3.0 ARCHITECTURE OF THE AMPP

3.1 Physical Overview

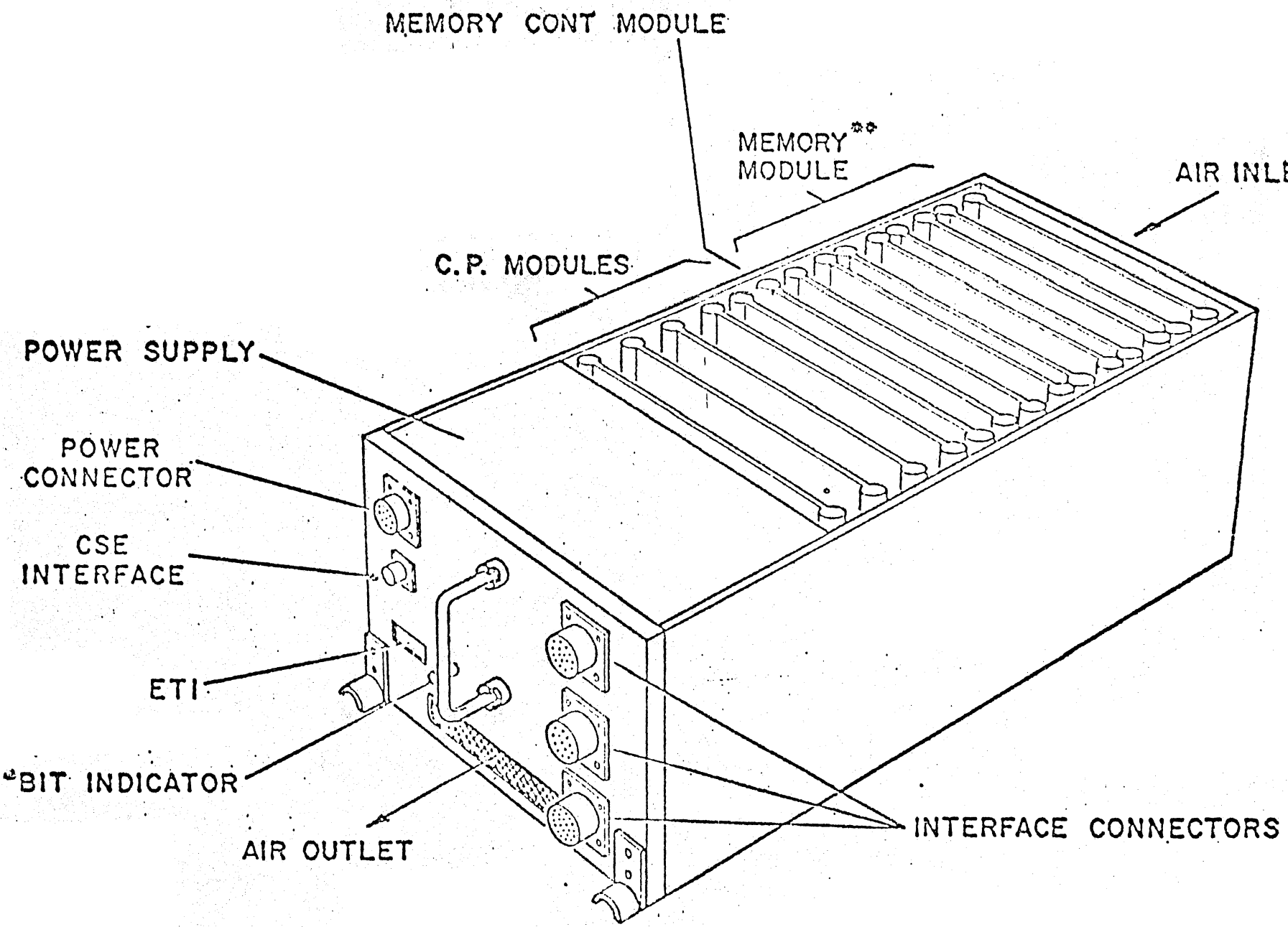
A sketch indicating the physical arrangement of the pluggable modules within the AMPP unit is shown in Figure 3-1. The AMPP unit is packaged in an airborne militarized full ATR enclosure having dimensions of approximately 10.1 x 7.6 x 19.6 inches. The unit contains a single pluggable power supply module, six pluggable logic modules, and four pluggable core storage modules. All of these modules interconnect electrically through a single multilayer motherboard mounted in the bottom of the unit. Flexible flat printed circuit cables provide interconnection from this board to the front panel connectors.

