EAI®
ELECTRONIC ASSOCIATES, INC.

640 *DIGITAL COMPUTING SYSTEM INFORMATION MANUAL*

CONTENTS

section

1

EAI 640 SYSTEM

EAI 640 Digital Computing System

INTRODUCTION

The EAI 640 Digital Computing System provides the user with an outstanding, general purpose, computing capability for handling a wide range of scientific applications. It includes flexible input/output and interrupt features that make it particularly useful in combined hybrid and special systems for simulation, hybrid computation, on-line monitoring/control and other uses.

The EAI 640 is a parallel, binary computer that operates with a fixed length 16 bit instruction and data word. It has a protected core memory with a maximum storage capacity of 32,768 words. Each word in memory has a protect bit and violations are immediately recognized through interrupt procedures in the computer.

The EAI 640 offers an extensive list of instructions, multi-level interrupt capability and high speed input/output operations for communication with up to 64 peripheral devices.

The EAI 640 System is of modular design. Memory can be expanded from the basic 8K to 16 or 32 K. The basic EAI 640 contains interfaces for connection to interval timers. A Direct Memory Access Channel can be added to allow high speed input/output interleaved with computation on a cycle stealing basis. Advanced integrated circuitry makes the EAI 640 a member of the third generation computer family combining reliability with high speed logic.

640 SYSTEM SUMMARY

.... Stored program, general purpose digital computer
.... Monolithic integrated circuitry
.... 64 Instructions including:

- Add/Subtract
- Multiple precision hardware features for Add/Subtract
- Multiply/Divide
- Square Root
- Logical
- Inter-Register Exchanges
- Condition Tests

.... Fixed word length of 16 data or instruction bits

.... 16-bit Storage Word - plus a memory Protect bit

.... Magnetic Core Storage

1.65 Microseconds cycle time

Basic Size 8,192 words

Expandable to 16,384 or 32,768 words

.... Multi-level Indirect Addressing

.... Two's Complement Binary Arithmetic

.... Index Registers

One Hardware register

An indefinite number of memory words can be used as index counters.

.... 7 Internal Interrupts and 64 External Interrupts

.... Input/Output

Single Word Mode

Record Mode

Direct Memory Access Channel (Optional)

.... Four Interval Timer Interfaces

CHARACTERISTICS OF THE BASIC 640 COMPUTER

Arithmetic Operations

The EAI 640 performs all arithmetic operations using two's complement binary arithmetic.

ADD/SUBTRACT OPERATIONS

Addition and Subtraction can be performed in either single or multiple precision format. When the multiple precision bit is on, the carry/borrow bit automatically provides a carry or a borrow when operating on the next higher precision word. When a multiple precision bit is off, single precision arithmetic is performed.

MULTIPLY/DIVIDE OPERATIONS

A Multiplicand contained in the Accumulator is multiplied by the multiplier as specified by the effective address. The resulting product is double length and is contained in the Accumulator and the Accumulator Extension (Q Register). The most significant portion (with sign) is in the

Accumulator, the least significant portion is in the Q register. In the Divide operation, the dividend contained in the Accumulator and the Accumulator Extension is divided by the divisor contained in the memory location specified by the effective address. After division, the quotient with sign is contained in the Accumulator and the remainder is in the Accumulator Extension.

Basic Computer

The basic EAI 640 Computer System is available with 8,192 words of core storage. The basic 640 Computer with 8K of memory and an upright console is the 640/008 and the same computer with a desk mounted operator console is the 640/018.

MEMORY EXPANSIONS

The 640/008 and 640/018 can be expanded to 16K in the field with the 640/208 Memory Conversion Unit. In addition, the 640/216 allows field modification of a 16K computer system to the full 32K version.

Memory Data and Address

A 16-bit Memory Data Register (M) holds data written into, or read out of core memory. The Memory Address Register (S) holds the address specifying where a word is to be read from or written into core storage.

Indexing Operations

HARDWARE INDEX REGISTER

The EAI 640 has an Index Register (X) for high speed address modification without increasing instruction execution time. The X Register includes 15 address bits and a post-index indirect bit. This post-index indirect bit specifies whether the calculated address is to be the effective address or will result in a further indirect address. The Index Register can be transferred directly to or from memory without affecting the Accumulator. Increment or decrement of the Index Register followed by a Skip is performed in one instruction.

MEMORY COUNTERS

The EAI 640 allows counting operations with the content of any memory location. With one instruction, a memory location can be incremented and tested. This capability allows the programmer an unlimited number of counters which can be serviced and tested with one instruction each.

Memory Protection

Each core memory location in the EAI 640 Computer has a protect bit. This bit can be set or reset under program control only if the protect switch on the console is in the ON position. Violations of the protect system generate an interrupt and the contents of the memory location involved in the violation are not changed.

I/O System

The EAI 640 Computer can handle input and output in two modes. The single word mode transfers one word at a time into or out of the Accumulator. In the Record Mode, consecutive memory locations are automatically accessed for input or output. Block transfers can be made easily by specifying a starting and the final word address. After completion of the record transfer, a terminating address is available which can be examined against the last word address which has been specified. The extensive interrupt capabilities of the 640 allows I/O operations to proceed without constant need for status testing of the peripheral devices.

Bootstrap Operation

After clearing the computer (setting the hardware registers and memory to zero), the Execute Run switch will cause a bootstrap program to be read from the teletype paper tape reader which, after loading itself, will turn control over to the program which was loaded. This hardware feature allows the user to get off from a dead start with a minimum of inconvenience and loss of time.

Direct Memory Access Channel (640/310)

The basic EAI 640 Computer can expand its input/output capabilities through the addition of an optional Direct Memory Access Channel. Direct memory access operates on a cycle stealing basis which allows computation to proceed simultaneously with buffered input/output operations.

The EAI 640 Computer performs all calculations and processes data in a parallel, binary mode through the execution of individual instructions. Both instructions and data are stored in the magnetic core storage of the 640 Computer system. The 640 Computer is divided functionally into three sections, arithmetic, control, and memory.

ARITHMETIC SECTION

The Arithmetic Section of the EAI 640 Computer performs the arithmetic and logical operations necessary for execution of instructions. This section has several operational registers. Figure 1 shows a generalized logic block diagram of the 640 Computer. It should be emphasized that not all registers shown on this diagram are available to the programmer for direct operations. For example, the Half-Add Register (H) is necessary for proper operation of the Arithmetic Section, but cannot be directly manipulated by any instruction in the repertoire of the 640 Computer.

Accumulator (A Register):

The Accumulator is a 16-bit register used in arithmetic, logical and input/output operations.

Accumulator Extension (Q Register):

The Accumulator Extension is a 16-bit register used in double shifting operations, multiply, divide, square root and record mode input/output operations.

Arithmetic Register (O Register):

The Arithmetic Register is a 16-bit register used in the execution of certain instructions. It is not available to the programmer, except for console display and manipulation.

Half-Adder (H Register):

This 16-bit register is used in the execution of certain instructions but is not available to the programmer for direct manipulation or display.

