SYSTEM/4 Pi

MODEL TC

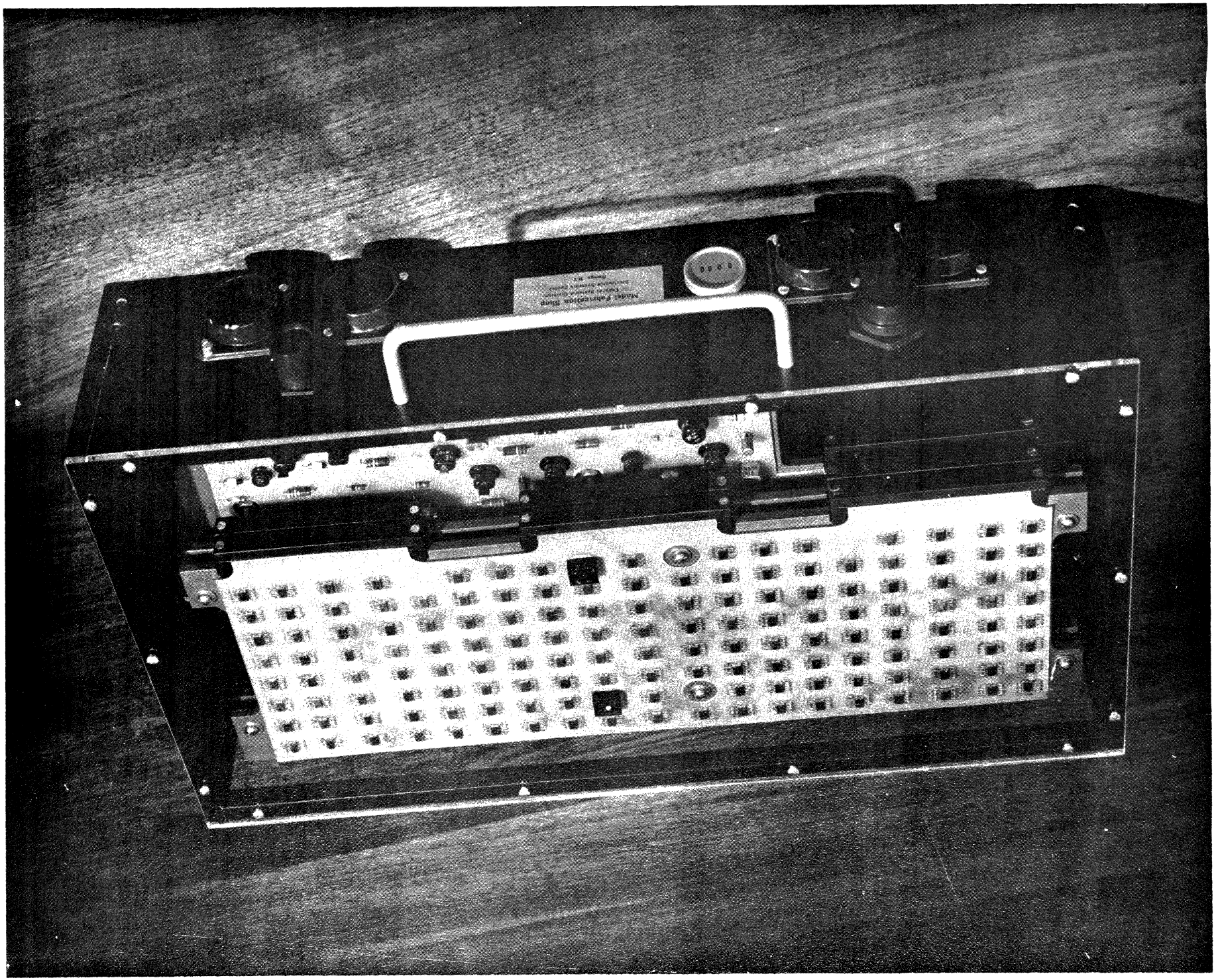


TABLE OF CONTENTS

FOREWORD

The IBM Federal Systems Division has developed a family of militarized, general purpose, digital computers called System/4 Pi. The name 4 Pi is derived from the number of steradians in a sphere, symbolizing the capability of these computers to address the entire spectrum of military applications.

The models in the System/4 Pi family currently under contract include:

TC - Tactical Computer
CP - Cost Performance
CP-2 - Cost Performance - Model 2
EP - Extended Performance

These models satisfy the low-cost, limited processing requirements and expand upward to meet the requirements of the very largest data processing applications. Deliveries on all models begin in early 1967.

MODEL TC

SUMMARY

The Model TC is a general purpose, stored program digital computer. It is designed for the low cost, medium range of computational performance. The hardware requirements are minimized by utilizing a byte (8 bits) data transfer format and efficient utilization of hardware registers to perform multifunctions.

The computer is subdivided into a memory, processor, and I/O section. All signals entering or leaving the computer are digital in nature. In the configuration presently being fabricated, the power is supplied from an external source.

The 2.5 microsecond, 8192 byte core memory has a parallel 8 bit transfer path to the processor section and operates within a basic 3 usec machine cycle. The computer has the capability to address 65K bytes of main store.

The processor is organized as a bus system design with an 8 bit fully parallel data flow. The processor contains the facilities for addressing main storage, for fetching or storing information, for arithmetic and logical processing of data, instruction sequencing, timing, and for initiating the communication between storage and external equipment.

Working registers are located both in hardware and memory. A total of 54 instructions are implemented. Processing rates on the order of 48,500 operations/second are attained by optimizing the machine organization, monolithic logic circuitry, and the high speed core memory.

A summary of Model TC characteristics is contained in Table 1.

DATA AND INSTRUCTION FORMATS

The TC computer transmits information between main storage and the processor in an 8 bit unit of information called a byte, the basic building block of all formats.

The basic arithmetic operand is the 16 bit fixed point binary word (sign + 15 bits) and is formed by two successive 8 bit memory words. There is no parity check of the memory byte. A four byte operand is used for double precision as well as products and dividends. Two byte operands must be located in main storage on

SYSTEM/4 Pi MODEL TC CHARACTERISTICS

TYPE	General Purpose, Digital
ORGANIZATION	Byte parallel, fixed point, fractional, two's complement notation
INSTRUCTIONS	54
INSTRUCTION WORD	1, 2, or 3 bytes
DATA WORD	16 or 32 bits
COMPUTER CYCLE TIME	3 usec
STORAGE CYCLE	2.5 usec

TYPICAL EXECTUION TIMES

	Short Format	Long Format
ADD	15 usec	18 usec (12 usec if BR=0)
MULTIPLY	51 usec	54 usec (48 usec if BR=0)
DIVIDE	N/A	54 usec (48 usec if BR=0)
OPERATIONS/SEC	48,500 typical	

PHYSICAL

SIZE	9.75" x 17.12" x 4" (.37 cu. ft.)
WEIGHT	17.3 lbs.
MAIN STORAGE	8192 bytes (8 bits) expandable to 65K bytes
TECHNOLOGY	TTL Integrated Circuits
POWER	60 Watts
ENVIRONMENT	MIL-E-5400 Class 2
MTBF	6,300 hours predicted.

even byte boundaries. Four byte operands must be positioned so that the two least significant bits of the binary operand address are zero. Figure 1 shows the basic word formats used in the TC Computer.

Three instruction word formats are used in the computer to provide ease of programming and to achieve good bit efficiency. These instruction formats are compatible with the 8 bit memory organization, being 8, 16, or 24 bits in length. All instructions of 16 or 24 bits in length cause a reference to main store. Some of the 8 bit instructions (e. g., SKIP ON CARRY, ZERO Q) make no reference to main store.

The short instruction format is utilized for the most common instructions, thereby minimizing the program memory size while the longer (16 bit) instructions provide greater flexibility and larger operand addressing. The 24 bit short format-immediate provides 16 bits of immediate data along with an 8 bit operation code and displacement field.

In each format, the first bit is a flag bit denoting a short or long instruction format. The short format also contains a three bit instruction code, and a four bit field which is used as either a displacement for operand addressing or to specify the number of shifts. Short format operand addressing is to a displacement of 16 syllables with respect to soft base Register 1 (operand fetching only).

The long format instruction consists of a flag bit to designate the long format, a five bit instruction code, a two bit base register field, and an 8 bit displacement field. This instruction requires two memory bytes. Operand addressing is to a displacement of 256 syllables with respect to one of three base registers. The displacement is a direct address to the first 256 syllables of memory (for most instructions) when the base register field equals zero.

ADDRESSING

Byte locations in storage are consecutively numbered starting with 0; each number is considered the address of the corresponding byte. A group of bytes in storage is addressed by the left most byte of the group. The addressing capability permits a maximum of 65, 546 bytes, using a 16 bit binary address. The TC computer memory addressing hardware permits the addressing of 16, 384 bytes, although 8, 192 bytes are presently implemented. This set of main storage addresses includes some locations reserved for special purposes.

For addressing purposes, operands can be grouped in two classes: explicitly addressed operands in main storage and immediate operands placed as part of the instruction stream in main storage.

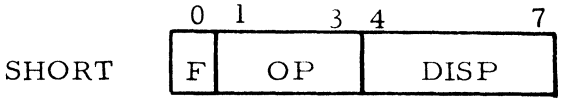
To permit the ready relocation of program segments and to provide for the flexible specifications of input, output, and working areas, all instructions referring to main storage have been given the capacity of employing a full address. The address used to refer to main storage is generated as follows:

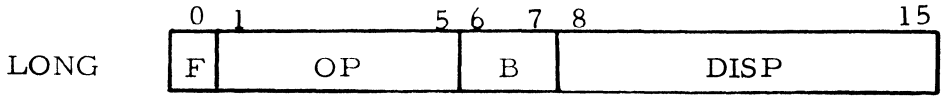
FIGURE 1

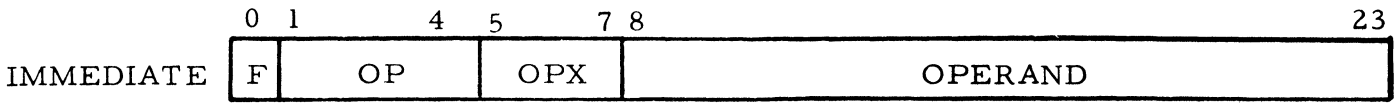
WORD FORMATS

DATA

INSTRUCTIONS







Base Address (B) is a 16 bit number contained in a soft general register specified by the program in the B field of the instruction. The base address can be used as a means of static relocation of programs and data. The base address provides for addressing the entire main storage. A total of three base registers are located in main store and contain base addresses used in effective address generation. A special base register designated linkage register (L) is also used but only for branch type instructions.

Displacement (D) is a number contained in the instruction formats. It is included in every address computation. The displacement provides for relative addressing up to 256 syllables for long format instructions and 16 syllables for short format instructions.

In forming the address, the base address is treated as unsigned 16 bit positive binary integers. The displacement is similarly treated as a positive binary integer. The two are added as 16 bit binary numbers ignoring overflow after the displacement syllable address is converted into an effective byte address. Since every address includes a base, the sum is always 16 bits long although only 14 bits are implemented in the computer. The resulting address is the effective address of the left most operand byte.

PROCESSOR ORGANIZATION

The data flow in the processor section is via one of three data buses: the X Bus, the Y Bus, and the Main Bus. The X bus gates one of eight registers to the left side of the adder. The Y bus gates one of three registers to the right side of the adder. Output data from the adder is distributed to any one of seven registers via the main bus. Data inputs and outputs to and from the I/O are gated onto the main bus.

The 8 bit adder features the use of binary full adders packaged one bit per flatpack, thereby improving performance and cost packaging effectiveness. The adder executes all arithmetic and effective address update functions. In addition to the add function, the adder also has the capability to perform AND, OR, and exclusive OR functions.

The hardware registers of the processor section are also packaged very efficiently with two register bits contained in a flatpack without the need for external gating hardware.

The following hardware registers are included in the design:

1. Accumulator (A) - The 16 bit accumulator is the principal arithmetic register.

2. Multiplier-Quotient Register (Q) - The Q register is a 16 bit low order extension of the accumulator.

3. Memory Buffer Register (MBR) - The MBR is an 8 bit register that receives the output from the memory.

4. Displacement Register (D) - The D register is an 8 bit register that stores the address displacement. The TC computer possesses a displacement field of up to 256 syllables for long format instructions and up to 16 syllables for short format instructions. It is also used as a temporary storage register on various arithmetic operations.

5. Instruction Register (I) - The instruction register is a 14 bit register that holds the address of the next sequential instruction.

6. Operation Register (O) - The operation register is an 8 bit register used to store the instruction code.

7. Memory Address Register (MAR) - The Memory Address Register is a 14 bit register used to store and select the proper memory address. The two least significant bits are configured as a counter.

8. Shift and Multiply Counter (S/M) - The S/M counter is used to count iteration during shift, multiply, and divide instruction. The five bit counter allows for a maximum of 32 shifts in either direction.

9. Status Register (SR) - The status register contains the wait state bit, the carry indicator, and the two interrupt masks.

The interruption system used in the TC computer permits a change in state of the computer in response to conditions external to the system. Two interrupts have been designed:

1. Interrupt #0 - Major priority for such interrupts as power off interrupt.

2. Interrupt #1 - Non-critical priority interrupts that can be interrogated by program control. An example of this is a real time clock interrupt.

An interrupt consists of storing the updated Instruction Counter (IC) and executing an automatic indirect branch. The IC is saved in a location permanently assigned to the interrupt.

MAIN STORAGE

The TC computer utilizes an 8,192 word by 8 bit destructive readout core memory. The memory has a 2.5 microsecond read-write cycle time. Each cycle consists of a read followed by a write operation. The computer initiates a start memory every three microseconds. The access time of the memory is approximately .9 usec. A coincident current (3-D) selection scheme is utilized. The memory array requires four memory mats packaging two bits per mat. Operation is possible over the ambient temperature range of -55°C to +100°C.

The memory uses external data buffer (memory buffer) and memory address lines (14) sent from the processor section. The system is organized such that the "X" half select driver is clocked first since it drives the long dimension. The "Y" current thus has time to reach full amplitude at a slower rate, thereby reducing driver supply voltage requirements.

Timing is generated internally to the memory section. Address decoding is carried out by logic circuits internal to the memory section.

The monolithic circuits used in the memory are designed to "state-of-the-art" groundrules obtained from integrated circuit manufacturers. The high voltage, high current address driver and inhibit driver consist of monolithic pre-driver transformers coupled to discrete chip output stages packaged in flatpacks. The sense amplifier is a monolithic circuit with additional resistors added for threshold setting.

The storage element is an IBM 0.0137" by 0.021" by 0.0045" (nominally 13/21) lithium nickel ferrite toroidal core that is designed to operate over a wide temperature range.

INPUT/OUTPUT

The Input/Output section of the TC computer performs the following functions:

1. Data Formatting - The I/O section provides for a byte to serial data conversion to allow the processor to interface with other systems. The 8 bit main data bus is converted to a 1 mc serial readout of one syllable burst duration.

2. Real Time Clock Generation - A real time clock is located in the I/O section and is used to generate timed interrupts. The contents of the counter are periodically interrogated by the processor. Two low frequency clocks are also generated by the real time clock.