The power supply module receives 400 Hz, 3 phase, 208V input power and furnishes the required DC power forms for the logic and core storage modules. It contains power sequencing, transient detection, over-voltage and under-voltage protection, over-current protection, and thermal over-temperature detection with automatic shutdown.

The six logic modules are each constructed from two printed wiring assemblies bonded one to each side of an air cooling plate. Each printed wiring assembly consists of a 10 layer multilayer board with a 180 pin connector at the bottom and 123 feed-through/test-point connection points at the top of the card. A maximum of 170 fourteen or sixteen lead integrated circuit flat packs are mounted on each printed wiring assembly giving a total of 340 integrated circuits maximum per logic module. Twenty-four lead flat packs are also used on the modules and consume the area of about 1-1/2 sixteen lead packages. The feed-through/test-point connection points along the top of the board are used for signals between the two printed wiring assemblies of the module while the connectors at the bottom provide a total of 320 signal pins, 20 power pins, and 20 ground pins for the module. The first logic module, the one closest to the front of the unit directly adjacent to the power supply is termed logic module assembly A1A1. The second module is A1A2, and so forth, with A1A6 adjacent to the core storage modules which are in the rear of the unit. The front side of each module is termed the A side and the rear the B side. The following is a list of the six logic modules and their functional names:

A1A1	Micro Module
A1A2	I/O Module
A1A3	Arithmetic Module
A1A4	Option Module
A1A5	Control Module
A1A6	Memory Control Module

During the discussion of the architecture, functional portions of the organization will be related to the physical location in the unit using the above module names.



** *SEE SKETCH OF TYPICAL MEMORY MODULE FOR DETAILS*

Figure 3-1. AMPP Computer

The four core storage modules are identical and interchangeable. Their construction is slightly more complex than that of the logic modules. Each core storage module contains two air cooling plates. Each of the plates have two printed wiring assemblies bonded to them, one to each side, and are further hinged to two core board assemblies. Figure 3-2 shows a sketch of the core storage module in a partially unfolded position. In this position, access to all components, with the exception of the cores themselves, may be obtained. In the fully unfolded position, the cores are also accessible. Each of these core storage modules contains 16K of 18 bit words. The core storage modules are designed such that they may be partially populated with cores, digit drive circuits, and sense electronics to provide shorter word lengths than 18 bits. For 16 and 32 bit word length versions of the AMPP, the last 18 bit words provide a parity bit and protect bit for each 16 bits of information. For a 24 bit word length version, 14 bit core storage module words provide 12 bits of information plus a parity bit and protect bit.

The AMPP unit shown in Figure 3-1 has a maximum internal core memory storage capacity of 32,768 words of 32 bits plus parity and protect bits (131,072 bytes). The internal core memory is divided into two independent and simultaneously accessed memories, each having four access channels. Two of these channels may be used internally by the processor and two are available for external access. Memory capacity of processors is expanded by adding additional identical units with five of the six logic modules removed. The Memory Control Module, A1A6, is retained in the Memory Expansion Unit, and contains priority resolution, storage protect keys, parity generation/checking, control, and interface circuitry. The addressing capability of the processors permits expansion to a maximum core storage capacity of 1,048,576 bytes.

