

*reference  
manual*

**UNIVAC**  
S C I E N T I F I C

**1103A**

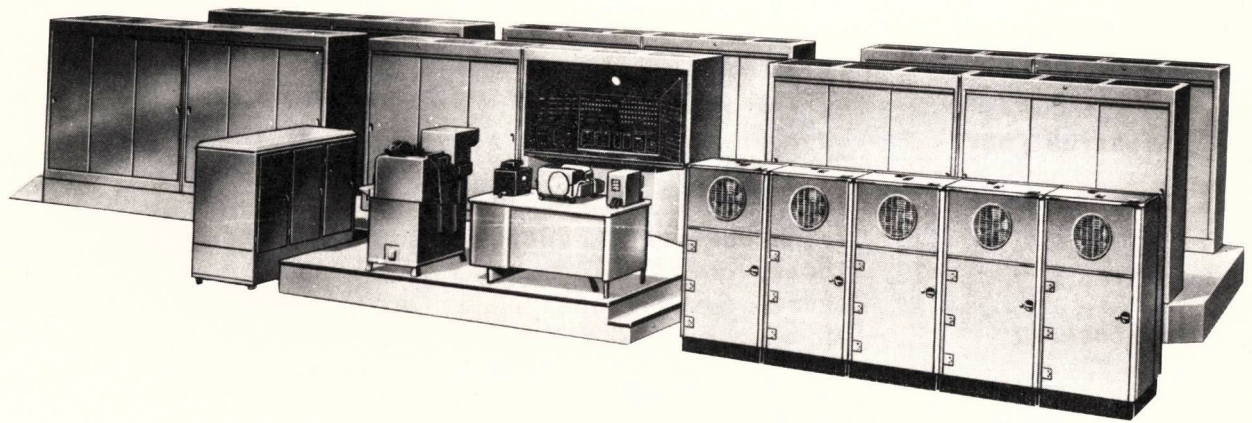
**COMPUTER**

*Remington Rand Univac*  
DIVISION OF SPERRY RAND CORPORATION

*reference manual*

**UNIVAC**  
**S C I E N T I F I C**  
**1103A**  
**COMPUTER**

COPYRIGHT © 1956  
SPERRY RAND CORPORATION



## PREFACE

The Univac Scientific, Model 1103A, is the computer element of a large scale data reduction-computation system. It is the newest of a series of type 1103 computers which were first placed in operation several years ago. The Model 1103A was developed to satisfy the increasing need of computer users for a large rapid access storage, fast input-output, and greater speed of internal operations. The features incorporated in this computer are a result of continuing research in Remington Rand laboratories and extensive application experience gained at various installations.

All type 1103 computers have the same general characteristics: They are large-scale, parallel, binary computers generally intended for applications requiring extremely high internal operating speeds and a large repertoire of logical as well as arithmetic commands.

The first type 1103 computer was delivered to the Bureau of Ships several years ago: in 1954 several commercial versions which contained 1,024 words of electrostatic storage were delivered to Consolidated Vultee Aircraft, Elgin Air Force Base, White Sands Proving Ground, Ramo-Wooldridge Corporation, and Westinghouse. These were known as the ERA 1103 computers. In 1955, delivery of a later version of the ERA 1103 computer began. These later versions were logically the same as the first computers, but incorporated an important engineering development: magnetic cores replaced the 1,024 word electrostatic memory. The introduction of magnetic cores made the ERA 1103 the first computer system to be delivered using this improved form of high speed memory. Among the organizations receiving this version of the 1103 were the Operations Research Office of Johns Hopkins University, Wright Air Development Center, Lewis Flight Propulsion Laboratory of the National Advisory Committee for Aeronautics and Remington Rand Univac, St. Paul, Minnesota.

The Univac Scientific Model 1103A is an expansion of the earlier ERA 1103 computer system with several significant engineering and logical improvements. One of the improvements made to the 1103A is that the magnetic core storage has been expanded to 4,096 words of magnetic core storage as standard equipment, with one or two additional banks of 4,096 word storage available as optional equipment. Thus, 4,096, 8,192, or 12,288 words of high speed storage are available at the option of the user. It should be noted that the improvements incorporated into the 1103A do not make obsolete the programming work already completed by users of the 1103. Organizations which will receive an 1103A installation are the Missile Systems Division of Lockheed Aircraft Company, Boeing Airplane Company, Holloman Air Force Base (two 1103A's), Ramo-Wooldridge Corporation, Applied Physics Laboratory of Johns Hopkins University, and Wright Air Development Center.

This manual is intended to provide those persons familiar with large scale computers with a comparison of the ERA 1103 and the Univac Scientific Model 1103A.

## Table of Contents

	page
INTRODUCTION . . . . .	1
GENERAL DESCRIPTION . . . . .	3
GENERAL THEORY OF OPERATION . . . . .	7
REPertoire OF INSTRUCTIONS . . . . .	27
INSTRUCTION EXECUTION TIMES . . . . .	35
FLOATING POINT ARITHMETIC . . . . .	37

## INTRODUCTION

The Univac Scientific, Model 1103A, is a general-purpose digital computer system similar to the Univac Scientific, Model 1103, but differing in the following features:

MODEL 1103	MODEL 1103A
1. Rapid Access Storage - 1,024 words of electrostatic or magnetic core storage.	1. Rapid Access Storage - 4,096 words of magnetic core storage (8,192 or 12,288 words of magnetic core storage are optional).
2. Magnetic Tape Storage - four Raytheon or Potter units.	2. Magnetic Tape Storage - up to ten Uniservo units.
3. No equivalent.	3. Program Interrupt.
4. No equivalent.	4. Left Transmit Instruction.
5. No equivalent.	5. Floating Point Arithmetic (optional).

In order to provide a set of consecutive addresses for the larger core storage of the 1103A, the address structure differs from that of the 1103 as follows:

Storage Class	Addresses in Octal Notation	
	1103	1103A
Magnetic Core	00000-01777	00000-07777 (4,096 registers ) 00000-17777 (8,192 registers ) 00000-27777 (12,288 registers)
Q-Register	10000-17777	31000-31777
Accumulator	20000-27777	32000-37777
Magnetic Drum	40000-77777	40000-77777

Up to ten Uniservo units are available for bulk storage of information or fast input-output. The transfer rate between magnetic tape and magnetic core storage is 2,130 words per second. The various reading, writing, and positioning operations of the Uniservos are initiated by an External Function instruction. Use of the Univac magnetic tapes permits off-line processing of information by such Univac equipments as: the Unityper II, the Univac High-Speed Printer (600 lines per minute), Card-to-Tape and Tape-to-Card Converters, etc. Thus the computer is freed for other problems while much of the data processing is performed. Since all instructions found in the earlier 1103's (except magnetic tape instructions) are present in the 1103A, it is

a simple matter to modify programs written for the 1103 computers for running on the 1103A. The computer itself may be used to change references to the magnetic tapes and the A & Q registers.

By means of the Program Interrupt which is a standard feature on all 1103A computers, a program which is running on the computer may be interrupted and control automatically transferred to a different program. Upon completing the interrupting program, the computer will resume execution of the original program at the point of interruption. The interrupting program may be that of a high priority problem; in this case the process of stopping the running program. Storing it in its current state, loading the high priority program and finally re-loading the interrupted program is reduced from minutes to microseconds. Use of the program interrupt by asynchronous input-output equipment eliminates the need for careful timing of computing operations and input-output operations, and permits maximum use of the time available for computation between external operations. Among other things, the program interrupt may be used to alter quickly and easily the course of a running program on the basis of displayed output.

As an optional feature of the 1103A, an additional floating point package will be provided to perform floating point arithmetic. Automatically sequenced instructions are available for addition, subtraction, multiplication, division, polynomial multiply and inner product. Three other instructions facilitate floating point-fixed point conversion and double precision floating point operations.

## GENERAL DESCRIPTION

The Univac Scientific, Model 1103A, is a general-purpose digital computing system used to perform a wide variety of large-scale calculations. It is suitable for applications requiring large-storage capacity, fast input and output, high operating speed, and programming versatility. The system is adaptable to many special purpose applications, including simulation and control in real time.

The computer's internal memory consists of 16,384 registers of magnetic drum storage, 4,096, 8,192 or 12,288 registers of magnetic core storage. Each of these registers is individually addressed and directly accessible. Supplementary storage is provided by up to ten Uniservo magnetic tape units.

The computer performs 41 different arithmetic and logical operations. It is fully automatic in that the sequence of operations is determined by a program of internally stored instructions capable of self-modification. To attain high computing speed, the computer operates in the parallel mode, i. e. all digits of a number are operated upon simultaneously. Internal arithmetic operations are in the binary system. The basic word size is 36 binary digits, or 'bits'. A word may be an instruction, a number, or an arbitrarily coded quantity.

A system of 'two address' logic is employed. An instruction word consists of a 6-bit operation code and two 15-bit execution addresses. The operation code specifies which of the 41 possible operations is to be performed. The functions of the execution addresses are different for the various types of instructions, but, in general, they specify registers in the memory from which operands are obtained or in which results are stored.

The computer employs a 'one's complement' system of notation in which the left-most bit of a number is the sign bit of that number and the binary point is considered to be to the right of the lowest order bit. Thus, if the left-most bit of a number is '0' the number is said to be positive; if it is a '1' the number is said to be negative. In the one's complement system a negative number can be obtained by complementing all the bits (including the sign bit) of the corresponding positive number. In the single length, or 36-bit, register, integers from  $1-2^{35}$  up to  $2^{35}-1$  may be represented; in the double length, or 72-bit, Accumulator integers from  $1-2^{71}$  up to  $2^{71}-1$  may be represented. In one's complement notation, each of these integers has a unique representation with the exception of zero; zero has both a negative and a positive notation; however, because of the nature of the arithmetic operations in the Univac Scientific, a negative zero cannot be generated as the result of arithmetic operations.



Integers not lying in the above range as well as fractional quantities that may occur in certain problems can also be handled by suitably scaling such quantities so that the resulting quantity can be represented by machine numbers. This scaling can either be accomplished using the Scale Factor instruction or by using floating point arithmetic routines.

Standard input and output equipments of the Univac Scientific system are a photoelectric paper tape reader, a typewriter, and a high speed paper tape punch. Communication with a wide variety of optional devices is made possible by use of an 8-bit input-output register, IOA, and a 36-bit input-output register, IOB. The choice of optional equipment is determined by the user's needs. Frequently included are: Univac magnetic tape units, a card reader and card punch, a line printer, and an oscilloscope display unit. Directly connected devices may also include: Teletype communication circuits, analog-to-digital converters, signal circuits to process-actuating mechanisms, etc. The computer's input-output system permits simultaneous use of several external units and allows computation to proceed while such terminal equipments are operating. The use of Unitapes permits off-line processing of information by a variety of Univac auxiliary equipments.