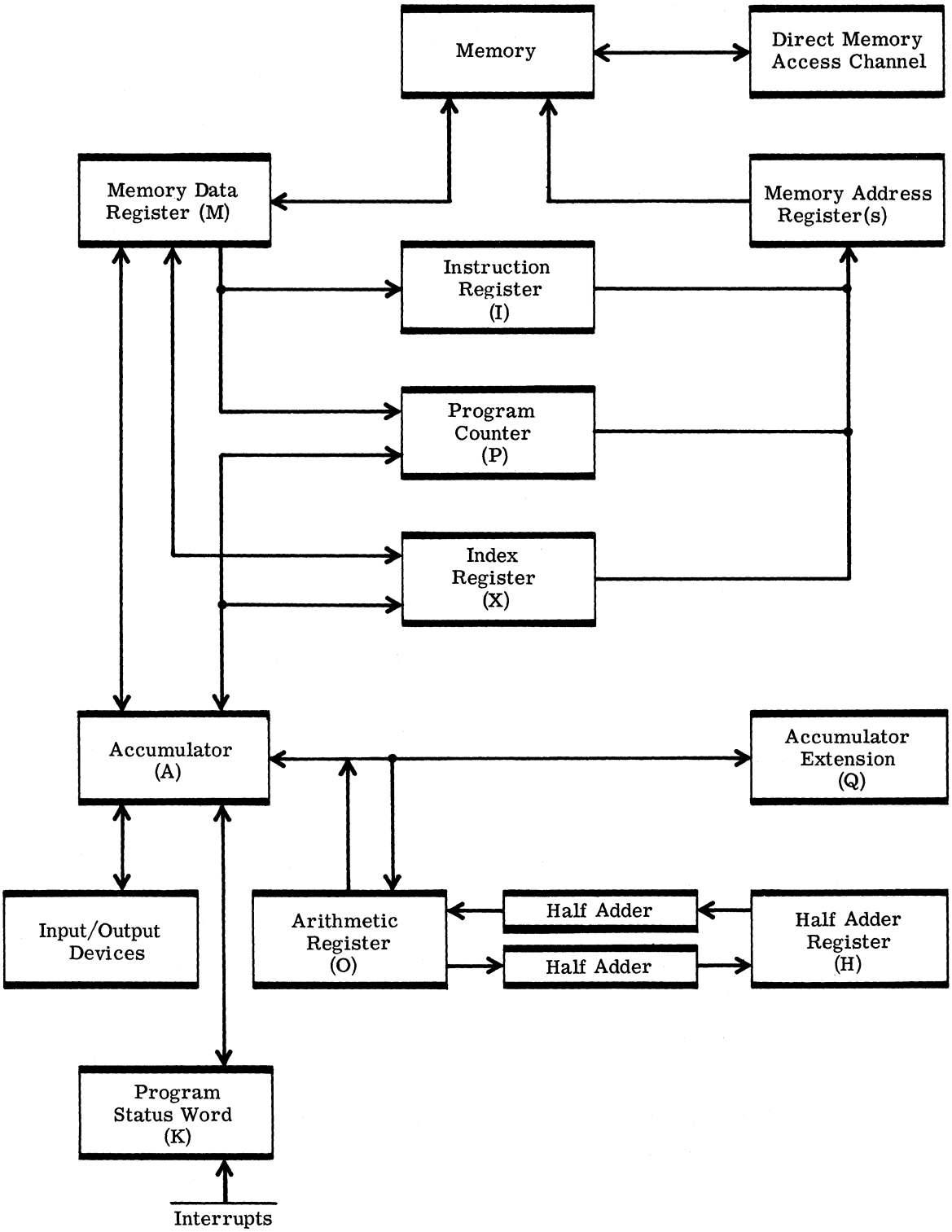


Figure 1. Block Diagram - EAI 640 Computer

Program Counter (P Register):

The Program Counter is a 15-bit register containing the memory address of the next instruction to be executed. After the instruction to be executed is placed in the Instruction Register the P register is incremented by one. The P register may be further changed by the instruction being executed. The P register has 15 bits and can generate addresses from 00000 to 32,767 which is the maximum memory capacity of the 640 Computer.

Index Register (X Register):

This 16-bit register is used in indexing operations. The index value can be positive or negative in two's complement form and contains an Indirect Addressing bit position.

Instruction Register (I Register):

The Instruction Register is a 17-bit register which contains the instruction currently being executed and its program protect bit. The register is not directly available to the programmer, except for console display and manipulation.

Program Status Word (K Register):

The Program Status Word is a 16-bit register which contains the enable/disable bits for each interrupt level, the condition codes, the multiple precision and carry/borrow bits.

CONTROL SECTION

The control section of the EAI 640 Computer directs the operations required to execute instructions and establishes the timing relationships needed to perform these operations in their proper sequence. It also controls memory usage, interrupt processing, program protection, and input/output operations. The control section acquires an instruction from storage, interprets it, and sends the necessary commands to other sections. The Program Counter (P) provides program continuity by generating in sequence the storage addresses which contain the individual instructions.

MEMORY SECTION

The Memory Section of the EAI 640 Computer consists of two registers and the Storage Modules. The basic computer may have 8,192 words of high-speed, random access, 2-1/2 D, magnetic core storage which can be expanded to 16,384 or 32,768 words. Storage cycle time is 1.65 microseconds which is defined as the shortest possible time between successive read/write operations in core memory.

Memory Data Register (M Register):

The Memory Data Register is a 17-bit register from which information is written into memory and to which information is read from memory (16 information bits and a protect bit).

Memory Address Register (S Register):

The Memory Address Register is a 15-bit register which contains the address in memory from which information is to be read or into which information is to be written.

STORAGE WORD

A Storage Word may be a 16-bit instruction, a 16-bit operand, one-half of a 31-bit operand (divide or square root) or a 16-bit portion of a multiple precision operand. A Program Protect bit is appended to each 16-bit storage word, thus a storage word is 17 bits long. If the Protect Bit is a "1", it indicates that the word is part of a protected program.

STORAGE ADDRESSING

The location of each word in storage is identified by an assigned number (address). An address consists of 15 bits. Instructions, which reference core memory, have three fields: 4-bit Operation Code, 3-bit Address Mode (E field), and a 9-bit Displacement Field. As a matter of convenience, an instruction may be expressed as a six digit octal number, where the first digit may only be a 0 or 1. The first two octal digits will be the operation code, the third will be the E field and the last three digits the displacement address.

The terms used to describe memory addressing methods are:

EA - effective address: the destination location containing the data which is to be used for computation.

IA - indirect address: a location containing an address which indicates where the intended operand may be found. Multi-level indirect addressing is defined as a series of memory locations containing indirect addresses which are referenced in sequence by the computer.

D Field - displacement: bits 7 to 15 of a memory reference instruction.

Treated as either:

(a) a 9-bit positive number in the range of 0 to 511;

(b) an 8-bit signed number in the range of -256 to +255.

The value in the D Field *always* participates in address computation.

E Field: bits 4, 5 and 6 of a memory reference instruction.

Bit 4 - indicates indirect addressing will take place (I).

Bit 5 - indicates index register addressing will take place (X).

Bit 6 - indicates relative-to-program-counter addressing will take place (P).

I: Indirect addressing will take place followed (optionally) by Indexing and further Indirect Addressing.

X: the contents of the index register. The most significant bit (zero) determines whether the resulting address is the effective address or is to be interpreted as a further indirect address. If bit 0 of the index register is "0" the resulting address is the effective address.

P: the content of the program counter register which is the location of the current instruction.

ADDRESSING CONCEPT OF THE 640 COMPUTER

To understand the addressing concept of this computer, it is necessary to consider the memory of the computer in relation to the bits available in an instruction for addressing. The Displacement field, when treated as a 9-bit positive number, yields a range of 0 to 511 addresses. Therefore, the "Zone Zero" area from location 00000 to 00777 (octal) becomes available to the programmer directly.

The method of addressing these low core locations is commonly referred to as Absolute Addressing. It often is convenient to store counters or to provide temporary storage for intermediate results in this low core area because of the convenience of addressing through the displacement portion of the instruction.

The other method of addressing is commonly called "Relative" because it references locations which are relative to the current contents of the Program Counter (P Register) or the Index Register (X). It was stated earlier that the displacement portion of the instruction can also be an 8-bit signed number which gives a range of -256 to +255. This method allows a program to reference core forward or backward from its current P location and/or a value in the Index (X) Register.

Both the Absolute and Relative form of addressing by themselves would not offer the flexibility of programming desired. For this reason, the 640 provides indirect addressing. Through the use of the program counter register, the index register, indirect addressing, and any combination thereof, the programmer has available any location in core storage for addressing.

Memory Address Mode Table

Address Mode	E Field	Initial Address	Indirect Addressing Follow?	Post Index Indirect Addressing?
Absolute	E is 0	D (Zero Zone)	No	No
Absolute Indirect	E is 4		Yes	No
Absolute Indirect Indexed	E is 6		Yes	Yes
Relative	E is 1	(P) ± D	No	No
Relative Indexed Direct	E is 3	(P + X) ± D	No	No
Relative Indexed Indirect	E is 3		Yes	No
Relative Indirect	E is 5	(P) ± D	Yes	No
Relative Indirect Indexed	E is 7		Yes	Yes
Indexed Relative Direct	E is 2	(X) ± D	No	No
Indexed Relative Indirect	E is 2		Yes	No

GENERAL DESCRIPTION

The Interrupt System of the EAI 640 Computer allows the programmer a choice of interrupt priorities and actions. As the name implies, an Interrupt forces the computer to temporarily abandon its processing of the main program to service an anticipated condition.

Interrupts are signals which are generated internally or externally to the computer by certain conditions. An internal interrupt, for example, is caused by a memory protect violation. An example of an external interrupt would be the completion of an output through a peripheral piece of equipment causing an interrupt on that particular channel. The 640 Computer has a Master Interrupt Bit which enables or disables the interrupt system of the machine. The programmer can set or reset the Master Interrupt Bit. The exceptions to this interrupt procedure are the power failure interrupt and the interrupt test instructions which are programmed interrupts and will be processed regardless of the setting of the Master Interrupt Bit (Figure 2).

The Program Status Word (K Register) contains the enable/disable bits for each interrupt level. The programmer can "mask" the interrupts he desires by setting or resetting the bits in the K Register. Priority of interrupts is on the basis of bit position in the K Register. The PSW has the highest priority interrupts in the most significant bit positions.

The ability to mask interrupts through the Program Status Word allows the programmer to handle more than one interrupt. An interrupt automatically forces the computer to store the content of the Program Counter in the memory address specified by the Pointer. The content of the Accumulator is stored in the next memory location. The interrupt acknowledge procedure requires 5.8 microseconds. The first instruction executed to process the interrupt condition is located at the pointer address plus two. The Pointer itself is a fixed memory location assigned to each particular condition permitted to cause an interrupt and to each of the 64 possible external devices.