3.2 AMPP Word Length

The AMPP architecture is designed not around a 16, 24, or 32 bit word length, but around all three. In other words, the machine can operate in any of the above three word length modes. The philosophy incorporatedin the design enables the standard printed circuit boards on the logic modules to be populated with components according to different assembly drawings to yield the different word length versions of the machine. When the boards are assembled with parts in those locations necessary to produce a 16 bit version of the machine, there are many other component mounting locations which are unused and would be used for other word length versions of the machine. These modules may also be fully populated with components such that the processor, under firmware control, may dynamically change its word length from one operating mode to another. One of the primary design goals of the AMPP was to use common circuit boards for the various word length versions without requiring an excessive amount of components for a particular word length machine. The number of integrated circuits required in any one word length version of the machine has been kept to a minimum. Figure 3-3 shows the technique employed in the arithmetic and other data paths in the AMPP architecture. All word length parallel paths are 32 bits in width. The 32 bit machine uses, of course, all of these data paths. The 24-bit version, however,

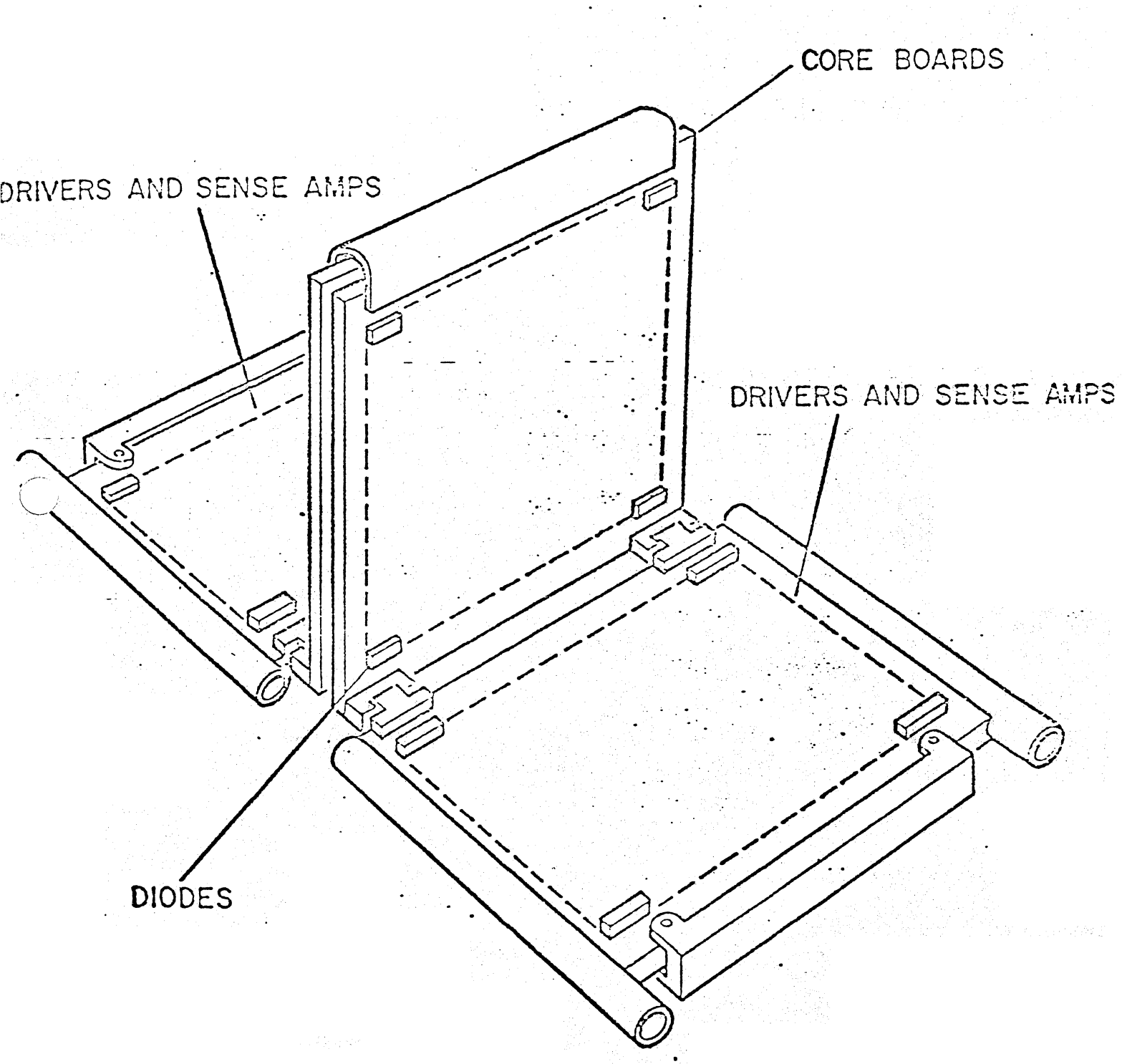


Figure 3-2. Memory Module-Component Access

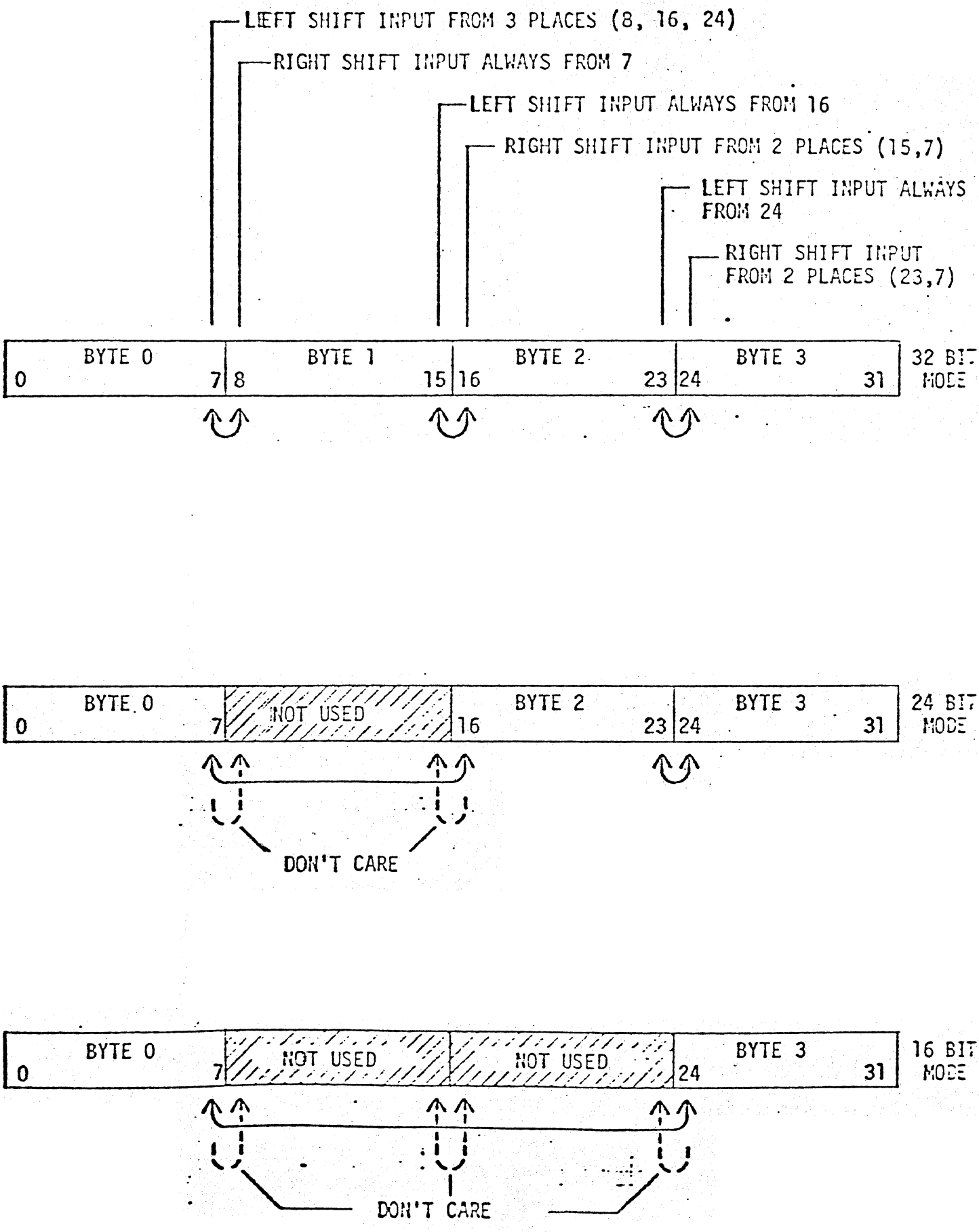