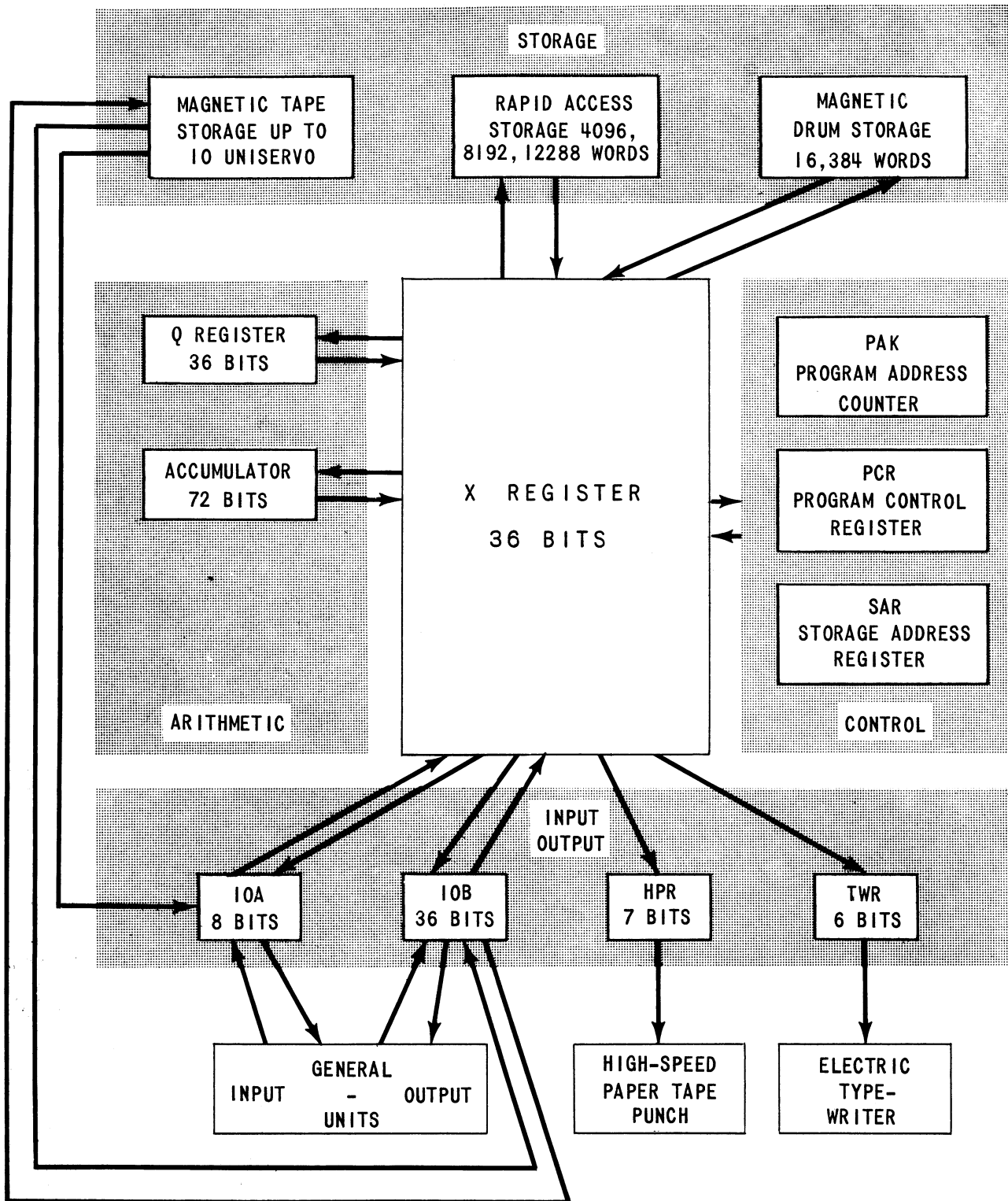


DIAGRAM OF THE I103A COMPUTER SYSTEM

## GENERAL THEORY OF OPERATION

The computer is divided into four basic sections as follows:

- (1) *INPUT-OUTPUT* - Effects insertion of information into the Storage Section and delivers the results of a computation to external devices.
- (2) *STORAGE* - Stores information within the computer.
- (3) *ARITHMETIC* - Performs digital manipulations to accomplish arithmetic and logical operations.
- (4) *CONTROL* - Interprets the program and directs its execution.

The general features of each of the above systems and the manner in which each system functions with the others in the execution of a program is discussed in the following paragraphs. The reader is also referred to the diagram of the 1103A computer system on page 5.

*PRINCIPAL MULTIPURPOSE REGISTERS.* - The X-Register, the Q-Register, and the Accumulator each perform several different functions. These registers collectively are referred to as the '*arithmetic registers*'; however, each has additional functions not associated with arithmetic functions. Brief descriptions of the registers and their functions are given below.

(1) *X-REGISTER* - The X-Register, X, is a 36-bit flip-flop register which can be complemented. It has two functions:

(a) As an '*exchange register*', X handles nearly all internal transmissions of words between various sections of the computer.

(b) As an '*arithmetic register*', X holds the addend, subtrahend, multiplicand, and divisor in the corresponding arithmetic operations.

(2) *Q-REGISTER* - The Q-Register, Q, is a 36-bit flip-flop register with shifting properties. It has two functions:

(a) As a '*storage register*', Q provides storage for a single 36-bit word. It is individually addressed; its address is 31 --- (only the first two octal digits are of significance).

(b) As an '*arithmetic register*', Q holds the multiplier, quotient, and logical multiplier in the corresponding arithmetic operations.

(3) *ACCUMULATOR* - The Accumulator, A, is a 72-bit flip-flop register with shifting properties. This register is subtractive in nature. A is logically divided into two halves termed  $A_R$  (A-right) and  $A_L$  (A-left). The 36-higher-order stages give A double precision properties. The algebraic sign normally is carried by  $A_L$ . The transmissive links to X are from  $A_R$  and  $A_L$ ; in the Left Transmit instruction the transmission from  $A_L$  to X is accomplished. A has two functions:

(a) As a 'storage register', A provides storage for either the single-extension (36 bits) or double-extension (72 bits) of a single computer word. A is individually addressed: its address is 32--- (only the first two octal digits are of significance).

(b) As an 'arithmetic register', A holds the sum, difference, product, and dividend (and remainder) in the corresponding arithmetic operations. In additions and multiplications, sums and products may be accumulated up to 72 bits.

### INPUT - OUTPUT SECTION

Certain equipment is considered basic. This includes a photoelectric paper tape reader, a directly-connected electric typewriter, and a high-speed paper tape punch. In addition, communication with optional devices is possible through use of the Input-Output Registers IOA and IOB. Information is transmitted between these buffer registers and the X-Register under program control. In addition to transferring information to and from the computer, the Input-Output System synchronizes the computer with external equipment which is operating basically under its own control. The Input-Output System permits simultaneous use of many external units and allows computation to proceed while such terminal equipment is operating. The system will also detect faults in operation due to programming errors or failures in the external equipment.

#### (1) INPUT-OUTPUT SYSTEM

The Input-Output System consists of the IOA and IOB registers, lockout circuits for both IOA and IOB, the external equipment, and the external equipment control. Program control of the external equipment is achieved with the following instructions:

##### *External Function (EF-v)*

This instruction selects a unit of external equipment and designates the operating mode in case the unit has several modes of operation. The selection is by means of a code word transmitted to IOB.

*External Read (ERjv)*

This instruction transfers information from IOA or IOB (j=0 or j=1) to storage location v.

*External Write (EWjv)*

This instruction transfers information from storage location v to IOA or IOB (j=0 or j=1).

(a) *Input-Output Registers, IOA and IOB.*

Both IOA and IOB serve as buffer storage registers to receive data from and supply data to the external equipment. In addition, IOB supplies SELECT enable signals used to control the operation of the external equipment.

(b) *Lockout Circuits*

A functional part of each of the Input-Output Registers is a flip-flop interlock. During transmissions from storage to external equipment via IOA or IOB, this interlock holds up execution of a second External Write instruction until the external equipment has obtained the previous word from Input-Output Register. However, if the external equipment calls for the next word from IOA or IOB before IOA or IOB has been loaded by an External Write instruction, a Fault will occur.

During transmissions from the external equipment to internal storage, this lockout holds up execution of an External Read instruction until IOA or IOB has been loaded by the external equipment. If the external equipment attempts to transmit a word to IOA or IOB before the previous word has been read by the program, a Fault will occur.

In general, the interlock system achieves this: if the computer is proceeding at a rate faster than the peripheral equipment, the computer waits at each input or output for the peripheral equipment; if the computer does not keep up with the peripheral equipment, a Fault occurs.

(c) *External Equipment*

The external equipment may be any device capable of processing binary digital information.

(d) *External Equipment Control*

One external equipment control system is provided for each external equipment. The control selects the mode of operation as specified by the program, produces the pulses to synchronize the computer with the external equipment, and shapes and amplifies the data pulses passing between the computer and the external equipment.

## (2) SUPERVISORY CONTROL PANEL

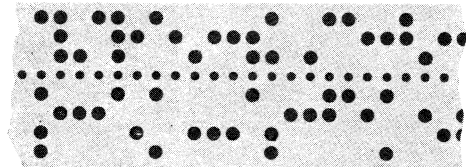
The Supervisory Control Panel affords direct communication between the operator and the 1103A. All arithmetic and control registers are displayed; every reference to Rapid Access Storage is indicated on the Monitoring Oscilloscope. Together these provide a continuous picture of all computer operations. The control panel contains all switches required for starting and stopping computer operation and for interruption of a program. The panel also provides for:

1. displaying the contents of any storage location
2. altering the contents of any storage location or arithmetic register
3. selecting the rate of operation such as high speed, step-by-instruction, etc.
4. testing the memory and functional circuitry
5. indicating and locating circuit failure.

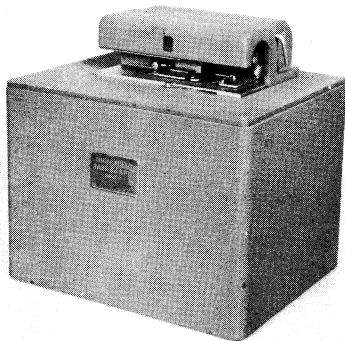
## (3) BASIC INPUT-OUTPUT EQUIPMENT

### (a) Punched Paper Tape -

The program of instructions and operands is presented to the computer via standard seven level perforated tape. The '1' digits are represented by holes in the tape, the '0' digits by the absence of holes. Usually six of the tape levels store data and the seventh level stores loading codes which direct the insertion of the data into storage. Any code may be employed for punched tape input. The tape may be in flexo-writer code, biocatal code, binary-coded decimal, or any arbitrary code. The computer can be programmed to translate data from any input code to the desired form for internal manipulation.



