JNZ	Jump Nonzero	Jump to U if (A) = ±0; go to NI if (A) ≠ ±0	3.0/1.5
74	02	JP	Jump Positive	Jump to U if (A) ₃₅ = 0; go to NI if (A) ₃₅ = 1	3.0/1.5
74	03	JN	Jump Negative	Jump to U if (A) ₃₅ = 1; go to NI if (A) ₃₅ = 0	3.0/1.5
74	04	JK J	Jump Keys Jump	Jump to U if a=0 or if a=lit SELECT JUMP indicator; go to NI if neither is true	1.5
74	05	HKJ HJ	Halt Keys and Jump Halt Jump	Stop if a=0 or if [a OR lit SELECT STOP indicators] ≠ 0; on restart or continuation, jump to U	1.5
74	06	NOP	No Operation	Proceed to next instruction	1.5

*See notes at end of table.

Function Code (Octal)		Mnemonic	Instruction	Description	Execution Time* in μ sec. With Overlap
f	j				
74	07	AAIJ	Allow All I/O Interrupts and Jump	Allow all I/O interrupts and jump to U	1.5
74	10	JNB	Jump No Low Bit	Jump to U if $(A)_0 = 0$; go to NI if $(A)_0 = 1$	3.0/1.5
74	11	JB	Jump Low Bit	Jump to U if $(A)_0 = 1$; go to NI if $(A)_0 = 0$	3.0/1.5
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 $(X_a)_{17-0} + (X_a)_{35-18} \rightarrow X_a_{17-0}$	3.166/1.5
74	13	LMJ	Load Modifier and Jump	(P) – Base Address Modifier [BI or BD] $\rightarrow X_a_{17-0}$; jump to U	1.666
74	14	JO	Jump Overflow	Jump to U if D1 of PSR = 1; go to NI if D1 = 0	3.0/1.5
74	15	JNO	Jump No Overflow	Jump to U if D1 of PSR = 0; go to NI if D1 = 1	3.0/1.5
74	16	JC	Jump Carry	Jump to U if D0 of PSR = 1; go to NI if D0 = 0	3.0/1.5
74	17	JNC	Jump No Carry	Jump to U if D0 of PSR = 0; go to NI if D0 = 1	3.0/1.5
75	00	LIC	Load Input Channel	For channel [a OR CSR]: (U) \rightarrow IACR; set input active; clear input monitor	1.5
75	01	LICM	Load Input Channel and Monitor	For channel [a OR CSR]: (U) \rightarrow IACR; set input active; set input monitor	1.5
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	1.5
75	03	DIC	Disconnect Input Channel	For channel [a OR CSR]: clear input active; clear input monitor	1.5
75	04	LOC	Load Output Channel	For channel [a OR CSR]: (U) \rightarrow OACR; set output active; clear output monitor; clear external function (ISI only)	1.5

*See notes at end of table.

Function Code (Octal)		Mnemonic	Instruction	Description	Execution Time* in μ sec. With Overlap
f	j				
75	05	LOCM	Load Output Channel and Monitor	For channel [a OR CSR]: (U) \rightarrow OACR; set output monitor; clear external function (ISI only)	1.5
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	1.5
75	07	DOC	Disconnect Output Channel	For channel [a OR CSR]: clear output active; clear output monitor; clear external function	1.5
75	10	LFC	Load Function in Channel	For channel [a OR CSR]: (U) \rightarrow OACR; set output active (ISI only), external function, and force external function; clear output monitor (ISI only)	1.5
75	11	LFCM	Load Function in Channel and Monitor	For channel [a OR CSR]: (U) \rightarrow OACR; set output active (ISI only), external function, force external function, and output monitor (ISI only)	1.5
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	1.5
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	1.5
75	14	AACI	Allow All Channel External Interrupts	Allow all external interrupts	1.5
75	15	PACI	Prevent All Channel External Interrupts	Prevent all external interrupts	1.5
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	1.5
75	17	-	Illegal Code		
76	00	FA	Floating Add	(A) + (U) \rightarrow A; RESIDUE \rightarrow A + 1	3.0
76	01	FAN	Floating Add Negative	(A) - (U) \rightarrow A; RESIDUE \rightarrow A + 1	3.0
76	02	FM	Floating Multiply	(A) * (U) \rightarrow A, A + 1	4.0
76	03	FD	Floating Divide	(A) \div (U) \rightarrow A; REMAINDER \rightarrow A + 1	11.5**
76	04	LUF	Load and Unpack Floating	{(U)} ₃₄₋₂₇ \rightarrow A ₇₋₀ , zerofill; (U) ₂₆₋₀ \rightarrow A + 1 ₂₆₋₀ , signfill	1.5

