:020

RICHARD CASTANIAS

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

41 FIRST STREET
SAN FRANCISCO 5, CALIFORNIA
PHONE: DOUGLAS 2-8600

PRELIMINARY

Functional Description

CLASS X COMPUTER

(XU-72E)

15 February 1960

Remington Rand Univac ®
DIVISION OF SPERRY RAND CORPORATION
UNIVAC PARK, ST. PAUL 16, MINNESOTA

PRELIMINARY FUNCTIONAL DESCRIPTION

CLASS X COMPUTER (XU-72E)

PX 1472

15 February 1960

Remington Rand Univac®
DIVISION OF SPERRY RAND CORPORATION
UNIVAC PARK, ST. PAUL 16, MINNESOTA

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

LIST OF TABLES

CLASS X COMPUTER (XU-72E)

1. INTRODUCTION

The Class X Computer is an advanced large-scale solid-state data processing system. This computer is intended primarily for military applications, or wherever reliable high-speed solutions are required for complex off-line or real-time on-line data processing problems. A powerful instruction repertoire in conjunction with a compiler assists the user in the solution of a wide range of problems in fields such as data reduction and analysis, scientific computation, logistics, information retrieval, language transformation, matrix solution, military control and data systems, war games, damage assessment, and military intelligence.

Using advanced computer technology, this computer provides very high computation capability with great economy. Modular construction, in particular the storage capacity and input-output options, enables this computer system to be tailored to the particular requirements of the application.

The central computer is a parallel, binary, 48-bit, two-address system. Options of memory capacity up to 65,536 words are directly addressable. Direct communication can be scheduled between the memory and up to sixteen peripheral systems at a given time. Concurrent input-output word transfers occur without program intervention.

The Class X Computer is equipped with an advanced line of standard peripheral equipment and accommodates the use of specialized peripheral equipment. The complete line of standard Univac peripheral equipment can be

utilized. The generality of the computer input-output permits direct incorporation of future peripheral systems without obsolescence of the central computer.

2. GENERAL DESCRIPTION

The primary system features of the Class X Computer are given in Table I.

The main memory consists of arrays of ferrite cores with a cycle time of 1.5 microseconds. The capacity options are 8,192, 16,384, 32,768, and 65,536 48-bit words in one or two banks with 32,768 words maximum per bank. Two-bank operation provides an effective cycle time of approximately one microsecond.

The control memory comprised of magnetic films contains 128 words with a cycle time of 0.5 microsecond. The control memory provides 32 index registers special control registers, and high-speed auxiliary storage.

Up to sixteen input-output channels provide direct communication between the main memory and peripheral systems. An input or output operation is activated by a computer instruction. Thereafter, the input or output word transfers between memory and the peripheral equipment proceed until completion without any further concern required by the program over the external equipment. The peripheral control unit demands access to the main memory for each input or output word transfer as required by the peripheral system. The computer input-output access control multiplexes the demands and services up to 16 concurrently operating channels. The maximum gross instantaneous transfer rate is 1,333,000 words per second.

The standard peripheral systems include:

Magnetic Drums

Disc Files

TABLE 1. PRIMARY FEATURES OF THE SYSTEM

Item	Description
Type	General-purpose, solid-state, parallel, binary
Word Length	48 bits
Arithmetic	Fixed and floating point
Input-Output Channels	16
Peripheral Equipment	Options: Magnetic drum systems Magnetic tape systems Disc file systems Punched card processors High-speed printer Paper tape units Displays and plotters Teletypewriters Real-time equipment
Main Memory: Capacity	Options: 8,192, 16,384, and 32,768 words in one bank, 16,384, 32,768, and 65,536 words in two banks separately accessed
Cycle Time	1.5 microseconds per bank
Use	Program storage; operand storage
Control Memory: Capacity	128 words
Cycle Time	0.5 microsecond
Use	Index registers; input-output access control; high-speed operand storage
Instructions	105 two-address instructions
Average Instruction Times	3.0 microseconds add* 5.8 microseconds multiply* 21 microseconds divide*

* Time shown includes instruction access, indexing, operand access, and arithmetic. See Appendix III.

Magnetic Tapes

Punched Cards Units

High-Speed Printers

Supervisory Console and Auxiliaries

The specialized peripheral systems include:

A/D and D/A Converters

Electronic Printers

Displays/Plotters/Keysets

Digitized Radar and Communication Links

Multiplex and Switching Units

Other Off-line Systems

Other Real-Time On-Line Systems

Other Computers

The arithmetic is fractional with both stated point and floating point instructions. Double length accumulations and computer masking facilities are provided. The basic add time is 0.8 microsecond, and the basic multiply time is 3.6 microseconds.

A real-time clock with increments of 100 microseconds can be set to interrupt the program after any desired time lapse. Other interrupts mark the completion of input and output operations and afford a safeguard to the accuracy of these transmissions. Breakpoint interrupt enables the operator and executive routines to interrupt and monitor a running program at any specified point.

The instruction format is designed for maximum efficiency. Some examples of the flexibility available to the programmer are listed below:

1) In many two-address operations, the accumulator may be used to store a third operand, thus giving in effect the power of three-address instructions.

2) Instructions that do not reference the arithmetic registers may be executed concurrently with extended sequence operations such as divide and multiply. Therefore, no time need be lost while these operations are being carried out.

3) A special divide step instruction, coupled with a repeat designation, permits the automatic conversion of binary numbers to numbers with any desired radix. A complementary multiply step facilitates conversion in the reverse direction.

Customer services provided with the computer include an automatic coding system and a central library of routines. (See Appendix IV, Application Note 3.)

3. CENTRAL COMPUTER

Figure 1 is a simplified block diagram of the central computer, showing the four major sections: input-output, storage, arithmetic, and control. The abbreviations on the diagram are explained as the operation of the computer is discussed.

a. INPUT-OUTPUT SECTION. - Up to 16 in-out transfer registers (T1, T2, ... T16) communicate with the peripheral equipments. The T-registers are plug-in modules, and only the number and type required by the associated peripheral equipment are installed. Normally, one register, T16, is assigned exclusively to communication with the supervisory control console, including the monitor typewriter and paper tape units. The remaining 15 T-registers are tailored