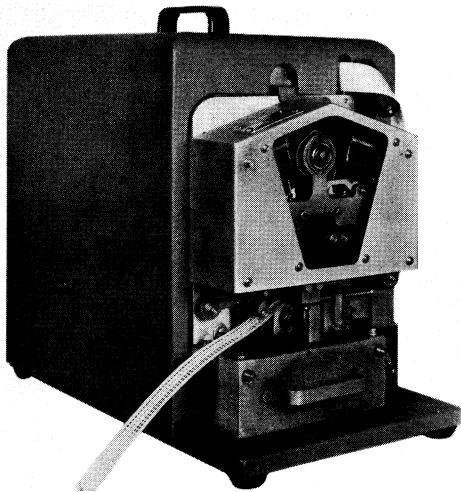
- (b) *The Tape Reader* - A photoelectric tape reader is used to transform the data on the perforated tape into corresponding patterns of pulses for input to the computer. Information is transferred to the computer via IOA. The reader operates in either of two modes: reading one frame on each command or reading continuously on a single command. The punched tape is read at the rate of 200 frames per second. Thus 2,000 binary-coded computer words per minute can be read with the photoelectric tape reader.



(c) *The Typewriter* - A basic output device for the 1103A is a directly-connected electric typewriter. By means of the PRINT instruction, PR-v, a character is typed which corresponds to a six-bit typewriter code held in storage location v. Output to the typewriter is via the Typewriter Register (TWR). The Typewriter Control (TWC) receives the order to print from the control section, translates the six-bit character in TWR and activates the proper one of 50 different typewriter operations. Typing is at the rate of 10 characters per second.

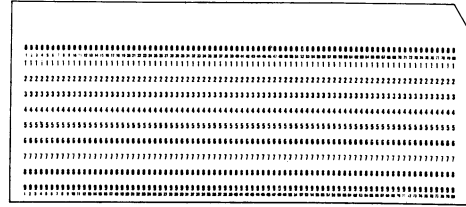


(d) *The High Speed Punch* - The other basic output device is a high speed paper tape punch. Execution of a PUNCH instruction, PUjv, causes the lower-order six bits of memory location v to be transferred to the High Speed Punch Register (HPR). HPR also stores the factor 'j' which controls the punching of a seventh-level hole. The High Speed Punch Control (HPC) receives the order to punch and directs the transfer of the information onto a single frame of tape. If 'j' = 1, HPC also directs the punching of a seventh-level hole. Information is punched at the rate of 60 frames per second.

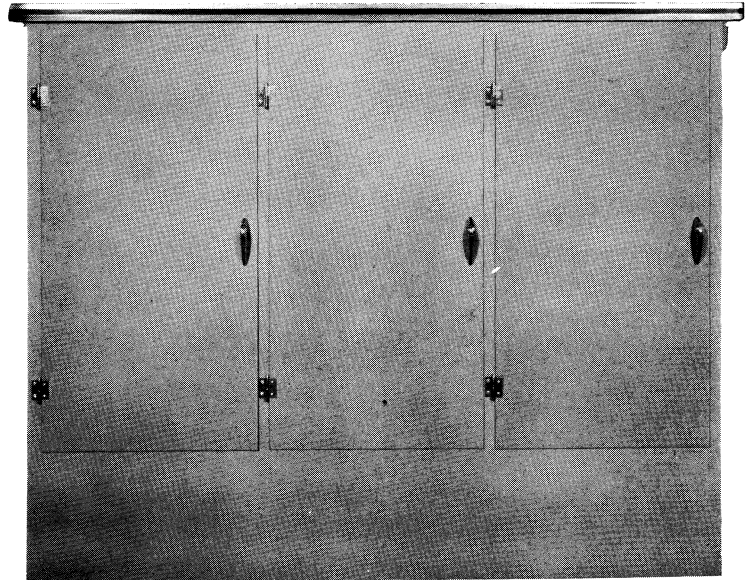
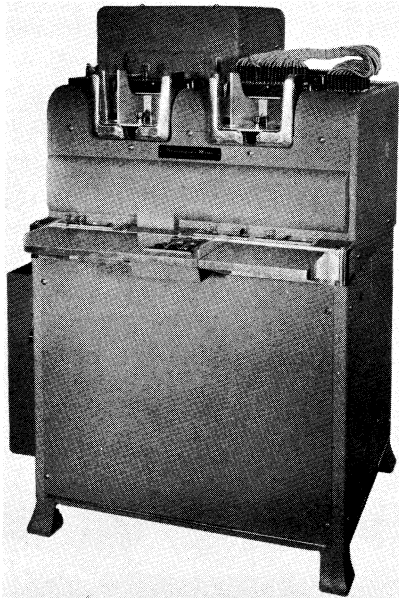


(4) OPTIONAL INPUT-OUTPUT EQUIPMENT

(a) *Punched Card Input-Output* - Information may be transferred to and from the computer by means of standard 12-row 80-column punched cards. The '1' digits are represented by rectangular holes in the card, the '0' digits by the absence of holes. Information is interpreted by the computer program and hence may be represented in any arbitrary code. Usually an alphanumeric character is noted by one or two punches in a single column and a decimal character by a single punch per column. Binary information may be represented either row-wise or column-wise.

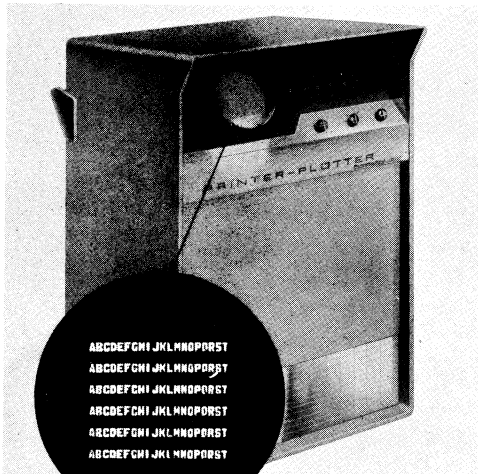


The Card Unit and Card Unit Control provide for both reading and punching tabulating cards. The card unit can read cards only, punch cards only or simultaneously read and punch. The unit operates at a rate of 120 cards per minute. Information is transferred between the card equipment and the computer through the Input-Output Registers IOA and IOB.



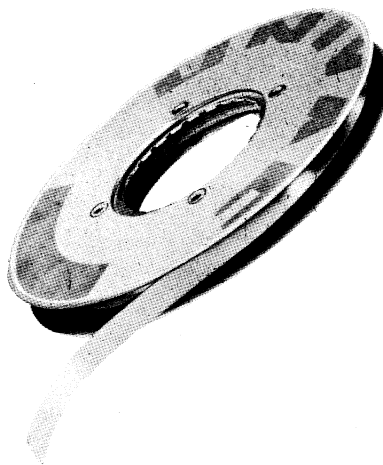


(b) *Oscilloscope Display* - The Remington Rand Oscilloscope Printer-plotter is a low-cost output display unit. Digital information may be displayed as graphs, animated drawings, words and numbers on the screen of a cathode ray tube. The Printer-Plotter is available with a 35 mm. automatic camera or a manually operated Polaroid-Land Camera. Thus a permanent, reproducible record may be made of the results of computation. The Charactron, a very high speed oscilloscope display unit manufactured by Stromberg-Carlson, is also available for use with the 1103A.



(c) *Lineprinter Output* - Two lineprinters are available for printed output from the 1103A: The directly-connected Output Printer (150 lines per minute) and the Univac High Speed Printer (600 lines per minute). The Output Printer is a low-cost printer which operates at a maximum rate of 230 characters per second. 35 different characters are available in each of 92 columnar positions. Spacing between lines is automatic and may be varied by means of manual settings on the Format Switchboard. Information is transmitted to the Output Printer through the Input/Output Register IPB.

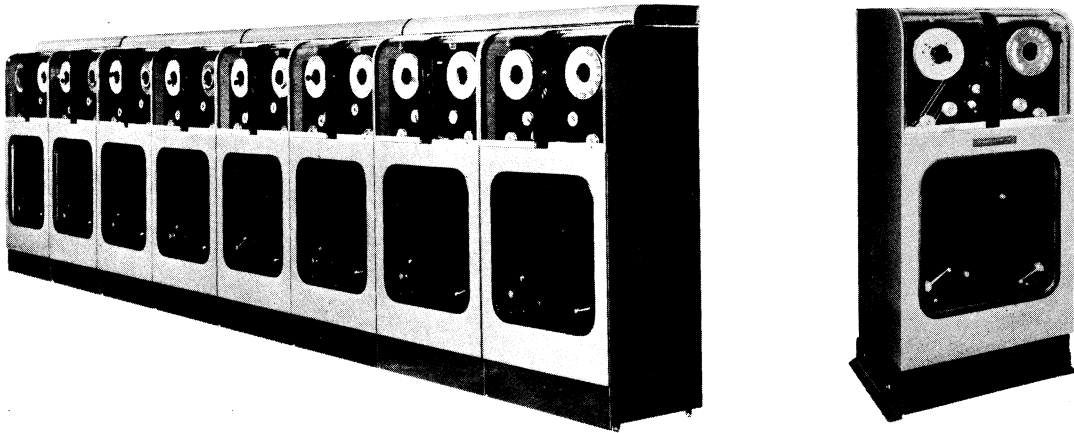
(d) *Magnetic Tape Input-Output* - The use of Univac magnetic tape, or UNITAPE, makes possible exceedingly fast input and output. 36-bit words may be transferred between the magnetic tape and computer memory at rates up to 2,130 words per second. The time-consuming processes of recording input information on the magnetic tape and printing or punching output from the tapes is performed off-line by a variety of Univac peripheral equipments.



*Unitape* - The standard magnetic tape used on all Univac equipment is  $\frac{1}{2}$  inch wide metallic tape. This tape is available in reels of varying lengths. Tapes are carefully tested to ensure proper recording surfaces. Holes are punched before and after any imperfect areas and additional tape length is allowed for the punched areas. The punched holes are sensed and passed over by all components on which the tape is used.

*Uniservos* - Uniservos are the magnetic tape-handling devices. Each Uniservo contains a magnetic 'read-write' head and a mechanism for moving the magnetic tape past the head. The Uniservos are operated under the program control by use of the External Function instruction; up to ten Uniservos may be directly connected to the 1103A. By means of manual selections the unit designations for the servos may be assigned in any manner to the functional units. Thus programs referencing particular units may in fact be run using any of the available Uniservos. Reels of magnetic tape may be put on a Uniservo or taken off in a few seconds.

