# 7300 Processing Unit

## Design Description Manual

## Volume I: Overview

2501.001

MEMOREX

Computer System Products

**October 1972 Edition**

**Memorex Corporation**
**Santa Clara, California 95052**

Requests for copies of Memorex publications should be made to your Memorex representative or to the Memorex branch office serving your locality.

This publication is provided for informational purposes. Contact Memorex for the latest periodic enhancement.

A readers' comments form is provided at the back of this publication. If the form has been removed, comments may be addressed to the Memorex Corporation, Publications Dept., Santa Clara, California 95052.

# PREFACE

This volume provides an introduction to the hardware in the MEMOREX 7300 Processing Unit. It is a continuation of the information presented in the 7300 Processing Unit Reference Manual, and is based on the assumption that the reader is familiar with the functions and special features of the computer as described in that publication.

This volume is part of a four volume set comprising the 7300 Processing Unit Design Description Manual. The set of four volumes is assembled as a continuem of section numbers containing the following information:

**Volume 1, Overview (2501.001)**

Section 1. A general description of the 7300 Processing Unit.

**Volume 2, Shared Resources (2501.002)**

Section 2. A detailed operational description of main storage, control, timing and arithmetic parts of the 7300 Processing Unit.

Section 3. A detailed description of the formats, characteristics and implementation of the micro instructions associated with the 7300 Processing Unit.

**Volume 3, Dedicated Resources (2501.003)**

Section 4. A detailed operational description of the two basic data (selector) channels for the 7300 Processing Unit.

Section 5. A detailed operational description of the communications (multiplex) channel (ICA) for the 7300 Processing Unit.

Section 6. A detailed operational description of the disc channel (IFA) for the 7300 Processing Unit.

**Volume 4, Power System (2501.004)**

Section 7. A detailed description of the power supplies, voltage regulators and associated hardware for the 7300 Processing Unit.

# TABLE OF CONTENTS


| Section | | Page |
|---|---|---|
| 1 | GENERAL DESCRIPTION | 1-1 |
| | Introduction | 1-1 |
| | Physical Description | 1-3 |
| | Cooling | 1-3 |
| | Power | 1-3 |
| | Component Locations | 1-3 |
| | Functional Description | 1-8 |
| | Shared Resources | 1-8 |
| | Resource Allocation Network | 1-8 |
| | Time Slicing | 1-8 |
| | Arithmetic – Logical Unit | 1-10 |
| | Main Storage | 1-11 |
| | Control Storage | 1-11 |
| | Dedicated Resources | 1-13 |
| | Register File | 1-13 |
| | Basic Register File | 1-13 |
| | Group I, Extended | 1-14 |
| | Group II, Extended | 1-14 |
| | Group III, Extended | 1-14 |
| | Communications Processor (COMM) | 1-15 |
| | Basic Data Channel 2 (BDC2) | 1-16 |
| | Basic Data Channel 1 (BDC1) | 1-17 |
| | Disc Processor | 1-18 |
| | Executive Processor (EXEC) | 1-19 |

# LIST OF FIGURES


| Figure | | Page |
|---|---|---|
| 1-1 | Computer Layout | 1-2 |
| 1-2 | Computer Locations, Overall | 1-4 |
| 1-3 | Computer Locations, Front | 1-5 |
| 1-4 | EPO/Power Sequence Connector Panel | 1-6 |
| 1-5 | Logic Chassis Layout | 1-7 |
| 1-6 | Time Slicing | 1-9 |
| 1-7 | ALU Configuration | 1-10 |
| 1-8 | Main Storage Configuration | 1-12 |
| 1-9 | Control Storage Network, Simplified | 1-12 |
| 1-10 | Register File Grouping | 1-13 |
| 1-11 | Communications Processor | 1-15 |
| 1-12 | Basic Data Channel 2 | 1-16 |
| 1-13 | Basic Data Channel 1 | 1-17 |
| 1-14 | Disc Processor | 1-18 |
| 1-15 | Interaction of Exec with Other Processor States | 1-19 |


# LIST OF TABLES


| Table | | Page |
|---|---|---|
| 1-1 | Physical and Electrical Characteristics | 1-3 |
| 1-2 | Register File IC's | 1-14 |

# 1. GENERAL DESCRIPTION

## INTRODUCTION

The MEMOREX 7300 Processing Unit consists of a main cabinet and a permanently attached console table that holds the optional console card reader.

The main cabinet contains the computer logic and integrated adapters for communicating with disc drives, card equipment, and various terminal devices — the latter including a keyboard printer for manual entry and "hard copy" display of data. Binary entry and display is also possible from the System Control Panel.