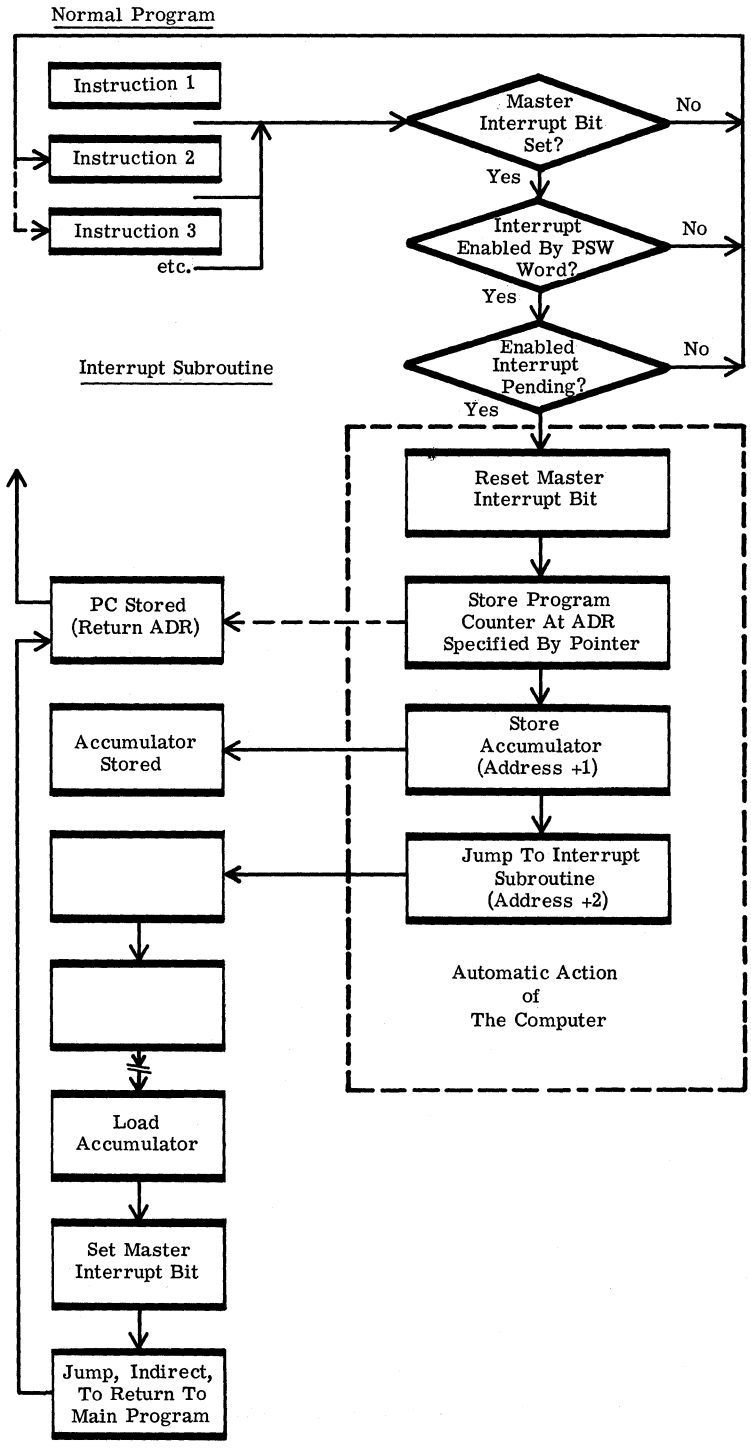


Figure 2. Diagram of 640 Interrupt System

MASTER INTERRUPT

The Master Interrupt Bit controls operation of the interrupt system. If the bit is set interrupts are recognized as specified by bits set in the PSW. Two instructions control setting and resetting of the Master Interrupt Bit.

The Master Interrupt is turned off (reset) upon recognition of an interrupt. The Master Interrupt can be set again as soon as the mask in the PSW has been modified to take into consideration the processing of the interrupt presently under consideration. The last instructions of the interrupt subroutine would normally be to set the Master Interrupt Bit followed by an indirect jump to the previously stored program counter.

PROGRAM STATUS WORD (K Register)

The K Register has 16 bits of which bit 0 through 10 are used to enable or disable corresponding levels of interrupts. If a bit in the PSW is a "1", the corresponding interrupt is enabled. If the bit is a "0", the interrupt is masked (disabled). Bit 12 is the Carry/Borrow bit, bit 13 is the Multiple Precision bit, bits 14 and 15 are condition codes which are set by arithmetic operations, and bit 11 is vacant.

The Exchange Accumulator and PSW instruction permits the programmer to store; modify it, or replace the present status word.

PRESERVATION OF REGISTERS DURING INTERRUPTS

On recognition of an interrupt, the Program Counter Register content is stored at the address designated by the Pointer and the content of the Accumulator is stored in the pointer address plus 1. The only exception to this is the TRAP instruction which modifies the Accumulator before it is stored. In the case of the TRAP, bits 8 through 15 of the Accumulator are set to equal the same bits of the instruction. The instruction will set bits 0 through 7 of the Accumulator to zero. Trap instructions can thus be numbered up to 256 (377_8).

PRIORITY

Priority is established by bit position in the PSW, with the PSW bit zero having the highest priority. A further priority assignment is made in the case of peripheral equipment by the position of the device on the standard data channel or the buffered data channel.

INTERNAL INTERRUPTS

Internally generated interrupts have the highest priority. Internal interrupts are ignored and do not remain pending if the Master Interrupt Bit is reset or bits 0 through 4 of the PSW are reset. The single exception to this is the interval timer interrupt (bit 5 of PSW) which will remain pending until serviced.

The Internal Class of interrupts consists of:

Bit	
0	Trap Instruction
1	Illegal Instruction
2	Illegal Procedure in Setting/Resetting Protect Bit
3	Memory Protect Violation
4	Machine Check and I/O Parity
5	Interval Timer (Up to Four, Optional)

EXTERNAL INTERRUPTS

There may be up to 64 peripheral devices, each with its own unique interrupt service, on the I/O data channel. Data channel class interrupts are divided into four groups (Data Channel Interrupt 0 through 3). These interrupts will remain pending even though the Master Interrupt Bit is reset or bits 6 through 10 of the PSW are reset. Priority among Data Interrupt Channels is established by their bit position in the PSW

The external class of interrupts consists of:

Bit	
6	Data Interrupt Channel 0
7	Data Interrupt Channel 1
8	Data Interrupt Channel 2
9	Data Interrupt Channel 3
10	Alarm Channel Interrupt

On each of the four Data Interrupt Channels, an additional ordering of priority is provided for peripheral equipment by the position of the 16 possible devices on each interrupt line. On each interrupt line, that device which is connected nearest to the computer is given the highest priority on that Data Interrupt Channel. This provides for future re-ordering of priorities and adding additional peripheral devices.

ABSOLUTE PRIORITY

The interrupt test instructions and the power failure interrupt have absolute priority in that neither the PSW nor the Master Interrupt can prevent them from occurring. The interrupt test instructions are programmed interrupts placed at pre-determined locations in a program and thus will not interfere with normal processing.

POINTERS

The fixed assignment of a core memory location which is made for each interrupt is designated as Pointer. The pointer registers must be loaded by the program with an address to which the computer will automatically go upon recognition of an interrupt.

DESCRIPTION OF INTERRUPTS

Power Failure Interrupt

The absolute priority interrupt is given to the power failure condition. If the computer should sense line voltages above 135 volts or below 100 volts for more than 100 milliseconds, this interrupt will be generated and the I/O instruction in progress will be immediately interrupted. The content of the Program Counter Register and Accumulator will be automatically stored. There is sufficient time (2 milliseconds minimum) before computer operation stops for the interrupt routine to retrieve and store the content of the three additional hardware registers.

Trap Instruction

The Trap instruction modifies Accumulator bits 8 through 15 and sets bits 0 through 7 to zero. Bits 8 through 15 can be used to code the instruction from 0 to 256_{10}. The Trap instruction is considered a program generated interrupt.

Illegal Instruction

This interrupt occurs when the control section of the 640 Computer cannot decode the operation code of an instruction as a legal bit combination.

Illegal Procedure in Setting/Resetting Protect Bit

An interrupt is generated when the Memory Protect Switch on the console is in the "OFF" position and the program attempts to set or reset a protect bit. Neither instruction will be executed before the interrupt occurs.

Memory Protect Violation

A program in an unprotected area of core memory has attempted to alter a core location which has the protect bit set. An interrupt occurs and no storage alteration will take place.

Machine Check or I/O Parity

This interrupt occurs on a parity failure in data transmission from a peripheral device or a device fails to respond properly.

Data Interrupt Channel 0 through 3

A data channel interrupt is generated by one of the peripheral devices on the data channel.

Alarm Channel

The interrupt on Alarm is generated by a peripheral device which is attached to the Alarm Channel. Only special features of the device are recognized as alarm conditions and are signaled through this channel. Generally, it can be stated that an alarm condition is one that would defeat the operation of the peripheral device unless attended to promptly.

EAI 640 MEMORY PROTECT SYSTEM

The program protection system of the EAI 640 Computer makes it possible to prevent storage locations in core memory from being modified by a non-protected program. The system is based on the presence of a "protect bit" for each storage location in core memory. Operation of the protect system involves a switch on the control panel console, two instructions from the repertoire, the contents of the Index Register (X) and the Accumulator. Violations of memory protection are signaled through the interrupt system.