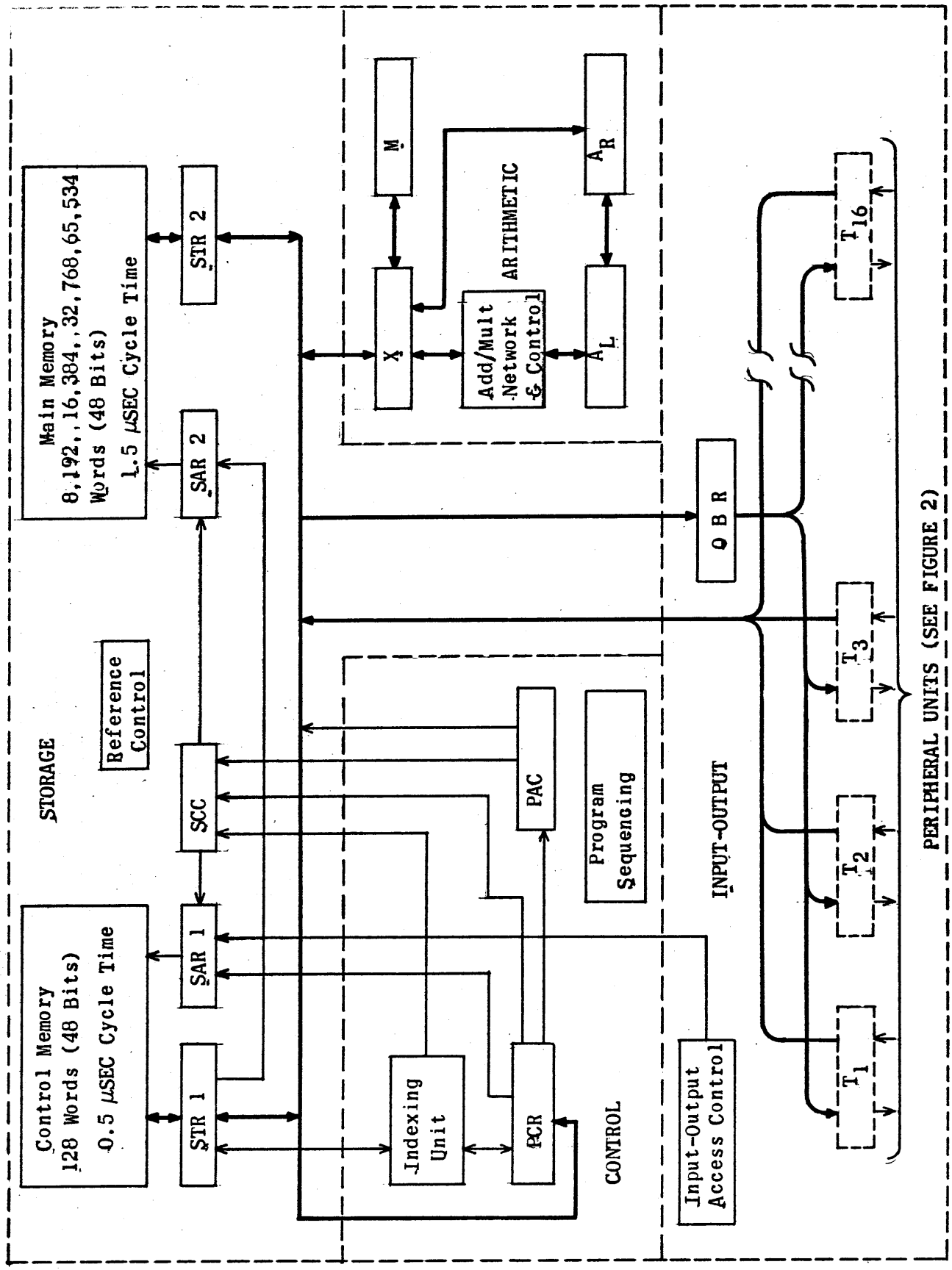

Figure 1. Simplified Block Diagram of Centrall Computer

to the requirements of the particular units to which they are connected. In general, the registers have shifting properties such that output characters of the correct bit length are disassembled from the 48-bit computer word, and input characters are assembled into full computer words. Figure 2 shows peripheral equipment.

Transfer of data between the computer and peripheral equipment need only be initiated by the program. Thereafter, the transfer is governed by the input-output access control, which monitors the current number of words to be transferred and specifies the addresses in main memory to or from which the data are transmitted. The access control thus frees the remaining computer elements, enabling them to continue with the execution of the main program.

Any number of the channels may be in concurrent operation; however, the number that can be handled efficiently depends upon the data transfer rate of the peripheral equipment. For example, if one magnetic drum channel is in use, up to 14 magnetic tape channels can be operated efficiently at the same time. If a second drum channel is used, tape channel capacity is reduced to seven for a given bank of main memory. (See Appendix IV, Application Note 2.)

Working in conjunction with the access control, priority control circuits resolve situations in which two or more external equipments simultaneously attempt to communicate with the computer. In each case, priority is given to the one with the lower channel number.

A 48-bit word originating in memory passes through an output buffer register (OBR) before entering the assigned T-register, where it is disassembled into appropriate character lengths. Input data originating outside the computer are transferred to the storage section directly after being assembled into a 48-bit word in the T-register.

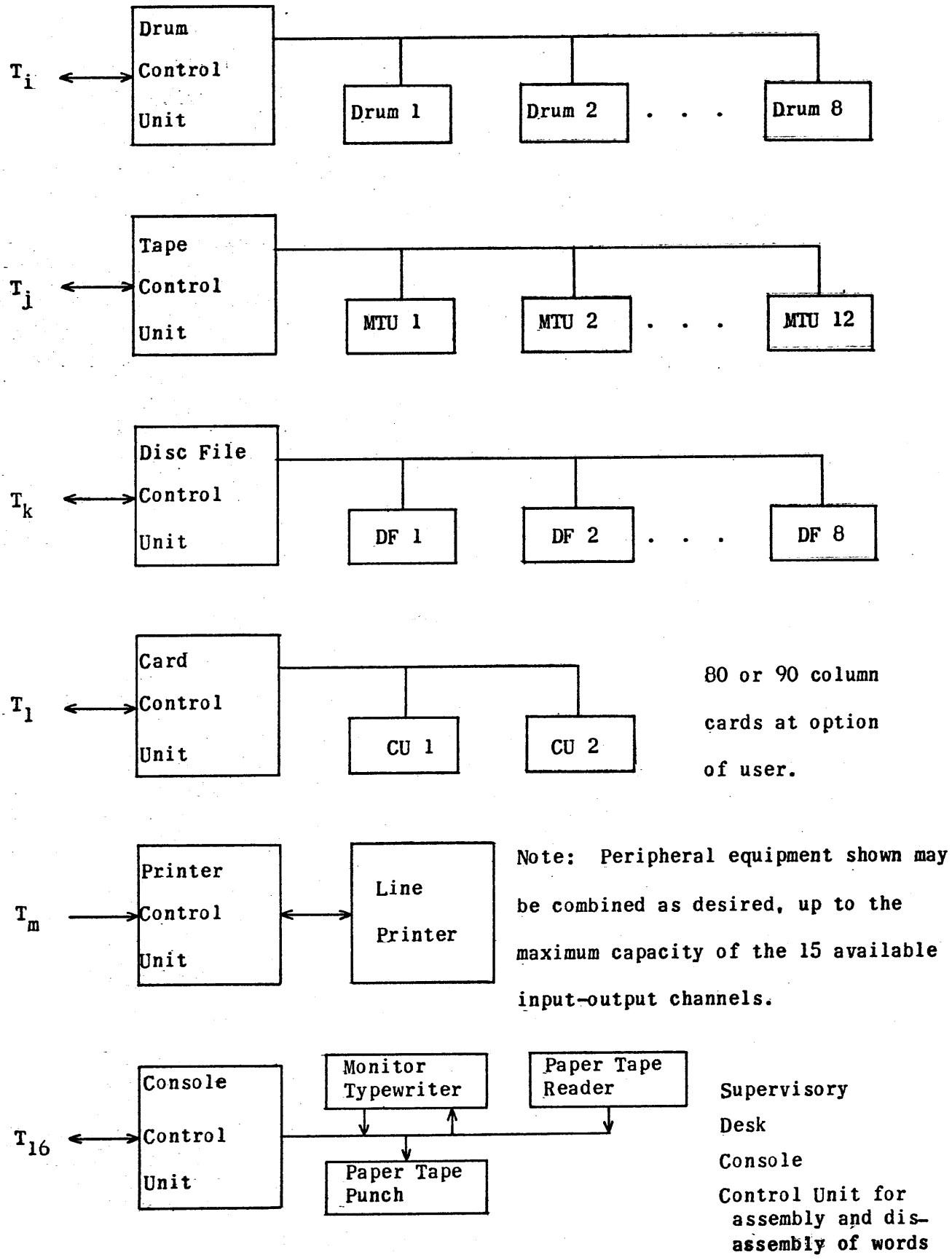

Figure 2. Peripheral Equipment Options

b. STORAGE SECTION. - The storage section consists of main memory, control memory, and their associated address, transfer, and control circuits.

(1) MAIN MEMORY. - The main memory consists of modular arrays of ferrite cores coincident current driven. Capacity options are 8,192, 16,384, and 32,768 words in single-bank operation and 16,384, 32,768, and 65,536 words in two-bank operation. Read access time in a memory bank for any address is 0.6 microsecond; complete cycle time is 1.5 microseconds. Two banks function as independent asynchronous units which are overlapped to provide the effect of faster memory. Two-bank operation provides an effective cycle time of 1 microsecond approximately.

(2) CONTROL MEMORY. - The 128-word control memory consists of deposited magnetic films. The film array is word organized with destructive readout providing 48-bit parallel operation. Read access for any address is 0.2 microsecond; complete cycle time is 0.5 microsecond. The write portion of the cycle on some operations is delayed to allow a read-out item to be modified before return to the same address.

(3) STORAGE ALLOCATION. - The first 194 of the 65,536 addresses have the assignments shown in Table 2.

(4) MEMORY ACCESS. - Memory is addressed via storage class control (SCC) from the indexing unit, program control register (PCR), or program address counter (PAC). The address is first decoded in SCC to determine whether the reference is to main or control memory. From SCC, the address is transferred to the appropriate storage address register (SAR 1 for control memory and SAR 2 for the main memory). SCC also controls access to the five addressable program control and arithmetic registers.