A cabinet under the console table provides for future expansion of the integrated communications facility beyond the present maximum of sixteen lines, and may also contain additional adapters that cannot be physically accommodated by the main cabinet. Space has been reserved in this console cabinet for storing manuals and ledgers.

MEMOREX 7300 Features and Options

- Generalized Business Data Processing instruction set
- 16-bit Arithmetic/Logical Unit (ALU) using two's complement arithmetic
- 14-bit alterable control storage (CS) with word parity; available in 4K-word modules to 16K* words
- Floating point option, micro instruction implemented
- 18-bit Main Storage (MS) with byte parity; 16K bytes basic, up to 128K bytes available in 8K-byte increments
- Relocation and Protection feature for dynamic relocation and address expansion of MS beyond 64K bytes
- Main Storage Bounds Protection (Write only) for processors 5, 6, and 7 in lieu of Relocation and Protection feature
- Error Correction Code (ECC) option for MS
- Job Accounting Aids (optional)
- Communications processor with integrated communications adapter (ICA) serving a console keyboard-printer and up to 15 additional communication lines
- Disc processor with integrated file adapter (IFA) for up to eight MEMOREX 3664 disc drives (96 megabyte capacity)
- Basic Data Channel (BDC) processor, with options for integrated adapters or external controllers operating in byte-wide (8-bit) transfer mode

  Integrated adapters include:

  - 300, 600, 1000 CPM readers (80-column)
  - 500/100 CPM reader/punch (80-column)
- A second Basic Data Channel processor for communicating with peripheral devices using controllers outside the main cabinet, plus optional two-byte-wide (16-bit) transfers between the BDC and shared resources for increased throughput
- Three general-use processors
- System Monitor (Executive) processor
- Integral power supplies
- System Control Panel

*K=1024 in this type usage.