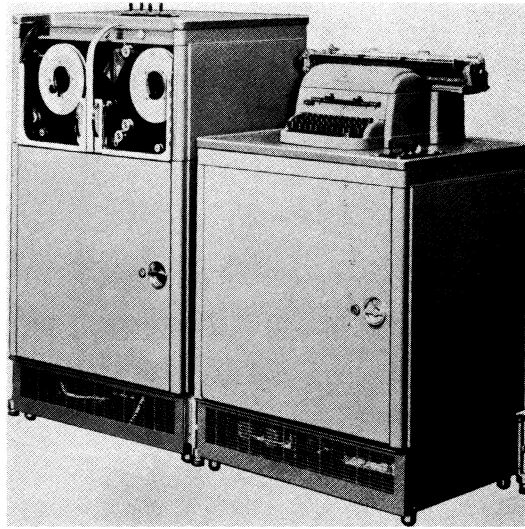
14



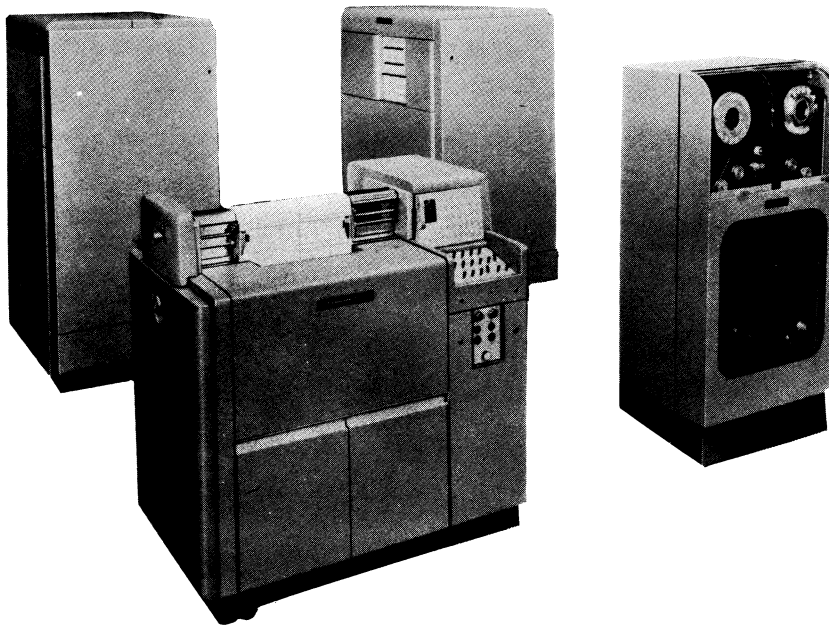
*Unityper II* - This is a desk-top electric typewriter with a special mechanism for the direct recording of alphanumeric information on magnetic tape. As a key is struck by the operator, a corresponding pulse code pattern is recorded on the tape. Special keys are provided for correcting improperly recorded information as well as filling with zeros or spaces.



*Uniprinter* - The Uniprinter consists of a tape reader and a printing unit. The pulse codes are read from the magnetic tape and the corresponding typewriter keys activated. These codes may represent letters, numbers or symbols for such typewriter functions as carriage return, tabulate, and shift. Printing is at the rate of 10 characters per second. A skip switch allows skipping over unwanted data at a faster rate in order to print selected areas.

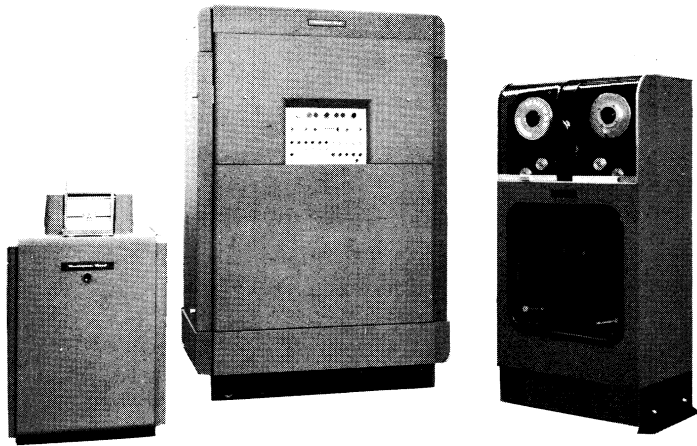


*High-Speed Printer* - This printer reads information from magnetic tape and prints it at the rate of 600 lines per minute. There are 130 printing positions on the line with 51 different characters available in each position. Flexibility of format is attained by means of a plugboard which provides for multiline printing, multiple character printing, columnar rearrangement, zero suppression, and single, double or triple spacing. In addition to its conventional use for volume printing, the High-Speed Printer makes an excellent plotting device for representing graphically the results of computation. As noted above, as an optional feature the High-Speed Printer may be equipped for on-line as well as off-line use with the 1103A.



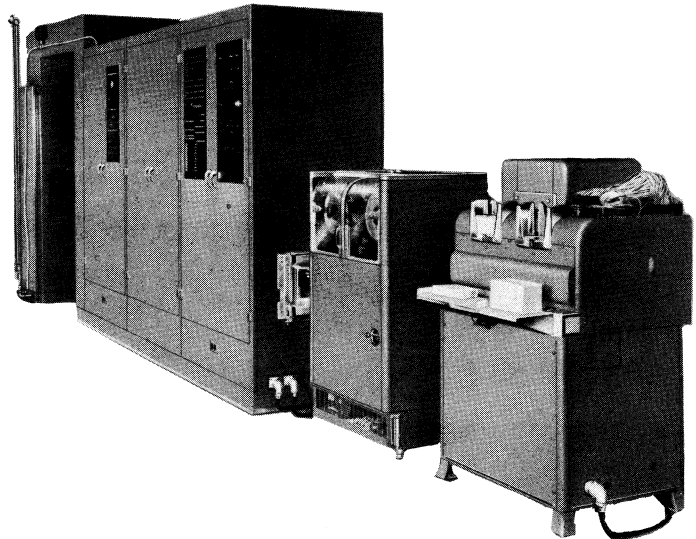
*Card-to-Tape Converter* - The 80-Column Card-to-Magnetic

Tape Converter automatically transcribes data from 80-column punched cards to Univac magnetic tape. When the conversion process has been completed the tape may then be mounted on a Uniservo and the information read into the computer. Data from the punched cards is transcribed to tape at the rate of 240 cards per minute.



16

*Tape-to-Card Converter* - The Tape-to-Card Converter transfers data from Univac magnetic tape to standard 80-column punched cards. The conversion rate is 180 cards per minute.



*Additional Equipment* - Only the major Univac Scientific auxiliaries have been described above. Many more specialized equipments are available. These include the Paper Tape-to-Magnetic Tape Converter, the Magnetic Tape-to-Paper Tape Converter, the Verifier, and Unitape Transmission equipment. And, of course, use of Univac magnetic tapes with the 1103A provides communication with other Remington Rand computers such as the UNIVAC and the UNIVAC FILE COMPUTER.

## STORAGE SECTION

The principal functions of the Storage Section, or computer memory, is to provide the Arithmetic and Control Sections with the operands and instructions required during the execution of a program. A secondary function of the section is to provide temporary storage for the intermediate and final results of the computations.

The Storage Section is composed of four classes of uniquely addressed storage locations (Magnetic Drum Storage, Magnetic Core Storage, the Accumulator, and the Q-Register). In addition, Magnetic Tapes may be used for bulk storage of information. The information is transmitted to and from tapes in blocks of either a fixed number of words or a variable number of words. Individual addresses are used to identify the storage positions of the four storage classes MD, MC, A, and Q. A total of 20,482 15-bit addresses are assigned to these classes, as follows:

STORAGE CLASS	ADDRESSES IN OCTAL NOTATION
MC: First bank of 4,096 words	00000-07777
Second bank of 4,096 words	10000-17777
Third bank of 4,096 words	20000-27777
Q	31000-31777
A	32000-37777
MD (Group 4)	40000-47777
MD (Group 5)	50000-57777
MD (Group 6)	60000-67777
MD (Group 7)	70000-77777

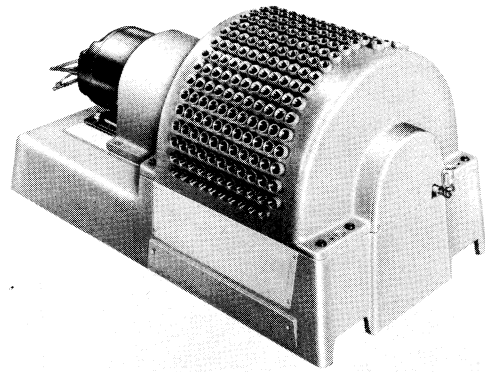
A storage reference to MD, MC, A or Q is made on the basis of the 15-bit storage address held in the Storage Address Register, SAR, at the time of the reference. The address held in SAR is interpreted by the Storage Class Translator and Storage Class Control. The proper one of the four addressed classes is then selected and the necessary signals sent to the proper locating circuit. The locating circuit then locates the specific storage position called for by (SAR).

(1) *STORAGE ADDRESS REGISTER.* - The Storage Address Register (SAR) holds the 15-bit storage address during any storage reference. SAR receives addresses from PAK, UAK and VAK and communicates with Storage Class Control, MD, MC, the X-Register and PAK.

(2) *STORAGE CLASS CONTROL.* - The Storage Class Control (SCC) and its translator (SCT) determine the class of storage referenced by translating the higher-order bits of the address in SAR. Thus it may determine that MD, MC, A or Q is being referenced.

(3) *MAGNETIC DRUM STORAGE SYSTEM.* - The Magnetic Drum Storage System is a medium access speed storage medium consisting of a magnetic drum, a locating circuit, and reading and writing circuits. The system stores, in parallel, 16,384 words in four groups of 4,096 words each. The maximum access time is approximately 34 milliseconds, the time taken in one drum revolution. The transfer rate from magnetic drum to magnetic cores is 32 microseconds per word once the first word in a sequence has been accessed.

(a) *MAGNETIC DRUM* - The Magnetic Drum is a precision aluminum cylinder 17 inches in diameter and 12 inches long. Its surface is coated with a magnetizable iron oxide. A housing in which the magnetic heads are mounted covers the surface of the drum. The magnetic heads which read and write on the drum's surface have their gaps about 0.002 inches from the surface and provide non-contact recording of the digital data. At one end of the drum a soft steel band containing the milled Timing Track and Mark Track is affixed. The timing notches are spaced 80 to the linear inch and the mark notch (one only) corresponds to one of the timing notches. The mark notch denotes the electrical beginning of the Timing Track. The data heads are arranged in a spiral fashion to permit an axial spacing of 16 to the inch. As a customer option, the Univac Scientific Model 1103A System may be rented or purchased without the Magnetic Drum Memory.

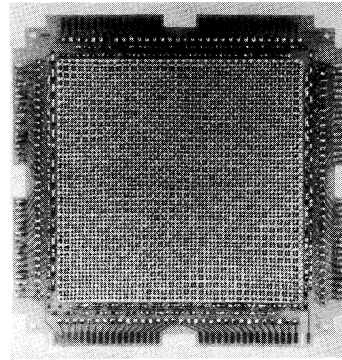


(b) *LOCATING CIRCUIT*. - The MD Locating Circuit is composed of the Group Detector and Group Selector, the Angular Index counter (AIK), the Angular Coincidence Detector, and the location control unit. The functions of these circuits are to identify the referenced group (one of four) and then to select the proper angular address (one of 4,096). An interchangeable Address Interlace Chassis (4-, 8-, 16-, 32-, and 64-interlace patterns may be chosen) is interposed between the Angular Coincidence Detector and the lower-order 12 stages of SAA for the purpose of selecting proper time intervals between consecutive MD storage references (to accommodate inter-reference sequences) rather than having to wait a full drum revolution between references.