TABLE 2. ADDRESS ASSIGNMENT

Decimal Address	Octal Address	Function
00000-00031	000000-000037	Index registers (32)
00032-00047	000040-000057	Input-output access control (16)
00048	000060	Repeat count
00049	000061	Real-time clock
00050-00127	000062-000177	Unassigned auxiliary parking registers (78)
The above addresses are in the Control Memory (0.5 microsecond cycle). The addresses immediately below are for the addressable registers.		
00128	000200	Accumulator (A_L)
00129	000201	Accumulator extension (A_R)
00130	000202	Mask register (M)
00131	000203	Breakpoint register
00132	000204	Jump-stop register
The addresses below are in the Main Memory (1.5 microsecond cycle).		
00133	000205	Breakpoint interrupt
00134	000206	Real-time clock interrupt
00135	000207	Real-time synchronization interrupt
00136-00151	000210-000227	Input-output completion interrupts (16)
00152-00167	000230-000247	External input-output interrupts (16)
00168-00183	000250-000267	Input-output error interrupts (16)
00184-00193	000270-000301	Other error interrupts (10)
00194-65535	000302-177777	Unassigned

(Note: The interrupt addresses are subroutine entrances.)

Memory access for input-output transfers is somewhat different. Input-output access control inserts in SAR 1 the address which corresponds to the channel being serviced so that the corresponding input-output access control word is read out into STR 1. The address-portion is transferred directly to SAR 2 and the reference to the main memory is then made for the input-output data transfer. (See discussion of word formats in paragraph 5.)

c. ARITHMETIC SECTION. - The arithmetic section comprises four registers, each 48 bits long. Associated with the registers are arithmetic sequence control circuits and counters which govern the execution of the algorithms for addition, subtraction, multiplication, division, and shifting. Circuits for sensing equality and relative magnitude of the contents of the registers determine if the jump conditions specified by particular instructions are fulfilled.

Operands enter the arithmetic section via the exchange register, X. For the corresponding operations, X contains the addend, the subtrahend, the multiplicand, and the divisor. X is not addressable.

The double-length accumulator consists of two halves designated A-left, A_L, and A-right, A_R. In addition to the normal 48 arithmetic bits, A_L has two overflow bits. This register retains the results of most arithmetic operations: sums, differences, and the higher order bits of products. It contains the dividend for division and the remainder after completion of division. The remainder in A_L is in position to become the dividend if a second division follows the first. A_R is a 48-bit register which contains the lower order bits of the product after a multiplication and the low order bits of the dividend before a division.

Both A_L and A_R have shifting properties. Their contents may be shifted left or right, end-off or end-around, and separately or together (as one 96-bit register). Both registers are addressable.

The mask register, M, contains the masking quantity during logical instructions, the multiplier during multiplication, and the quotient after division. The M register has left shift properties and is addressable.

A complete range of instructions for both fixed and floating point arithmetic, together with a highly flexible indexing capability, simplifies the coding of complex calculations. Appendix I contains a summary of the instructions.

Add time, excluding memory references, is 0.8 microsecond, multiply time 3.6 microseconds. (Times including memory access appear in Table 1, also Appendix III.) Divide time ranges from a minimum of 10.4 microseconds to a maximum of 28.8 microseconds, excluding memory references.

d. CONTROL SECTION. - The main control and timing circuits supply control signals which synchronize the execution of the instructions. The particular instruction to be executed is determined by the contents of the program control register, PCR. The exact organization of the instruction word is explained in paragraph 5; it is sufficient here to note that, in addition to the operation code, the instruction contains the addresses (termed "u" and "v") of the two operands involved. Both addresses may be indexed if the programmer so specifies. An adder independent of the arithmetic section is provided for indexing. After indexing, the modified addresses are referred to as U and V, to distinguish them from the unmodified base addresses, u and v.

The address of the next instruction is contained in the program address counter, PAC. Usually, PAC is incremented by one, so that the instructions are read from sequential addresses in main memory. Jumps inhibit the normal sequence and force new addresses into PAC. An interrupt causes an instruction to be read out of the fixed address of main memory which corresponds to the type of interrupt. If this instruction is a return jump, the address in PAC is stored at the u portion of the fixed address and the v portion becomes the new program address.

The control section contains two supplementary control registers, not shown in Figure 1. These are the breakpoint register and the jump-stop register; both are addressable. The breakpoint register contains the program address at which a breakpoint interrupt is desired. At a time during execution of each instruction, the address in PAC is compared with the breakpoint address. If coincidence occurs, a breakpoint interrupt takes place after execution of the instruction obtained from the breakpoint address. The jump-stop register contains conditional control bits which are compared with the U portion of the Manual Jump-Stop instruction (see Appendix I).

4. PERIPHERAL EQUIPMENT

A variety of external storage units may be used with the computer. The characteristics of some of the major types are described below. Figure 2 contains a summary of options. Because of the modular construction and adaptable design of the T registers, the computer can also communicate with real-time devices such as analog-to-digital and digital-to-analog converters, digital communication terminals, digitized radar and tracking systems, displays, and keysets.

Up to 16 channels are provided for direct communication with peripheral equipment. Normally, channel 16 is reserved for the supervisory console. Whenever communication is desired to more than 15 groups of peripheral units, additional switching and multiplexing units may be employed. These are tailored to the particular requirements of the application.

a. MAGNETIC DRUM SYSTEM. - A magnetic drum system comprises one drum control unit and from one to eight high-density flying-head drums. This system provides the computer with a large-volume, medium-access memory in optional capacities of from 524,288 to 4,194,304 forty-eight bit words. Up to 15 such drum systems may directly communicate with the computer, giving the computer more than 62 million words of supplementary storage.

A single External Function instruction (explained in paragraph 5) initiates the transfer of any number of words to or from a drum. A transfer begins at a specified location and continues sequentially. Words may be transferred between consecutive drum and core memory addresses since the drum word interval is 8 microseconds and the main memory cycle time is 1.5 microseconds. Transfers between drum and main memory begin at the time the angular position of the drum coincides with the specified starting address. The average access time for a single word is 17 milliseconds, with an 8-microsecond interval between succeeding words. The input-output section of the computer allows time sharing of the flying-head drum with magnetic tape units and other peripheral devices.

b. DISC FILE STORAGE SYSTEM. - The disc file storage system is a mass storage device employing from one to eight disc file units and one control unit. The capacity of one file unit is 8,388,608 forty-eight bit words; the maximum capacity of one system is 67,108,864 words. Up to 15 systems may be

used directly with the computer. This allows as many as 120 disc files or a capacity of over one billion words of storage.

A single External Function instruction initiates the reading or writing of any number of words to or from a disc file. Reading begins at any word location and continues until the specified number of words has been transferred to main memory. Writing must begin at the start of a disc sector, of which there are 256 per file unit.

Each file unit has one read-write head for each disc surface. Maximum head positioning time is 300 milliseconds. Transfer rate is 24 microseconds per word. Minimum access time to any given word is 372 microseconds, and the maximum is 350 milliseconds.

c. MAGNETIC TAPE SYSTEM. - A magnetic tape system consists of a control unit and from one up to 12 high-performance tape units. Up to 15 such systems (180 tape units) may be directly connected to the computer. A tape system performs the following operations: read forward or backward, write, move tape without reading or writing, move tape with read checking only, and rewind on the specified tape unit. The particular operation is initiated by the External Function instruction (see paragraph 5). Only one tape unit per tape system on a given input-output channel may read, write, or move tape at any given time, but any number of units may rewind while a read, write, or move tape operation is in process. Once initiated by the program, transfer of data between tape and main memory proceeds without further intervention by the program. The transfer rate is up to 100,000 six-bit characters per second. Eight characters form one computer word. An arbitrary number of words may be recorded per block. Each character on tape has one parity bit.

Several commonly used tape formats can be provided. Compatibility is achieved at the tape control unit, which combines the instruction decoders, word register, counters, data transmission facilities, and other control circuits with the appropriate amplifiers and heads of the tape units which are required to handle the various formats.

d. PUNCHED CARD EQUIPMENT. - Punched-card equipment, either 80-column or 90-column, may be used for on-line operation. One or two card units of the same type may be connected by means of a card control unit to one input-output channel. If operation with both 80- and 90-column cards is desired, two control units and two computer channels are used. The standard card units process cards at a sensing rate of 450 per minute and a punching rate of 300 per minute.

e. HIGH-SPEED PRINTER. - An alphanumeric line printer may be operated on-line via a control unit connected to an input-output channel. Printing speeds up to 72,000 characters per minute may be achieved. Fifty-one different alphanumeric characters may be printed. The line printer control unit may alternately control either of two line printers.

f. SUPERVISORY CONTROL CONSOLE AND AUXILIARY EQUIPMENT. - The supervisory control console with conventional indicator lights and controls includes an input-output typewriter for monitoring computer replies and modifying programs. The typewriter operates at a rate of ten characters per second. The console may be placed in any location convenient to the central computer cabinets.

Auxiliary equipment available separately or as an adjunct to the console includes two options of paper-tape reader and two options of paper-tape punch. The readers operate at 330 or 1000 characters per second; the punches operate at 60 or 300 characters per second.