3. Device Address Generation - The external device addresses are decoded in the I/O section.

4. Interrupt Control - The buffer and control logic associated with interrupts is a part of the I/O section.

COMPUTER PACKAGING

The TC computer is packaged in a ruggedized aluminum structure that contains three major subassemblies: Processor, Memory, and Input/Output and Wiring Harness. The structure has dimensions 9.75" x 17.12" x 4.0" and occupies .37 cubic feet. The heat generated from the subassemblies is removed by an air cooled heat exchanger located in the bottom of the computer structure.

Electrical interfacing is accomplished with four input/output connectors (118 pins/connector) located on the front panel. AC and DC power is applied through separate connectors. Lightweight magnesium covers at the top and back of the unit facilitates removal of the subassemblies, provides radio frequency attenuation, and prevents entrance of foreign materials. The subassemblies are mounted at the bottom of the structure.

The computer is designed to meet the requirements of the MIL-E-5400 Class 2 specification. The three subassemblies are all of the same basic construction differing only in the type of electronic components that are mounted.

Each subassembly (page) normally consists of two rectangular shaped large SLT boards, a thermal mounting plate, the I/O connectors, and the circuit components.

The processor page contains all the logic circuitry for the computer as defined in previous sections. It consists of two SLT-type printed circuit boards on which transistor-transistor logic (TTL) integrated circuit flatpacks are soldered. These PC boards are sandwiched about a metal supporting structure which contains three 98 pin Saturn type connectors. An insulator separates each board from the supporting structure. Feedthroughs are provided to allow electrical connection between the two boards of the page. The SLT board contains two levels of signal wires and two internal ground-voltage planes.

The I/O page is identical in design to the CPU page except that due to less electronics only one SLT board and two Saturn connectors are provided.

The memory subassembly contains an 8,192 x 8 bit memory array and the necessary drive and sense circuitry packaged on two SLT boards with one Saturn connector for I/O communication.

The memory array consists of the following subdivisions:

- 4 memory planes
- 2 diode printed circuit boards
- 3 resistor termination boards
- 1 X line jumper board

The array subassembly is disconnectable from the memory page, thereby allowing flexibility in production and test.

SOFTWARE SUPPORT

The support programming system provided with the Model TC will assist the problem programmer in the production of operational software for the computer by providing:

- Programming aids that will enable a program to be expressed in a language which is readily understood
- Facilities for storing, modifying, and accessing precoded programs
- A program checkout system to ensure operational program readiness
- Effective documentation manuals.

The assembler program and language will allow a programmer to make efficient use of all the features of the proposed computer while relieving him of the details of machine coding. The assembler will translate symbolic instructions into machine language instructions, assign storage locations, and perform auxiliary functions necessary to produce an executable machine language program.

Service programs will assist the programmer by providing routines for performing frequently used functions, including system tape generation and source program edit facilities. The service programs consist of a source program editor and a set of utility programs. The source program editor is designed to operate with the assembler and is, in fact, called by the assembler. The function of the editor is to update and alter existing source programs at the source statement level. The utility programs consist of the following:

- System job control program
- System tape maintenance programs
- Preparation program.

The system job control program provides for standard transition of control from one support program to another; for example, from assembler to dynamic program checkout system. This control is accomplished through user-supplied control cards specifying the processing programs desired.

The dynamic program checkout system (DPCS) will provide dynamic program analysis of the computer at the instruction word level of operation.

The DPCS will execute any given set of computer instructions as they appear on the symbolic output listing of the assembler. DPCS control will be exercised by user-provided programs written in a high level computer language (e.g. FORTRAN IV). These programs will call simulator subroutines which will manage the operation of the simulator. To aid in program debugging and checkout, the DPCS will provide a number of outputs concerning data flow, internal operation, and results of computations. These outputs will be generated by various trace and dump routines within the simulator.

The DPCS will provide an accurate simulation of the functional operation of the computer and associated input/output devices. Every mode of computer operation subject to program control will be exercised with DPCS, just as on the actual equipment. The DPCS will operate with the same arithmetic constraints as the actual computer; likewise, the DPCS will have the same program restrictions as the actual computer. The DPCS will aid in providing a near real-like situation to the simulated object program, and will greatly assist the problem programmers in debugging the operational programs.

A self-test routine to detect and isolate Model TC computer malfunctions is designed to be co-resident with the operational program. At periodic intervals the operational program transfers control to the self-test routine to begin a functional go/no-go indication. The routine is intended to exercise main working registers, data flow and controls, and instruction control logic. Failure of any active computer circuitry, as well as use of invalid data, will cause an error indication. The storage requirements of the self-test routine for Model TC are variable, depending on the degree of testing desired, and the interference allowable with the operational program.

INSTRUCTION LIST

A complete list of Model TC instructions and execution times is contained in Table 2.

TABLE 2

INSTRUCTION LIST

OPERATION	FORMAT	EXECUTION TIME
ADD	Short	15
ADD	Long	18*
ADD	Immediate	9
ADD DOUBLE	Long	24*
AND	Immediate	9
AND	Long	18*
ARITHMETIC COMPLEMENT	Short	3
BRANCH	Immediate	9
BRANCH	Long	12
BRANCH BACKWARD ON MINUS	Long	6
BRANCH BACKWARD ON PLUS	Long	6
BRANCH BACKWARD UNCONDITIONAL	Long	6
BRANCH BACKWARD ON ZERO	Long	6
BRANCH FORWARD ON MINUS	Long	6
BRANCH FORWARD ON PLUS	Long	6
BRANCH FORWARD UNCONDITIONAL	Long	6
BRANCH FORWARD ON ZERO	Long	6
BRANCH INDIRECT	Long	12
BRANCH INDIRECT & STORE IC	Long	18
COMPARE	Immediate	9
COMPARE	Long	18*
COMPARE DOUBLE	Long	24*
DIVIDE	Long	54*
EXCHANGE A & Q	Short	3
EXCLUSIVE OR	Long	18*
INPUT/OUTPUT CONTROL	Long	24*
LOAD A	Short	15
LOAD A	Long	18*
LOAD A	Immediate	9
LOAD BASE	Long	18
LOAD BASE 1	Immediate	15
LOAD BASE 2	Immediate	15
LOAD BASE 3	Immediate	15
LOAD DOUBLE	Long	24*
LOAD STATUS WORD	Long	12
MODIFY BASE	Long	18
MULTIPLY	Short	51
MULTIPLY	Long	54*
NO OPERATION	Short	3
OR	Long	18
SHIFT LEFT DOUBLE	Short	6-18

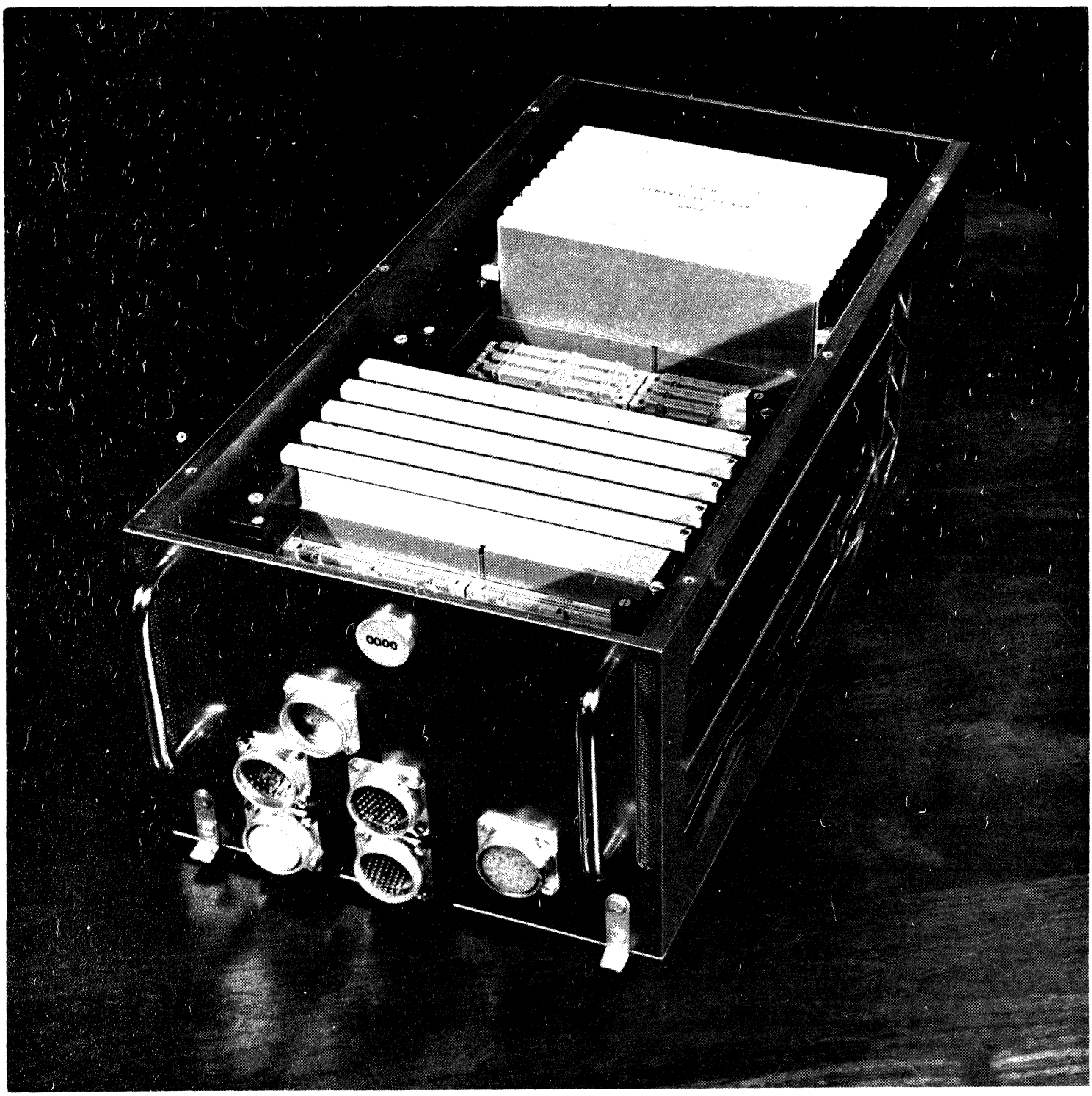
TABLE 2
(Cont.)

OPERATION	FORMAT	EXECUTION TIME
SHIFT RIGHT ARITHMETIC DOUBLE	Short	6-18
SKIP ON CARRY	Short	3
STORE A	Short	15
STORE A	Long	18*
STORE BASE	Long	18
STORE DOUBLE	Long	24*
STORE IC	Long	18
STORE STATUS WORD	Long	12
SUBTRACT	Short	15
SUBTRACT	Long	18*
SUBTRACT DOUBLE	Long	24*
TALLY	Long	12-24
ZERO Q	Short	3

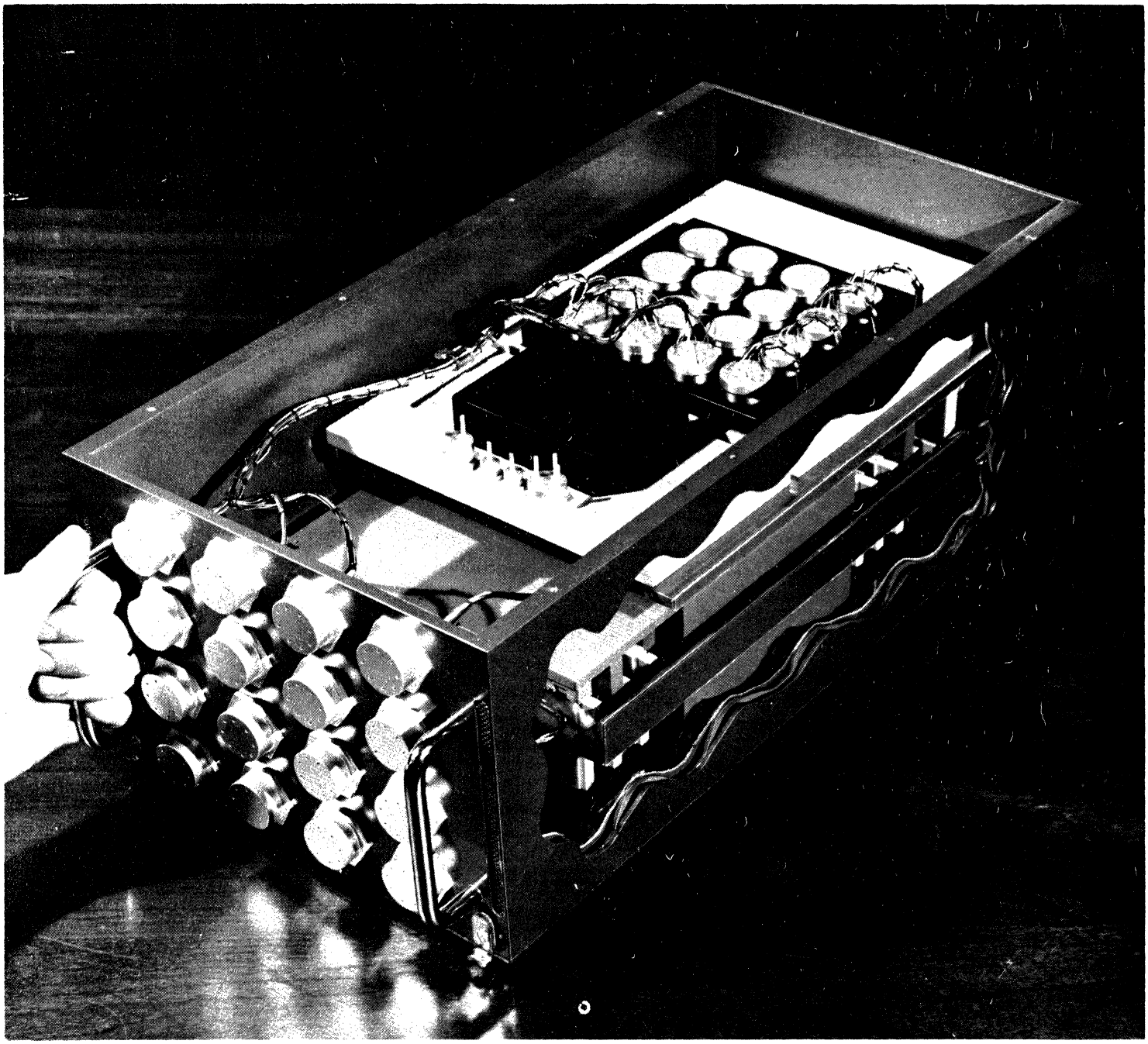
*Execution Time is 6 usec Less if B=00.

SYSTEM/4 Pi

MODEL CP



SYSTEM/4 Pi MODEL CP PROCESSOR MOCKUP



SYSTEM/4 Pi MODEL CP INTERFACE ADAPTER MOCKUP

TABLE OF CONTENTS

FOREWORD

The IBM Federal Systems Division has developed a family of militarized, general purpose, digital computers called System/4 Pi. The name 4 Pi is derived from the number of steradians in a sphere, symbolizing the capability of these computers to address the entire spectrum of military applications.

The models in the System/4 Pi family currently under contract include:

TC - Tactical Computer
CP - Cost Performance
CP-2 - Cost Performance - Model 2
EP - Extended Performance

These models satisfy the low-cost, limited processing requirements and expand upward to meet the requirements of the very largest data processing applications. Deliveries on all models begin in early 1967.