Figure 3-3. Data Paths

does not use the second most significant byte, or bits 8-15, in any of the registers in the machine. In a like manner, the 16-bit version does not use either of the two center bytes, or bits 8-23 of the registers in the machine. Using this scheme, the most significant bits and the least significant bits of all word length machines are the same physical bits in all of the registers. This means that the sign bit, or most significant bit, is always physical bit 0, and the least significant bit is always physical bit 31. All sign tests, or "end bit" manipulations use these physical bits regardless of the word length mode in which the machine is operating. All shift registers, arithmetic circuits, and counter-type registers in the AMPP utilize circuitry to "by-pass" unused bytes when the 16-bit or 24-bit mode is selected. The circuitry required for by-passing unused bytes has been minimized.

3.3 Block Diagram of AMPP

A simplified block diagram of the processor is shown in Figure 3-4. The blocks shown with dotted lines on the block diagram are either optional features or tailored features to improve emulation speeds for particular machines. For example, the block containing the single-pass, 64 bit shift matrix and the block containing floating-point functions, together with a high speed TTL memory (the latter is not shown on the block diagram) are all contained on the Option Module, A1A4. In the System/360-75 emulation, this module would be used with the circuitry for the optional TTL memory removed.

The blocks labeled FILE 1 and FILE 2 each contain register storage for 256, 32-bit general registers. These two files are contained on the Arithmetic Module, A1A1. In the System/360-75 emulation, the circuitry for the optional FILE 2 would be removed. A portion of FILE 1 would be used for the sixteen general arithmetic registers and the eight floating-point registers in the System/360-75 emulation.

The block labeled WORKING REGISTERS contains the six registers A, Q, X, F, P and I. The F register is used for writing into FILE 1 and/or FILE 2. One bit fields are used in the micro command word of the micro program to specify that the contents of F are to be written into the files during the first part of the next micro command. The P register is normally interpreted by the firmware as the program address register for the machine being emulated. In the System/360 emulation, P will contain the address of the next full word (32 bits) to be referenced in core storage for instructions. The I register is normally interpreted by the firmware as zero or sign extended portions of the instruction word for the machine being emulated. Both P and I may be used for other purposes during the emulation of certain complicated instructions, in which case, a copy of the original P and I contents is retained in FILE 1. The A and Q registers have right and left shift properties which may be specified by fields in the micro command. The working registers, along with the two files, the arithmetic logical unit, and the multiplexers shown on the block diagram are all contained on the Arithmetic Module, A1A1.