5. INSTRUCTIONS

a. ANALYSIS BY TYPE. - A brief description of the major types of instructions is given in the following paragraphs. See Appendix I for instruction repertoire.

(1) TRANSMIT. - Fifteen transmissive instructions enable the contents of one memory address to be transferred to a second address. A complete 48-bit word, half words (24 bits), or only the u or v portion (16 bits) of the word may be so transmitted. The left and right half words may cross over or transfer in-line. Similarly the u and v portions may cross over or transfer in-line. In the case of complete words, positive, negative, or absolute values may be transferred. The arithmetic registers are not affected by the transmission unless they are addressed by the instruction. One transmissive instruction permits the contents of two memory addresses to be interchanged.

(2) ARITHMETIC. - The arithmetic instructions may be divided into three groups: arithmetic-transmit, replace, and hold. All three groups leave the results in the arithmetic registers; the first two also place the results in memory.

Addition, subtraction, multiplication, and division instructions are available in both fixed and floating point form; there are 16 fixed and 12 floating point instructions. Two additional instructions provide for conversion between fixed point and floating point forms.

(3) SHIFT. - Twenty-nine types of shift instructions afford great versatility in shifting and replacing the contents of memory and in shifting the contents of the arithmetic registers. A family of instructions combines

shifting with addition, subtraction, and rounding. Two scale factor instructions are also provided.

(4) JUMP. - Nineteen jump instructions give the programmer a powerful means of conditional branching in program logic. Among the jump conditions are tests of sign, equality and inequality, and absolute magnitude. Four tests of logical products facilitate programming for information retrieval.

(5) LOGICAL. - Eight logical instructions give the programmer the ability to change, extract, or complement any desired portion of a word. Some of the instructions include the addition or subtraction of the contents of the accumulator to or from the logical result. In all these instructions, the location specified by V receives the result as well as A_L.

(6) SPECIAL. - Of the four special instructions, two are used for converting numbers between any radix and binary. A third special instruction, External Function, initiates and controls input-output transmissions. An Internal Function instruction assists in special internal functions, such as real-time clock and internal errors.

b. WORD FORMATS. - All word formats described below are illustrated in Figure 3.

(1) GENERAL INSTRUCTION FORMAT. - Although some differ in minor details, the instructions in general are organized in five parts:

GENERAL INSTRUCTION WORD					
F	u	L	B	v	
00 07	08 23	24 26	27 31	32 47	

F - Function Code
u - "u" Address; v - "v" Address
L - Index Mode Designator
B - Index Register Designator

INDEX REGISTER WORD					
±	Δu	Iu	±	Δv	Iv
00	07 08	23 24	31	32	47

Δu - Iu Modifier
Iu - u Increment
Δv - Iv Modifier
Iv - v Increment

FIXED POINT OPERAND	
±	OPERAND
00	47

± - Sign Bit

FLOATING POINT OPERAND		
±	CHARACTERISTIC	MANTISSA
00	01 11	12 47

± - Sign Bit

EXTERNAL FUNCTION INSTRUCTION WORD				
F	U	-	B	V
00 07	08 23	24 26	27 31	32 47

F - Function Code
U - Address of Input-Output Function Word
B - Channel Designator
V - Exit Address if Channel Is Busy

INPUT-OUTPUT ACCESS CONTROL WORD							
N	R	S	H	C'	W	C	V
00	01	02	03	04 07	08 23	24 31	32 47

N - Inhibit Main Memory Transfer
R - New Access Control Word Follows
S - Increment or Decrement V
H - Interrupt when W = 0
C,C' - Address of Next Access Control Word
W - Number of Words to be Transferred
V - Address for Next Data Transfer

Figure 3. Word Formats

Bits	Code	Description
00-07	F	The function code, specifying the particular operation to be performed, such as add or subtract.
08-23	u	The u address, being the base (unmodified) location of the first operand.
24-26	L	The index mode designator, signifying how the u and the v addresses are to be modified.
27-31	B	The index register designator, selecting one of the 32 index registers.
32-47	v	The v address, being the base (unmodified) location of the second operand.

(2) INDEX REGISTER FORMAT. - The contents of the 32 index registers are identical in format, consisting of:

Bits	Code	Description
00-07	Δ u	The increment added to the u index, if so specified in the instruction.
08-23	Iu	The u index which, when added to the base address u, forms U, the absolute address of the first operand.
24-31	Δ v	The increment added to the v index, if so specified in the instruction.
32-47	Iv	The v index which, when added to the base address v, forms V, the absolute address of the second operand.

The indexing unit performs address modifications as 16-bit one's complement operations which treat the indices and increments as positive or negative quantities depending upon the left-most bit.

Indexing is performed according to the mode specified by the L-portion of the instruction:

L	Index Mode
0	No indexing.
1	u + Iu = U.
2	v + Iv = V.
3	u + Iu = U; v + Iv = V.
4	Repeat mode (see following paragraph).
5	u + Iu = U; Iu + Δ v $\rightarrow$ Iu.
6	v + Iv = V; Iv + Δ v $\rightarrow$ Iv.
7	u + Iu = U; v + Iv = V; Iu + Δ u $\rightarrow$ Iu; Iv + Δ v $\rightarrow$ Iv.

(3) REPEAT MODE. - If the L portion of any instruction (except the external function instruction) is 4, the instruction will be executed k times. This assumes the repeat count, k, and the index register to be used have been set up initially. After each instruction, R is reduced by one. When k reaches zero, the repeat is terminated and the program proceeds to the next instruction. The index register selected by B in the repeated instruction is used as if L were 7 on the initial execution of the instruction (i.e., both u and v are indexed with Iu and Iv, and the indices Iu and Iv are incremented with Δ u and Δ v, respectively). Thereafter, for each subsequent execution of the instruction, Δ u is added to U, Δ v is added to V, Δ u is added to Iu, and Δ v is added to Iv. The running absolute addresses, U and V, are retained in PCR and the running relative addresses are retained in STR-1. Whenever the repeat mode is terminated, the index register receives the relative addresses, which are the original indices plus the sum of the increments used.

Termination of the repeat mode before k equals zero leaves the relative addresses in the index register, the remainder of k in the repeat count of control memory, and the unmodified instruction at its original location in

memory. If the termination was caused by a jump condition having been met, the relative addresses of the operands causing the condition may be extracted from the index register before reinitiating the repeated jump test instruction. If termination is caused by an interrupt, the current program address (of the repeated instruction) can be stored by means of a return jump instruction located at the address corresponding to the particular interrupt. Upon re-entering at the repeated instruction location, the repeat process begins with the remaining k count and the index register with the Iu and Iv portions of the index register giving the relative addresses of the next set of operands to be used; i.e., the repeat mode picks up at precisely the place it left off when interrupted.

(4) FIXED POINT FORMAT. - Fixed point operands are represented by 47 bits and one sign bit. The sign bit occupies the highest order position.

(5) FLOATING POINT FORMAT. - Floating point operands consist of a sign bit (highest order) 11 bits of characteristic including one bias bit, and 36 bits of mantissa.

(6) EXTERNAL FUNCTION INSTRUCTION FORMAT. - The external function instruction has a special format:

Bits	Code	Description
00-07	F	The operation code, specifying External Function.
08-23	U	The absolute address of the input-output function word (see below).
24-26	-	Not used
27-31	B	The channel designator, selecting one of the 16 input-output channels.

Bits	Code	Description
32-47	V	The absolute address of the instruction to which a jump is made if the selected channel is busy.

Because the B-portion of the instruction is used for the channel designator, neither U nor V can be indexed. The program must be coded directly with the absolute addresses involved. (Although unindexed, the addresses are capitalized since they are absolute addresses.)

The External Function (EF) instruction tests the specified input-output channel (designated by B) to determine if it is busy. If the channel is busy, a jump is made to the address specified by the V portion of the EF instruction. If the channel is not busy, the contents of the address specified by the U portion of the EF instruction are transferred to the T register at the specified channel. The word so transferred is an input-output function word which specifies those operations to be performed by the peripheral control unit connected to that channel. The peripheral control unit interprets the input-output function word for selection of external unit and initiates the specified operation (e.g., in case of a tape control unit, initiate read, write, move tape, or rewind the specified one of the 12 tape units).

After an input or output operation is initiated in the peripheral control unit, this unit requests the transfer of words between the main memory and the peripheral device via the T register which is functionally a part of the peripheral control unit as the data assembly register. The input-output access control along with its priority control performs the required references to main memory for all 16 channels without any intervention of the running program. Address and other access control information for each channel are

contained in an input-output access control word which is retained in the control memory at a location which corresponds to the channel number. The input-output access control uses the STR of the control memory along with the indexing unit to interpret and modify the access control words.