MODEL CP

SUMMARY

IBM System/4 Pi Model CP (Cost Performance) is a general purpose, single address, fixed point, digital computer. The ruggedized design and flexible logic organization provide the Model CP with the capability to operate in aircraft, aerospace, ground-based, and shipboard environments.

The Model CP is comprised of a Central Processing Unit (CPU), Main Storage (MS), Read Only Storage (ROS), Interface Adapter, and Power Supply. In the configuration currently being fabricated, the units are packaged in two separate cases.

The 2.5 microsecond main memory has a 36 bit parallel data transfer to the CPU. The data flow within the CPU is 16 bits in parallel.

The unique feature of the Model CP is the control section of the CPU. Each instruction available to the programmer is implemented as a short subroutine of "microinstructions" read from ROS. The movement of data between hardware registers and main storage, and between the registers and I/O hardware, is entirely under control of these micro-commands. The flexibility provided by this novel design feature permits adding application oriented instructions and microprogramming highly repetitive routines for increased speed and efficient performance without hardware redesign.

A summary of Model CP characteristics is contained in Table 1.

DATA AND INSTRUCTION FORMATS

The Model CP is implemented with a standard 32 bit data word. A 16 bit halfword can also be utilized when the accuracy afforded by a full word is not required (See Figure 1).

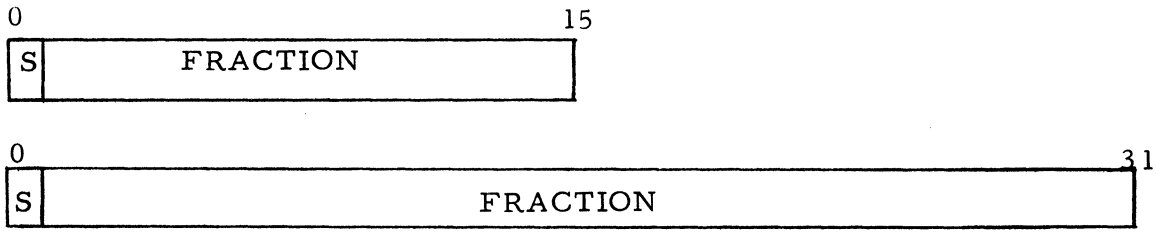


Figure 1. Halfword and Fullword Data Formats

Positive numbers are always in true binary form. Negative numbers are in two's complement form. The sign bit (position 0) is always 0 for positive numbers and 1 for negative numbers.

SYSTEM/4 Pi MODEL CP CHARACTERISTICS

TYPE	General Purpose, Digital
ORGANIZATION	Parallel, fixed point, fractional, two's complement notation
INSTRUCTIONS	36
INSTRUCTION WORD	16 or 32 bits
DATA WORD	16 or 32 bits
COMPUTER CYCLE TIME	417 nsec.
STORAGE CYCLE	2.5 usec
TYPICAL EXECUTION TIMES	
ADD	5.0 usec
MULTIPLY	29.58 usec
DIVIDE	45.42 usec
OPERATIONS/SECOND	91,000*
PHYSICAL	
SIZE	9.97" x 7.38" x 19.40" (.82 cu ft. each case)
WEIGHT	80 lbs. (Total)
MAIN STORAGE	8192 - 36 bit words expandable to 32,768 words
TECHNOLOGY	TTL Integrated Circuits
POWER	350 Watts
ENVIRONMENT	MIL-E-5400, Class 2
MTBF	2,500 hours predicted

*Reference value only, higher speeds possible with application oriented microprograms.

The instruction words in Model CP are 16 or 32 bits long (See Figure 2).

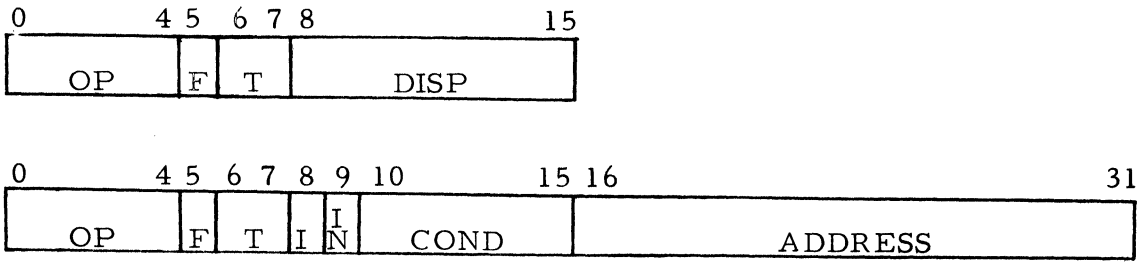


Figure 2. Instruction Formats

The bits within the instruction words are used as follows:

OP — These five bits define which operation is to be performed.

F — This format bit controls the instruction format. A "zero" indicates a halfword instruction and a "one" indicates a fullword instruction.

T — These two Index Tag bits address the Index Register or Instruction Register used in the address modification.

DISP — These eight bits are called the displacement (halfword format only) and are added to the Instruction Register or the Index Register specified by T to define the effective address (EA). If T is 00, the displacement is added to the Instruction Register. The displacement is in two's complement form if negative, the sign being in bit 8. The bit in position 8 is automatically extended to the higher ordered bits (0-7) when the displacement is added to an Index Register.

I — This is an indirect bit in the two word instruction format. If "zero", addressing will be direct. If a "one", addressing will be indirect.

IN — This bit is used to specify that the Branch or Skip on Condition (BSC) instruction is to be interpreted as a "Branch Out" of an interrupt routine.

COND — Specifies the condition indicators to be interrogated on a BSC or BSI instruction.

ADDRESS — These 16 bits contain the address in a full word instruction. It can be modified by the contents of an index register or is an indirect address.

ADDRESSING

All core storage locations are directly addressable under program control. The effective address may be determined by examining the tag, flag and indirect address bits shown in Figure 2. The effective address is formed as shown in Table 2.

TABLE 2

EFFECTIVE ADDRESS

F = 0 (Direct Addressing)	F = 1, IA = 0 (Direct Addressing)	F = 1, IA = 1 (Indirect Addressing)
T = 00 EA = I[1] ∠ DISP[2]	EA = Address	EA = C[3] (Address)
T = 01 EA = XR1 ∠ DISP	EA = Address ∠ XR1	EA = C (Address ∠ XR1)
T = 10 EA = XR2 ∠ DISP	EA = Address ∠ XR2	EA = C (Address ∠ XR2)
T = 11 EA = XR3 ∠ DISP	EA = Address ∠ XR3	EA = C (Address ∠ XR3)

1 - Contents of instruction count register or index register

2 - May be true positive quantity or negative 2's complement quantity (-128 to +127)

3 - C specifies " contents" at location specifed by address of address ∠ XR1, XR2, or XR3

The recognition in the instruction of an indirect Address Control bit will cause the address portion to be treated as an indirect address. The address, after indexing (if specified), gives the location of the effective address. An additional memory cycle is required.

PROCESSOR ORGANIZATION

The CPU is a single address, fixed point, two's complement, fractional arithmetic system. It consists of the basic timing, register organization, arithmetic-logic unit, I/O interface and interrupt interface.

The Model CP Data Flow contains ten 16-bit storage registers, one 18-bit buffer register, one 16-bit storage address register, one 17-bit adder, a bus system, and associated gates and controls. The data flow of the central processor is 16 bits wide. Data flow is organized around a two-clock system. A two-clock cycle is approximately 417 nanoseconds. Timing is generated from a frequency source internal to the CPU.

The computer is comprised of three main buses - the left adder bus, the right adder bus, and the buffer bus. The left adder bus will gate any one

of eight storage registers to the left side of the adder; and the right adder bus controls which storage register is gated into the right side of the adder. Output data from the adder is distributed to any of the ten storage registers via the buffer register bus (BFR). Data inputs to the central processor are gated from the I/O onto the BFR bus. Output data is read into the I/O from this same bus. Data to and from AGE equipment is also taken from this bus.

The major portion of the data flow consists of the ten storage registers. All of these are provided with appropriate circuitry to enable any register to have access to the bus system. These registers are general purpose and can be used as necessary for any function.

In the configuration currently being fabricated, the ten primary storage registers are assigned as follows:

B_1	Operand Buffer Register	16 bits
B_2		16 bits
L	Mask Register	16 bits
G	Instruction Register	16 bits
A_1	Accumulator	16 bits
A_2		16 bits
Q_1	M-Q Register	16 bits
Q_2		16 bits
PC	Program Counter	16 bits
STAT	Status Register	16 bits

The STAT register has the additional capability of recording machine status and controlling the ROS sequencing. Additional gating is provided for the STAT register to enable unique combinations of bits to be read for control and arithmetic operations. Three Index Registers (XR) are provided, two in storage and one in high-speed circuitry.

In addition to the previously described registers, the following registers are also used by the computer.

SAR	Storage Address Register	16 bits
SDR	Storage Data Register	36 bits
BFR	Buffer (Adder output register, common second rank register)	18 bits

READ ONLY STORAGE

The processor is controlled by a permanently recorded microprogram which is stored in a read only store (ROS) and is supplemented by conventional control logic. Each time the ROS is addressed, a microinstruction is read. When decoded, the microinstruction controls the routing of data in the CPU and provides means of selecting the next microinstruction.

To understand the operation of ROS, it is helpful to note its relationship to conventional controls. Conventional controls may be characterized by sequence triggers and by the control lines activated by the sequence triggers as a function of the operation to be performed and data conditions. Each cycle that the CPU may take represents a state of the CPU as defined by the control circuitry. Each state, in turn, specifies which control lines are to be activated during that cycle and which state is to follow next. The defined state will cause the next sequence trigger to be set in the following cycle. In some cases the next state may be contingent upon a branch condition in which one of two or more sequence triggers must be selected.

In ROS-controlled processors, the sequence triggers are replaced by microinstructions or ROS words. Each ROS word consists of a predetermined bit pattern and represents a state of the CPU. The addressed ROS word controls the CPU during the particular machine cycle in use. When decoded, the ROS word defines all control lines that are to be activated during the machine cycle. Also contained in the ROS word is the address of the next ROS word to be used. If the address of the next ROS word is dependent on data conditions (for example, branch if overflow occurs), a base address and the conditions to be tested (branch test) are specified in the ROS word. In this case, one ROS word is associated with each of the possible data conditions; the ROS word whose associated conditions are satisfied is the next to be addressed. Thus, ROS eliminates the need for most of the complex sequencing networks.

Each ROS word consists of unique predetermined bit configuration grouped into control fields. The number of bits within a field determines the number of unique control signals (micro-orders) available within that field. (In a 4-bit field, for example, 16 distinct micro-orders can be defined, only one of which can be activated at any one time.) The micro-orders are grouped functionally within the fields according to two rules:

1. All micro-orders grouped in a field must be mutually exclusive since only one micro-order within that field may be specified at a time.

2. Micro-orders that are functionally similar (such as micro-orders that control outgating to the adder right input bus) are grouped in one field for ease of decoding.

Usually, rule 1 results in rule 2.

When decoded, each micro-order activates one or more control lines that condition logic gates to perform the function specified by the micro-order. Each micro-order is assigned a mnemonic code that defines the control function performed.

MAIN STORAGE

The MS is a three-dimensional coincident current 8,192-word, (standard) 36-bit ferrite core array. Thirty-two bits contain data, two bits are allocated to parity, and two bits are used for storage protection. Access time for this system is approximately 0.9 usec., and it can operate continuously at a 2.5 usec. cycle. Core storage addresses begin at 0000 and end at 8,191. Storage wrap around exists; i. e., the next sequential address above the highest numbered address is always 0000.

A storage protection capability has been implemented through the use of storage protection tests in every instruction which modifies storage. Each 36-bit storage word contains two storage protection bits; one for each half-word. These bits differentiate between protected and scratch pad locations, and are assigned when core storage is loaded to prevent inadvertent modification of the data by the program. Any attempt to store data in a protected location will be prevented by an Invalid Address interrupt in ROS control.

The main storage core plane (Figure 3) used in Model CP is a militarized version of a commercial IBM plane. The wiring is done by machine on high volume production equipment.

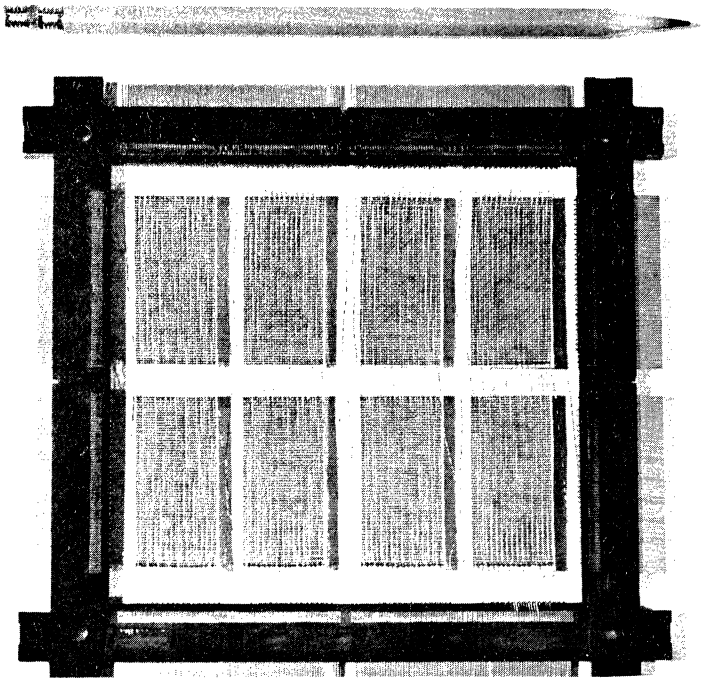


Figure 3. Main Storage Core Plane

INPUT/OUTPUT

The Interface Adapter assembly, included as part of Model CP, contains the necessary buffering and interface circuits to handle the following I/O channels:

- 3 high-speed, parallel, input channels
- 1 high-speed, parallel, output channel
- 1 serial output channel
- 24 discrete inputs
- 144 discrete outputs
- 24 interrupt lines

Table 3 includes the characteristics of the Interface Adapter.

TABLE 3
INTERFACE ADAPTER CHARACTERISTICS

Type	Parallel and serial operation
Modes of Operation:	Multiplex (interleave mode) Multiplex (block transfer mode) Direct I/O Synchronous servicing of interface adapter channels
Transfer rate:	
Parallel	260,000 words/sec (Burst Mode)
Serial	200 KC bit rate

Three fully buffered parallel input channels are provided. Each channel is capable of transferring a data word equal in length to the memory word size (32 bits plus check bits). Each channel consists of a data buffer register, an address register, and a count register. In addition, each channel shares the use of common circuitry consisting of the incrementer, control unit, and drivers.

A parallel output channel is provided for high speed block transfers to an external device. The output channel is functionally identical to a parallel input channel except that data is read from storage and placed in the buffer whenever a strobe pulse is received from the external device. A maximum burst rate of 260,000 words per second is provided. Provision is made for optionally adding two more parallel output channels.

A serial output channel is provided for transmission to an external device of serial data at a 200-kc bit rate. A 32-bit plus parity shifting register, loaded in the same manner as the parallel output channel, is provided. The output is fully buffered and converts data to the return to zero form. A shift clock is provided, the leading edge delayed from the leading edge of the data by 1/4 of the shift clock period. The parity of the serial output message is checked.