The Memory Protect Bit can be set or reset only if the "Memory Protect" switch on the control panel console is in the ON position at the time the instruction is executed. The least significant 15 bits of the Index Register (X) added into the contents of the Accumulator specify the address of the core memory location which is to have the protect bit set or reset.

If the "Memory Protect" switch on the control panel is in the OFF position when a set or reset protect bit instruction is attempted, an Illegal Procedure interrupt will be generated. The protect bit of the specified memory location will not be changed.

A protected instruction (one which has its protect bit set) may reference another protected instruction or data operand. The protected instruction is therefore privileged. An unprotected instruction may not change a protected core memory location. An attempt to do so will result in a Memory Protect Violation interrupt.

INTRODUCTION

The EAI 640 Computer can handle input/output in three modes:

---- one word at a time through the Accumulator
---- a complete record through the Accumulator and Q Registers directly to or from memory
---- a buffered record directly to memory (Optional feature)

The standard I/O data channel handles transfers in Single Word Mode or Record Mode. The single word mode and the record mode stop computation because the Accumulator is used for transfers. The difference between single word mode and record mode lies in the convenience of programming. The former requires a loading or storing of the Accumulator for each one word transfer. The latter runs to completion once the program has specified the beginning and end of the memory area to be input or output. An optional buffered I/O channel provides for Direct Memory Access. The buffered data channel operates on a "cycle stealing" basis permitting computation to continue while I/O operations proceed independently. The EAI 640 can operate both the buffered data channel and the standard data channel simultaneously.

Operation of peripheral devices is initiated through Device Function instructions. The data path is established and the equipment instructed on the action to be taken. Through the Status Input instruction the peripheral equipment can inform the computer about existing conditions. Interrupts are used to signal termination of I/O either normally, or through some abnormality such as a parity or protect violation.

Should the interrupt system be disabled by the program, either through the masking of the interrupt or the resetting of the Master Interrupt, the peripheral device can be tested for an interrupt through the Test Device Interrupt instruction.

SINGLE WORD MODE INPUT/OUTPUT

The two instructions, Data In and Data Out, cause a 16-bit parallel word transfer between the Accumulator and a specified device on the standard data channel.

These two instructions, together with store or load Accumulator and skip on index permit successive word transfers from or to all core memory locations. The single word input/output mode is generally used in communications with low speed devices. This mode of input/output permits many instructions to be executed between data transfers.

RECORD MODE INPUT/OUTPUT

The two instructions, Record In and Record Out, cause 16-bit parallel word transfer between core memory and a specified device on the standard data channel. When operating in the Record Mode, the running program is halted and the processor devotes full time to information transfer from or to a peripheral device. This mode of input/output is generally used in communication with devices having a high transfer rate.

Record Packets

The Record Mode instructions operate under the control of four input/output packets each consisting of four assigned core memory locations. The packets are numbered zero through three. The four core memory locations assigned for each packet are used as follows:

Device Function Word
Final Record Address plus 1
Starting Record Address
I/O Termination Address

Before execution of a Record Mode instruction, the program must initialize the Device Function Word, the Final Record Address, and the Starting Record Address words. When a Record Mode instruction is executed, the Device Function Word is sent to the specified device and the desired mode of operation begins. Data is then transferred from or to the Starting Record Address location and continues a word at a time until the Final Record Address is reached unless the device terminates transmission of data. The Terminating Address of the last word transmitted is then stored in the last word of the packet.

The length of a record may range from one to 32K words; a record length of zero is illegal. If the actual record length is shorter than called for, the Terminating

Address will be less than the Final Address. If the actual record length is equal or longer than called for, the Terminating Address will be the same as the Final Address. To sense a longer length record, the program must sample the Device Status Word.

DIRECT MEMORY ACCESS CHANNEL (DMAC)

General

The Direct Memory Access Channel (640/310) provides a direct data path between various peripheral equipment and the memory of the 640 Computer. As a logical device the DMAC will control the input/output of data, the core memory locations accessed, the record length as well as check for correct transmission parity, violation of memory protection and control over device interrupts. The DMAC performs these functions independent of program control.

Functional Description

The DMAC operates on a cycle-stealing basis, stealing memory cycles from normal program execution when input/output data is ready for transfer to or from memory. Data is held in the DMAC's data register and the memory address in an address register while the DMAC steals a memory cycle. Instruction execution in the computer is suspended during a cycle steal but resumes automatically when the steal cycle is complete. The address register is incremented at the end of each steal and compared with a second address register in the DMAC containing the final address of the record. When the two address registers compare equal the DMAC becomes inactive and an interrupt is generated.

The DMAC will steal a cycle if the computer has not already entered into a memory read cycle. The data path through the DMAC is a full 16-bit word and is transferred in parallel. While the DMAC is active, the computer may access its current address register to determine how much progress has been made on the data transfer.

Initialization

The DMAC is initialized by the execution of a Buffered Record Input or a Buffered Record Output instruction. These instructions will load starting and final ad-

dresses into the DMAC and set it in motion as specified by the Device Function word. The four core memory locations of record packets 00 through 03 are assigned as follows for the Buffered Record Mode:

Device Function Word
Final Record Address
Starting Record Address
Not Used

Buffered Record Transfer

The Buffered Record Mode instructions cause an autonomous record transfer. The transfer operates in a cycle-stealing mode interleaving high speed input/output operations with computation in the central processor. Transfer is initiated by the computer and is terminated by the DMAC. Record transfer termination is accomplished when a full record is transferred as defined by the Starting and Final Addresses.

Peripheral Equipment

All operating functions of a particular peripheral device work normally on the DMA channel. The only apparent difference between systems with and without the DMA channel is the overall decrease in program running time due to increased efficiency of input/output housekeeping operations.

Only one DMAC peripheral device can be active at any one time but may be either input or output. Standard I/O operations do not interfere with an active DMAC. The DMAC will continue to transmit data when the computer is in PAUSE condition.

No more than one DMAC can be installed in a 640 System. The maximum number of peripheral devices has not doubled with the addition of the DMAC. The total remains at 64, with the further requirement being that no device on the standard channel may have the same device address as a device on the DMA channel. A device may not be connected to both channels at once. A requirement of the DMAC is that the interrupt line on the DMA channel be separate from the standard device channels.

Interrupts

If the DMAC option is installed, data interrupt channel 2 is assigned to those devices attached to the DMAC I/O Interface, while interrupt channels 0, 1, 3, and Alarm remain assigned to devices on the standard channel.

An interrupt on the DMAC is processed in the same way as a standard interrupt, and at the end of the next instruction if the interrupt is not masked, the computer replies with an interrupt acknowledge. The first device on the DMA device channel that has an interrupt pending sends back its device address, from which the computer derives the pointer address. Important to note here is that the interrupt may have originated in a device other than one with which the DMAC is transferring data, and that action on the interrupt will remain pending until the DMAC goes inactive.

Parity

Like the standard channel, the DMA channel allows for the transmission and checking of parity along with the data. The DMAC generates parity on output and the device may or may not check it according to its design. The device, if it is so designed, will generate parity on input and the DMAC will check it. Parity checking is controlled by the parity inhibit line. An interrupt on parity error from the DMAC is the result of a parity error on input data. An interrupt on parity error from a device controller would be the result of a parity error on output data. There is no parity transmitted between the DMAC and the computer, nor between the DMAC and memory.

Memory Protect

Buffered Record Mode Instructions may be flagged as privileged, and upon execution, a flag is set in the DMAC. If so, any input data is stored into memory without violating memory protection. If the instruction is not privileged, an attempt to store data into a protected cell will result in a protect violation. This situation is detected by the DMAC and an interrupt occurs if the interrupt on protect violation was selected. The content of the protected cell is preserved and the input data that would have been stored there is lost; the DMAC will not allow the protect bit itself to be altered, whether the instruction is privileged or not.

Data Rates

The DMAC will transfer data at a maximum rate of 600,000 words per second. For any free running device to operate at, or near this data rate, sufficient buffering must be provided within the device itself to cover a missed memory cycle. The DMAC differs from the Record Mode of computer input/output in this respect. Word transfer during Record Mode may be delayed a half cycle (825 nsec) while transfer through the DMAC may be delayed a full cycle (1.65 μsec). Ideally, the DMA channel is best suited for those medium speed devices that would otherwise have a high interrupt priority on the standard channel. Interrupt after record transfer is normally a lower priority than an interrupt on data.

EAI 640 INSTRUCTION REPERTOIRE

The EAI 640 Computer has a flexible repertoire of 64 hard-wired instructions. These can be grouped conveniently into twelve functional categories:

Transfer instructions which communicate between core memory and the accumulator or index register. Transfer between memory and the index register does not affect the contents of the accumulator.

Arithmetic instructions include full word add and subtract; double word result after multiply, divide and square root; increment accumulator, clear accumulator, clear and add one, and two's complement accumulator. Any core memory register can become a counter by use of the Add One to Memory and Skip instruction.

Logical instructions include OR, XOR, AND, Compare and One's Complement Accumulator.

Shift instructions operate on the accumulator alone or across the combined accumulator and accumulator extension register. Shifts are right or left, single or double, arithmetic or logical.

Control instructions set or reset the multiple precision bit or set the sign of the accumulator positive or negative. Pause, Trap and No Operation are included in this group.