The initial input-output access control word must be loaded into the control memory location corresponding to the channel activated. This is accomplished by a transmit instruction which must immediately follow the EF instruction.

Formats of the EF instruction and the input-output access control word are shown in Figure 3. The input-output function word consists of those arbitrary bit patterns required by the type of peripheral control unit employed.

(7) INPUT-OUTPUT ACCESS CONTROL WORD FORMAT. - The make-up of this control word is as follows:

Bits	Code	Description
00	N	If "1", suppress transfer of data to or from main memory.
01	R	If "1", obtain new input-output access control word from address C when W=0.
02	S	If "0", increment V by one; if "1", decrement V by one.
03	H	If "1", interrupt main program when W=0.
04-07	C′	Used with C to make up a 12-bit address.
08-23	W	Number of words to be transferred.
24-31	C	Address of next input-output access control word.
32-47	V	Absolute address for next data transfer (main memory only, 65,536 locations).

When a peripheral control unit n demands a memory access, the access control word corresponding to channel n is read out to STR 1. The V portion is transferred to SAR 2 for the memory read or write with T_n. V is incremented or decremented according to S. The word count W is decremented and tested for zero. If not zero, the access control word is restored in the auxiliary memory. If W is zero, H is sensed for program interrupt or not, and R is sensed for procurement of a new control word or not. If R is 1, C is transferred to SAR 2 for a memory transfer of a new control word. When W is 0 and R is 0, the peripheral control unit receives a stop signal.

c. SPECIAL FEATURES

(1) EASE OF PROGRAMMING. - The instruction repertoire is designed to facilitate programming in the following areas:

(a) Powerful indexing and repeat control.

(b) Functional symmetry of the repertoire; e.g., conditional jumps on both zero and non-zero, equality and inequality; arithmetic and shift instructions, divide step and multiply step.

(c) Directly addressable arithmetic registers as well as all special addresses. These are all a part of a uniform addressing structure.

(d) Easily managed input-output operations. After initiation, further program intervention is not required. Termination of any input-output operation can interrupt the program as desired. The program, if desired, can also directly monitor

any input-output operation by examination of the access control word. Input-output data transfers can utilize scattered memory locations without any required program intervention.

(e) The ability to cascade interrupts, overflow detection, breakpoint, real-time clock and real-time synchronization. Specifically these features may be utilized in debugging routines, diagnostic procedures, program tracing, program interrogation, executive routine monitoring, multiple program operation, and other on-line and off-line uses.

(2) OVERFLOW DETECTION. - Upon detection of an overflow (add, divide, shift, floating point, overflow or underflow, etc.) control is transferred to an appropriate error address (according to the error type) where a return jump instruction is executed. This records the address at which the error occurred, appropriate action is taken by a subroutine, and control is then returned to the main program.

(3) BREAKPOINT FEATURE. - An addressable 16-bit register, which may be loaded either from the console or by the program, is monitored by PAC. When this register equals the contents of PAC an interrupt takes place and control is transferred to the fixed octal address 205. A return jump instruction then enters an appropriate subroutine the programmer may have arranged.

This breakpoint feature has two major uses: 1) to assist in program debugging tracing, program interrogation, etc., and 2) to assist executive routines in various monitoring operations.

(4) REAL-TIME CLOCK. - The right-hand 24 bits of octal address 61 in the Auxiliary Memory provides the real-time clock feature. This 24-bit count is automatically incremented by one in the indexing unit. The time interval between increments is 100 microseconds as provided by an internal oscillator. When overflow of the 24-bit count is detected in the indexing unit, an internal interrupt occurs which transfers control to octal address 206. The oscillator can be turned on or off manually or by the internal function instruction. The direct addressability of the clock count allows presetting or time readout as desired. The interrupt routine may, if desired, provide a programmed extension of the real-time clock into the left-hand 24 bits of the clock address.

(5) INTERRUPTS. - Many levels of control can be exercised by utilization of the numerous forms of interrupts provided. Occurrence of an interrupt causes control to be transferred to a fixed address with a return jump. Each form of interrupt has an explicit fixed address; such fixed addresses are provided. Each fixed address serves as an entrance to a routine which corresponds to the type of interrupt. Deferred or immediate action, relative priority of action, and any other degree of sophistication can be programmed.

Several classes of interrupts are provided. The 28 internal interrupts include 16 for input-output termination, one for breakpoint, one for real-time clock, and 10 for internal error conditions. Overflow detection is included in these internal errors. The 33 external interrupts include 16 for error detection in the peripheral control units, 16 for external requests, and one for real-time system synchronization. The 16 external request interrupts enable external devices connected to the input-output channels to demand the attention of the computer as required by the external devices. The real-time

synchronization interrupt is independent of the 16 input-output channels. The synchronization interrupt accepts signals from an external generator of any desired frequency. The external generator may be a supplementary real-time clock for the computer or a master clock of a multi-computer complex.

APPENDIX I

INSTRUCTION REPERTOIRE

(See Appendix II for Glossary of Symbols and Terms)

Transmissive Instructions

TPE	Transmit Positive Entire	$(U) \longrightarrow V$
TNE	Transmit Negative Entire	$-(U) \longrightarrow V$
TPM	Transmit Positive Magnitude	$\|(U)\| \longrightarrow V$
TNM	Transmit Negative Magnitude	$-\|(U)\| \longrightarrow V$
TIC	Transmit Interchange	$(U) \longrightarrow V$, $(V) \longrightarrow U$
TUA	Transmit u-Portion	$(U)_u \longrightarrow V_u$
TVA	Transmit v-Portion	$(U)_v \longrightarrow V_v$
TLL	Transmit Left Half	$(U)_l \longrightarrow V_l$
TRR	Transmit Right Half	$(U)_r \longrightarrow V_r$
TLR	Transmit Left Half to Right Half	$(U)_l \longrightarrow V_r$
TRL	Transmit Right Half to Left Half	$(U)_r \longrightarrow V_l$
TUV	Transmit u-Portion to v-Portion	$(U)_u \longrightarrow V_v$
TVU	Transmit v-Portion to u-Portion	$(U)_v \longrightarrow V_u$
TCL	Transmit Constant Left	$(PCR)_r \longrightarrow U_l$
TCR	Transmit Constant Right	$(FCR)_r \longrightarrow U_r$

Fixed Point Arithmetic Instructions

APT	Add Transmit	$(A_L) + (U) \longrightarrow V$ and A_L
SPT	Subtract Transmit	$(A_L) - (U) \longrightarrow V$ and A_L
RAT	Replace Add Transmit	$(U) + (V) \longrightarrow V$ and A_L
RST	Replace Subtract Transmit	$-(U) + (V) \longrightarrow V$ and A_L
CAA	Clear Add Add	$(U) + (V) \longrightarrow A$

HAA	Hold Add Add	$(A_L) + (U) + (V) \longrightarrow A_L$
CSA	Clear Subtract Add	$-(U) + (V) \longrightarrow A$
HAS	Hold Add Subtract	$(A_L) + (U) - (V) \longrightarrow A_L$
HSA	Hold Subtract Add	$(A_L) - (U) + (V) \longrightarrow A_L$
HSS	Hold Subtract Subtract	$(A_L) - (U) - (V) \longrightarrow A_L$
MLP	Multiply	$(U) \cdot (V) \longrightarrow A$
HMA	Hold Multiply Add	$(A_L) + \left[(U) \cdot (V)\right] \longrightarrow A$
HMS	Hold Multiply Subtract	$(A_L) - \left[(U) \cdot (V)\right] \longrightarrow A$
HPA	Hold Polynomial Add	$\left[(A_L) \cdot (U)\right] + (V) \longrightarrow A$
DPT	Divide Transmit	$(A) \div (U) \longrightarrow$ V and M, remainder in A_L
DIV	Divide	$(U) \div (V) \longrightarrow$ M, remainder in A_L

Shift Instructions*

Replace Shifts

RLC	Replace Left Circular	$(U) \longrightarrow A_L$, shift (A_L), $(A_L) \longrightarrow U$
RLL	Replace Left Long, Circular	$(U) \longrightarrow A_L$, shift (A), $(A_L) \longrightarrow U$
RRL	Replace Right Long, Open	$(U) \longrightarrow A_L$, shift (A), $(A_L) \longrightarrow U$
RRO	Replace Right, Open	$(U) \longrightarrow A_L$, shift (A_L), $(A_L) \longrightarrow U$

A-right Transmissive Shifts

TLC	Transmit left circular	Shift (A_R), $(A_R) \longrightarrow U$
TLL	Transmit left long	Shift (A), $(A_R) \longrightarrow U$

*V specifies the shift count; 7 bits, 41-47, are the shift count; when bit 32 is 1, the replace at U is suppressed.