Twenty-four discrete inputs are sampled in the interface adapter. The input signal logic levels will be compatible with the interface adapter logic circuit. One hundred forty - four discrete outputs are provided. Each discrete output is latched, and a TTL gate is provided to drive the interface line, thereby isolating the latch from the capacitive load of the line.

Twenty-four external interrupt lines are provided with four levels of priority provided. The system consists of the mask register, the demand registers, and the priority resolution network. Demands which arise from external sources set individual latches in the demand registers. The outputs from groups of these latches are combined into four different priority levels. The status of the demands at each level is forwarded to the priority resolution network.

This network compares the demands to the status of the mask register. If the current demands are of higher level than that of the mask register, a 2 bit code is generated and sent to the computer to indicate the existence of an interrupt and its level.

The demand register will record all interrupt signals which are sent to it. The status of the demand registers may be read into the computer for inspection by the program.

The demand registers retain information until it is revised under program control. If the program revises the setting of the mask register to a lower level, interrupts which had previously been masked off may be revealed and will immediately generate signals which are sent to the computer. A special command will permit general resetting of all demand registers.

The interrupt system has been configured to allow considerable flexibility for the programmer. In general, once the basic systems decisions are made, large parts of the routines associated with the handling of interrupts are either microcoded or put into monitor programs where, in either case, the operational programmer does not handle the details of interruption but merely refers to the standard routines.

COMPUTER PACKAGING

The Model CP is packaged in two cases. The contents of each case are as follows:

Computer Case	Interface Adapter Case
Central Processing Unit	Interface Adapter
Read Only Storage	Power Regulator
Power Supply	RFI Filter
Main Storage	

The cases are dip-brazed aluminum structures which conform to standardized ATR case dimensions. Heat generated from the subassemblies within each unit is conducted to the side walls of the structure. These walls consist of a hollow rectangular extrusion which forms a passage for the cooling air. Thin corrugated sheeting is brazed within this passage to provide the necessary surface area for forced air convection cooling and to rigidize the structure. The cooling air enters the passage on the rear panel and exits at the front panel.

Electrical interfacing between units is accomplished with input-output connectors located on the front panels. Lightweight covers at the top and bottom of the units facilitate removal of subassemblies, provide radio frequency attenuation, and prevent entrance of foreign materials. The units are mounted to the vehicle structure at the bottom edges of the front and rear panels.

The electronics for Model CP is implemented with integrated circuits mounted on pluggable electronic subassemblies called pages (Figure 4). A page consists of two multilayer interconnection boards (MIBS) bonded to a metal frame. An insulator separates each board from the frame. Two 98-pin connectors,

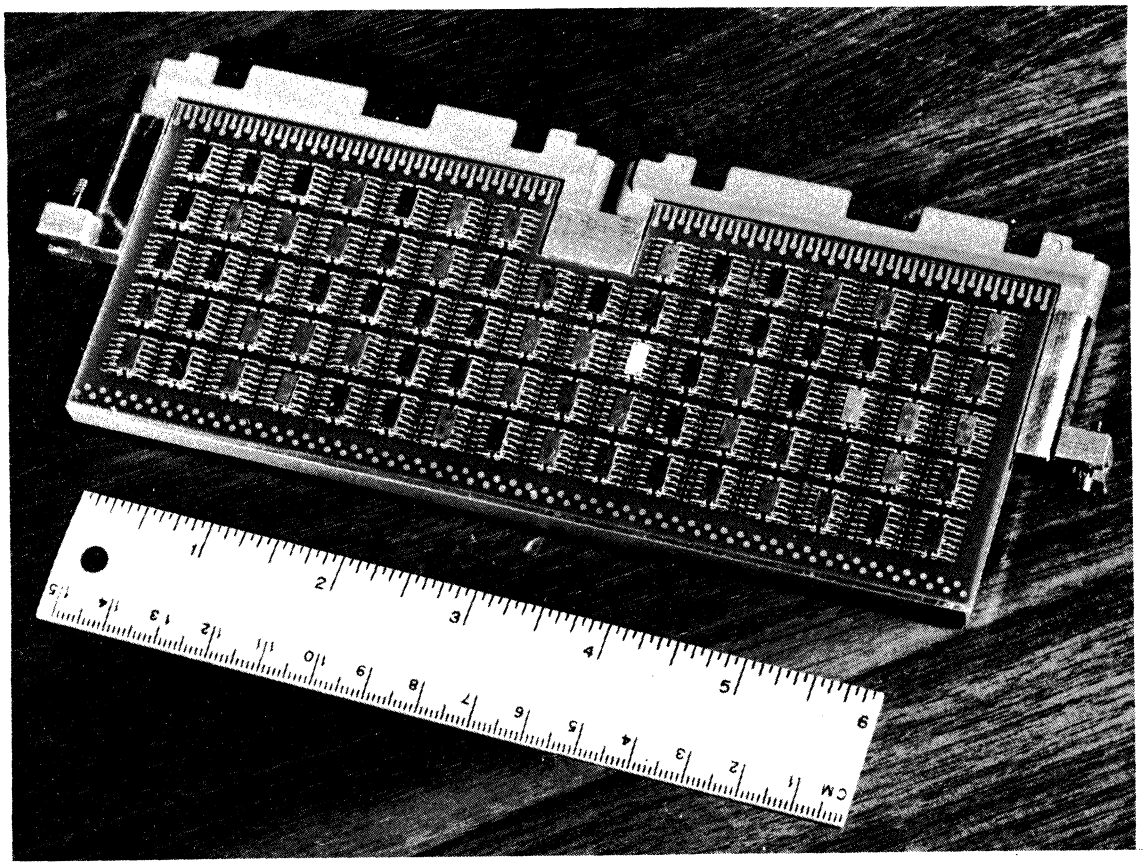


Figure 4. Typical Integrated Circuit Subassembly

developed for the Saturn V Guidance Computer subassembly, are fastened along the lower edge of the frame. The connector was developed for this application and has demonstrated its performance and reliability in extensive tests and field service. It has been qualified for use in man-rated space vehicles. Feedthrough connections and test points are provided along the upper edge of the frame. The page is fastened to the supporting structure at the two mounting flanges along an axis through the page center of gravity. Additional support is provided by the connectors. The page will be cooled by thermal conduction through the frame and mounting flange thermal interface. Keyed guide pins project from the lower edge of the page to prevent mislocation during installation. A tool is furnished for page insertion and withdrawal. Guides will be provided to facilitate handling during page insertion and withdrawal.

The integrated circuit flatpacks are soldered to etched patterns on the surface of the multilayer printed circuit boards. These multilayer boards allow the flatpacks to be closely spaced. Figure 4 shows one side of a typical page with flatpacks mounted. Each page contains two multilayer boards with up to 78 flatpacks per board. Discrete components are soldered in plated holes or on the board surface depending on terminal configuration. A conformal coating is applied for component support and environmental protection.

The main storage array is fabricated from core planes which are a militarized version of planes used on the IBM System/360. Each plane has 16, 384 cores. Planes are conformally coated and foam padding is placed between planes for environmental protection of cores.

The array is mounted in a housing which serves also to mount pluggable electronic subassemblies and acts as a thermal path. Pluggable electronic subassemblies similar to those in the central computer are used to mount the memory circuits. A distribution multilayer printed circuit board is used to interconnect memory circuits and the array. The main memory assembly is removable as a module. Electrical connections to the assembly are made through pluggable input-output connectors.

The Interface Adapter section is constructed using the pluggable electronic subassemblies with some utilization of discrete components because of the nature of the circuits.

Read Only Storage is implemented with a microminiature linear ferrite 7/12 (ID/OD in mils) magnetic core. The basic core plane contains 512 by 70 cores. A core is located where a "1" bit is stored and a core is missing where a "0" bit is stored. Two wires thread each core: one drive winding and one sense winding. Two planes are required for the ROS. Monolithic circuits are used in the decode, driver, detector and latch portion of the ROS.

SOFTWARE SUPPORT

A sophisticated package of computer programs will be provided for operation on

an IBM System/360. In addition, assemblers will be delivered which will operate on System/360 and the Model CP. This support package will assist the programmer in producing operational programs for the Model CP by providing:

- Programming aids that will enable a program to be expressed in a language which is readily understood
- Facilities for storing, modifying, and assessing precoded programs
- A program checkout system to ensure central computer operational program readiness

The software support programs delivered with the Model CP are the following:

- The assemblers will assemble programs executable on the Model CP. Programs will translate symbolic instructions into machine language instruction, assign storage locations, and perform auxiliary functions which allow the programmer to make efficient use of all of the features of the Model CP while relieving him of the details of machine language coding. This program provides for:
  - The assembly of relocatable object programs
  - Linkage between programs assembled at different times
  - Various listings under programmer control
  - A significant amount of error detection
  - An assemble-and-run capability when operating with the dynamic program checkout system described below
  - Automatic selection of long and short format instructions
  - Macro processing
- Service Programs which assist the programmer by performing frequently used functions such as:
  - Combining and relocating programs assembled at separate times
  - Handling the transition from assembly to execute as described above
  - Maintaining the library tape
- A dynamic program checkout system (simulator) which will provide for dynamic program analysis of the Model CP at the instruction word level of operation. This program will execute any instructions and aid in program debugging by providing:
  - Simulator control through a user's FORTRAN IV program

- User program access to simulated Model CP object program computed data
- Object program correction
- Program debugging options such as dumps, snaps, and traces
- Input/output and interrupt initiation and response

The test programs supplement the processor detection circuitry to provide a comprehensive self-check. The self-check capability is used to provide three separate functions in Model CP: error detection, fault isolation, and verification of operational readiness when required.

The methods used in the operational test program are a combination of special ROS diagnostic words (micro-instructions) and programmed routines (macro instructions) contained in the main storage section. It is a resident program within the problem program and its prime purpose is to isolate to the replaceable assembly by verifying the operations of the following:

- ROS addressing
- ROS decoding
- Instruction operation and flow through the Instruction execution micro-orders
- Processor data flow
- Register, adder, and shifter functions
- Channel data transfer where possible in the operational environment
- Interrupt feature where possible under the operational confines
- Limited main storage addressing
- Limited main storage data transfer

It is designed such that it may be entered without any special initialization and can be interrupted at any time to return to the problem program.

In addition to this, conventional program routines, located in main storage, will be cycled to insure proper Instruction operation and that all paths through the instruction execution micro-order sets have been exercised.

INSTRUCTION LIST

A complete list of Model CP instructions and execution times is contained in Table 4.

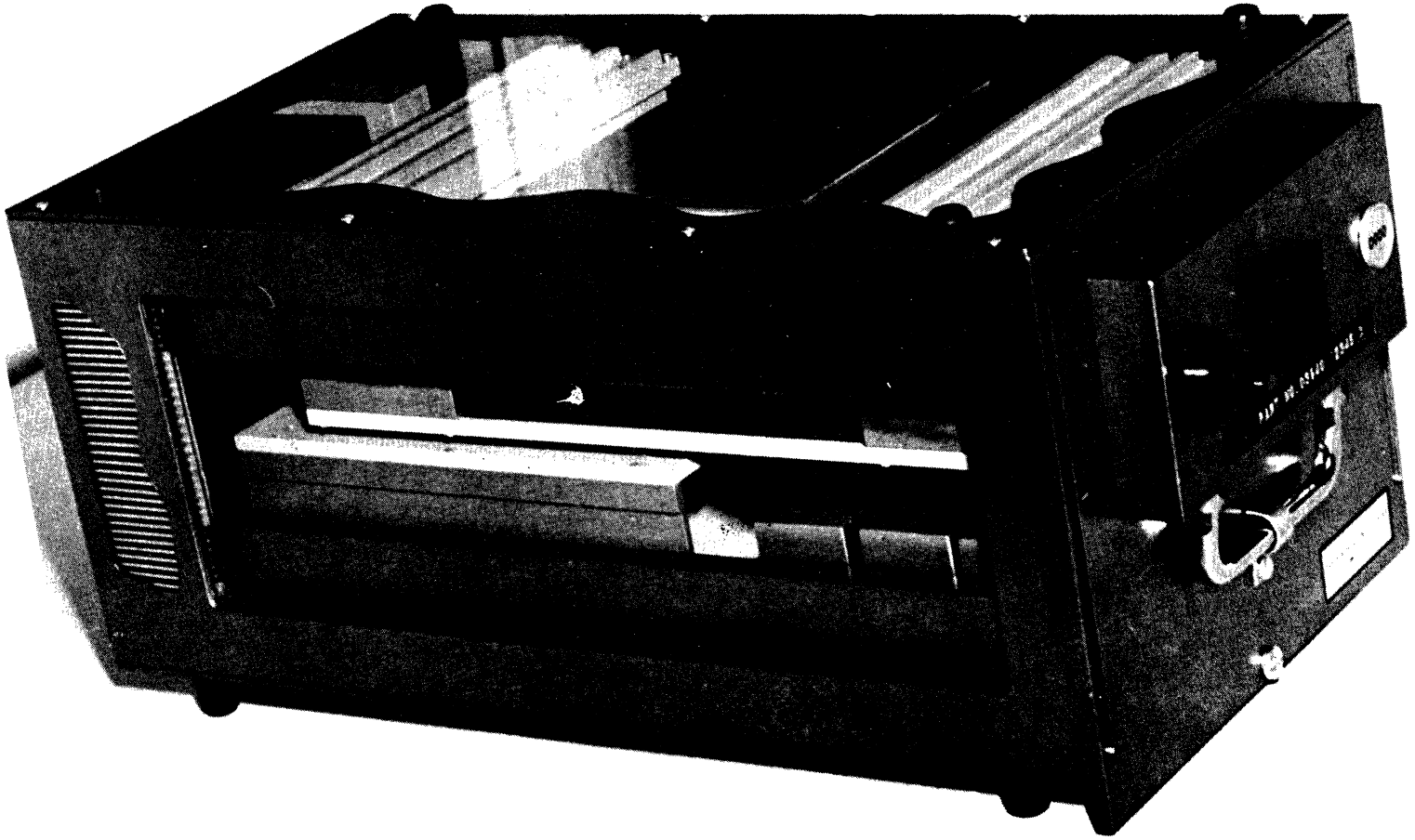
TABLE 4

INSTRUCTION LIST

OPERATION	TYPICAL EXECUTION TIMES (USEC) SHORT FORMAT	LONG FORMAT
ADD	5.00	5.00
ADD HALFWORD	5.00	5.00
AND	5.00	5.00
BRANCH ON CONDITION	N/A	4.58
BRANCH OUT ON CONDITION	N/A	3.75
BRANCH AND STORE INST. CTR.	5.83	8.75
COMPARE	5.00	5.00
COMPARE HALFWORD	5.00	5.00
DIRECT INPUT OUTPUT CONTROL	9.17	9.17
DIVIDE	45.42	45.42
EXCLUSIVE OR	5.00	5.00
INSERT STORAGE PROTECT BIT	N/A	6.66
LOAD A	5.00	5.00
LOAD A HALFWORD	5.00	5.00
LOAD INDEX	2.92	3.33
LOAD Q	5.00	5.00
MODIFY INDEX	5.00	5.00
MODIFY STORAGE HALFWORD	N/A	6.25
MULTIPLY	29.58	29.58
OR	5.00	5.00
SHIFT LEFT	VARIES	N/A
SHIFT LEFT AND COUNT	VARIES	N/A
SHIFT LEFT AND COUNT DOUBLE	VARIES	N/A
SHIFT LEFT DOUBLE	VARIES	N/A
SHIFT RIGHT ARITHMETIC	VARIES	N/A
SHIFT RIGHT ARITHMETIC DOUBLE	VARIES	N/A
SHIFT RIGHT LOGICAL	VARIES	N/A
SHIFT RIGHT ROTATE DOUBLE	VARIES	N/A
SKIP ON CONDITION	4.58	N/A
STORE A	5.83	5.83
STORE A HALFWORD	5.42	5.42
STORE INDEX	6.25	6.25
STORE Q	5.83	5.83
SUBTRACT	5.00	5.00
SUBTRACT HALFWORD	5.00	5.00

SYSTEM/4 Pi

MODEL CP-2



SYSTEM/4 Pi MODEL CP-2 DIGITAL COMPUTER MOCKUP

TABLE OF CONTENTS

FOREWORD

The IBM Federal Systems Division has developed a family of militarized, general purpose, digital computers called System/4 Pi. The name 4 Pi is derived from the number of steradians in a sphere, symbolizing the capability of these computers to address the entire spectrum of military applications.