Interrupt instructions set or reset the Master Interrupt Bit to permit or inhibit interrupts according to the structure of the Program Status Word (K register).

Program Protect instructions set or reset the program protect bit of the core memory location specified by the index register if the console memory protect switch is ON.

Exchange instructions interchange the content of the accumulator with the accumulator extension register, the index register, the program counter, or the program status word.

Jump and Skip instructions include jump with link to provide a stored program counter register for subroutine entry, and skip on sense switch or multiple sense switches. Unconditional jump and skip instructions are also included.

Skip on Register Condition instructions operate on accumulator positive or negative and accumulator or accumulator extension register even. Skip instructions also increment or decrement the index register and skip if the sign of the index register changes.

Skip on Condition Code instructions test two bit positions of the program status word. These instructions are only valid following certain arithmetic, shift, and logical operations.

Input/Output instructions are grouped into single word mode, record mode, and buffered record mode. Single word mode causes 16-bit parallel transfer of data between the accumulator and a specified device on the standard data channel. Record mode causes 16-bit parallel transfer of data directly between core memory and a specified device on the standard data channel. Record mode data transfer continues a word at a time until all data between the starting and final record addresses has been transmitted.

The buffered record mode operates in a similar manner except that transfer of data from core memory to a specified device on the buffered data channel occurs on a cycle stealing or "interleaving" basis. The Direct Memory Access Channel performs the logical input/output control functions independent of program supervision.

FUNCTION LISTING OF EAI 640 INSTRUCTIONS

Mnemonic Code	Instruction Description	Time In Microseconds
TRANSFERS		
LA	Load Accumulator	3.30
STA	Store Accumulator	3.30
LX	Load Index Register	3.30
STX	Store Index Register	3.30
ARITHMETIC		
A	Add	3.30
S	Subtract	3.30
M	Multiply	18.15
D	Divide	18.975
SQR	Square Root	16.5
AOA	Add One to Accumulator	1.65
AOM	Add One to Memory and Skip	4.95
TCA	Two's Complement Accumulator	1.65
CLR	Clear Accumulator	1.65
CAO	Clear Accumulator and Add One	1.65
LOGICAL		
OR	Or (Logical Sum)	3.30
XOR	Exclusive Or (Logical Subtract)	3.30
AND	And (Logical Product)	3.30
C	Compare	3.30
OCA	One's Complement Accumulator	1.65
SHIFTS		
ARS	Arithmetic Right Single	See Note (1) Below
ALS	Arithmetic Left Single	
ARD	Arithmetic Right Double	
ALD	Arithmetic Left Double	
LRS	Logical Right Single	
LLS	Logical Left Single	
LRD	Logical Right Double	
LLD	Logical Left Double	

Note (1): 1.65 μsec minimum for a single shift.

FUNCTION LISTING OF EAI 640 INSTRUCTIONS (Cont)

Mnemonic Code	Instruction Description	Time In Microseconds
CONTROL		
SMP	Set Multiple Precision Bit, Reset Carry/Borrow Bit	1.65
RMP	Reset Multiple Precision Bit	1.65
SSP	Set sign of Accumulator Positive	1.65
SSN	Set sign of Accumulator Negative	1.65
NOP	No operation	1.65
P	Pause	1.65
T	Trap	1.65
INTERRUPT		
SMI	Set Master Interrupt Bit	1.65
RMI	Reset Master Interrupt Bit	1.65
PROGRAM PROTECT		
SPB	Set Protect Bit	3.30
RPB	Reset Protect Bit	3.30
EXCHANGES		
EX	Exchange Accumulator and Index Register	1.65
EQ	Exchange Accumulator and Q Register	1.65
EP	Exchange Accumulator and Program Counter	1.65
ES	Exchange Accumulator and Program Status Word	1.65
JUMPS and SKIPS		
J	Jump Unconditional	1.65
L	Link	3.30
SSW	Skip on Sense Switch	2.475
	SW A+200 E+10 B+100 F+ 4 C+ 40 G+ 2 D+ 20 H+ 1	
SKU	Skip Unconditional	2.475
SKIP ON REGISTER CONDITION		
ICX	Increment Index and Skip	2.475

Mnemonic Code	Instruction Description	Time In Microseconds
SKIP ON REGISTER CONDITION (Cont)		
DCX	Decrement Index and Skip	2.475
SKN	Skip if Accumulator Negative	1.65
SKP	Skip if Accumulator Positive	1.65
SAE	Skip if Accumulator Even	1.65
SQE	Skip if Q Register Even	1.65
SKIP ON CONDITION CODE		
Class 1: Valid following A, S, AOA, TCA		
SZ	Skip if result was Zero	2.475
SP	Skip if result was Plus	2.475
SM	Skip if result was Minus	2.475
SO	Skip if result caused Overflow	2.475
SNZ	Skip if result was Not Zero	2.475
SNP	Skip if result was Not Plus	2.475
SNM	Skip if result was Not Minus	2.475
SNO	Skip if result did Not cause Overflow	2.475
SPZ	Skip if result was Plus or Zero	2.475
SMZ	Skip if result was Minus or Zero	2.475
SZO	Skip if result was Zero or causes Overflow	2.475
SPM	Skip if result was Plus or Minus	2.475
SPO	Skip if result was Plus or caused Overflow	2.475
SMO	Skip if result was Minus or caused Overflow	2.475
Class 2: Valid following M and D		
SO	Skip if result caused Overflow	2.475
SNO	Skip if result did Not Cause Overflow	2.475
Class 3: Valid following OR, XOR, and AND		
SZ	Skip if result was Zero	2.475
SNZ	Skip if result was not Zero	2.475

Mnemonic Code	Instruction Description	Time In Microseconds
Class 4: Valid following C		
SE	Skip if Operands Equal	2.475
SG	Skip if accumulator was Greater	2.475
SL	Skip if accumulator was Less	2.475
SNE	Skip if accumulator was Not Equal	2.475
SGE	Skip if accumulator was Greater or Equal	2.475
SLE	Skip if accumulator was Less or Equal	2.475
Class 5: Valid following ALS and ALD		
SO	Skip if result caused Overflow	2.475
SNO	Skip if result did Not cause Overflow	2.475
SAO	Skip if result is About to Overflow	2.475
NAO	Skip if result is Not About to Overflow	2.475
INPUT/OUTPUT		
DI	Data Input	3.30
DO	Data Output	3.30
RI	Record Input	6.6 +1.65 per Word
RO	Record Output	6.6 + 1.65 per Word
DF	Device Function	3.30
SI	Status Input	3.30
	Test Device Interrupt (Not Pending)	2.47
	(Pending)	7.42
TTI	Timer Channel	
TDI	Channel Zero	
	Channel One	
	Channel Two or DMAC Devices	
	Channel Three including Teletype	
	Direct Memory Access Controller	
TAI	Alarm Channel	

Mnemonic Code	Instruction Description	Time In Microseconds
INPUT/OUTPUT (Cont)		
BRI	Buffered Device Input	6.6 + 1.65 per Word
BRO	Buffered Device Output	6.6 + 1.65 per Word
BDF	Buffered Device Function	3.30
BSI	Buffered Status Input	3.30

section

2

EAI 640 PERIPHERALS DESCRIPTION

EAI 640 PERIPHERALS DESCRIPTION

INTRODUCTION

Selection of peripheral devices for an EAI 640 Digital Computing System is optional on the part of the user with a single requirement. The 640 Programming Systems require a keyboard/typer and a paper tape reader and punch as a minimum for operation. This requirement can be met in one of two ways:

(1) By an ASR Teletype (Model 33 or 35) to provide a keyboard/typer, paper tape reader and punch in a single unit.

(2) Or by a KSR Teletype (Model 33 or 35) to provide the keyboard/typer along with a high speed paper tape reader and punch separately housed.

A maximum of 64 device controllers can be attached to the EAI 640 Computer. These may be distributed within the four data channel interrupt levels to meet the individual system requirements.

640/960 FLOATING POINT PROCESSOR

The 640/960 Floating Point Processor (FPP) is a peripheral device that augments the 640 Digital Computing System with high speed 32-bit single precision floating point capability. The FPP operates independently of the 640 Central Processor (CPU) and is controlled through the standard Input/Output Channel.

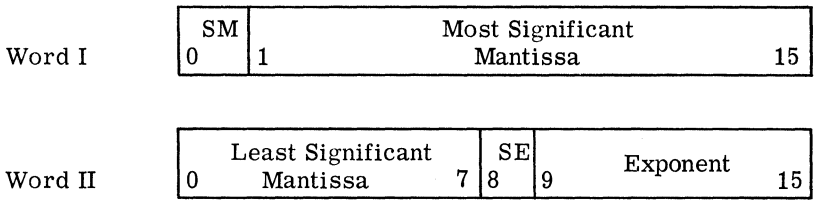
A 32-bit (24-bit mantissa, 8-exponent) accumulator is included in the FPP which can be loaded by two Data Output (DO) instructions and then operated upon. Control of the FPP operation is achieved by the Device Function (DF) instruction. Once an operation is initiated, the FPP will compute independently of the Central Processor which is then free for other computation.

For maximum high speed performance, the FPP contains a buffer register for data words to permit overlapping an FPP computation with data loading for the next computation.