TRL	Transmit Right Long	Shift (A), $(A_R) \longrightarrow U$
TRS	Transmit Right	Shift (A_R), $(A_R) \longrightarrow U$

Arithmetic Shifts (Add then Shift)

ALC	Add Left Circular	$(A_L) + (U) \longrightarrow A_L$, shift (A_L)
ALL	Add Left Long	$(A_L) + (U) \longrightarrow A_L$, shift (A)
ARL	Add Right Long	$(A_L) + (U) \longrightarrow A_L$, shift (A)
ARS	Add Right	$(A_L) + (U) \longrightarrow A_L$, shift (A_L)
SLC	Subtract Left Circular	$(A_L) - (U) \longrightarrow A_L$, shift (A_L)
SLL	Subtract Left Long	$(A_L) - (U) \longrightarrow A_L$, shift (A)
SRL	Subtract Right Long	$(A_L) - (U) \longrightarrow A_L$, shift (A)
SRS	Subtract Right	$(A_L) - (U) \longrightarrow A_L$, shift (A_L)

Arithmetic Shifts (Shift, then Add)

LCA	Left Circular and Add	Shift (A_L), $(A_L) + (U) \longrightarrow A$
LLA	Left Long and Add	Shift (A), $(A_L) + (U) \longrightarrow A$
RLA	Right Long and Add	Shift (A), $(A_L) + (U) \longrightarrow A$
RSA	Right Shift and Add	Shift (A_L), $(A_L) + (U) \longrightarrow A$
LCS	Left Circular and Subtract	Shift (A_L), $(A_L) - (U) \longrightarrow A$
LLS	Left Long and Subtract	Shift (A), $(A_L) - (U) \longrightarrow A$
RLS	Right Long and Subtract	Shift (A), $(A_L) - (U) \longrightarrow A$
RSS	Right Shift and Subtract	Shift (A_L), $(A_L) - (U) \longrightarrow A$

Shift and Round

TLR	Left Long and Round	Shift (A), round, $(A_L) \longrightarrow U$
TRR	Right Long and Round	Shift (A), round, $(A_L) \longrightarrow U$

In the above two instructions rounding occurs if $(A_R)_{00}$ and $(A_L)_{00}$ are different.

Replace Shift in M

RML	M-replace Left Circular	$(U) \longrightarrow M$, shift (M), $(M) \longrightarrow U$

Scale Factor

SFU	Scale Factor U	$(U) \longrightarrow A_L$, Scale, $k \longrightarrow V_v$

Left circular shift (U) in A until $(A_L)_{00}$ and $(A_L)_{01}$ are different; number of places shifted, k, to V_v.

SFA	Scale Factor A	Scale (A), $k \longrightarrow V_v$ and $(A_L) \longrightarrow U$

Left long circular shift (A) until $(A_L)_{00}$ and $(A_L)_{01}$ are different; $k \longrightarrow V_v$ and $(A_L) \longrightarrow U$.

Logical Instructions

LMT	Logical Multiply Transmit	$L\,(U)\,(M) \longrightarrow V, A_L$
LMR	Logical Multiply Replace	$L\,(U)\,(V) \longrightarrow V, A_L$
LAR	Logical Add Replace	$(U) \oplus (V) \longrightarrow V, A_L$
LAA	M-controlled Add	$(A_L) + L(U)\,(M) \longrightarrow V, A_L$
LAS	M-controlled Subtract	$(A_L) - L(U)\,(M) \longrightarrow V, A_L$
LDA	M-controlled Substitute*	$L(U)\,(M) + L(V)\,(M)' \longrightarrow V, A_L$
LRR	Logical OR Replace	$(U) \bar{\oplus} (V) \longrightarrow V, A_L$
LSR	Logical Subtract	$(U) \ominus (V) \longrightarrow V, A_L$

Note the uniformity involved, in that in all cases V is the recipient of the information, as well as A_L.

Jump Instructions

JPE	Jump Positive Equality	If (A) = (U) go to V
JIE	Jump Inequality	If $(A) \neq (U)$ go to V
JPT	Jump Positive Threshold	If (A) < (U) go to V

*Here the prime on (M) denotes complement.

JMT	Jump Magnitude Threshold	If (A)<|(U)| go to V
JPL	Jump Positive Less Than	If (A)> (U) go to V
JML	Jump Magnitude Less Than	If (A)<|(U)| go to V
JLE	Jump Logical Product Equal	If (A) = [L (M) (U)] go to V
JLT	Jump Logical Product Threshold	If (A)< [L (M) (U)] go to V
JLI	Jump Logical Product Inequality	If (A) ≠ [L (M) (U)] go to V
JLL	Jump Logical Product Less Than	If (A) > [L (M) (U)] go to V
JZQ	Jump Zero Quantity	If (U) = 0 go to V
JPQ	Jump Positive Quantity	If (U) ≥ 0 go to V
JNQ	Jump Negative Quantity	If (U) < 0 go to V
JTB	Jump Test Bit	If $(U)_{47}$ = 1 go to V
JTN	Jump Test No Bit	If $(U)_{47}$ = 0 go to V

In the above two jumps, in either case shift (U) left circularly one place before jumping or taking next instruction.

JIE	Jump Index Equality	If (u) = (B), go to v
JII	Jump Index Inequality	If (u) ≠ (B), go to v

No indexing on these two instructions. The designator L is used to specify the following:

L = 1 $(u)_u$: Iu

L = 2 $(u)_v$: Iv

L = 5 $(u)_u$: Iu and then Δu + Iu

L = 6 $(u)_v$: Iv and then Δv + Iv

L = 0, 3, 4, 7 not used

In all the above instructions if the jump conditions are not satisfied i.e., the V address is <u>not</u> taken as the next instruction, then the next instruction in the sequence will be taken.

JRS	Return Jump	Y + 1→U_v go to V;

That is, address of current instruction plus one, is sent to the V portion of the address specified by U, and then V is taken as the next instruction.

MJS Manual Jump-Stop

An addressable 15-bit register is provided for selective jump or stop commands. The left-most (16th) bit of the U portion of this instruction specifies whether the condition is to apply to the jump or the stop with a zero or a one respectively. If any of the other 15 bits in the U portion are non-zero it is implied that this is a conditional instruction. If the condition is satisfied, i.e., if there is a one in the lower 15-bit positions of the U portion corresponding to a one in the register, a jump or a stop according to the sixteenth bit will be carried out. There are four cases:

a. $U_{16} = 0$, $U_i = 0$ for all $i = 1, 2, \ldots, 15$. This means unconditional jump and no stops.

b. $U_{16} = 0$, $U_i = 1$ for some i (for $1 \leq i \leq 15$). This means conditional jump and no stops.

c. $U_{16} = 1$, $U_i = 0$ for all $i = 1,2,\ldots, 15$. This means conditional stop.

d. $U_{16} = 1$, $U_i = 1$ for some $i(1 \leq i \leq 15)$. This means conditional stop.

In the cases c and d an unconditional jump occurs whether or not the stop takes place.

For each of the 15 bits in the register there is a corresponding three-position switch on the console. This switch can be in the normal, manual or off position. If the switch is set to manual, then a one is set in that position of the register. If the switch is set to off, a zero is set in that position of the register. If the switch is set to normal, then the register bit is under control of the computer program. The programmer can allocate the 15 positions for jumps and stops.

Floating Point Instructions

FAT	Floating Add Transmit	$(A_L) + (U) \longrightarrow V$ and A_L
FST	Floating Subtract Transmit	$(A_L) - (U) \longrightarrow V$ and A_L
FCA	Floating Add	$(U) + (V) \longrightarrow A_L$
FRA	Floating Replace Add	$(U) + (V) \longrightarrow V$ and A_L
FCS	Floating Subtract	$-(U) + (V) \longrightarrow A_L$

FRS	Floating Replace Subtract	$-(U) + (V) \longrightarrow V$ and A_L
FML	Floating Multiply	$(U) \cdot (V) \longrightarrow A_L$
FMA	Floating Hold Multiply Add	$(A_L) + [(U) \cdot (V)] \longrightarrow A_L$
FMS	Floating Hold Multiply Subtract	$(A_L) - [(U) \cdot (V)] \longrightarrow A_L$
FPA	Floating Polynomial Add	$[(A_L) \cdot (U)] + (V) \longrightarrow A_L$
FDT	Floating Divide Transmit	$(A_L) \div (U) \longrightarrow V$ and M
FDV	Floating Divide	$(U) \div (V) \longrightarrow V$ and M
STF	Fixed To Floating	Given a fixed point number N as (V) and the scaling S as (U). Convert to floating point number as (V).
FTS	Floating To Fixed	Given a floating point number N as (V). Convert to fixed point number with number as (V) and the scaling S as (U).