\* (c) *ACCESS CONTROL*. - The MD Access Control (MDAC) provides 125 kc TIMING PULSES to the Location Control and determines whether a reading or writing operation is ordered; it reacts accordingly on receipt of a COINCIDENCE PULSE from the Angular Coincidence Detector and sends a RESUME signal to the Control Section at the completion of the storage reference.

(4) *MAGNETIC CORE STORAGE SYSTEM*. - The Magnetic Core Storage System is a non-volatile rapid-access storage medium consisting of 36 magnetic core matrices, an address selection system, a 36-bit input register, and an access control. The system stores 4,096, 8,192, or 12,288 36-bit words in parallel. Any word in this memory is available in 10 microseconds.

(a) *MAGNETIC CORE MATRICES* - Each matrix consists of 4,096 cores wired in a 64 by 64 array so that the wires all lie in the same plane. The cores are held in position by the wires which are terminated on a square printed-circuit frame. Each core is a small toroid of material exhibiting a nearly rectangular hysteresis loop. Five wires pass through each core: a horizontal X wire, a vertical Y wire, a diagonal S wire, a horizontal  $I_1$  wire, and a vertical  $I_2$  wire. Each core, thus wired, forms a bistable device, which can store a '1' or a '0' depending on the direction of magnetization. The '1' state is 'written' in a core when the resultant sum of the magnetizing forces of coincident currents in the X and Y wires is sufficient



to magnetize the core to the opposite polarity from that remaining after a 'read' operation. The '0' state is obtained by not altering the polarity after a 'read' operation. The core is 'read' by introducing coincident current simultaneously in the X and Y wires in the opposite direction from the current used in a 'write' operation. If, before reading, the core was in the '1' state, the resultant changes in flux produced by the 'read' current induces a current in the diagonal S wire which after amplification indicates the presence of a '1'. If, before reading, the core was in the '0' state, no change in flux is produced, and, hence, no signal is induced in the S wire.

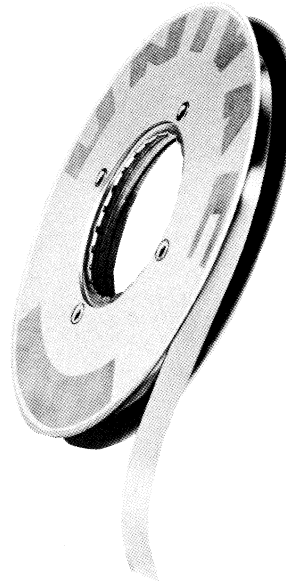
(b) *ADDRESS SELECTION*. - The circuits used in address selection consist of the Address Register, the Read/Write Pulse Generators, and the Read/Write Enable Generators. These circuits select a particular pair of X and Y wires so that reading or writing is executed at a particular coordinate location.

(c) *INPUT REGISTER*. - The 36-stage Magnetic Core Input Register acts both as an input register and restoration register. During a writing operation, a 36-bit word from the X-Register is transmitted to the Input Register from which the word is entered into the Magnetic Core Storage at the address specified by SAR. During a reading operation, the word being transferred from a Magnetic Core Storage address to the X-Register is held temporarily in the Input Register so that the information at the reference address can be rewritten (the reading operation destroys the information by driving all 36 cores to the '0' state).

(d) *ACCESS CONTROL*. - The Magnetic Core Access Control receives INITIATE READ MC or INITIATE WRITE MC signals from the Control Section of the computer. Upon receipt of such a signal, the Access Control generates a sequence of operations to effect the incoming command. These sequences control the reading, writing and restoration operations within the Magnetic Core Storage System.

(5) REGISTER STORAGE. - In many programs it is convenient to use the Q-Register and the Accumulator as one-word storage media. SAR through SCC and the Arithmetic Registers Access Control (ARAC) can select A or Q as a storage register if the u-address or v-address of an instruction so dictates. Q has the address 31 --- and A the address 32 ---.

(6) MAGNETIC TAPE STORAGE SYSTEM. - The Magnetic Tape Storage System employs a number of Uniservo tape handling mechanisms. The units are located externally to the computer and the number used is optional up to a maximum of ten. Four to six units are considered appropriate for the average installation. The electronic control portion of the system is located within the basic computer structure. Use of the Uniservo units makes possible communication between the computer and a variety of Univac peripheral equipments on an off-line basis.



(a) METHOD OF RECORDING. Eight-channel recording is employed with the Unitapes, six channels contain data, one channel contains a parity check bit on the six data channels, and one channel is used to record a sprocket or timing signal which controls the read-back operation.

A recording density of 128 lines per inch and a tape speed of 100 inches per second are standard. The free-running transfer rate of information between computer and tapes is therefore 2,130 computer words per second. More than 382,000 words may be stored on a standard 1,500 ft. tape.

Information may be recorded on magnetic tape in three forms for input to the 1103A:

- 1. Fixed Block Length Recording
  - 2. Variable Block length Recording
  - 3. Continuous Recording
- } Optional variable format

Information may be recorded for output from the 1103A in either Fixed Block Length or Variable Block Length form. It is not possible to record information continuously for output.

Fixed Block Length Input-Output is a standard feature of the 1103A. As an optional feature control circuitry will be added to provide both Variable Block Length Input-Output and Continuous Recording Input.



## 1 FIXED BLOCK LENGTH RECORDING

All data are recorded in blocks of 720 characters or lines on the tape, each block thus containing 120 complete 36-bit computer words. Blocks are separated by a 'dead' space of one inch on the tape (a 2.4 inch block spacing is also available). Optionally, a block may be subdivided into six blockettes with a spacing of either one inch or one-tenth inch between blockettes. The purpose of this sub-division is to provide tapes suitable for use in the Univac peripheral units. If the tape is to be used only as storage for the computer, no blockette spacing is provided.

## 2 VARIABLE BLOCK LENGTH RECORDING

The information is arranged in blocks of variable length. However, the number of words in any one block must not exceed the capacity of the high speed memory. Blocks are separated by a one inch dead space to allow for stopping and starting the Uniservo.

## 3 CONTINUOUS RECORDING

Information to be entered into the computer is recorded continuously on the magnetic tape. The length of the magnetic tape reel is the only limitation on the size of a block of information. This type of recording may be used for real time observations which will not permit interruptions in order to format the information in fixed or variable length blocks. In this method of recording, the eight tape channels are used as follows: one sprocket channel, one parity check channel, four data channels, and two block control channels. The block control channels contain a Block Control Code which allows for stopping and starting the Uniservo. The block control channels may also contain codes identifying the data.

(b) *SELECTION AND INSTRUCTION CONTROL.* - The External Function instruction (17-v) is used to initiate the various reading, writing and positioning operations of the Uniservos through the Selection and Instruction Control portion of the system. Coded information provided by the External Function instruction includes the following: (1) Designation of the selected Uniservo, (2) Type of recording format, (3) Type of operation to be performed.

Eight operations are possible in the Fixed Block Length or Variable Block Length modes: Read Forward, Write Forward, Read Backward, Move Forward n Blocks, Move Backward n Blocks, Stop, Rewind, and Rewind Interlock. In the case of a reading or writing operation, the External Function instruction which initiates the operation must be followed by an appropriate number of External Read or External Write instructions to transfer the information between the Input-Output Register and the computer memory.

(c) *TRANSFER CONTROL*. - The Transfer Control portion of the system controls the transfer of information between the computer and Unitapes through a 36-bit Tape Input-Output Register (TR). The writing operation involves the breakdown of the 36-bit computer word into six 6-bit characters and the generation of a parity check bit for each character. During the reading operation, the parity check is made on each line read from the tape and the 36-bit word assembled in TR. If a parity error occurs, an indication of this error is entered in IOA. This indication can be used to initiate a repetition of the operation in which the error occurred. When information is recorded on tape in fixed blocks of 120 words, a check is also made during reading to determine if the proper number of lines are recorded in the block.

## ARITHMETIC SECTION

The Arithmetic Section performs the arithmetic operations of addition, subtraction, multiplication, and division as well as some strictly logical operations such as shifting, logical addition (bit-by-bit addition), and logical multiplication (bit-by-bit multiplication). The section contains the X-Register, Q-Register, Accumulator, the Shift Counter and its control, and the Arithmetic Sequence Control.

(1) *X-REGISTER*. - As an arithmetic register, X holds the addend, subtrahend, multiplicand, and divisor in the corresponding arithmetic operations. In addition, during a logical multiplication, X may be considered as holding the multiplier; the actual bit-by-bit multiplication is carried out by a transmission from Q, Q'X'.

(2) *Q-REGISTER*. - As an arithmetic register, Q holds the multiplier and quotient in the corresponding arithmetic operations. During a logical multiplication, Q may be considered as holding the multiplicand during the transmission Q'X'. The shifting property of Q is utilized in both multiplication and division.

(3) *ACCUMULATOR*. - As an arithmetic register, A holds the sum, difference, and product in the corresponding arithmetic operations. In the division operation, A initially holds the dividend, and at the completion of the division operation A holds the non-negative remainder R. The shifting property of A is utilized in multiplication, division and in the scale factor operation. The Accumulator is basically subtractive; an operand in being transferred from X to A is automatically complemented and subtracted. During a transfer from X to A the modulus is converted from  $2^{36} - 1$  to  $2^{72} - 1$  except in 'split' operations.

(4) *SHIFT COUNTER*. - The Shift Counter SK is used to keep track of the number of shifts in arithmetic operations. Physically SK is the seven lower order stages of SAR (since SAR is not used as an addressed register during an arithmetic sequence, SAR is free to perform this second function). Associated with SK is the Shift Counter Control SKC, which governs its sequence.

(5) *ARITHMETIC SEQUENCE CONTROL*. - The Arithmetic Sequence Control, ASC, controls the operations of the Arithmetic Section upon command from the Control Section. ASC generates sequences of subcommands, each subsequence being dependent on the command received from the Control Section. At the end of the subsequence, control is returned to the Control Section.

## *C O N T R O L   S E C T I O N*

The Control Section exerts the directing influence over the activities of the computer by controlling the timing of the various operations of the computer. The Control Section receives the instructions which the computer is to carry out, interprets them, and directs their execution upon the operands specified. The computer must be manually started, but can stop automatically or be manually stopped. Besides being automatically controlled by a program of instructions, it can be manually controlled from the Supervisory Control Panel which contains all the necessary controls and indicators for manually operating (and maintaining) the equipment.

(1) *MASTER CLOCK*. - All of the activities which take place within the computer, except for certain ones in the output section, are synchronized by a central timing system called the Master Clock. During normal computer operation, the clock generates 500 kc *CLOCK PULSES* based on *TIMING PULSES* from the Magnetic Drum Storage System and, after exerting certain controlling influences over them, supplies them to circuits throughout the computer. During testing operations, a 500 kc oscillator may be used instead of the drum as the basic source of timing pulses.

The principal circuits of the Master Clock system, and the functions they serve, are as follows:

(a) *OSCILLATOR*. - Generates a continuous series of timing pulses at a rate of 500 kc.