The models in the System/4 Pi family currently under contract include:

TC - Tactical Computer
CP - Cost Performance
CP-2 - Cost Performance - Model 2
EP - Extended Performance

These models satisfy the low-cost, limited processing requirements and expand upward to meet the requirements of the very largest data processing applications. Deliveries on all models begin in early 1967.

MODEL CP-2

SUMMARY

The Model CP-2 is a general purpose, stored program, digital computer. It is designed for real-time control applications such as navigation, guidance, or weapons delivery.

Major subsections of the computer are storage, processor, and I/O section. All signals entering or leaving the computer are digital in nature. In the configuration presently being fabricated, a power supply is included within the processor structure.

The 2.5 microsecond, 8,448 word core storage has a parallel 36 bit transfer path to the processor section. Two parity bits and two storage protect bits insure accurate transfer and control of data located in storage. The storage section may be expanded up to 33,792, 36 bit words.

The processor is organized as a bus system design with a 16 bit fully parallel data flow. The processor contains the facilities for addressing main storage, for fetching or storing information, for arithmetic and logical processing of data, instruction sequencing, timing, and for initiating the communication between storage and external equipment.

Working registers are located in high speed hardware. A total of 35 instructions are implemented presently. Processing rates on the order of 125,000 operations/second are attained by optimizing the machine organization, monolithic logic circuitry, and the high speed core storage.

A summary of Model CP-2 characteristics is contained in Table 1.

DATA AND INSTRUCTION FORMATS

The standard or single precision data word is 32-bits long (Figure 1)

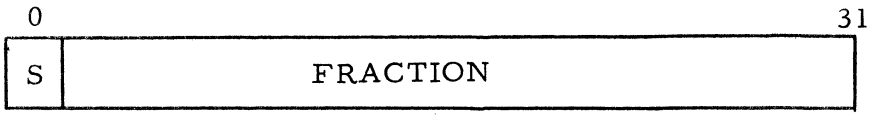


Figure 1. Data Format

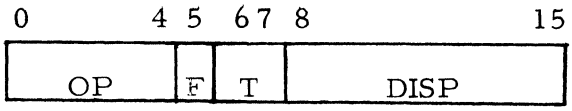
SYSTEM/4 Pi MODEL CP-2 CHARACTERISTICS

TYPE	General Purpose, Digital	
ORGANIZATION	Parallel, fixed point, fractional, two's complement notation	
INSTRUCTIONS	36	
INSTRUCTION WORD	16 or 32 bits	
DATA WORD	16 or 32 bits	
COMPUTER CYCLE TIME	417 nsec	
STORAGE CYCLE	2.5 msec	
TYPICAL EXECUTION TIMES		
	Short Format	Long Format
ADD	3.75 usec	5 usec
MULTIPLY	18.13 usec	19.38 usec
DIVIDE	47.29 usec	48.54 usec
OPERATIONS/SEC	125,000 typical	
PHYSICAL		
SIZE	10" x 7.6" x 19.6" (.86 cu. ft.)	
WEIGHT	47 lbs.	
MAIN STORAGE	8,448 - 36 bit words expandable to 33,792 words	
TECHNOLOGY	TTL Integrated Circuits	
POWER	240 Watts	
ENVIRONMENT	MIL-E-5400, Class 2	
MTBF	4,000 hours predicted	

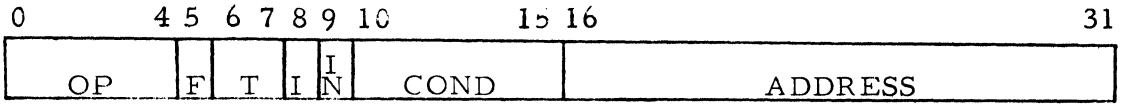
Positive numbers are always in true binary form. Negative numbers are in two's complement form. The sign bit, position 0, is always 0 for positive numbers and 1 for negative numbers.

A half-precision data word of 16 bits is provided to increase processing speeds in areas where full accuracy is not required. Addressing is to the half-word level, but 32-bit data words must have an even address.

Two basic instruction word formats are used (Figure 2).



Short Instruction Format



Long Instruction Format

Figure 2. Instruction Word Formats

In a typical program, long format instructions are used infrequently, allowing a larger program to be contained in storage. The bits within the instruction words are used in the following manner:

OP — These 5 bits define which operation is to be performed by the computer.

F — This format bit controls the instruction format. A "zero" indicates a normal instruction, and a "one" indicates a long instruction.

T — These two Index Tag bits address the Index Register or Instruction Register (I) used in the address modification.

DISP — These bits are called the displacement and, with normal format instructions only, are usually added to the Instruction Register or the index register specified by T. The modified address is defined as the effective address (EA). If T is 00, the displacement is added to the

Instruction Register. The displacement is in 2's complement form if negative.

IA — Except for the Modify Index and Skip instruction with T=00, the Indirect Addressing bit is used in the long instruction format. If "zero", addressing will be direct. If a "one", addressing will be indirect.

INT — This bit specifies that the Branch or Skip on Condition (BSC) instruction is to be interpreted as a "Branch Out" when used in an interrupt routine.

COND — These six bits specify the condition of indicators that are interrogated on a BSC or Branch and Store Instruction Counter (BSI) instruction.

ADDRESS — These 16 bits usually specify a core storage address in a long instruction. The address can be modified by the contents of an index register, or used as an indirect address if the IA bit is on.

ADDRESSING

All core storage locations are directly addressable under program control. The effective address may be determined by examining the tag, flag, and indirect address bits shown in Figure 2.

The recognition in the instruction of an indirect address control bit will cause the address portion to be treated as an indirect address. The address after indexing, if specified, gives the location of the effective address. An additional storage cycle is not required if index register one is used. One level of indirect addressing is provided.

The effective address for operands is formed as shown in Table 2.

TABLE 2

EFFECTIVE ADDRESS GENERATION

INDEX REG.	RELATIVE ADDR.	DIRECT ADDR.	INDIRECT ADDR.
0	Program Counter + DISP*	ADDRESS	C (ADDRESS)**
1	XR1 + DISP	ADDRESS + XR1	C (ADDRESS + XR1)
2	XR2 + DISP	ADDRESS + XR2	C (ADDRESS + XR2)
3	XR3 + DISP	ADDRESS + XR3	C (ADDRESS + XR3)

*DISP may be positive or negative (2's complement)
**C means "contents of" ADDRESS field in instruction

PROCESSOR ORGANIZATION

The processor is organized as a single address, fixed point, two's complement, fractional arithmetic system. It consists of the basic timing, register organization, arithmetic logic unit, I/O interface, and interrupt interface.

The data flow contains eight 16-bit storage registers, two 16-bit buffer registers, one 16-bit storage address register, one 32-bit storage data register, one 16-bit operation code register, one 17-bit adder, a bus system, and associated gates and controls. The data flow of the central processor is 16 bits wide. Data flow is organized around a two-clock system with a cycle of approximately 417 nanoseconds. Timing is generated from a frequency source internal to the CPU.

Functions of the hardware registers are as follows:

A1 and A2 - These two 16-bit registers comprise the accumulator, the principal arithmetic register. The quotient is also contained in the accumulator following a division operation.

B1 and B2 - These two 16 bit auxiliary registers are used to hold the multiplicand or divisor.

IX1 - A 16-bit register used in calculating an effective address.

Program Counter - This 16-bit register is used to determine the location of the next instruction to be fetched from storage.

Storage Address Register - A 16-bit register used to determine the next storage location fetched. May be loaded from either the Program Counter or with the calculated effective operand address.

Storage Data Register - A 32-bit register used to hold the word read from the storage array.

Operation Register - A 16-bit register which holds the current instruction to be executed.

Next Instruction Register - A 16-bit register containing the next instruction to be executed, or part of the current instruction for long formats.

Buffer Register - A 16-bit register which contains the output of the adder. Its contents are distributed over the bus system to other hardware registers.

MAIN STORAGE

The MS is a three dimensional, coincident current, 8,448 word, 36-bit ferrite core array. Thirty-two bits contain data, two bits are allocated to parity, and two bits are used for storage protection. Access time for this system is approximately 0.9 usec, and it can operate continuously at a 2.5 usec cycle. Core storage addresses begin at 0000 and end at 8,447.

The nonalterable storage capability has been implemented by storage protection tests in every instruction which modifies storage. Each 36-bit storage word contains two storage protection bits, one for each half-word. These bits differentiate between protected and scratch pad locations, and are assigned when core storage is loaded from either the AGE or from the I/O Unit to prevent inadvertent modification of the data by the program. Any attempt to store data in a protected location will be prevented by an Invalid Address interrupt in computer control.

DRO cores with storage protection and power sequencing effectively provide program protection equivalent to most NDRO storage elements. DRO also allows complete flexibility in assigning storage between program and scratch pad, thus simplifying programming and allowing maximum use of the total storage capacity.

The storage element used is an IBM 13/21 (ID/OD in mils) lithium nickel ferrite. It is designed to operate over a temperature range of -55°C to +105°C.

The plane (Figure 3) used in the core array is a militarized version of a commercial IBM plane. The plane is approximately 6" x 6" x 0.156" and contains 16,896 cores.

Welded connections are used throughout the memory array. This plane is machine-wired on IBM high volume production equipment. Foam pads are placed between the coated planes to meet the vibration requirements.

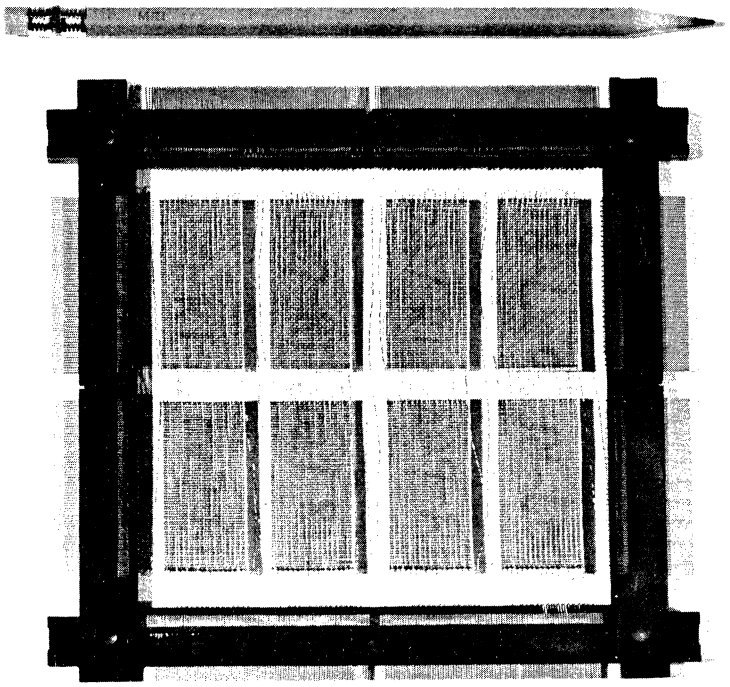


Figure 3. Core Plane

INPUT/OUTPUT

The input-output section of the computer provides high-speed data transfer on two independent two-way, 17-bit parallel channels. The number of channels can be expanded to three without redesign or engineering modification to the computer. Input-output data transfer on two of the channels is accomplished under external control. Data transfer on the third channel is accomplished under program control. This growth capacity consists of one external controlled output channel and one program controlled input channel.

In addition, the input-output section provides capacity for discrete inputs and outputs. Storage cycle requests for externally controlled channels have priority over all other normal storage cycle requests and are honored at the completion of the existing storage cycle.

Each externally-controlled input channel has a block of 2048, 17-bit words of storage reserved for input data. Direct input to storage will occur under control of the external device, but only when the automatic input/output mode has been enabled by a program instruction, which specifies the channel to be enabled. Each externally controlled input channel is capable of data transmission at a word rate of 60 KC interleaved with information transmission on the other externally controlled channels.

Externally-controlled output channels function in much the same manner, referencing a 2048 word block of storage and transferring data out at a 60KC word rate.

The processor may handle up to 40 discrete input lines, with eight of these operating as interrupt lines.

The interrupts are tested upon the completion of each instruction.

Interrupts are divided into two levels. The highest level consists of one interrupt, and the other level contains the remaining seven. The recognition of an interrupt in the lowest level causes all other interrupts in that level to be inhibited until the interrupt processing program is completed. The high-level interrupt is not inhibited while a low-level interrupt is being processed. Occurrence of the high priority interrupt causes a program branch. Upon completion of the high priority interrupt program, the interrupted program is resumed. The interrupted program could have been the main program or any one of seven lower priority interrupt programs.

Provisions are also included in the I/O Section for handling 16 discrete outputs.

COMPUTER PACKAGING

The Model CP-2 is packaged using pluggable construction. The pluggable electronic subassembly, called a page, consists of two multilayer printed circuit boards bonded to a metal frame. An insulator separates each board from the frame. Two 98-pin connectors, identical to the ones being used for the Saturn Guidance Computer subassembly, are fastened along the lower edge of the frame. The connector was developed for this application and has demonstrated its performance and reliability in extensive tests and field service. It has been qualified for use in man-rated space vehicles. Feed-through connections and test points are provided along the upper edge of the frame. The page is fastened to the supporting structure at the two mounting flanges along an axis through the page center of gravity. Additional support is provided by the connectors. The page will be cooled by thermal conduction through the frame and mounting flange thermal interface. Keyed guide pins prevent installation in the wrong locations.

The multilayer printed circuit boards are made of several layers of etched, copper-clad, epoxy-glass laminates which are bonded together under heat and pressure. These boards have been in production since 1962 and have been successfully used in many systems produced by IBM including the Titan II, Titan III, Gemini, Saturn I, and Saturn V Guidance Computers.

The integrated circuit flatpacks are soldered to etched patterns on the surface of the multilayer printed circuit boards. Figure 4 shows one side of a typical page with flatpacks mounted. Each page contains two multilayer boards with up to 78 flatpacks per board. Discrete components are soldered in plated holes or on the board surface, depending on terminal configuration. A conformal coating is applied for component support and environmental protection.