Special Instructions

EFU External Function

This instruction transfers an input-output function word from address U in main memory to the peripheral control unit connected to the input-output channel designated by B, except when the selected channel is active; then a jump is made to address V.

MPS Multiply Step

The primary purpose of this instruction is to speed up decimal to binary conversion (any radix to binary). The number to be converted is assumed to be in A_R and the instruction is ordinarily used in a repeat mode. For each execution of this instruction, when $(A_R)_{47} = 1$, (U) is added to A_L; when $(A_R)_{47} = 0$, (A) remains unchanged; then, in either case (A) is given a long open right shift at one place. The U address modified by the contents of the index register scans a table of appropriate constants.

DVS Divide Step

The purpose of this instruction is to facilitate binary to decimal conversion routines (binary to any radix). The number to be converted is present in A_L. The repeat mode is used and the result is formed in A_R. For each execution of this instruction, (U) is compared with (A_L); if $(U) \leq (A_L)$, $(A_L) - (U)$ is formed and $(A_R)_{47}$ is set to one; if $(U) > (A_L)$,

(A) remains unchanged; then, in either case, (A) is given a long open left shift of one place. Here, as in the multiply step, the U address, modified, scans a table of appropriate constants.

IFU Internal Function

The format of this instruction is not defined at this time. It is used in setting and clearing certain non-addressable elements. Specifically, it is used to turn the real time clock oscillator on or off and to sense the overflow bits of A_R for recovery purposes.

APPENDIX II

GLOSSARY OF SYMBOLS AND TERMS

( ) - Indicates "the contents of" the register when address is contained within the parentheses.

V,U - Effective address or field portion.

v,u - Written address before modification by index registers.

$( )_u$ - the "u" field of 16 bits of the contents of the indicated address.

$( )_v$ - the "v" field portion.

$( )_l$ - the left hand 24 bits of the contents of the indicated address.

$( )_r$ - the right hand field portion.

$\rightarrow$ - "transmit to".

L( )( ) - the logical product (or logical AND, or intersection). It follows the table.

	0	1
0	0	0
1	0	1

( )$\oplus$( ) - the logical sum (or union), also the inclusive OR.

	0	1
0	0	1
1	1	1

( )$\bar{\oplus}$( ) - the logical difference, exclusive OR, controlled complement, add without carry, etc.

	0	1
0	0	1
1	1	0

( )$\ominus$( ) - Complements those bits in U where there are zeros in V. This is the equivalence operation or NOT exclusive OR

	0	1
0	1	0
1	0	1

Long shift - Shift both A_L and A_R together as one 96-bit register.

Circular shift - Causes the bits which are shifted out of one end of a register to re-enter at the other end of that register.

Open end shift - Causes the bits which are shifted out of one end of a register to be lost.

( ) $_{6-0}$ - Digits as subscripts indicate the range of bit positions of concern.

A_L - A-left accumulator

A_R - A-right accumulator

A - Total accumulator (96 bits)

B - Index register, or channel designator

PAK - Program address counter

APPENDIX III

INSTRUCTION TIMES

The execution times cited for the instructions below include access for the instruction and the operands. For a given instruction the time for execution depends upon whether the instruction is procured from main memory or control memory or whether it is performed in the repeat mode. Similarly, time is dependent upon whether the operands are procured from main memory, control memory, or the addressable registers. On some of the instructions the execution time is dependent on the amount of indexing specified; i.e., whether none, one, two, or four index additions are performed. Execution time is also dependent upon whether references are made to alternate banks of main mamory. Use of alternate memory banks saves up to one complete memory cycle time.

TPE Transmit Positive Entire (U) $\longrightarrow$ V

4.5 microseconds, main memory, indexed

3.0 microseconds, main memory, repeated

1.0 microsecond, control memory and registers

The maximum time assumes instruction and both operands are in the same main memory bank; maximum indexing is included. With the instruction and U address in one bank and the V address in the other bank, the time becomes 3.0 microseconds. The minimum time assumes the instruction is in the control memory and the operands are addressable registers. A repeated transfer takes 3.0 microseconds between main memory locations of one bank; 2.2 microseconds, between

main memory and auxiliary memory; 1.5 microseconds, between overlapped main memory banks. For the instructions below the main memory references are in the same bank. Overlapped operations occur with a shorter time.

TIC Transmit Interchange $(U) \rightarrow V$, $(V) \rightarrow U$

7.5 microseconds, main memory, indexed

6.0 microseconds, main memory, repeated

1.4 microseconds, control memory and registers

APT Add Transmit $(A_L) + (U) \rightarrow V$ and A_L

4.5 microseconds, main memory, indexed

3.0 microseconds, main memory, repeated

1.0 microsecond, control memory and registers

RAT Replace Add Transmit $(U) + (V) \rightarrow V$ and A_L

6.0 microseconds, main memory, indexed

4.5 microseconds, main memory, repeated

1.8 microseconds, control memory and registers

HAA Hold Add Add $(A_L) + (U) + (V) \rightarrow A_L$

4.5 microseconds, main memory, indexed

3.0 microseconds, main memory, repeated

2.0 microseconds, control memory and registers

MLP Multiply $(U) \cdot (V) \rightarrow A$

5.8 microseconds, main memory, indexed

4.0 microseconds, control memory and registers

HPA Hold Polynomial Add $\left[(A_L) \cdot (U) \right] + (V) \rightarrow A$

6.7 microseconds, main memory, indexed

4.8 microseconds, main memory, repeated

4.8 microseconds, control memory and registers

DIV Divide $(U) \div (V) \rightarrow M$, remainder in A_L

12.2 to 31.0 (average 22.2) microseconds, main memory, indexed

10.8 to 29.2 (average 20.4) microseconds, control memory and registers

DPT Divide Transmit $(A) \div (U) \rightarrow V$ and M, remainder in A_L

13.6 to 32.0 (average 23.2) microseconds, main memory, indexed

12.1 to 30.5 (average 21.7) microseconds, main memory, repeated

11.4 to 29.8 (average 21.0) microseconds, control memory and registers

FAT Floating Add Transmit $(A_L) + (U) \rightarrow V$ and A_L

5.2 to 7.0 microseconds, main memory, indexed

3.7 to 5.5 microseconds, main memory, repeated

2.4 to 4.2 microseconds, control memory and registers

FML Floating Multiply $(U) \cdot (V) \rightarrow A_L$

5.8 microseconds, main memory, indexed

4.0 microseconds, control memory and registers

FPA Floating Polynomial Multiply Add $\left[(A_L) \cdot (U) \right] + (V) \rightarrow A_L$

6.4 to 8.2 microseconds, main memory, indexed

6.2 to 8.0 microseconds, main memory, repeated

6.0 to 7.8 microseconds, control memory and registers

FDV Floating Divide $(U) \div (V) \longrightarrow M$

10.8 to 24.8 microseconds, main memory, indexed

9.0 to 23.0 microseconds, control memory and registers

FDT Floating Divide Transmit $(A_L) \div (U) \longrightarrow V$ and M

11.8 to 25.8 microseconds, main memory, indexed

10.3 to 24.3 microseconds, main memory, repeated

9.6 to 23.6 microseconds, control memory and registers

APPENDIX IV

APPLICATION NOTES

1. EQUIPMENT CONFIGURATION

A typical system configuration may consist of:

Central Computer

32,768 Words of Magnetic Core Memory

16 Magnetic Tape Units

10 Magnetic Drum Units

High-Speed Printer

Supervisory Control Console with Paper Tape Units

Six Peripheral Control Units

In such a configuration, the Central Computer and memory occupies less than 50 square feet and requires less than 15KW of power. The peripheral equipment and supervisory control occupies about 300 square feet of a 1500 square foot area designed for efficient system operation, and requires approximately 60KW of power.

a. CABINET AND FLOOR LAYOUT. - The computer is comprised of modular cabinets, each approximately 25 inches wide, 36 inches deep, and 72 inches high. Five cabinets are required to house the Central Computer, one for the 32,000 word magnetic core memory and about one-half cabinet for each peripheral control unit. Power supplies and cooling fans are an integral part of each cabinet, so that additions to the system can be easily accomplished.