The Central Processor can test the status of the FPP at any time through the Status In (SI) instruction. After the FPP has completed the current operation, the result is stored in its accumulator. This result can either be operated upon or input to the Central Processor through two Data Input (DI) instructions.

Data Format

Each single precision floating point word consists of two 640 data words which have the following format.



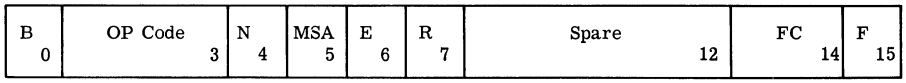
Since the single precision floating point format requires two 640 data words, the execution of 2 Data Output Instructions is required to transmit data to the FPP, and the execution of 2 Data Input Instructions is required to receive results from the FPP.

Commands and Timing

Command	Execution Time* In Microseconds Min	Max
Floating Clear and Add	1.0	1.5
Floating Clear and Subtract	1.0	1.5
Floating Add	1.0	6.0
Floating Subtract	1.0	6.0
Floating Multiply	5.5	11.0
Floating Divide	9.0	11.0

**NOTE: Execution times are for the FPP computation only and do not include function and data fetch times which are a function of CPU program execution. The times given are for unnormalized operations. A post-normalize option exists for every instruction and a rounding option is provided for floating multiply.*

Control Word Format



B = busy
N = normalize
MSA = 2nd operand is accumulator
E = enable exponent fault interrupt
R = round bit
PC = fault code
F = fault occurred, generate interrupt if E is set

Software

A modified Run Time Library will be supplied with the FPP, so that FORTRAN object programs can take advantage of the FPP when performing single precision floating point addition, subtraction, multiplication and division.

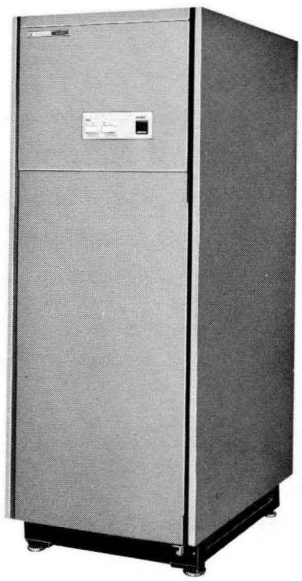
Note that the use of subroutines in the modified Run Time Library can be called from Assembly language programs which will utilize the higher speed of the FPP for the following functions.

(1) Floating Clear and Add
(2) Floating Store
(3) Floating Add
(4) Floating Subtract
(5) Floating Multiply
(6) Floating Divide

Physical Requirements

The Floating Point Processor is housed in the 640/910 Expansion Rack. Has self-contained power supply capable of operating on 50/60 Hz, 115/230 volts.

640/260 DATA DISC STORAGE SYSTEM



The 640 Data Disc Storage System provides bulk storage of information with rapid and random access. Each 640/262 Data Disc Drive provides direct access to 360, 448 sixteen-bit words of information. The disc drive utilizes the design principle of a permanent read/write head positioned over each information track. The result is an all-electronic track access, significantly faster and more flexible than conventional disc files. Average access time is 17 milliseconds.

The 640/260 Data Disc Controller operates up to four 640/262 Data Disc Drives for a total capacity of 1,441,792 sixteen-bit words per controller. Data validation is provided in the controller by calculating and recording a 16-bit cyclic check word for each block written and calculating and comparing the cyclic check word for each block read. In addition, the controller generates and checks the parity of data transmission between the controller and computer.

Protection of information on the disc is provided by allowing 16-sector blocks of data to be recorded as protected. Writing on these blocks is prohibited as long as the unprotect switch on the disc drive console is not set. Writing on protected tracks is permitted only when the unprotect switch is set by the operator and a protect override command is furnished by the computer program. This feature provides for the protection of system software and user libraries.

The disc has 64 tracks, each track has 128 sectors which are individually addressable, and each sector holds 44 words plus one cyclic check word.

Transfer Rate is 3 million bits/second.

Average Access (latency) time is 17 milliseconds.

The Data Disc Controller and up to two Data Disc Drives are mounted in one disc storage cabinet. Communications with the 640 computer may be through either the Standard or Direct Memory Access Channel.

EAI®

TELETYPE WRITER STATION

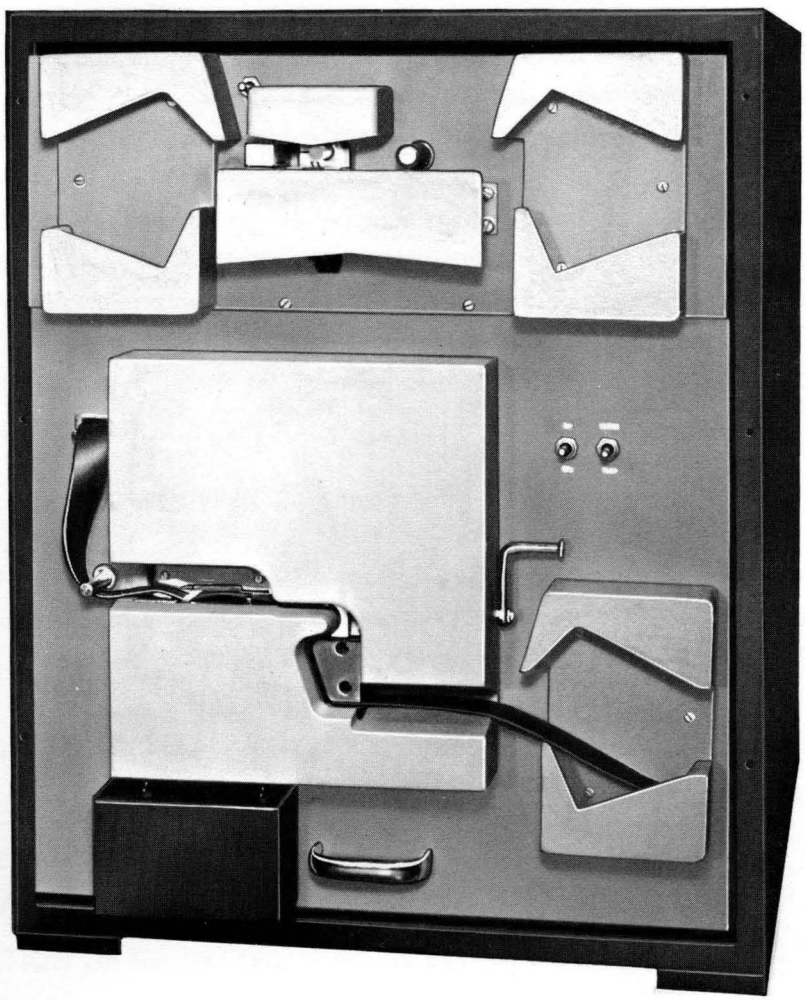
640/410	TELETYPE UNIT (33 KSR)
640/411	TELETYPE UNIT (33 ASR)
640/412	TELETYPE UNIT (35 KSR)
640/413	TELETYPE UNIT (35 ASR)

A selection of four Teletype units is available with the EAI 640 System. Model 33 is the light duty Teletype unit, while Model 35 is the ruggedized version. The KSR (keyboard send-receive) Teletype units provide for keyboard input and typer output, while the ASR (automatic send-receive units include a paper tape reader and paper tape punch.

The keyboard and typer operate in ASCII code at 10 characters per second. The tape reader and punch operate in any code at 10 characters per second.

EAI®

PAPER TAPE STATION



640/420	PAPER TAPE CONTROLLER
640/421	PAPER TAPE READER
640/422	PAPER TAPE PUNCH

The high speed Paper Tape Reader provided with the EAI 640 System is capable of reading 5, 7, or 8 level punched paper tape at 300 characters per second in a forward direction only.

The high speed Paper Tape Punch is capable of punching 5, 7, or 8 level tape at 120 characters per second.

The Paper Tape Controller handles a maximum of one tape reader and one tape punch. The reader or punch can be purchased separately at the customer's option.

The high speed paper tape units can be housed in a standard equipment cabinet or in the desk console equipment cabinet.

EAI®

640/520 CARD READER

The Punched Card Reader provided with the EAI 640 System reads standard 12 row, 80 column cards at a rate of 400 cards per minute. Cards are read column by column; the original deck orientation is maintained in the output hopper after reading. Cards may be loaded or unloaded concurrent with card reader operation so as not to interfere with overall system efficiency. The card reader has an input hopper capacity of 1400 cards and an output hopper capacity of 1000 cards.

EAI®

640/610 LINE PRINTER

The Line Printer provided with the EAI 640 System prints 120 columns (136 optional) at a speed of 300 lines per minute, from a full line buffer. The Line Printer has 64 characters printing 6 lines per inch; vertical format is controlled by a closed loop tape moving synchronously with the forms.

EAI®

640/720 MAGNETIC TAPE CONTROLLER & TRANSPORT

The Magnetic Tape Controller is capable of operating up to four magnetic tape transports. The Tape Transports are available for either of the two most common recording formats - 7 track IBM compatible and 9 track which is compatible with the IBM 360 System.