(b) *CLOCK SOURCE SELECTOR (CSS)*. - Selects either the MD system or the Oscillator, depending upon manual selections made by the operator, as the source of timing pulses for the clock.

(c) *CLOCK RATE CONTROL (CRC)*. - Controls the rate at which pulses leave the clock. CRC permits the operator to select any of six different pulse rates. During normal operation, the 500 kc. pulse rate is selected. The other rates are of use in program debugging and computer maintenance.

(d) *PULSE DISTRIBUTOR CONTROL (PDC)*. - Starts and stops the flow of *CLOCK* pulses from CRC to the Main Pulse Distributor in response to signals received from other sections of the computer.

(2) *MAIN PULSE DISTRIBUTOR*. - The Main Pulse Distributor, MPD, receives CLOCK pulses from PDC and distributes them, in successive cycles of from four to eight pulses, to the Command Timing Circuits. The distributor supplies each of the pulses cyclicly on its eight output lines. In an eight pulse cycle, all of the output lines are used and the pulses are designated, in the order of their generation,  $MP_0$  through  $MP_7$ . The selection of a particular cycle is made on the basis of the operation code held in the Main Control Register, MCR. Each code selects the cycle which will permit its performance in the least possible time.

(3) *PROGRAM ADDRESS COUNTER*. - The Program Address Counter PAK, is a 15-stage additive counter which is used to generate successive addresses at which the instructions of the Computer's program can be found. During computation, the address in PAK is referred to each time an instruction word is to be obtained from the computer memory; PAK can thus be thought of as guiding the computer through the instructions of the program. The starting address of a computation may be manually inserted into PAK before the START button is pressed; if this is done, computation will begin by selecting the instruction stored at that address. If PAK is not manually preset, it will automatically be set to MD address 40000. Once computation is started, PAK generates consecutive addresses at which succeeding instructions in the program can be found. If a jump instruction appears in the program and its execution calls for a jump to be made, the following events occur: (1) The jump address (i.e., an address to which the Control Section must now refer in extracting the next instruction) is inserted into PAK; (2) the Control Section selects the next instruction from the jump address specified by PAK and advances PAK by one; and (3) thereafter, PAK generates consecutive addresses, starting from the jump address, until another jump occurs or until a stop instruction occurs.

(4) *PROGRAM CONTROL REGISTERS*. - The Program Control Registers, PCR, receive each instruction and temporarily store it during its execution. The Registers consist of the Main Control Register, MCR, the U-Address Counter, UAK, and the V-Address Counter, VAK. Each instruction sent to PCR consists of a 6-bit operation code, which is stored in MCR; a 15-bit u-address portion, which is stored in UAK; and a 15-bit v-address portion, which is stored in VAK. Each instruction is obtained from some 36-bit storage location as specified by PAK.

(5) *MAIN CONTROL TRANSLATOR*. - The Main Control Translator, MCT, is composed of a principal and an auxiliary translator. Its principal translator receives a 6-bit operation code from MCR and produces a single prime operation code enable on one of its 41 output lines. Its auxiliary translator receives enables from the principal translator, from MCR, and from various other circuits in the computer and produces composite or conditional enables. Output enables from both translators in MCT are utilized throughout the Control Section, but mainly in the Command Timing Circuits, CTC, and the Main Pulse Distributor, MPD. In CTC, the MCT enables are used in the selection

of the sequence of commands which are needed to execute the instruction currently in MCR. In MPD, the MCT enables are used in the selection of the MAIN PULSE, MP, cycle required for operation.

(6) *COMMAND TIMING CIRCUITS.* - The Command Timing Circuits, CTC, combine each operation enable, from the Main Control Translator, with the corresponding MP cycle, from MPD, to produce a discrete sequence of commands which executes the specified operation. CTC receives two or more of the pulses MP<sub>0</sub> through MP<sub>5</sub> and MP<sub>6</sub> and MP<sub>7</sub>, along with each operation enable. It distributes the pulses MP<sub>0</sub> through MP<sub>5</sub> as the commands which execute the operation. It reads the succeeding instruction from storage into X on MP<sub>6</sub>, then transfers the instruction from X into PCR on MP<sub>7</sub>.

(7) *REPEAT SEQUENCE CONTROL.* - When a Repeat Instruction (75jnw) is executed, a repeat sequence is set up in the Repeat Sequence Control, RSC. The controlling functions of this repeat sequence depend on the values of 'n' and 'j' in the Repeat Instruction and the nature of the instruction following the Repeat. In general, the RSC sequence (1) notifies Control that a repeat operation is in progress; (2) causes additional commands to be generated on MP<sub>5</sub> during the execution of the repeated instruction (these commands determine whether another execution of the instruction should be carried out and therefore, the usual MP<sub>6</sub> omitted, or they determine that the repeat operation should be terminated), and (3) initiates the Repeat Terminations routines. (RSC may be rendered inoperative by certain jump and stop instructions.) The nature of the repeat operation and the specific functioning of RSC is outlined in the following paragraph.

During the execution of the Repeat Instruction, the right-hand 15-bits of (F<sub>1</sub>) are replaced by 'w'; the instruction to be repeated is transmitted to PCR (and its operation code held there until the repeat operation is terminated); the factor 'jn' is stored in PAK; PAK is then complemented; and a repeat sequence installed in RSC. The RSC sequence then tests to see if the 'n' of the Repeat Instruction is zero or not. If n = 0, the RSC sequence is immediately terminated, and no execution of the instruction in PCR is carried out; instead, the next instruction is taken from F<sub>1</sub>. If n ≠ 0, the RSC sequence advances MPD to 7, SAR is cleared, MPD advances and the instruction in PCR is executed. Whether it is then repeated or not depends on the value of 'n' and the nature of the instruction itself. If the execution of an instruction does not create a condition which can terminate the repeat sequence (as a jump or stop), RSC advances PAK and tests the new value of 'n'. If n ≠ 0, RSC advances the execution addresses of the instruction as specified by the initial value of 'j' in the 75jnw instruction:

- If j = 0, RSC does not advance the 'u' or 'v' address,
- If j = 1, RSC advances the 'v' address by one,
- If j = 2, RSC advances the 'u' address by one,
- If j = 3, RSC advances both the 'v' and 'u' addresses by one.

RSC also sets MPD to 7 (omits  $MP_6$ , thereby retaining the instruction to be repeated in PCR). On  $MP_7$ , SAR is cleared; MPD then advances, and another execution of the instruction is carried out. This procedure continues until 'n' executions have been made, or until some condition arises in the execution of the instruction which terminates the repeat sequence.

## REPERTOIRE OF INSTRUCTIONS

The complete list of instructions which the computer performs is presented below. The instructions are arranged in 11 groups according to their basic characteristics. In each listing a code representing the instruction is enclosed in parenthesis after the name of the instruction. The operation code portion is designated by a two-character combination and the execution addresses by the letters u and v. In some cases u is replaced by the conditioning factors j, n, or k, as in the Repeat instruction or the Left Transmit instruction. In other cases v is replaced either by the repeat termination address, w, or in some shifting operations by the factor k. The repertoire of instructions is summarized in Table 1. The meanings of symbols used to describe the instructions are presented in the following glossary of terms and abbreviations.

27

- Word. - A combination of 36 bits.
- Instruction. - A word, represented by  $i_{35}, i_{34}, \dots, i_0$ , which causes the computer to perform one or more of its operations. The instruction consists of an operation code and, usually two execution addresses.
- Operation Code. - That six-bit part of an instruction, represented by  $i_{35}, i_{34}, \dots, i_{30}$ , designating the operation to be performed.
- u. - The first execution address of an instruction, represented by  $i_{29}, i_{28}, \dots, i_{15}$ .
- v. - The second execution address of an instruction, represented by  $i_{14}, i_{13}, \dots, i_0$ .
- operand. - A word on which an operation is performed.
- ( ). - Parenthesis, denoting 'the contents of'.
- A. - The 72-bit Accumulator,  $A_{71}, A_{70}, \dots, A_0$ .
- $A_L$ . - The left-hand (most significant) 36 bits of A.
- $A_R$ . - The right-hand (least significant) 36 bits of A.
- Q. - A 36-bit shifting register,  $Q_{35}, Q_{34}, \dots, Q_0$ .
- X. - A 36-bit exchange register,  $X_{35}, X_{34}, \dots, X_0$ .
- ( )<sub>i</sub>. - The initial contents of a register before an operation has taken place.

- ( )<sub>F</sub>. - The final contents of a register after an operation has taken place.
- D(u). - A 72-bit word whose right-hand 36 bits are (u) and whose left-hand 36 bits are all alike and equal to the left-most bit of (u).
- S(u). - A 72-bit word whose right-hand 36 bits are (u) and whose left-hand 36 bits are all zero.
- L(Q)(u). - A 72-bit word whose left-hand 36 bits are zeros and each of whose right-hand 36 bits is given by the product of the corresponding bits of (u) and (Q).
- L(Q)'(v). - A 72-bit word whose left-hand 36 bits are zeros and each of whose right-hand 36 bits is given by the product of the corresponding bits of (v) and the complement of (Q).
- j. - A one-digit octal number, represented by  $u_{14}$ ,  $u_{13}$ ,  $u_{12}$ .
- n. - A four-digit octal number, represented by  $u_{11}$ ,  $u_{10}$ , ...,  $u_0$ .
- k. - The shift count, usually represented by  $v_6$ ,  $v_5$ , ...,  $v_0$ , but in the Left Transmit instruction, represented by  $u_6$ ,  $u_5$ , ...,  $u_0$ .
- F<sub>1</sub>. - Fixed octal address 00000 (or address 40001 if so manually selected in the TEST mode).
- F<sub>2</sub>. - Fixed octal address 00001.
- F<sub>3</sub>. - Fixed octal address 00002.
- TWR. - A typewriter register of six bits.
- HPR. - The high-speed punch register of seven bits.
- CI. - The current instruction.
- NI. - The next instruction.