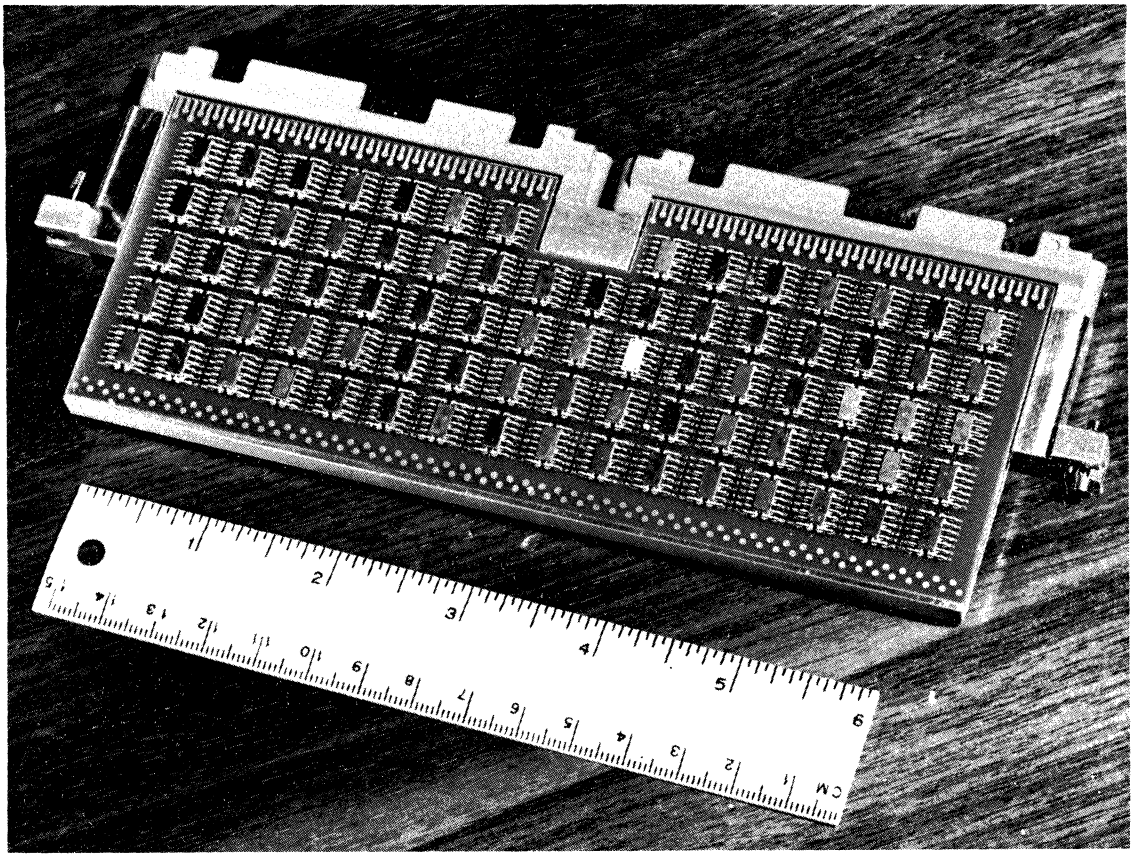


Figure 4. Typical Integrated Circuit Subassembly

The storage array is mounted in a housing which serves also to mount pluggable electronic subassemblies and acts as a thermal path. Pluggable electronic subassemblies similar to those in the computer are used to mount the storage circuits. A distribution multilayer printed circuit board is used to interconnect storage circuits and the array. The main storage assembly is removable as a module. Electrical connections to the assembly are made through pluggable input/output connectors.

The power supply assembly consists of three pluggable regulator modules and a transformer - RFI filter assembly. High-power components are fastened directly to the base plates for maximum heat transfer. A conformal coating is applied to the printed circuit board assemblies for environmental protection.

SOFTWARE SUPPORT

A sophisticated package of computer programs will be provided for operation on an IBM System/360. This support package will assist the programmer in producing operational programs for the Model CP-2 by providing:

- Programming aids that will enable a program to be expressed in a language which is readily understood
- Facilities for storing, modifying, and accessing precoded programs
- A program checkout system to ensure central computer operational program readiness.

The software support programs delivered with the Model CP-2 are the following:

- The assemblers will assemble programs executable on the Model CP-2. These programs will translate symbolic instructions into machine language instruction, assign storage locations, and perform auxiliary functions which allow the programmer to make efficient use of all of the features of the Model CP-2 while relieving him of the details of machine language coding. This program provides for:
  - The assembly of relocatable object programs
  - Linkage between programs assembled at different times
  - Various listings under programmer control
  - A significant amount of error detection
  - An assemble-and-run capability when operating with the dynamic program checkout system described below
  - Automatic selection of long and short format instructions
  - Macro processing.
- Service Programs which assist the programmer by performing frequently used functions such as:
  - Combining and relocating programs assembled at separate times
  - Handling the transition from assembly to execute as described above
  - Maintaining the library tape.

- A dynamic program checkout system (simulator) which will provide for dynamic program analysis of the Model CP-2 at the instruction word level of operation. This program will execute any instructions and aid in program debugging by providing:

  - Simulator control through a user's FORTRAN IV program
  - User program access to simulated Model CP-2 object program computed data
  - Object program correction
  - Program debugging options such as dumps, snaps, and traces
  - Input/output and interrupt initiation and response.

The self-test program is designed to be co-resident with the operational program. At periodic intervals, the operational program transfers control to the self-test routine to begin a functional go/no-go computer check. The self-test routine, when entered, cycles until interrupted by the operational program. Test results are progressive --results may depend upon prior test action, but can be interrupted at any point. There is no requirement for system restoration upon entering or exiting the test routine. The operational program also contains checks for reasonableness of data. The self-test routine covers the processor registers, data flow and controls, and the I/O interface circuitry to the extent that it will not interfere with the operational programs.

Tests performed by the self-test routine include patterns to check all possible adder combinations, carry paths, and carry look-ahead paths. Similarly, patterns and processor actions are used to check the function of the internal registers, shifter and associated data paths, and instruction control logic.

INSTRUCTIONS

A complete list of Model CP-2 instructions and execution times is contained in Table 3.

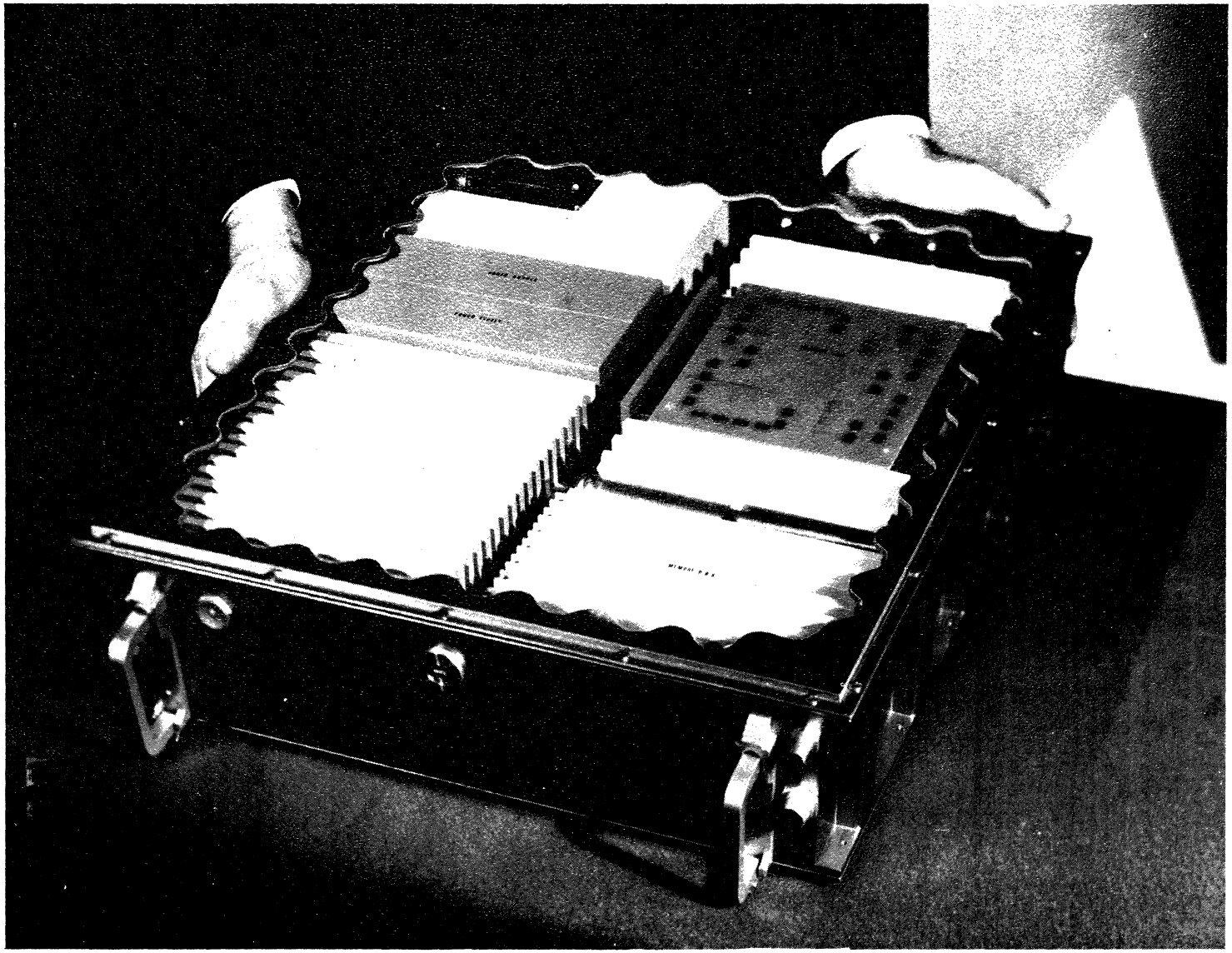
TABLE 3

INSTRUCTION LIST

OPERATION	TYPICAL EXECUTION TIMES	
	Short Format	Long Format
ADD	3.75	5.0
ADD HALFWORD	3.75	5.0
AND	3.75	5.0
BRANCH ON CONDITION	----	4.58
BRANCH OUT ON CONDITION	----	5.42
BRANCH AND STORE INSTRUCTION COUNTER	----	7.71
COMPARE	4.79	6.04
COMPARE HALFWORD	3.96	5.21
DIRECT INPUT OUTPUT CONTROL	2.50	3.75
DIVIDE	47.29	48.54
EXCLUSIVE OR	3.75	5.0
INSERT STORAGE PROTECT BIT	----	5.42
LOAD A	3.75	5.0
LOAD A HALFWORD	3.75	5.0
LOAD INDEX	2.08	3.33
LOAD Q	3.75	5.0
MODIFY INDEX	2.08	3.33
MODIFY STORAGE HALFWORD	----	5.83
MULTIPLY	18.13	19.38
OR	3.75	5.0
SHIFT LEFT	Varies	----
SHIFT LEFT AND COUNT	Varies	----
SHIFT LEFT AND COUNT DOUBLE	Varies	----
SHIFT LEFT DOUBLE	Varies	----
SHIFT RIGHT ARITHMETIC	Varies	----
SHIFT RIGHT ARITHMETIC DOUBLE	Varies	----
SHIFT RIGHT LOGICAL	Varies	----
SHIFT RIGHT ROTATE DOUBLE	Varies	----
SKIP ON CONDITION	2.71	----
STORE A	4.58	5.83
STORE A HALFWORD	5.42	6.67
STORE INDEX	5.42	6.67
STORE Q	4.58	5.83
SUBTRACT	3.75	5.0
SUBTRACT HALFWORD	3.75	5.0

SYSTEM/4 Pi

MODEL EP



MODEL EP MOCKUP

TABLE OF CONTENTS

FOREWORD

The IBM Federal Systems Division has developed a family of militarized, general purpose, digital computers called System/4 Pi. The name 4 Pi is derived from the number of steradians in a sphere, symbolizing the capability of these computers to address the entire spectrum of military applications.

The models in the System/4 Pi family currently under contract include:

TC - Tactical Computer
CP - Cost Performance
CP-2 - Cost Performance - Model 2
EP - Extended Performance

These models satisfy the low-cost, limited processing requirements and expand upward to meet the requirements of the very largest data processing applications. Deliveries on all models begin in early 1967.

MODEL EP

SUMMARY

The Model EP is a high-performance computer with a 32-bit wordlength and data path, and a Main Storage cycle time of 2.5 microseconds. It features microprogram control, enabling great flexibility in the definition of additional special instructions as desired. The microprograms control the data flow, and allow changes to be made to the standard computer instruction set without changing computer hardware. These microprograms are stored in a Read-Only-Storage (ROS) which is a high speed (417 nanosecond) core storage. The ROS has been "personalized" to implement the IBM System/360 Model 44 instruction set. Additional instructions may be added if further problem definition indicates a need.

The Input/Output interface uses an IBM System/360 Multiplexor Channel augmented with a high speed interface to handle high data rates. A multiplexor channel provides a flexible and economical means of attaching multiple devices to the computer by time-sharing the channel hardware between the various sub-channels. Up to 128 sub-channels (or devices) can be attached to the channel.

Since the Model EP computer uses the IBM System/360 instruction set, an extensive System/360 programming library can be made available. Also, user programs can be checked-out on commercial machines, simplifying the overall programming effort.

A summary of Model EP characteristics is contained in Table 1.

DATA AND INSTRUCTION FORMATS

The basic arithmetic operand is the 32-bit fixed-point binary word. Sixteen-bit halfword operands may be specified in most operations for improved performance or storage usage. To preserve precision, some products and all dividends are 64 bits long. The sign bit occupies the left-most digit position, with successive digits extending from bit position 1 in decending order of magnitude. Negative numbers are represented in two's complement representation, with a sign bit represented by a binary "one". Figure 1 shows the data word format.

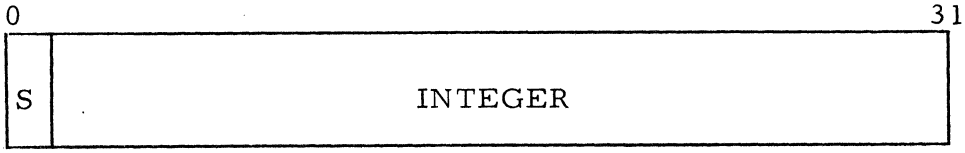


Figure 1. Fixed-Point Number Format

SYSTEM/4 Pi MODEL EP CHARACTERISTICS

TYPE	General Purpose, Digital	
ORGANIZATION	Parallel, fixed point, integer, two's complement notation (floating point optional)	
INSTRUCTIONS	70 (compatible with System/360 Model 44)	
INSTRUCTION WORD	16 or 32 bits	
DATA WORD	32 bits	
COMPUTER CYCLE TIME	417 nsec	
STORAGE CYCLE	2.5 msec	
TYPICAL EXECUTION TIMES		
	RR Format	RX Format
ADD	2.1 usec	5.0 usec
MULTIPLY	9.2 usec	10.4 usec
DIVIDE	20.0 usec	20.8 usec
OPERATIONS/SEC	190,000*	
PHYSICAL		
SIZE	8.25" x 17.5" x 22.5" (1.88 cu. ft.)	
WEIGHT	75 lbs.	
MAIN STORAGE	16,384 x 36 bit words, expandable to 131,072 words	
TECHNOLOGY	TTL Integrated Circuits	
POWER	365 Watts	
ENVIRONMENT	MIL-E-5400 Class 2	
MTBF	3,000 hours predicted	

*Reference value only, higher speeds possible with application-oriented microprograms.