640/720	Magnetic Tape Controller and one 9 track Magnetic-Tape Transport
640/721	Magnetic Tape Controller and one 7 track Magnetic Tape Transport
640/730	Magnetic Tape Transport (9 track)
640/731	Magnetic Tape Transport (7 track)

EAI®

Both the 9 or the 7 track Magnetic Tape Transports operate at 45 inches per second. Recording density is selectable at 200, 556 or 800 bpi with a switch and can be tested by the computer program with a Status Input.

The tape controller can be used with either the 7 channel or the 9 channel transport units exclusively, or intermixed. The program in the computer can test through a Status Input whether a given tape transport is 7 or 9 track. Tape units can be field modified from either configuration to the other by replacing the read/write head.

Two markers are sensed on the magnetic tape. The Beginning of Tape (BOT) also referred to as "Load Point" is indicated by a reflective marker on the edge of the tape. The tape can be positioned on the marker or it can be rewound to the marker under program control. The End of Tape (EOT) marker can also be sensed by the computer program through an Interrupt.

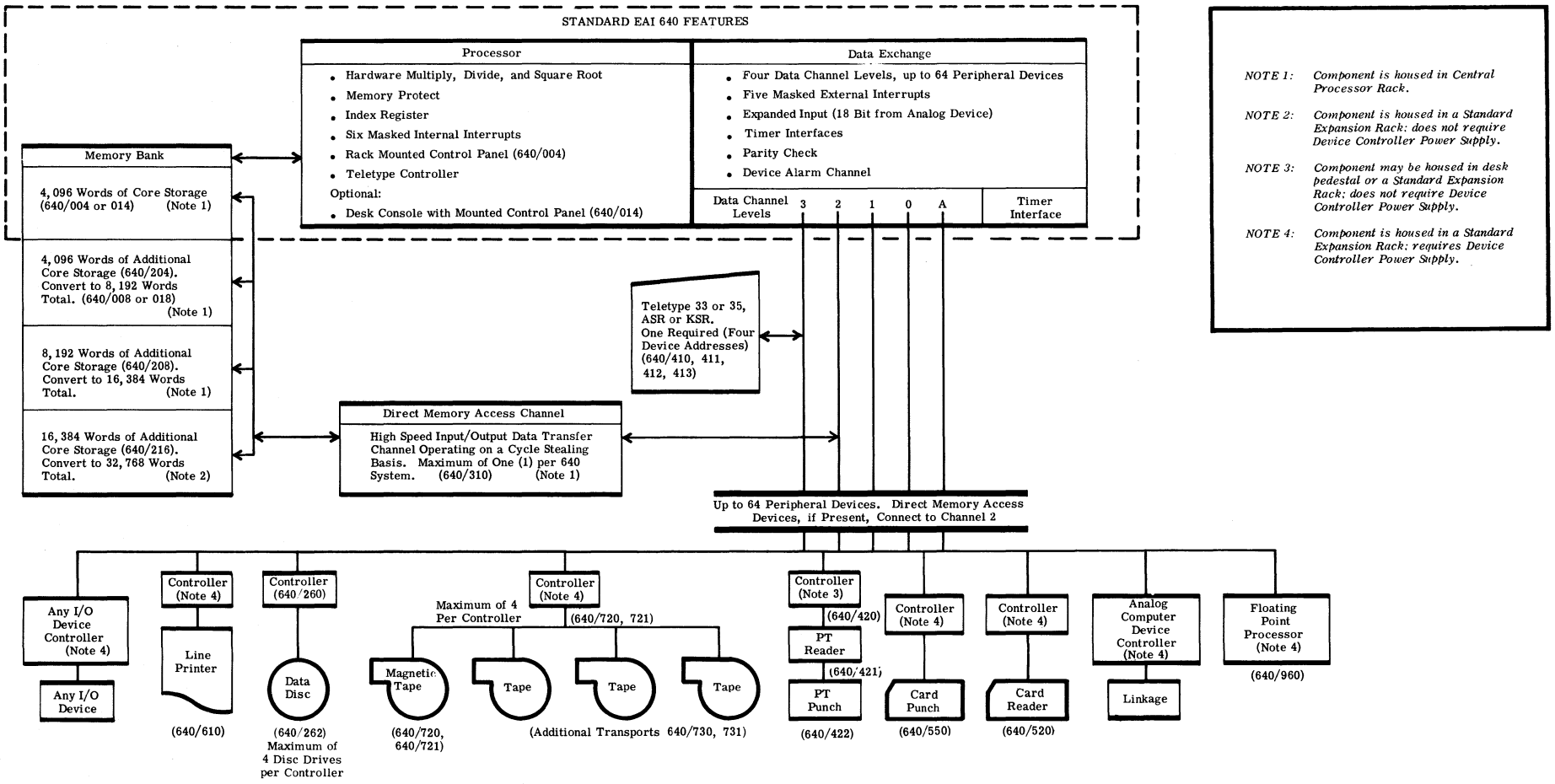
The computer can also request that interrupt be generated when the Magnetic Tape Controller detected an End of Record mark on the Tape.

The Magnetic Tape Transports can read in a forward or reverse direction. Writing is only permitted in the forward direction of tape travel. The controller is capable of transferring data in 8 or 16-bit mode. Assembly-disassembly occurs in the 16-bit mode. Vertical tape parity can be selected under program control as "Even" or "Odd".

The Magnetic Tape Transport can be commanded to skip one record on the tape in either a forward or a reverse direction. A complete file can also be skipped. Should the Magnetic Tape develop a bad spot, a command is available to skip 3.75 inches in a forward direction. An End of File mark can be written under program control after which the transport stops the tape movement.

The status of the selected Magnetic Tape Transport can be tested and the instruction to input this information to the computer can be executed at any time without affecting the operation of the controller or the tape unit.

EAI 640 SYSTEM CONFIGURATOR



NOTE 1: *Component is housed in Central Processor Rack.*

NOTE 2: *Component is housed in a Standard Expansion Rack; does not require Device Controller Power Supply.*

NOTE 3: *Component may be housed in desk pedestal or a Standard Expansion Rack; does not require Device Controller Power Supply.*

NOTE 4: *Component is housed in a Standard Expansion Rack; requires Device Controller Power Supply.*

section

3

EAI 640 SYSTEM SOFTWARE

EAI 640 SYSTEM SOFTWARE

The 640 Programming System is an integrated set of computer programs providing efficient, reliable operation of the EAI 640 Digital Computing System.

The descriptions which follow provide an understanding of the software capabilities available with the EAI 640 Computer and detail the minimum components required for operation.

SYMBOLIC ASSEMBLER

The Symbolic Assembler permits the user to code instructions in a symbolic language using mnemonic symbols of instruction codes and addresses. In addition to the 64 basic instructions, the Assembler also provides a powerful set of pseudo-operations for program control, data definition, storage allocation, input/output functions, and subroutine calls.

The Symbolic Assembler provides a two-pass operation and produces a relocatable object program, program listing, and diagnostic messages.

Available Assembly options include:

(1) inhibit program listings and provide error listing only

(2) output object code in relocatable form

(3) output object code in absolute form (binary)

A variety of peripheral device options are available besides the minimal teletype input/output device. These options are:

Input source program from:

- Teletype paper tape reader
- High speed paper tape reader
- Card Reader

Output object program to:

- Teletype paper tape punch
- High Speed paper tape punch
- Magnetic Tape
- Data Disc

Program listings may be printed on:

- Teletype printer
- Line printer

The determination of which devices are to be used by the assembler is designated by the assembler I/O control package.

Minimum Operating System

640/008 or 018 EAI 640 with 8K words of core memory.

640/411 or 413, Model 33 or 35 ASR Teletype

ASA STANDARD FORTRAN

FORTRAN permits the user who has little knowledge of the computer's organization or machine language to write programs in a language consisting of English words and mathematical symbols. EAI 640 FORTRAN IV is designed for maximum programming flexibility and operating efficiency. It conforms to all specifications

of ASA Standard FORTRAN as described in the October 1964 Communications of the ACM.

640 FORTRAN is a one-pass compiler designed to facilitate batch processing of source programs. Compilation of main FORTRAN source programs is performed separately from the associated subroutines. Therefore, when errors in FORTRAN coding are detected by the compiler only the main program need be recompiled. Ample diagnostics are included in the compiler for rapid identification of coding errors.

The major characteristics of EAI 640 FORTRAN are:

FIXED POINTS CONSTANTS:

1 - 5 decimal digits, absolute value up to 32, 767

FLOATING POINT CONSTANTS:

6 decimal digit single precision words and 15 decimal digit double precision words; decimal equivalent of the exponent range is $10^{\pm 38}$.

ARRAYS:

Up to 3 dimensional arrays are permitted.

STATEMENTS:

Mixed expressions containing both fixed and floating point variables are permitted. With minor restrictions, an unlimited number of continuation lines are allowed.

STATEMENT NUMBERS:

1 - 99999

FUNCTIONS AND SUBROUTINES:

Subroutines not contained in the FORTRAN library may be compiled by the use of FUNCTION and SUBROUTINE statements.

INPUT AND OUTPUT:

Any 640 peripheral equipment may be used as an input/output device. Data format is specified by use of the FORMAT statement.