*See notes at end of table.

Function Code (Octal) f j	Mnemonic	Instruction	Description	Execution Time* in μ sec. With Overlap
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] $\rightarrow A + 1_{34-27}$	2.0
76 06	MCDU	Magnitude of Characteristic Difference To Upper	$ (A)_{35-27} - (U)_{35-27} \rightarrow A + 1_{8-0}$; ZEROS $\rightarrow A + 1_{35-9}$	1.5
76 07	CDU	Characteristic Difference To Upper	$ (A)_{35-27} - (U)_{35-27} \rightarrow A + 1_{8-0}$; SIGN BITS $\rightarrow A + 1_{35-9}$	1.5
76 10	DFA	Double Precision Floating Add	$(A, A + 1) + (U, U + 1) \rightarrow A, A + 1$	4.5
76 11	DFAN	Double Precision Floating Add Negative	$(A, A + 1) - (U, U + 1) \rightarrow A, A + 1$	4.5
76 12	DFM	Double Precision Floating Multiply	$(A, A + 1) \cdot (U, U + 1) \rightarrow A, A + 1$	6.667
76 13	DFD	Double Precision Floating Divide	$(A, A + 1) \div (U, U + 1) \rightarrow A, A + 1$	24.0***
76 14	DFU	Double Load and Unpack Floating	$ (U)_{34-24} \rightarrow A_{10-0}$, zerofill; $(U)_{23-0} \rightarrow A + 1_{23-0}$, signfill; $(U+1) \rightarrow A + 2$	3.0
76 15	DFP	Double Load and Convert to Floating	$(U)_{35} \rightarrow A + 1_{35}$; [NORMALIZED $(U, U + 1)_{59-0} \rightarrow A + 1_{23-0}$ and $A + 2$. If $(U)_{35} = 0$, $(A)_{10-0} \pm$ NORMALIZING COUNT $\rightarrow A + 1_{34-24}$. If $(U)_{35} = 1$, ones complement of $[(A)_{10-0} \pm$ NORMALIZING COUNT] $\rightarrow A + 1_{34-24}$	3.830
76 16	FEL	Floating Expand and Load	If $(U)_{35} = 0$, $(U)_{35-27} + 1600_8 \rightarrow A_{35-24}$; if $(U)_{35} = 1$, $(U)_{35-27} - 1600_8 \rightarrow A_{35-24}$; $(U)_{26-3} \rightarrow A_{23-0}$; $(U)_{2-0} \rightarrow A + 1_{35-33}$; $(U)_{35} \rightarrow A + 1_{32-0}$	1.833
76 17	FCL	Floating Compress and Load	If $(U)_{35} = 0$, $(U)_{35-24} - 1600_8 \rightarrow A_{35-27}$; if $(U)_{35} = 1$, $(U)_{35-24} + 1600_8 \rightarrow A_{35-27}$; $(U)_{23-0} \rightarrow A_{26-3}$; $(U + 1)_{35-33} \rightarrow A_{2-0}$	3.167
77 0-17	-	Illegal Code	Causes illegal instruction interrupt to address 241_8	-

*See notes at end of table.

NOTES:

- * Execution times given are calculated using a core memory cycle time of 1.5 microseconds and a CPU clock cycle time of 166 nanoseconds.

For all comparison instructions, the first number represents the skip or jump condition, the second number is for a no skip or no jump condition.