TABLE 1

A SUMMARY OF THE  
REPERTOIRE OF INSTRUCTIONS

11	TRANSMIT POSITIVE (TP <sub>uv</sub> ) . . . . .	(u) → v
12	TRANSMIT MAGNITUDE (TM <sub>uv</sub> ) . . . . .	(u)   → v
13	TRANSMIT NEGATIVE (TN <sub>uv</sub> ) . . . . .	(u)' → v
14	INTERPRET (IP <sub>xx</sub> ) . . . . .	y+1 → F <sub>1</sub> , take (F <sub>2</sub> )
15	TRANSMIT U-ADDRESS (TU <sub>uv</sub> ) . . . . .	(u <sub>15-29</sub> ) → (v <sub>15-29</sub> )
16	TRANSMIT V-ADDRESS (TV <sub>uv</sub> ) . . . . .	(u <sub>0-14</sub> ) → (v <sub>0-14</sub> )
17	EXTERNAL FUNCTION (EF-v) . . . . .	Select Ext. Equipment and perform (v)
21	REPLACE ADD (RA <sub>uv</sub> ) . . . . .	(u) + (v) → u
22	LEFT TRANSMIT (LT <sub>jkv</sub> ) . . . . .	Shift (A) by k; j=0, (A <sub>L</sub> ) <sub>f</sub> → v; j=1, (A <sub>R</sub> ) <sub>f</sub> → v
23	REPLACE SUBTRACT (RS <sub>uv</sub> ) . . . . .	(u) - (v) → u
27	CONTROLLED COMPLEMENT (CC <sub>uv</sub> ) . . . . .	(u) ⊕ (v) → u
31	SPLIT POSITIVE ENTRY (SP <sub>uk</sub> ) . . . . .	S(u) → A, shift (A) by k
32	SPLIT ADD (SA <sub>uk</sub> ) . . . . .	(A) + S(u), shift (A) by k
33	SPLIT NEGATIVE ENTRY (SN <sub>uk</sub> ) . . . . .	S(u)' → A, shift (A) by k
34	SPLIT SUBTRACT (SS <sub>uk</sub> ) . . . . .	(A) - S(u), shift (A) by k
35	ADD AND TRANSMIT (AT <sub>uv</sub> ) . . . . .	(A) + D(u) → v
36	SUBTRACT AND TRANSMIT (ST <sub>uv</sub> ) . . . . .	(A) - D(u) → v
37	RETURN JUMP (RJ <sub>uv</sub> ) . . . . .	y+1 → u, take (v)
41	INDEX JUMP (IJ <sub>uv</sub> ) . . . . .	D(u) - 1 → A; (A) <sub>f</sub> + (A) → u & take (v)
42	THRESHOLD JUMP (TJ <sub>uv</sub> ) . . . . .	(u) > (A), take (v)'
43	EQUALITY JUMP (EJ <sub>uv</sub> ) . . . . .	(u) = (A), take (v)
44	Q-JUMP (QJ <sub>uv</sub> ) . . . . .	(Q) -, take (u); (Q) +, take (v); (Q) left 1
45	MANUALLY SELECTIVE JUMP (MJ <sub>jv</sub> ) . . . . .	j=0, or 1, 2, 3, & MJS =, take (v)
46	SIGN JUMP (SJ <sub>uv</sub> ) . . . . .	(A) -, take (u); (A) +, take (v)
47	ZERO JUMP (ZJ <sub>uv</sub> ) . . . . .	(A) ≠ 0, take (u); (A) = 0, take (v)
51	Q-CONTROLLED TRANSMIT (QT <sub>uv</sub> ) . . . . .	L(Q) (u) → v
52	Q-CONTROLLED ADD (QA <sub>uv</sub> ) . . . . .	(A) + L(Q) (u) → v
53	Q-CONTROLLED SUBSTITUTE (QS <sub>uv</sub> ) . . . . .	L(Q) (u) + L(Q)' (v) → v
53	LEFT SHIFT IN A (LA <sub>uk</sub> ) . . . . .	D(u) → A, shift (A) by k, (A) <sub>f</sub> → u
55	LEFT SHIFT IN Q (LQ <sub>uk</sub> ) . . . . .	(u) → Q, shift (Q) by k, (Q) <sub>f</sub> → u
56	MANUALLY SELECTIVE STOP (MS <sub>jv</sub> ) . . . . .	j=0, stop; j=1, 2, 3, & MSS =, stop
57	PROGRAM STOP (PS--)	Stop and Indicate
61	PRINT (PR-v)	Typewriter performs code in v <sub>0-5</sub>
63	PUNCH (PU <sub>jv</sub> ) . . . . .	Punch (v <sub>0-5</sub> ); j=1, 7th level
71	MULTIPLY (MP <sub>uv</sub> ) . . . . .	(u) (v) = (A)
72	MULTIPLY ADD (MA <sub>uv</sub> ) . . . . .	(A) <sub>i</sub> + (u) (v) = (A) <sub>f</sub>
73	DIVIDE (DV <sub>uv</sub> ) . . . . .	(A) <sub>i</sub> ÷ (u) = (Q) + (A) <sub>f</sub> , (Q) → v; (A) <sub>f</sub> = +R
74	SCALE FACTOR (SF <sub>uv</sub> ) . . . . .	Shift D(u) in A until A <sub>34</sub> ≠ A <sub>35</sub> , (SK) → v
75	REPEAT (RP <sub>jnw</sub> ) . . . . .	Execute NI 'n' times, jump to F <sub>1</sub>
76	EXTERNAL READ (ER <sub>jv</sub> ) . . . . .	j=0, (IOA) → v; j=1, (IOB) → v
77	EXTERNAL WRITE (EW <sub>jv</sub> ) . . . . .	j=0, (v) → IOA; j=1, (v) → IOB

## SEQUENCED INSTRUCTIONS

71 MULTIPLY (MPuv): Form in A the 72-bit product of (u) and (v), leaving in Q the multiplier (u).

72 MULTIPLY ADD (MAuv): Add to (A) the 72-bit product of (u) and (v), leaving in Q the multiplier (u).

73 DIVIDE (DVuv): Divide the 72-bit number (A) by (u), putting the quotient in Q, and leaving in A non-negative remainder R. Then replace (v) by (Q). The quotient and remainder are defined by:  $(A)_i = (u) \cdot (Q) + R$ , where  $0 \leq R \leq |u|$ . Here  $(A)_i$  denotes the initial contents of A.

74 SCALE FACTOR (SFuv): Replace (A) with D(u). Then left circular shift (A) by 36 places. Then continue to shift (A) until  $A_{34} \neq A_{35}$ . Then replace the right-hand 15 bits of (v) with the number of left circular shifts, k, which would be necessary to return (A) to its original position. If (A) is all ones or zeros, k=37. If u is the address of the Accumulator, (A) is left unchanged in the first step, instead of being replaced by D(A<sub>R</sub>).

75 REPEAT (RPjnw): This instruction calls for the next instruction, which will be called NIuv, to be executed n times, its 'u' and 'v' addresses being modified or not according to the value of j. Normally n executions are made and the program is continued by the execution of the instruction stored at a fixed address F<sub>1</sub>. The steps carried out are:

- (a) Replace the right-hand 15 bits of (F<sub>1</sub>) with the address w.
- \*(b) Execute NIuv, the next instruction in the program, n times.
- (c) If j=0, do not change u and v.  
       If j=1, add one to v after each execution.  
       If j=2, add one to u after each execution.  
       If j=3, add one to u and v after each execution.
- \*(d) On completing n executions, take (F<sub>1</sub>) as the next instruction.
- \*(e) If the repeated instruction is a jump or stop instruction, the occurrence of a jump or stop terminates the repetition. In addition, if NIuv is a Threshold jump or an Equality jump, and the jump to address v occurs, (Q) is replaced by the quantity j, (n-r), where r is the number of executions that have taken place.

\* See Repeat Sequence Control, page 25

## TRANSMISSIVE INSTRUCTIONS

- 11 TRANSMIT POSITIVE (TPuv): Replace (v) with (u).
- 13 TRANSMIT NEGATIVE (TNuv): Replace (v) with the complement of (u).
- 12 TRANSMIT MAGNITUDE (TMuv): Replace (v) with the absolute magnitude of (u).
- 15 TRANSMIT U-ADDRESS (TUuv): Replace the 15 bits of (v), designated by  $v_{15}$  through  $v_{29}$ , with the corresponding bits of (u), leaving the remaining 21 bits of (v) undisturbed.
- 16 TRANSMIT V-ADDRESS (TVuv): Replace the right-hand 15 bits of (v) designated by  $v_0$  through  $v_{14}$ , with the corresponding bits of (u), leaving the remaining 21 bits of (v) undisturbed.
- 22 LEFT TRANSMIT (LTjkv): Left circular shift (a) by k places. Then replace (v) with ( $A_L$ ) if  $j=0$  or replace (v) with ( $A_R$ ) if  $j=1$ .
- 35 ADD AND TRANSMIT (ATuv): Add D(u) to (A). Then replace (v) with ( $A_R$ ).
- 36 SUBTRACT AND TRANSMIT (STuv): Subtract D(u) from (A). Then replace (v) with ( $A_R$ ).

## Q-CONTROLLED INSTRUCTIONS

- 51 Q-CONTROLLED TRANSMIT (QTuv): Form in A the number L(Q) (u). Then replace (v) by ( $A_R$ ).
- 52 Q-CONTROLLED ADD (QAuv): Add to (A) the number L(Q) (u). Then replace (v) by ( $A_R$ ).
- 53 Q-CONTROLLED SUBSTITUTE (QSuv): Form in A the quantity L(Q) (u) plus L(Q)' (v). Then replace (v) with ( $A_R$ ). The effect is to replace selected bits of (v) with the corresponding bits of (u) in those places corresponding to 1's in Q.

## REPLACE INSTRUCTIONS

- 21 REPLACE ADD (RAuv): Form in A the sum of D(u) and D(v). Then replace (u) with ( $A_R$ ).
- 23 REPLACE SUBTRACT (RSuv): Form in A the difference D(u) minus D(v). Then replace (u) with ( $A_R$ ).

27 CONTROLLED COMPLEMENT (CC<sub>uv</sub>): Replace ( $A_R$ ) with ( $u$ ) leaving ( $A_L$ ) undisturbed. Then complement those bits of ( $A_R$ ) that correspond to ones in ( $v$ ). Then replace ( $u$ ) with ( $A_R$ ).

54 LEFT SHIFT IN A (LA<sub>uk</sub>): Replace ( $A$ ) with  $D(u)$ . Then left circular shift ( $A$ ) by  $k$  places. Then replace ( $u$ ) with ( $A_R$ ). If  $u$  is the address of the Accumulator, the first step is omitted, so that the initial content of  $A$  is shifted. (The value of  $k$  must not exceed seven bits if the result is to be replaced in  $u$ .)

55 LEFT SHIFT IN Q (LQ<sub>uk</sub>): Replace ( $Q$ ) with ( $u$ ). Then left circular shift ( $Q$ ) by  $k$  places. Then replace ( $u$ ) with ( $Q$ ). (The value of  $k$  must not exceed seven bits if the result is to be replaced in  $u$ .)

### SPLIT INSTRUCTIONS

31 SPLIT POSITIVE ENTRY (SP<sub>uk</sub>): Form  $S(u)$  in  $A$ . Then left circular shift ( $A$ ) by  $k$  places.

33 SPLIT NEGATIVE ENTRY (SN<sub>uk</sub>): Form in  $A$  the complement of  $S(u)$ . Then left circular shift ( $A$ ) by  $k$  places.

32 SPLIT ADD (SA<sub>uk</sub>): Add  $S(u)$  to ( $A$ ). Then left circular shift ( $A$ ) by  $k$  places.

34 SPLIT SUBTRACT (SS<sub>uk</sub>): Subtract  $S(u)$  from ( $A$ ). Then left circular shift ( $A$ ) by  $k$  places.

### TWO-WAY CONDITIONAL JUMP INSTRUCTIONS

46 SIGN JUMP (SJ<sub>uv</sub>): If  $A_{71}=1$ , take ( $u$ ) as NI. If  $A_{71}=0$ , take ( $v$ ) as NI.

47 ZERO JUMP (ZJ<sub>uv</sub>): If ( $A$ ) is not zero, take ( $u$ ) as NI. If ( $A$ ) is zero, take ( $v$ ) as NI.

44 Q-JUMP (QJ<sub>uv</sub>): If  $Q_{35}=1$ , take ( $u$ ) as NI. If  $Q_{35}=\text{zero}$ , take ( $v$ ) as NI. Then, in either case, left circular shift ( $Q$ ) by one place.