STATEMENTS AVAILABLE:

Arithmetic, Input/Output with FORMAT; DO, DIMENSION, COMMON, IF, GO TO, ASSIGN, CONTINUE, CALL, SUBROUTINE, FUNCTION, RETURN, END, PAUSE, STOP, EQUIVALENCE, EXTERNAL and DATA.

TYPE OF DECLARATIONS:

Variables may be declared as real, integer, logical, complex, and double precision. Variable names are 1 to 6 alphanumeric characters.

Minimum Operating System

640/008 or 018 - EAI 640 Computer with 8K words of core memory.

640/411 or 413 - Model 33 or 35 ASR Teletype.

OPERATIONS INTERPRETER

The Operations Interpreter represents an on-line interactive language system created specifically for scientists and engineers engaged in the preparation and execution of scientific digital computations. The user is afforded immediate access to both the computational process (software) and computational equipment (hardware) through use of the teletype. The language medium employs a dual syntax, combining in part a complete algebraic interpreter with an expandable command and control repertoire.

This programming system provides the digital computer user with on-line computational flexibility coupled with a substantial reduction in bothersome detail required by Assembly and compiler languages.

The primary consideration of this interactive language interpreter is that communication between the user and the system proceeds wholly on a request-response basis. The user initiates a request and the system responds. The response may

be passive, calling for another request; or active, calling for a user reaction to a system response. User requests are designed to be terse but reasonably readable. This affords high information density on input and is oriented to on-line use by an inexperienced typist.

Minimum Operating System

640/008 or 018 EAI 640 with 8K core memory

640/411 or 413 Model 33 or 35 ASR Teletype

640/420 and 422 High Speed Paper Tape Punch with controller

640 SYMBOLIC TEXT EDITOR

The Symbolic Text Editor is used to prepare punched paper tape containing alphanumeric information. The text is divided into characters, lines and pages. A line is a group of characters terminated by a carriage return. A page is a group of lines terminated by a special code. Any information may be edited or created for input into the Software System (i.e., Symbolic Assembler, FORTRAN, Operations Interpreter).

A page of text is entered from the teletypewriter keyboard, paper tape reader, or card reader. The contents of the page buffer can be altered by deleting, changing and inserting lines and then output to the teletypewriter, paper tape punch, or line printer.

Minimum Operating System

640/008 or 018, EAI 640 Computer with 8K words of core memory

640/411 or 413, Model 33 or 35 ASR Teletype

BASIC MONITOR (LIBRARIAN)

The BASIC MONITOR (LIBRARIAN) is a small executive program used in a disc-oriented computer system. The program provides automated operation of the computer by replacing manual loading and retrieving of programs with typed-in directives to the monitor which performs these functions.

The BASIC MONITOR (LIBRARIAN) makes use of the disc for storing segments of itself, a library index of programs, and the system programs. The main control portion of the BASIC MONITOR does not reside in memory during program operation; rather it resides on the disc and is called in when a program is completed. A small portion of the BASIC MONITOR containing information about the system's operational state is resident in the computer at all times.

The bootstrap loader will bring in the resident portion of the BASIC MONITOR (LIBRARIAN) which will in turn load the main BASIC MONITOR program into memory from the disc.

The Teletype keyboard is used to communicate directives to the BASIC MONITOR to load, delete, or insert new user programs on the disc. It will also be used to load system programs into core memory. The BASIC MONITOR will accept one directive at a time and will turn control over to a user program when directed and will expect control to be returned to it upon completion of the user program.

Minimum Operating System

640/008 or 018 - EAI 640 with 8K words of core memory

640/411 or 413 - Model 33 or 35 ASR Teletype

640/250 Data Disc Memory System

SYSTEM SUBROUTINE LIBRARY

The programs contained in the mathematics subroutine library facilitate use of the Symbolic Assembler and FORTRAN programming systems. More than seventy subroutines are available and are divided into three classes:

1. Various arithmetic and mathematical functions for real, integer, and complex numbers in single and double precision format.
2. Numeric conversion routines, such as decimal to binary, binary to decimal, and format conversion such as fixed to floating, single to double precision, absolute value, etc.
3. Input/Output subroutines for communication with standard peripheral devices.

Program Size

Variable, depending on subroutines

Minimum Operating System

See Symbolic Assembler and ASA FORTRAN

DIGITAL DEBUG

Digital Debug is a program diagnostic using the teletype unit as the control source and readout device. This routine allows the flexibility of reading the contents of memory locations, altering these contents, and inserting breakpoints which result in snapshot printouts of the hardware registers during user program execution.

Digital Debug is an interactive program directed by action characters to perform one of a number of defined operations. These characters are input to the Debug program via the teletype keyboard. Program responses are printed on the teletype printer.

The following list of control functions perform either a Debug program subfunction or else exercise control on the program being debugged.

Function

Display and change the contents of the accumulator register
Breakpoint
Search and compare with given value
Dump a portion of memory
Go to specified location in core and start execution of the user's program
List
Punch object program
Display and change the contents of the Accumulator Extension register
Read object program from paper tape
Verify contents of paper tape with memory
Display and change the contents of status word register
Display and change the contents of index register
Display contents of current open memory core
Modify the current open cell if a change is entered and display contents of cell +1.
Modify the current open cell if a change is entered and display contents of cell -1.
Modify the current open cell if a change is entered and display contents of the current open cell.

Minimum Operating System

640/008 or 018 EAI 640 Computer with 8K words of core memory
640/411 or 413 Model 33 or 35 ASR Teletype

HARDWARE DIAGNOSTICS

The program consists of sets of diagnostic routines designed to exercise computer operations and validate the result of these operations.

The diagnostic routines will exercise the complete instruction repertoire, the entire memory available, and the peripheral devices in the configuration. Failures will result in messages on the Teletype or Line Printer to aid the service technician in locating the fault.

Instruction Test will consist of testing the operation of all instructions except the I/O group which will be tested when exercising the peripheral devices. Indirect addressing and indexing with indirect addressing are also checked. The bootstrap loader is used to read in the diagnostic program; therefore, the instructions used by the loader are assumed to be working only for read-in purposes. These instructions are fully checked by the Instruction Test. In general, an instruction is not used before it is tested.

The console teletype writer is tested upon initiation of the diagnostic since it will be used as the media of communication between the program and the operator.

Diagnostic routines to exercise each peripheral device are provided and form part of the hardware diagnostics for the specific system configuration. Peripheral device testing will occur for each individual device as selected by sense switch settings on the control console.

Minimum Operating System

640/008 or 018 EAI 640 Computer with 8K words of core memory
640/411 or 413 Model 33 or 35 ASR Teletype

BOOTSTRAP LOADER

The Bootstrap Loader is a short sequence of instructions which will facilitate loading of the Monitor or Program Loader which will in turn load object programs for execution.

The Bootstrap Loader can be automatically loaded from paper tape into the 640 Computer utilizing the hardware Bootstrap feature of the 640 if the system teletype has a paper tape reader. If a teletype tape reader is not present, the Bootstrap Loader must be input manually via the switches on the console.

Minimum Operating System

640/008 or 018 EAI 640 Computer 8K words of core memory
640/411 or 413 Model 33 or 35 ASR Teletype (for automatic loading)

PROGRAM LOADER

The Program Loader loads object programs into core memory, and allows assembled or compiled relocatable programs to be relocated. The loader is also capable of handling non-relocatable programs. In addition, the loader will link programs that have been called by the program being loaded when the called program is loaded.

The loader resides in upper memory. The starting location of the loader varies with the core size of the machine. The Program Loader version is capable of handling programs assembled or compiled in relocatable and non-relocatable form by the Symbolic Assembler and the FORTRAN compiler. In addition, it can handle external function such as linking program calls and loading the called program and linking external references within other programs associated with the loaded program.

The Program Loader is input in the machine by the bootstrap loader.

Input of programs assembled or compiled in relocatable form will be via a starting address. Upon completion of loading the initial program, subsequent called programs will be loaded via the mode selected upon initialization of the loader. Linking of programs called within the loaded program will be done by the loader when called programs are loaded into memory. Upon completing the loading of all called programs control is turned over to the first program loaded.

Minimum Operating System

640/008 or 018 EAI 640 Computer with 8K words of core memory
640/411 or 413 Model 33 or 35 ASR Teletype

EAI®

ELECTRONIC ASSOCIATES, INC. *West Long Branch, New Jersey 07764*

ADVANCED SYSTEMS ANALYSIS AND COMPUTATION SERVICES/ANALOG COMPUTERS/DIGITAL COMPUTERS/HYBRID ANALOG-DIGITAL COMPUTATION EQUIPMENT/ANALOG AND DIGITAL PLOTTERS/SIMULATION SYSTEMS/SCIENTIFIC AND LABORATORY INSTRUMENTS/INDUSTRIAL PROCESS CONTROL SYSTEMS/PHOTOGRAMMETRIC EQUIPMENT/RANGE INSTRUMENTATION SYSTEMS/TEST AND CHECK-OUT SYSTEMS/MILITARY AND INDUSTRIAL RESEARCH AND DEVELOPMENT SERVICES/FIELD ENGINEERING AND EQUIPMENT MAINTENANCE SERVICES.

PRINTED IN U.S.A.

67206