The length of an instruction format can be one or two-halfwords. It is related to the number of storage addresses necessary for the operation. An instruction consisting of only one halfword causes no reference to main storage. A two-halfword instruction provides one storage-address specification. All instructions must be located in storage on integral boundaries for halfwords. Figure 2 shows four basic instruction formats.

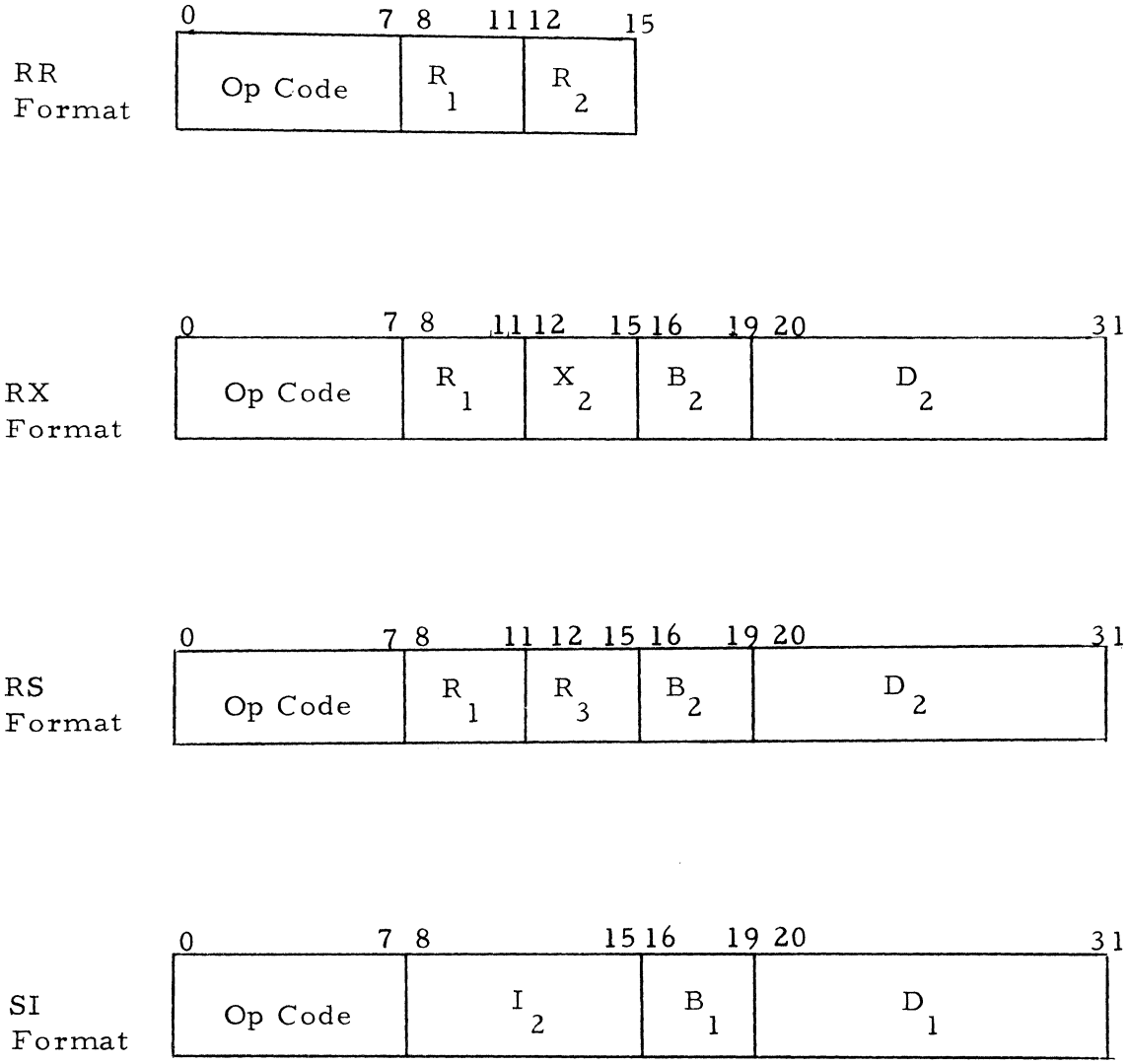


Figure 2. Basic Instruction Formats

The basic instruction formats are denoted by the format codes RR, RX, RS, and SI. The format codes express, in general terms, the operation to be performed. RR denotes a register-to-register operation; RX, a register-and-indexed-storage operation; RS, a register-and-storage operation; and SI, a storage-and-immediate-operand operation.

For purposes of describing the execution of instructions, operands are designated as first and second operands. These names refer to the manner in which the operands participate. The operand to which a field in an instruction format applies is generally denoted by the number following the code name of the field, for example, R_1, B_1, D_2.

The first byte of each instruction contains the operation code (op code). The length and format of an instruction are specified by the first two bits of the operation code. The second byte is used either as two 4-bit fields or as a single eight-bit field. This byte can contain the following information:

- Four-bit index register specification (X_2)
- Four-bit operand register specification (R_1, R_2, or R_3)
- Four-bit mask (M_1)
- Eight-bit byte of immediate data (I_2)

The second halfword always has the same format:

- Four-bit base register designator (B_1 or B_2), followed by a 12-bit displacement (D_1 or D_2).

ADDRESSING

Byte locations in storage are consecutively numbered starting with 0; each number is considered the address of the corresponding byte. A group of bytes in storage is addressed by the left most byte of the group. The addressing capability permits a maximum of 16,777,216 bytes, using a 24-bit binary address.

Fixed-length fields, such as halfwords and double words, must be located in main storage on an integral boundary for that unit of information. For example, words (four bytes) must be located in storage so that their address is a multiple of the number 4. A halfword (two bytes) must have an address that is a multiple of the number 2, and double words (eight bytes) must have an address that is a multiple of the number 8.

Instruction addressing is accomplished by referring to a register containing the current instruction address. The address in the register is then increased by the number of bytes (2 or 4) in the instruction to address the next sequential instruction. A change from sequential addressing may occur by branching, status switching or interruptions.

Operand addressing can be grouped into these classes: explicitly addressed operands in main storage, immediate operands placed as part of the instruction stream in main storage, and operands located in the general hardware registers.

The address used to refer to main storage is generated from the following three binary numbers:

- Base Address (B) is a 24-bit number contained in a general register specified by the program in the B field of the instruction. The B field is included in every address specification. The base address can be used as a means of static relocation of programs and data. The base address provides for addressing the entire main storage. The base address may also be used for indexing purposes.

- Index (X) is a 24-bit number contained in a general register specified by the program in the X field of the instruction. It is included only in the address specified by the RX instruction format. The index can be used to provide the address of an element within an array. Thus, the RX format instructions permit double indexing.

- Displacement (D) is a 12-bit number contained in the instruction format. It is included in every address computation. The displacement provides for relative addressing upto 4095 bytes beyond the element or base address. In array-type calculations the displacement can be used to specify one of many items associated with an element. In the processing of records, the displacement can be used to identify items within a record.

In forming the address, the base address and index are treated as unsigned 24-bit positive binary integers. The displacement is similarly treated as a 12-bit positive binary integer. The three are added as 24-bit binary numbers, ignoring overflow. Since every address includes a base, the sum is always 24 bits long. The address bits are numbered 8-31 corresponding to the numbering of the base address and index bits in the general register.

PROCESSOR ORGANIZATION

The processor contains the facilities for addressing main storage, fetching stored information, performing arithmetic and logical processing of data, sequencing instructions, and controlling the communications between main storage and external devices.

The processor has sixteen, 32-bit general registers called Local Store, implemented in active hardware elements for fixed point operands. The general registers can be used as index registers in address arithmetic, indexing, or as accumulators in fixed-point arithmetic and logical operations. Each register has a one word (32-bit) capacity, and can be selected by four-bit, R, B, or X fields in the instruction word.

The Model EP data flow consists primarily of two parallel paths which may be active simultaneously. One is a thirty-two bit wide adder/shifter path fed by several thirty-two bit registers. The other is an 8-bit wide mover path which performs logical operations on bytes of data selected from various processor registers. The computer operates on a 417 nanosecond cycle time which is the time required for a register-to-register transfer through the adder/shifter path or through the mover path.

Several registers make up the three main groups within the data flow. Some of the registers control instruction fetching, while others form the main working registers in the processor. The temporary storage registers containing results of instruction execution are the Local Store registers.

Some of the registers and their functions are as follows:

- Q Register (instruction buffer) - This is a 32-bit register which is used to buffer an instruction word. It is fed directly from the main storage data registers

- E Register (execution register) - This is a 16-bit register containing the first halfword of the instruction which is being executed. Bits 0 and 1 of the instruction specify the instruction length.

- IAR (instruction address register) - This is a 19-bit register which together with the instruction address counter is used to keep track of the address in main storage of the next instruction. Because all instructions are either 2 or 4 bytes long, and main storage delivers data in one word (4 byte) increments, the instruction address is updated by either two or four each time it is used.

- WA Register (working register A) - This is a 32-bit register which is a primary source of data for the adder left input. It can be gated direct true, direct complement, left 1 true, and left 1 complement to adder left.

- WE Register (working register E) - This is a 32-bit register which is a primary source of data for the adder right input. It is gated direct true to adder right.

- BR (Bus Register) - This is a 32-bit register which temporarily holds the shifted sum so that it may be returned to a working register or local storage. Data to be written into main storage is also gated from BR.

Programmed branches are normal means of altering the sequencing of instruction execution. Executive control and interrupt conditions require that data concerning the status of processor operations at the time of alteration be stored for future reference. This is accomplished by manipulation of the program status word (PSW). This double word includes the instruction address, condition code, and other fields. The active or controlling PSW is called the "Current PSW". By storing the current PSW during an interruption, the status of the computer at time of interruption is preserved. By loading a new PSW or part of a PSW, the state of the computer can be initialized or changed.

The interruption system permits the processor to change its state as a result of conditions in the I/O devices, in the computer or external to the system. An interrupt is defined as a demand for immediate computer response and is processed by a forced departure from the normal program instruction processing routine. Normal requests for I/O service are not classified as interruptions.

Six classes of interruptions are honored by the Model EP:

- Input/Output - conditions generated in the multiplexor channel or I/O devices
- External - conditions generated external to the computer. Six priority levels of external interrupts are provided.
- Program - unusual conditions encountered in a program
- Supervisor Call - result of executing the SUPERVISOR-CALL instruction
- Machine check - result of a machine malfunction
- Power Transient Detection - input power is at an abnormal level. This is a special sub-class of machine check.

READ-ONLY STORAGE

The processor is controlled by a permanently recorded microprogram which is stored in a read only store (ROS) and is supplemented by conventional control logic. Each time the ROS is addressed, a micro-instruction is read. When decoded, the microinstruction controls the

routing of data in the processor and provides means of selecting the next microinstruction.

To understand the operation of ROS, it is helpful to note its relationship to conventional controls. Conventional controls may be characterized by sequence triggers, and by the control lines activated by the sequence triggers as a function of the operation to be performed and data conditions. Each cycle that the processor may take represents a state of the processor as defined by the control circuitry. Each state, in turn, specifies which control lines are to be activated during that cycle and which state is to follow next. The defined state will cause the next sequence trigger to be set in the following cycle. In some cases, the next state may be contingent upon a branch condition in which one of two or more sequence triggers must be selected.

In ROS-controlled processors, the sequence triggers are replaced by microinstructions or ROS words. Each ROS word consists of a predetermined bit pattern and represents a state of the processor. The addressed ROS word controls the processor during the particular machine cycle in use. When decoded, the ROS word defines all control lines that are to be activated during the machine cycle. Also contained in the ROS word is the address of the next ROS word to be used. If the address of the next ROS word is dependent on data conditions (for example, branch if overflow occurs), a base address and the conditions to be tested (branch test) are specified in the ROS word. In this case, one ROS word is associated with each of the possible data conditions; the ROS word whose associated conditions are satisfied is the next to be addressed. Thus, ROS eliminates the need for most of the complex sequencing networks.

Each ROS word consists of unique predetermined bit configurations grouped into control fields. The number of bits within a field determines the number of unique control signals (micro-orders) available within that field. (In a 4-bit field, for example, 16 distinct micro-orders can be defined, only one of which can be activated at any one time.)

When decoded, each micro-order activates one or more control lines that condition logic gates to perform the function specified by the microorder. Each micro-order is assigned a mnemonic code that defines the control function performed.

The ROS generates micro-orders for arithmetic and other operations in a unit of time (a ROS cycle) which is 1/6 (417 ns) of the main storage memory cycle. The arithmetic operation to be executed is interpreted by a microprogram stored in the read-only-store, with each step of the microprogram distributing logic levels throughout the central processor. The sequence of steps implements the desired operation: arithmetic, logical, etc.

The distribution of control signals for a large instruction repertoire provides the potential for modification or creation of instructions through modification of the microprogram contained in ROS. The creation of macro-instructions specifically tailored to the application is technically feasible and readily implemented.

Following are some areas for exploiting ROS microprogramming flexibility: Implementation of trigonometric functions, special radar data processing, and logical processing.

MAIN STORAGE

The main storage is packaged in 8,192 word modules. All main storage words are 36 bits long. The cycle time, measured from select to the preceding select pulse, is 2.5 microseconds. The access time is 900 nanoseconds, maximum, measured from the select pulse.

There are two operating modes for storage - Read-Write and Split-Cycle.

- Read-Write - This mode is the normal read-restore mode.
- Split-Cycle - In this mode, the storage cycle consists of three parts read, dead time (compute time), and store. The select signal initiates the read portion. The dead-time (or compute time) is under the control of the computer and no limit is placed on this time. The store portion of the split-cycle operation is initiated by the split-cycle store command.

The inputs to Main Storage are controlled by the Storage Bus Control (SBC). The SBC determines the availability of storage and once a request for storage is honored, all other requests are held until the SBC is free to handle the request and storage is not busy.

Once a request has been granted access to storage, the SBC accepts the address, data, and control bits for processing and transmission to storage. The low order 13 bits of the address is multiplexed to the Storage Address Register (SAR) of each 8K storage module. These 13 bits (2^{13}) are sufficient to specify each of the 8,192 word locations. The SAR provides these address bits to the decode circuits.

The four remaining high order bits are sent to each 8K storage module. Each storage module determines if the specified storage module address is its own. If a module is selected, gating is established to accept the data from the computer and to ensure that data is returned to the computer. The storage module notifies the computer that its request has been honored, and generates a busy signal to prevent additional access until the storage cycle is completed.

After 900 nanoseconds, the data from the addressed storage location is available at the storage sense amplifiers and is strobed into the Storage Data Register (SDR). Prior to inserting the output data in the SDR, storage generates an advance pulse to the SBC. The SBC notifies the requesting area that data from storage is available.

The storage element used in the Model EP storage system is an IBM 13/21 (ID/OD in mils) lithuim nickel ferrite toroidal core that is designed to operate over a wide temperature range. The core planes are automatically wired and tested on IBM production lines. Figure 3 shows the core plane used in Model EP.