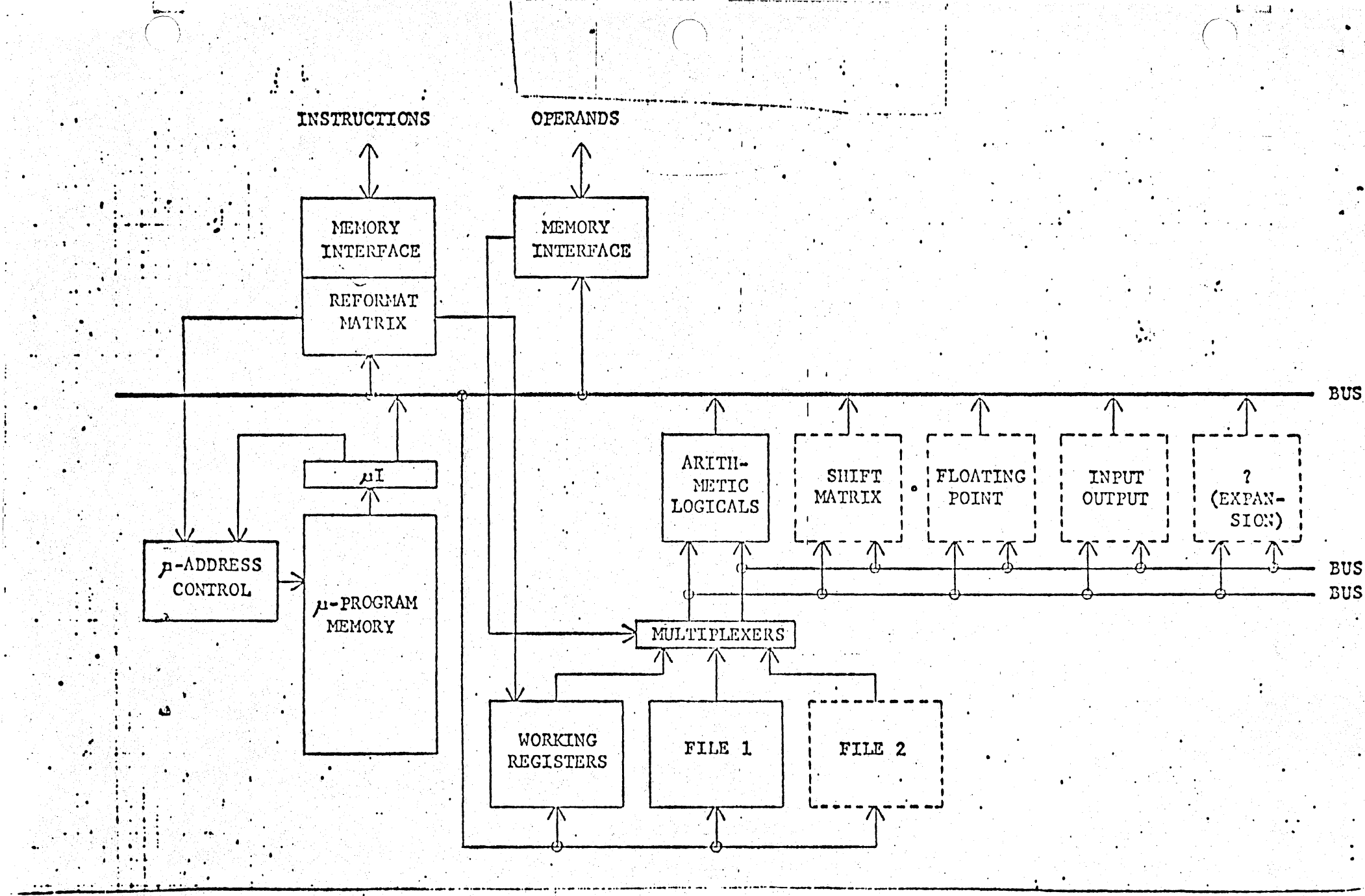


Figure 3-4. AMPP Simplified Block Diagram of Processor

The block labeled u-PROGRAM MEMORY contains up to 3072, 64-bit micro commands. The u-PROGRAM MEMORY is implemented with integrated circuit chips. A portion of the memory may be read/write (RAM), or all of the memory may be PROM or ROM. In the System/360-75 emulation, the entire u-PROGRAM MEMORY will be permanent (read only) with an estimated usage of two-thirds maximum capacity (1095 spare 64-bit words in the worst case for the central processor in subsystems B or C). The next micro command to be executed is read and held in the 64-bit uI register. The various fields of this micro command are used for micro program control of the processor and are explained in Section 3.4. Thenormal execution time for a micro command is 150 nanoseconds. The address of the next micro command to be referenced in the u-PROGRAM MEMORY is obtained through a multiplexer in the block labeled u-ADDRESS CONTROL. Inputs to this multiplexer are controlled by the micro command either unconditionally or conditionally by status specified during the present micro command. Address source is from the 12-bit V field of the microcommand (u-jump), the reformat matrix (instruction interpreted jump), or from one of three address registers uJ, uK, and uP. Various fields of the micro command are used for incrementing, transferring, and controlling these address registers which provide the capability of returnable micro program subroutines, repeat, and execution of a single micro command out of sequence (analogous to an EXECUTE INSTRUCTION in software). Additionally, up to four-way branches are provided. The u-PROGRAM MEMORY, u-ADDRESS CONTROL, and u-I register are contained on the MICRO MODULE, A1A2.