### ONE-WAY CONDITIONAL JUMP INSTRUCTIONS

41 INDEX JUMP (IJ<sub>uv</sub>): Form in  $A$  the difference  $D(u)$  minus 1. Then if  $A_{71}=1$ , continue the present sequence of instructions; if  $A_{71}=0$ , replace ( $u$ ) with ( $A_R$ ) and take ( $v$ ) as NI.

42 THRESHOLD JUMP (TJuv): If  $D(u)$  is greater than (A), take (v) as NI; if not, continue the present sequence. In either case, leave (A) in its initial state.

43 EQUALITY JUMP (EJuv): If  $D(u)$  equals (A), take (v) as NI; if not, continue the present sequence. In either case leave (A) in its initial state.

#### ONE-WAY UNCONDITIONAL JUMP INSTRUCTIONS

45 MANUALLY SELECTIVE JUMP (MJjv): If the number  $j$  is zero, take (v) as NI. If  $j$  is 1, 2 or 3, and the correspondingly numbered MJ selecting switch is set to 'jump', take (v) as NI; if this switch is not set to 'jump', continue the present sequence.

37 RETURN JUMP (RJuv): Let  $y$  represent the address from which CI was obtained. Replace the right-hand 15 bits of (u) with the quantity  $y$  plus 1. Then take (v) as NI.

14 INTERPRET (IP--): Let  $y$  represent the address from which CI was obtained. Replace the right-hand 15 bits of ( $F_1$ ) with the quantity  $y$  plus 1. Then take ( $F_2$ ) as NI.

#### STOP INSTRUCTIONS

56 MANUALLY SELECTIVE STOP (MSjv): If  $j=0$ , stop computer operation. If  $j=1, 2$ , or 3 and the correspondingly numbered MS selecting switch is set to 'stop', stop computer operation. Whether or not a stop occurs, (v) is NI.

57 PROGRAM STOP (PS--): Stop computer operation.

#### EXTERNAL EQUIPMENT INSTRUCTIONS

17 EXTERNAL FUNCTIONS (EF-v): As designated by (v) select a unit of external equipment and cause it to perform a function.

76 EXTERNAL READ (ERjv): If  $j=0$ , replace the right-hand 8 bits of (v) with (IOA); if  $j=1$ , replace (v) with (IOB).

77 EXTERNAL WRITE (EWjv): If  $j=0$ , replace (IOA) with the right-hand 8 bits of (v); if  $j=1$ , replace (IOB) with (v). Cause the previously selected unit to respond to the information in IOA or IOB.

61 PRINT (PR-v): Replace (TWR) with the right-hand 6 bits of (v). Cause the typewriter to perform the operation specified by the 6-bit code.

63 PUNCH (PUjv): Replace (HPR) with the right-hand 6 bits of (v). Cause the punch to respond to (HPR). If  $j=0$ , omit seventh level hole; if  $j=1$ , include seventh level hole.

## INSTRUCTION EXECUTION TIMES

Table II contains execution times for the non-repeated instructions in the Model 1103A computer. In each case the references are to the magnetic core storage.

With the exception of the Threshold Jump (42uv) and the Equality Jump (43uv), the execution times for repeated instructions can be determined by subtracting six clockpulses from the non-repeated execution times except for the last execution of the repeated instruction which takes the non-repeated duration. For example, the normal execution time for a Transmit Positive instruction is 17 clockpulses. The execution time,  $R_n$ , for a repeated Transmit Positive is therefore  $(n-1)(17-6)+17$  clockpulses. The execution time for the entire Repeat sequence is:

$$\begin{array}{rcl}
 75 \text{ jn w (Repeat)} & & 26 \text{ clockpulses} \\
 & & + \\
 11 \text{ u v (Transmit Positive)} & R_n: & 11(n-1)+17 \text{ clockpulses} \\
 & & + \\
 45 \text{ - v (Terminal Jump in } F_1) P: & & 10 \text{ clock pulses}
 \end{array}$$

In case of a repeated Threshold Jump or Equality Jump, 5 clockpulses must be added to the time of extracting the next instruction from v if a jump is executed (to be consistent with the counting from  $MP_0$  to  $MP_0$ ). If these instructions are executed  $n$  times without a jump termination, six clockpulses must be added to the last execution time as in the other instructions.

TABLE II

UNIVAC SCIENTIFIC MODEL 1103A

INSTRUCTION EXECUTION TIME\*

MC to MC

36

INSTRUCTION	MICROSECONDS	INSTRUCTION	MICROSECONDS
11uv	34	51uv	42
12uv	36	52uv	44
13uv	34	53uv	72
14--	30	54uk	42+2k
15uv	34	55uk	40+2k
16uv	34	56jv-Stop	4
17-v	28	56jv-No S	18
21uv	58	57--	2
22kv	30+2k	61jv	34(note 1)
23uv	60	63jv	34(note 2)
27uv	50	71uv	$116+14(u)_{35} + 8\sum_1^{35} (u)_i + 4(u)_0$
31uk	32+2k	72uv	$188+14(u)_{35} + 8\sum_1^{35} (u)_i + 4(u)_0$
32uk	32+2k	73uv-Max.	$480+8(A)_{71}$
33uk	34+2k	73uv-Min.	$474+8(A)_{71}$
34uk	34+2k	74jv	$120+Y$ (note 3)
35uv	42	75jnw	$52+R_n+P$ (note 4)
36uv	44	76jv	28 (note 5)
37uv	34	77jv	28 (note 5)
41uv-Jump	52		
41uv-No J.	44		
42uv-Jump	42		
42uv-No J.	42		
43uv-Jump	54		
43uv-No J.	54		
44uv	18		
45jv-Jump	18		
46jv-No J.	18		
46uv	18		
47uv	30		

NOTES:

1. Plus a lockout time of 105ms for successive prints.
2. Plus a lockout time of 16.7ms for successive punches.
3. Where  $Y=2[36-k \pmod{72}]$  and k is the scale factor:  $0 \leq k \leq 71$ . For  $k = 37$ , use value for  $k = 38$ .
4. Where  $P =$  time of terminal jump at  $F_1$  and  $R_n =$  execution time of repeated instruction; if  $n = 0$ ,  $R_n = 0$ .
5. Plus some lockout time for successive reads or writes.

\* Assuming MC access time to be as follows:

Initiate Read (Write) MC

MCP-1  
MCP-2

MC Resume on MCP-3

Total Time for these - 8 us

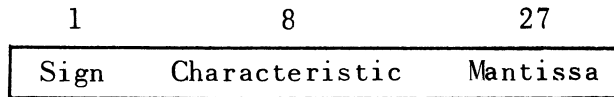
## FLOATING POINT ARITHMETIC

As an optional feature of the 1103A, additional circuitry will be provided to perform nine machine instructions for Floating Point operations. These include add, subtract, multiply, divide, polynomial multiply, inner product and three instructions which facilitate Fixed Point Floating Point conversion and double precision Floating point arithmetic.

### NUMBER REPRESENTATION

For non-zero numbers,  $x \cdot 2^y$ , the mantissa,  $x$ , is represented by a 28-bit ones complement fractional value such that  $\frac{1}{2} \leq |x| < 1$ . In binary, this consists of a sign bit followed by the binary point and then by a significant bit and the other 26 bits of the mantissa.

The integer characteristic,  $y$ , is restricted to  $-128 \leq y \leq 127$   $C = y + 128$ ,  $0 \leq C < 256$ . If the number is positive the mantissa sign-bit is positioned in  $X_{35}$ , the biased characteristic,  $C$ , in  $X_{34}$  through  $X_{27}$  and the mantissa in bits  $X_{25}$  through  $X_0$ . If the number is negative, the absolute value,  $|x| \cdot 2^y$ , is packed in this manner, and the entire word complemented.



Floating Point Word Structure

Zero is represented by the machine integer zero. These conventions preserve the significance of the machine instructions Transit Magnitude, Transmit Negative, Sign Jump, Threshold Jump, Equality Jump, and Zero Jump.

A fractional mantissa which has a significant bit in the first fractional place is said to be *normalized*. A floating point number is said to be represented in *packed form* if the mantissa and characteristic are stored in a single register. If the mantissa and characteristic are stored separately, the representation is termed *unpacked*. Two instructions are available for packing and unpacking a floating point number; these facilitate fixed point/floating point conversion.

### FLOATING POINT INSTRUCTIONS

These instructions assume all operands are floating point normalized packed in accordance with the above conventions. Each is repeatable.

1. Floating Point Add (FAuv).



Form in Q the normalized rounded packed floating point sum of the contents of u and the contents of v.

2. Floating Point Subtract (FSuv)

Form in Q the normalized rounded packed floating point difference of the contents of u and the contents of v.

3. Floating Point Multiply (FMuv)

Form in Q the normalized rounded packed floating point product of the contents of u and the contents of v.

4. Floating Point Divide (FDuv)

Form in Q the normalized rounded packed floating point quotient obtained by dividing the contents of u by the contents of v.

5. Floating Point Inner Product (FIuv)

Form in Q the normalized rounded packed floating point sum of the initial contents of Q and the floating point product of the contents of u and the contents of v. This instruction uses magnetic core location  $F_4 = 00003$  for storage of the initial contents of Q.

6. Floating Point Polynomial Multiply (FPuv)

Form in Q the normalized rounded packed floating point sum of the contents of v and the floating point product of the contents of u and the initial contents of Q. This instruction leaves the mantissa in A with the appropriate sign bits in the characteristic positions.

7. Floating Point Unpack (UPuv)

Store at address u the positive biased characteristic and store at address v the ones complement form of the mantissa.

8. Floating Point Pack Rounded (PRuv)

Replace the contents of u and Q with the normalized rounded packed floating point number obtained from the (possibly) un-normalized mantissa initially in u and the biased characteristic initially in the characteristic position of v.

9. Normalize Exit (NEj—)

If  $j = 1$ , set the NORMALIZE EXIT Flip Flop.  
If  $j = 0$ ; clear the NORMALIZE EXIT Flip Flop.

When this Flip Flop is set the normalization and round

after a floating point arithmetic operation is bypassed and the result is packed in Q.

This instruction facilitates double precision floating point operations, using a programmed combination of ordinary floating point commands and commands modified in this fashion.

### ERRATA

- Page 13, paragraph (c) - - read  
**IOB INSTEAD OF IPB.**
- Page 16, paragraph "Tape to Card  
Converter" - - read **120 CARDS PER MINUTE**  
instead of 180
- Page 18, last line - - read  
**8 MICROSECONDS** instead of 10.