For function codes 01 through 67, add .333 microseconds to execution times for 6-bit, 9-bit, and 12-bit writes.

Execution time for the Block Transfer and the search instructions depends on the number of repetitions of the instruction required. The variance is 3.0K microseconds for Block Transfer and 1.5K microseconds for searches where K equals the number of repetitions; that is, K equals the number of words in the block being transferred or the number of words searched before a match is found.

- ** If 28 instead of 27 subtractions are performed, add .333 microseconds.
- *** If 61 instead of 60 subtractions are performed, add .333 microseconds.

APPENDIX D. MATH-PACK ROUTINES

The following is a complete listing of STAT-PACK routines grouped according to function:

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

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

SOLUTION OF EQUATIONS

NEWTIT – Newton-Raphson Iteration

WEGIT – Wegstein Iteration

AITIT – Aitken Iteration

ROOTCP – Real and Complex Roots of Real or Complex Polynomial

DIFFERENTIATION

DERIV1 – First Derivative Approximation

DERIV2 – Second Derivative Approximation

NTHDER – Nth Derivative of a Polynomial

POLYNOMIAL MANIPULATION

GIVZRS – Polynomial Coefficients Given its Zeros

CVALUE – Complex Polynomial Evaluation

POLYX – Real Polynomial Multiplication

CPOLYX – Complex Polynomial Multiplication

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

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

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

MATRIX MANIPULATION: MISCELLANEOUS

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

PMXTRI – Polynomial Matrix Triangularization

SCALE – Polynomial Matrix Scaling

MXROT – Matrix Rotation

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

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

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

PSUEDO-RANDOM NUMBER GENERATORS

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

RANDU – Uniform Distribution

RANDN – Normal Distribution

RANDEX – Exponential Distribution

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 grouped according to function:

DESCRIPTIVE STATISTICS

- FREQP – Frequency Polygon
- HIST – Histogram
- MHIST – Multivariate Histogram
- GROUP – Grouping of Data

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
- SHEPCOR – Sheppard's Corrections
- KURSK – Skewness and Kurtosis

DISTRIBUTION, FITTING AND PLOTTING

BINOM – Binomial Distribution

POISON – Poisson Distribution

HYPER – Hypergeometric Distribution

PNORM – Normal Distribution

AFSER – Arne Fisher Series

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 Homogeneity

CHICNT – CHI-Square Test for Independence

GENGOF – CHI-Square Test of General Goodness of Fit

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

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

ANALYSIS OF VARIANCE

- ANOVA1 – One-Way Cross Classification
- ANOVA2 – Two-Way Cross Classification
- ANOVA3 – Three-Way Cross Classification
- MISDAT – Missing Data
- VTRANS – Variable Transformations
- ANOVRB – Randomized Blocks
- ANOVL5 – Latin Squares
- ANOVSF – Split Plot Design
- ANOSPF – Split-Split Plot Design
- ANOVN2 – Two-Way Nested Design
- ANOVN3 – Three-Way Nested Design
- ANOCO – Analysis of Covariance
- GLH – General Linear Hypotheses

REGRESSION ANALYSIS

- RESTEM – Stepwise Multiple Regression
- REBSOM – Back Solution Multiple Regression
- CORAN – Correlation Analysis

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

MULTI-VARIATE ANALYSIS

GENVAR – Generalized Variance
DISHOT – Hotelling's Distribution
DSQ – Mahalanobis' Distribution
SIGTMN – Significance of a Set of Means
DISCRA – Discriminant Analysis
FACTAN – Factor and Principal Component Analysis

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

INVERSE DISTRIBUTION FUNCTIONS

TINORM – Inverse Normal Distribution

STUDIN – Inverse Student's Distribution

FISHIN – Inverse Fisher's Distribution

CHIN – Inverse CHI-Square Distribution

MISCELLANEOUS

PLOT1 – Plot of One Line

JIM – Matrix Inversion

MXMTLT – Left Multiplication of a Matrix
by its Transpose