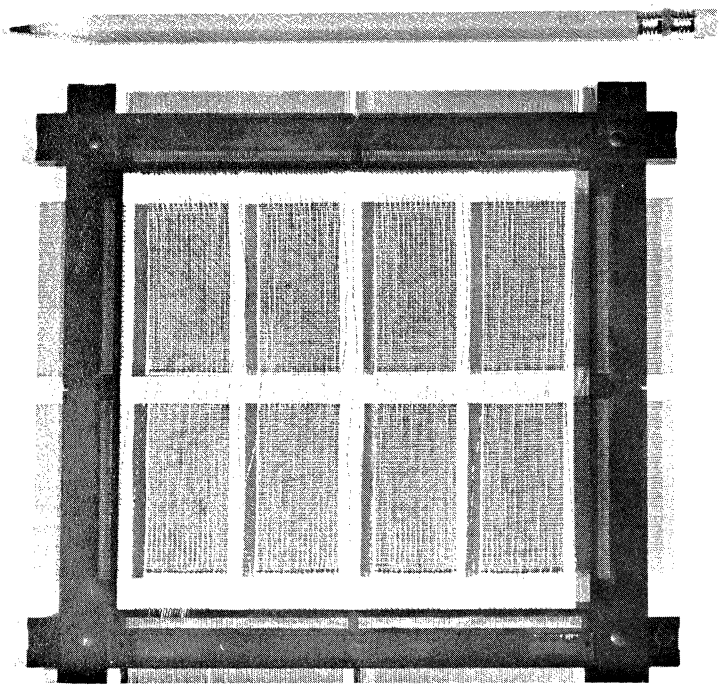


Figure 3. Core Plane

INPUT/OUTPUT

The model EP Computer input-output data transfers, as well as certain control signals, are routed between external devices and the main storage via the input-output multiplexor channel and control units. The only other external interface to the computer is accomplished through the external interrupt system. Each control unit provides interface adaptation between the multiplexor channel and several I/O devices. The devices accommodated by a single control unit can be a mixture of input and output devices.

The basic data flow is between an I/O device and the multiplexor channel via a control unit, and between multiplexor channel and storage via the processor. Areas are arbitrarily assigned in storage as input/output data buffering areas. The assignment of buffer areas is under control of the programmer, and a given area of any length can be established anywhere in storage.

The multiplexor channel has two basic modes of operation, the multiplexed mode and the burst mode. Capability to sustain several I/O operations on a time-shared basis in the multiplexed mode is the most important feature of the multiplexor channel. Figure 4 illustrates the multiplexed operation. In this mode, the channel services the I/O devices asynchronously as the input data becomes available, or when the output devices can accept more data. Each data transfer consists of 1 byte of data (8 bits) plus a parity bit to verify proper transmission.

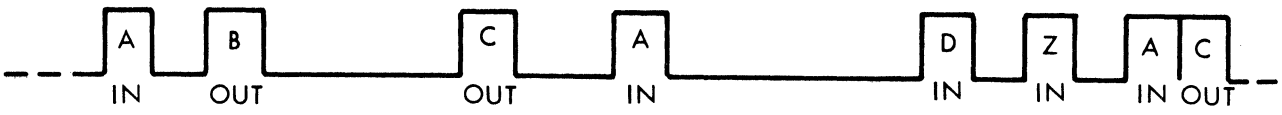


Figure 4. Multiplex Mode

The burst mode is an alternate multiplexor channel operation. In this mode, shown in Figure 5, the channel remains connected to a particular device until the transfer of a block of data is completed. All other devices are normally locked out of the channel until the end of the burst transmissions. The lengthy burst mode shown is usually reserved for high-speed, long-record transfers. As in all standard data transfers, the transmission is a byte at a time. The burst mode can also be used for lower speed program loading from a tape unit when it is desirable to block off any other I/O operation.

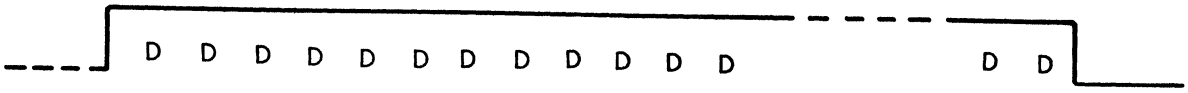


Figure 5. Burst Mode

Most frequently, the channel will be alternating between multiplexed and burst modes - a mixed mode (Figure 6). The burst mode can be controlled by the device requesting service and can be as short as two bytes long. For instance, consider an input device A which provides 32 bits of information at a time. It places the channel in a burst mode long enough to transmit 4 bytes, then disconnects from the channel. The channel is then returned to normal multiplex operations. Use of the short burst mode increases the channel time efficiency since only one sub-channel setup and disconnect cycle (10 microseconds) is needed; four such cycles would be needed if the bytes from device A were transmitted in the multiplex mode.

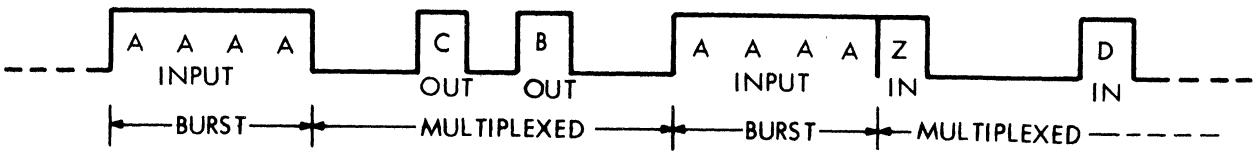


Figure 6. Mixed Mode

COMPUTER PACKAGING

The basic computer module is a pluggable electronic subassembly called a page. It consists of two multilayer printed circuit boards bonded to a metal frame. Electronic components are attached to the outer surface of each board. The computer circuits are composed mainly of monolithic integrated circuits, although conventional discrete components are used for some special applications.

Insulators separate the boards from the page frame. Two 98-pin connectors are fastened along the lower edge of the frame. The connector was developed for use in the IBM Saturn V Guidance Computer and has demonstrated its performance and reliability in extensive tests and field service. It has been qualified for use in man-rated space vehicles. Feed-through connections and test points are provided along the upper edge of the frame. The page is cooled by thermal conduction through the frame and mounting flange thermal interface. Guide pins project from the lower edge of the page to prevent mis-location during installation. Jack screws on the mounting flanges permit installation and removal without special tools.

The multilayer printed circuit boards are made of several layers of etched copper clad epoxy-glass laminates which are bonded together under heat and pressure. Connections between conductor layers are made through plated holes. These boards have been in volume production since 1962 and have been successfully used in many systems produced by IBM including the Titan II, Titan III, Gemini, Saturn I and Saturn V Guidance Computers.

The integrated circuit flatpacks are soldered to etched patterns on the surface of the multilayer printed circuit boards. These multilayer boards allow the flatpacks to be closely spaced. Figure 7 shows one side of a typical page with flatpacks mounted. Each page contains two multilayer boards with up to 78 flatpacks per board. Discrete components are soldered in holes or on the board surface, depending on terminal configuration. A conformal coating is applied for component support and environmental protection.

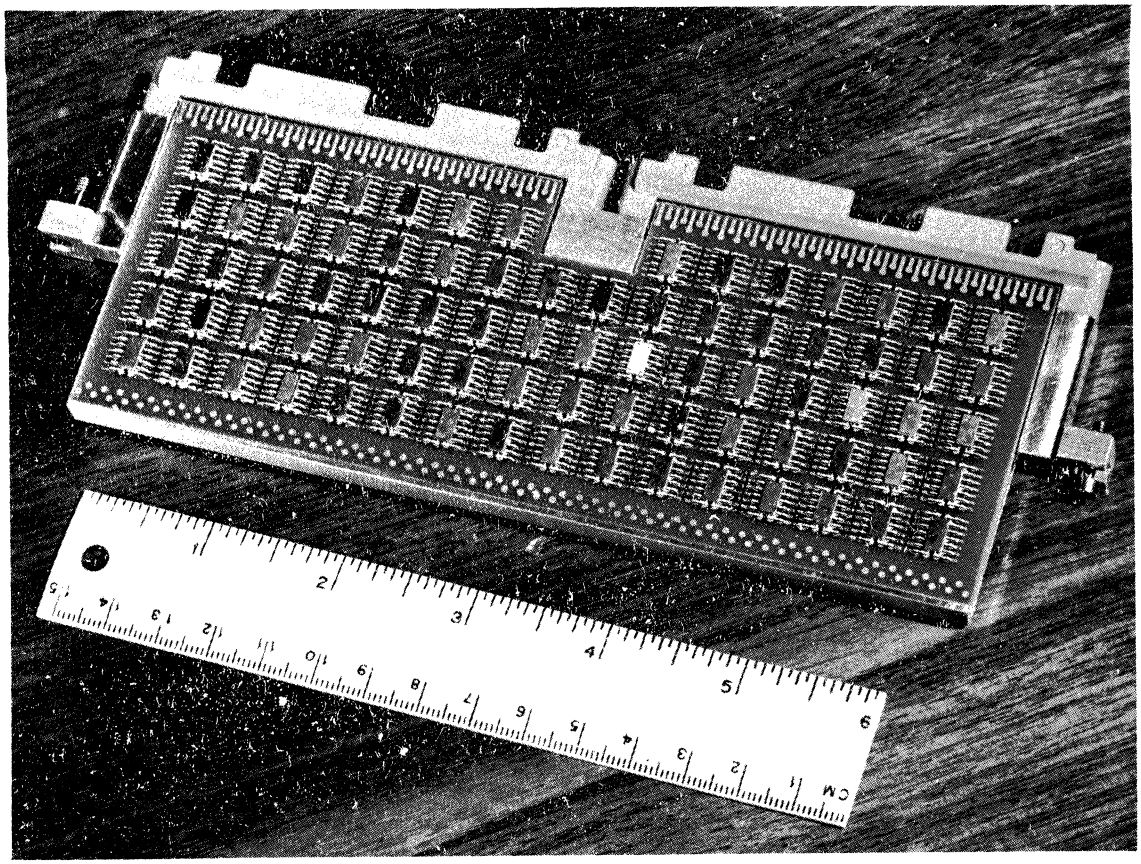


Figure 7. Pluggable Electronic Subassembly

Pluggable subassemblies similar to those in the processor are used to mount the storage electronic circuits. The storage array is removable as a module and is connected to the storage circuits distribution panel with pluggable connectors.

The ROS also has pluggable subassemblies similar to main storage.

Power supplies for the Model EP consist of pluggable modules. The RFI filter and main power transformer constitute one module, and the voltage regulators form the other module.

SOFTWARE SUPPORT

Computer programs are supplied with the Model EP computer to enable system programmers to transform their problem definitions efficiently and rapidly into computer programs. Included in the software package are:

- Symbolic Assembler
- Compiler
- Mathematical Subroutines
- Utility Programs
- Diagnostic Programs

The symbolic assembler converts an assembler language program into a machine code program. The assembler language is a set of mnemonic symbols which represent machine operation codes and operations to be performed by the assembler program. The language is augmented by other symbols, supplied by the programmer, that are used to represent storage addresses and data. The assembler provides auxiliary functions that assist the programmer in checking and documenting programs, in controlling address assignment, in data and symbol definition, in generating macro instructions, and in controlling the assembler itself.

Decimal, binary, hexadecimal, or character representation of machine language binary values may be employed by the programmer in writing source statements.

The Model EP addressing structure requires the designation of a base register and a displacement value in specifying a storage location. The assembler assumes the clerical burden of calculating storage addresses in these terms for the symbolic addresses used by the programmer. The programmer retains control of the base register usage and the values entered therein, thereby providing a relocatable feature.

The language also provides for partial control of the assembler output by the programmer; he may identify the assembly listing and designate how much detail is to be included in the assembly listing. The source program is analyzed for errors and detected errors are indicated in the assembly listing. The resultant object program is then processed by the relocatable loader and executed on the Model EP computer.

The Model EP compiler converts source language programs into machine language compatible with the real-time operation of the computer. The source language is PL/I. The object program is produced in relocatable code. Complete control of output formatting is provided. Source language error detection is performed and diagnostic messages provided to simplify debugging. The compiler provides Boolean algebra as well as conventional arithmetic statement capabilities.

Some trigonometric subroutines have been programmed in detail to provide storage and execution time for use in Model EP application programs. The functions which have been completed are sine, cosine, tangent, and arctangent. The routines have all been coded as closed subroutines and make extensive use of the general purpose registers. If all used registers are saved, the routines are re-entrant; only one copy of each need be in storage for all usage.

A library of utility programs are provided with this computer. Included will be a relocatable loader, input data conversion, and storage dump.

Two types of diagnostic programs are provided, functional test programs and fault location programs. The functional test programs are designed to exercise every possible computer circuit in a worst-case binary pattern using each computer instruction. Various parts of the computer are exercised in a progressive manner. Fault location programs are used in conjunction with the computer support equipment to isolate failures.

INSTRUCTION LIST

A list of the Model EP basic instructions and execution times is included in Table 2.

TABLE 2
INSTRUCTION LIST

(Fixed Point)

Instruction	Format	Execution Time (usec.)
Load	RR	1.9
Load	RX	5.0
Load Halfword	RX	5.0
Load and Test	RR	1.9
Load Complement	RR	2.1
Load Positive	RR	2.1
Load Negative	RR	2.1
Add	RR	2.1
Add	RX	5.0
Add Halfword	RX	5.4
Add Logical	RR	2.1
Add Logical	RX	5.0
Subtract	RR	2.1
Subtract	RX	5.0
Subtract Halfword	RX	5.4
Subtract Logical	RR	2.1
Subtract Logical	RX	5.0
Compare	RR	2.1
Compare	RX	5.0
Compare Halfword	RX	6.0
Multiply	RR	9.2
Multiply	RX	10.4
Multiply Halfword	RX	11.6
Divide	RR	20.0
Divide	RX	20.8
Store	RX	5.0
Store Halfword	RX	5.0
Shift Left Single	RS	Variable
Shift Right Single	RS	Variable
Shift Left Double	RS	Variable
Shift Right Double	RS	Variable

(Logical)

Compare Logical	RR	1.9
Compare Logical	RX	5.0
Compare Logical	SI	5.4
And	RR	3.4
And	RX	5.0

TABLE 2 (Continued)

Instruction	Format	Execution Time (usec.)
Or	RR	3.4
Or	RX	5.0
Exclusive Or	RR	3.4
Exclusive Or	RX	5.0
Test Parity	RX	5.0
Test Under Mask	SI	5.0
Sumcheck	RX	Variable
Insert Character	RX	5.0
Store Character	RX	5.0
Load Address	RX	2.9
Shift Left Single Logical	RS	Variable
Shift Right Single Logical	RS	Variable
Shift Left Double Logical	RS	Variable
Shift Right Double Logical	RS	Variable
	(Branching)	
Branch on Condition	RR	4.2
Branch on Condition	RX	4.4
Branch and Link	RR	4.0
Branch and Link	RX	4.1
Branch on Count	RR	4.2
Branch on Count	RX	4.4
	(Status Switching)	
Load PSW	SI	7.5
Load PSW Special	SI	9.0
Set Program Mask	RR	2.1
Set System Mask	RR	5.0
Change Priority Mask	SI	5.4
Supervisor Call	RR	15.0
Set Storage Key	RR	4.5
Insert Storage Key	RR	5.0
	(I/O)	
Start I/O	SI	Variable
Test I/O	SI	Variable
Halt I/O	SI	Variable
Test Channel	SI	Variable
Read Direct	SI	Variable
Write Direct	SI	Variable