Intercabinet connections in the system are made through a wiring duct in the rear of the cabinet along the base. In a proposed floor layout, all of the

main cabinets of the system abut each other in a single row to minimize cable lengths. Peripheral equipment such as magnetic drums, magnetic tape units, disc files, and card equipment may be arranged in rows parallel or perpendicular to the main cabinets.

b. POWER EQUIPMENT. - The primary voltage source is 208-volt, 60-cycle, 3-phase. A motor-alternator isolates the line voltage transients from the system and provides a 208-volt, 400-cycle voltage permitting use of compact, efficient direct-current power supplies. An alternator capacity of approximately 15KVA is required to supply primary power for the direct-current power supply system which includes the demand of the 32,768 words of core memory, two tape control units for 12 tape handlers, two magnetic drum control units for ten drum units and a printer control unit. Additional power of about 65KVA is required in the system to supply the 60-cycle requirement for magnetic tape, drum and printer unit motors, pumps, cooling fans, and miscellaneous items in the system.

c. COOLING REQUIREMENTS. - Each cabinet in the system contains air intakes, filters, and cooling fans. Controlled room air is forced through the cabinets cooling all components. Temperature control is provided by the customer to maintain a room ambient temperature in the range of 65 to 80 degrees fahrenheit, and maximum relative humidity of 60 per cent. The heat load from the system described above is approximately 60KW.

At the option of the customer, a closed loop air conditioning system can be supplied on the cabinets at installations which have extraordinary environmental conditions.

2. UTILIZATION OF PERIPHERAL EQUIPMENT

The maximum input-output transfer rate is limited by the cycle time of the main memory. With a cycle time of 1.5 microseconds, the transfer rate of a memory bank is 666.6 thousand words per second. This word rate is the maximum instantaneous input-output transfer rate for a memory bank. Two memory banks provide a maximum instantaneous word rate of 1.333 thousand words per second. However, in a practical system when memory cycles are required for other purposes, such as to obtain instructions, information for arithmetic and logic functions, and input-output control information, this transfer rate cannot be fully realized. In a specific computer system, the maximum concurrent peripheral input-output transfer rate must be considered so that it does not overload the system.

A representative example of a configuration of peripheral equipment operating concurrently is given below:

2 magnetic drum channels at 125,000 words/sec.	-	250,000 words/sec.
1 magnetic disc channel at 333,333 char/sec.	-	41,666 words/sec.
4 magnetic tape channels at 100,000 char/sec.	-	50,000 words/sec.
2 high-speed printer channels at 600 lines/min.	-	3,200 words/sec.
1 magnetic tape channel at 25,000 char/sec.	-	3,125 words/sec.
Total Transfer Rate		347,991 words/sec.

The computer can easily handle this transfer rate since it is less than one-half the maximum transfer rate of a single memory bank. When the input-output transfer rate is greater than one-half the maximum rate of a memory bank, special considerations must be given to factors such as the amount of internal data processing required, the data and control word transfers which are peculiar

to each peripheral unit, and the over-all duty cycle of such a combination of transfers. The transfer data rate given in the above example would not be sustained for more than a few thousand words unless the operation being performed was essentially buffering an external unit to external unit transfer with the input and output rates approximately in balance.

The above word rate for the printers is the instantaneous rate to fill the 120-character buffers. The average word rate for two printers is 300 words per second. Similarly, transfers with the drums, tapes, and discs are usually performed as finite block transfers which also make the average word rate significantly lower.

3. PROGRAM COMPILER

Integral with the computer is an automatic coding system which includes a program compiler and an associated library of data processing and algebraic statement generators. The compiler framework comprises the following subprograms:

(1) INPUT CONVERTER - The portion of the compiler routine which translates the mnemonic, computer-oriented input language (L_0) into the first of the intermediate, compiler language levels (L_1) by (1) converting input code to compiler code, (2) performing certain error detection steps, (3) making some minor initial translations, and (4) storing the language in table format in computer memory.

(2) GENERATOR EXECUTIVE - The compiler subroutine which translates, selects, and calls in specific generators, each of which produces a set of machine instructions.

(3) ALLOCATION - The process of assigning numerical values to symbolic labels and/or tags, each of which may represent a computer address or a constant.

(4) ASSEMBLER - The portion of the compiler routine which replaces the symbolic addresses (labels and tags) of the machine instructions with numeric addresses in the order required for program execution.

(5) OUTPUT CONVERTER - The portion of the compiler routine which converts compiler and octal codes to output codes and directs the planned conversion to printed copy and/or reloadable media.

(6) EDITOR - The compiler subroutine which selects and controls the output of informational material for the programmer's inspection.

(7) LIBRARIAN - The compiler subroutine mechanism which controls the storage and referencing of subroutines from a storage unit.

(8) COMPILER SUPPORT - The collection of auxiliary routines which perform such assistant functions to compiling as loading and dumping programs (Utility Package), generating debugging aids output, and accessing, updating, and manipulating compiler tables (Table Control System).

(9) ONE-ONE TRANSLATOR - The compiler subroutine which converts a mnemonic mono-operation into a machine instruction.

Other translators may be employed to provide compatibility between the above compiler and problem oriented languages such as the International Algebraic Language (ALGOL), the Common Business Oriented Language (COBOL), and others.

Vigorous customer support is provided with continuing services and programs which are kept updated. These include:

Program Diagnostic, Supervisor Routines

Subroutine Library

Maintenance Diagnostic Routines

Application Support

Users Manuals

Program Maintenance

Customer Assistance

4. PROGRAM APPLICATIONS

Many instructions given in the instruction repertoire (Appendix I) are effectively built-in subroutines, since each such instruction can:

1) Utilize two working addresses,
2) Designate repeated executions,
3) Specify indexing of either or both addresses, and
4) Specify incrementation (decrementation) of either or both address modifiers.

These instructions used as the key portions of programmed subroutines provide a very significant saving of execution time.

The following sample problems illustrate the use of some of the key instructions found in the repertoire.

a. DECIMAL TO BINARY CONVERSION

Assumptions:

1. A set of characters in A_R.

2. A set of bit weights stored in the main memory, i.e., each bit of each character has a specified weight value.
3. The starting location in the table is known, i.e., the scaling is established.

The conversion amounts to repeating the Multiply Step instruction the required number of times. A memory reference and an addition is performed only where a one is encountered in the coded number. This method will convert codes using any set of arbitrary bit weights, e.g., Gray codes. This conversion time for 4-bit binary coded decimal is on the average 2.25 microseconds per decimal digit. For a 12-digit decimal word 27 microseconds is required.

b. BINARY TO DECIMAL CONVERSION

Assumptions:

1. A binary number in A_L
2. A set of bit weights stored in main memory.
3. The scaling of this number is known; hence the starting location in the table is known.

The conversion proceeds with the repeated Divide Step instruction. A memory reference is made for each bit position of the binary number and a trial subtraction takes place. The conversion time per decimal digit is 1.5 microseconds per bit of the converted number, or 6 microseconds for each 4-bit binary coded decimal digit. A different table for other codes or other radix numbers can also be used.

c. FUNCTION EVALUATION. - For approximations using polynomial expansions, the polynomial multiply instruction with a repeated floating point time of 6.2 to 8 microseconds enables evaluation of a cosine function (using a 6-term Chebychev expansion) to be made in 52 microseconds, maximum.

d. MATRIX TRANSPOSITION. - Often in working in very large matrix operations the transposition is needed but additional working space is not available; hence, it must be performed "upon itself". The repeated interchange instruction (TIC) enables this to be accomplished in a very tight programmed loop of several instructions. The time for a 100 by 100 matrix is 30 milliseconds.

e. MATRIX MULTIPLY. - In the multiplication of two matrices many sums of products must be formed. The repeated inner product instruction (Multiply Add) can form each element of the new matrix each time the subroutine calls for the repeated inner product. The time for the product of two 100 by 100 element matrices of floating point numbers is 6 seconds.

f. LINEAR EQUATIONS. - Solution of a set of linear equations may utilize the Crout (modified Gauss Elimination) method*. This method uses sums of products, also (specifically the Multiply Subtract). The space requirements of this method are $n(n + 2)$ memory addresses where n is the number of unknowns. Assuming a 100 by 100 component system and floating point numbers, the time of solution is 3.4 seconds.

g. MATRIX INVERSION. - The method used here is an extension of the Crout method and again the repeated inner product is used. The space requirement is $2n^2$ memory addresses where n is the number of elements in a row or column of the matrix to be inverted.

Again assuming a 100 by 100 matrix and floating point entries, the time for inversion is 7 seconds.

* See Hildebrand, F. B. "Methods of Applied Mathematics", pp. 503 to 507.