The processor interfaces with core storage through two access channels (memory ports) indicated at the top of the block diagram by INSTRUCTIONS and OPERANDS. The two blocks labeled MEMORY INTERFACE contain separate address and data registers for each memory port. For the operand port, the address and data registers are OA and OD respectively. For the instruction port, they are IA and ID respectively. Thus, the processor has capability of overlapping instructions and operands by simultaneous access on these two ports (assuming the referenced addresses are located in different 16K by 32-bit blocks of core storage and no conflict exists with an external user during the reference). Various fields of the micro command are used for transferring information into and out of these registers and for initiating memory references on the two ports. When a memory reference has been initiated by a micro command for a fetch operation, data is automatically gated into the proper ID or OD register by asynchronous timing of the memory. When access to the information in the ID or OD register is desired by the processor, the D field of the micro command is used. This field contains codes which cause the present micro command to wait until information becomes available in the event that previously requested data has not been placed into the register by the memory or to proceed immediately if the data is available. Time-out delays are incorporated to assure non-stop operation and to detect memory failure or programmer error. In particular, an RNI (read next instruction) code is used in the D field of a micro command to inform the reformat matrix that the instruction in the ID register is to be interpreted for the <u>first time</u>. This code also normally causes the instruction in the ID register to be transferred to the ID* register, so that future references to the reformat matrix (not containing RNI codes) during the

emulation of this instruction are interpreted from the ID* register. Thus, the ID register becomes a buffer for holding the next instruction accessed during the emulation of the present instruction. The reformat matrix generates the address to be used in the micro program memory for the next micro command and the proper address for FILE 1 and 2 based upon the present instruction in ID or ID* and various fields of the micro command. A portion of the module containing the reformat matrix has been reserved for tailoring to the needs of particular machines for improved emulation speeds. The reformat matrix is considered in more detail in Section 3.5.

Three buses (A, B, and C) are shown on the block diagram. The various blocks shown with inputs from the BUS A and BUS B all have their outputs connected to the tri-state BUS C. Only one of these blocks at a time is active on the BUS C. The proper block is specified by the 3-bit F field (main function code) of the micro command. Additionally, the F-field may specify that none of these blocks are to be active on the BUS C, but that instead, fields of the uI register (the micro command itself for literal operations) are to be placed on the BUS C, or that the operation is an AQ shift function or a micro memory write operation. An eight bit S field (sub-function code) in the micro command supplies additional information for the particular operations of the block selected. For example, the particular function of the floating point hardware; the direction, length, and controls of the shift matrix; the sixteen functions of the arithmetic logical unit; the various selections on the I/O module; etc. are specified by S. The source of information placed on the BUS A and the BUS B through multiplexers is determined by two 3-bit fields, A and B respectively, in the micro command. These sources are the six working registers, FILE 1, FILE 2, the operand data register (OD), the X register shifted, the file address registers, a special output (T) from the reformat matrix, the shift counter (SC) and the iteration counter (IC). The latter two inputs and the file address registers are not shown on the simplified block diagram. The shift counter is used by the shift matrix to specify the number of places to be shifted in one pass. Both SC and IC are loaded, incremented, decremented, and tested by various fields of the micro command. The destination for information placed on the BUS C is specified in the micro command by the 4-bit C field. This information is gated to the specified destination at the end of the micro command.

Also not shown on the simplified block diagram is a mode and status register and the Bit File. These are used for holding various fixed and programmer defined modes, status, and conditions in the machine. The Bit File consists of 64 bits which by various fields of the micro command can be selectively cleared, set, toggled, or loaded with status, control, and flag information. Certain of these bits are programmer defined and others indicate control and status such as adder overflow, underflow, carry out, working register signs, BUS C signs, adder byte zero tests, storage byte write control, interrupt disables, provileged states, etc. Certain of the bits are automatically reset by completion of the instruction emulation (RNI code in D field). Fields of the micro command are also used for inspecting selective combinations of bits from the Bit File for next micro command branching. These and other fields of the micro command are defined in more detail in Section 3.4.