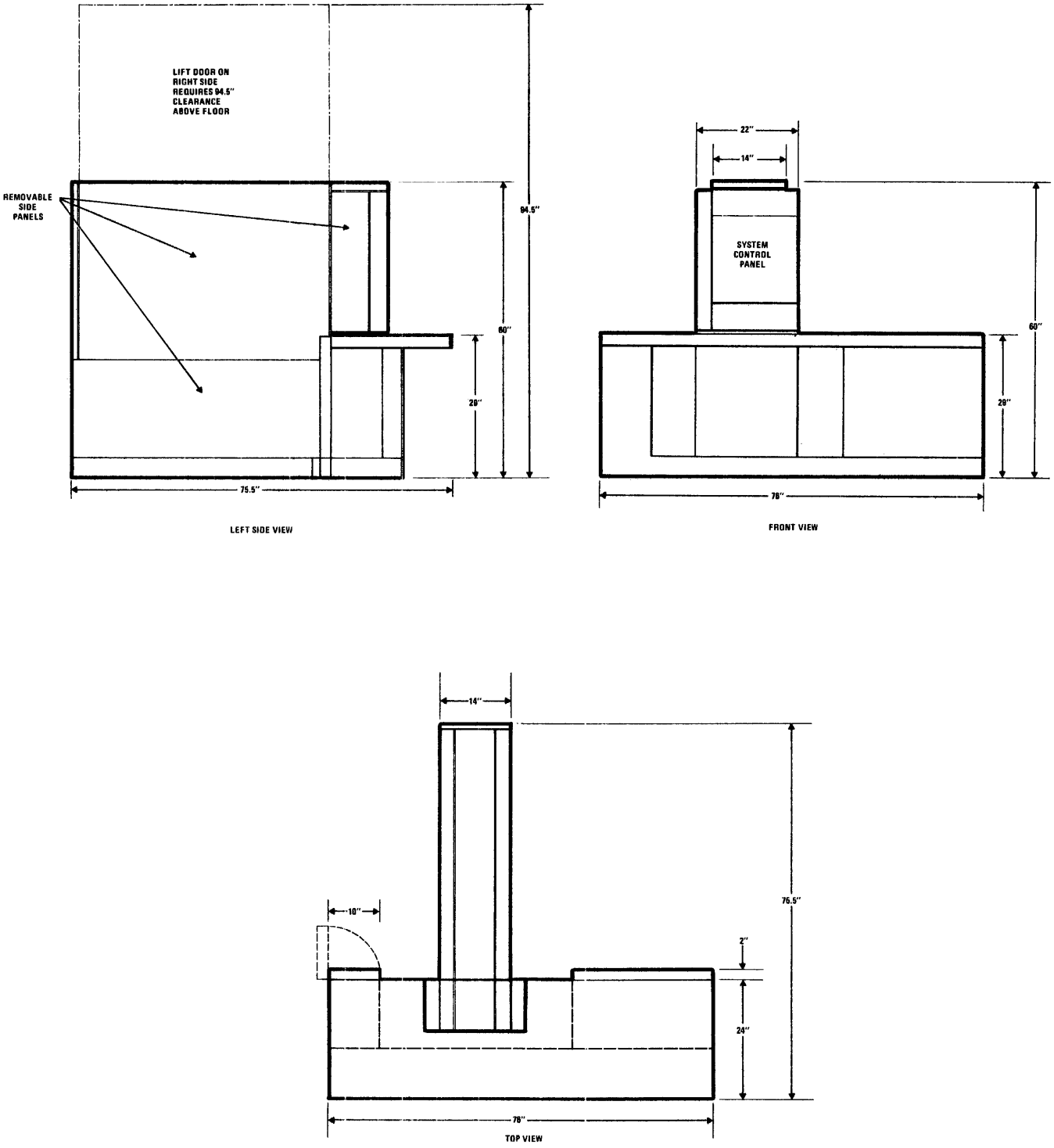



**Figure 1-1. Computer Layout**

## PHYSICAL DESCRIPTION

Basic dimensions of the MRX 7300 Processing Unit cabinet are given in Figure 1-1. Table 1-1 lists the salient physical and electrical characteristics.

**Table 1-1. Physical and Electrical Characteristics**

| | Processing Unit | Console Table |
|---|---|---|
| Height (inches) | 60 | 29 |
| Width (inches) | 14 | 76 |
| Depth (inches) | 75.5 (including console table) | 24 |
| Weight (pounds) | 900 | – |
| Heat Dissipation | 9000 BTU's/Hr | – |
| Power: | | |
| Voltage | 208/230V | |
| Phase | 3 | |
| KVA | 2.7 | |
| Ambient Ranges: | | |
| Temperature | 22°C. – 28°C. (72°F. – 82°F.) | |
| Rel. Humidity | 30% – 80% | |
| Barometric Pressure | 28 – 31 inches | |

Access to all serviceable components is facilitated by removable panels, and by a lift door on the right (printed-circuit-board) side. In addition, the normally front-facing System Control Panel may be swung out toward the right side of the processing unit, the position most useful when performing maintenance on the system.

Power supplies occupy the lower portion of the cabinet; the chassis bay, which holds two logic chassis and the power regulation chassis, is at the top. Cooling air is circulated by blowers situated beneath the chassis bay.

### COOLING

The blowers recirculate room air by moving cool air past the power supplies, up through the chassis, and out louvers at the top of the cabinet. Plenum cooling is not required; heat dissipation of the computer is about 9000 BTU's per hour, roughly twice that of an average line printer.

Ambient conditions for the computer room are determined by the comparatively stringent requirements of the various information media – magnetic tape, disc packs, punched cards, etc. The allowable ranges are given at the bottom of Table 1-1.

### POWER

The 7300 Processing Unit requires a 208/230-volt, 3-phase source, and provides power for a full complement of nine disc drives or its equivalent in power consumption. The processing unit controls the power sequencing for all disc drives as well as any devices connected to the external interface of Basic Data Channels 1 and 2.

Power sequencing of the disc drives is interlocked so that all disc drives having their power switches in the ON position must have come up to speed before power can be applied to the processing unit itself. Similarly, removing power from the processing unit cannot be effected until each disc drive has been successfully *powered down*. A sonic alarm in the Processing Unit will turn on if the power-down sequence for that drive has not fully retracted the heads. This alarm will also sound if a DC fault is detected in any of the power supplies, or if a high internal temperature is sensed. For either of the latter conditions, the system automatically shuts down within one minute.

### COMPONENT LOCATIONS

The locations of the major computer components are shown in Figure 1-2. Following is a brief description of some of the components.

The phase-lock control module monitors the current drain of the logic chassis and determines how many diodes are needed to maintain a relatively constant energy level in the +5v capacitor bank.

Diodes for the Auxiliary and Main Storage supplies are located in the capacitor/rectifier bank facing the right side of the cabinet. All three supplies (Logic, Auxiliary, Storage) deliver unregulated DC to the power regulation network in Chassis 3. The control voltage needed by the power regulation chassis is provided by the +24v supply. This supply also furnishes voltage to the power-sequence relays, and to the Data Access Arrangements (DAA) used to interface the ICA to the telephone network.

Power for the indicator lights on the system control panel comes from the +28v supply in the control panel recess.

Figure 1-3 shows in greater detail the components in the vicinity of the I/O connector panel. The I/O panel is used only for the disc drive subsystem and for subsystems attached to the two I/O channels. The "Unit" connectors

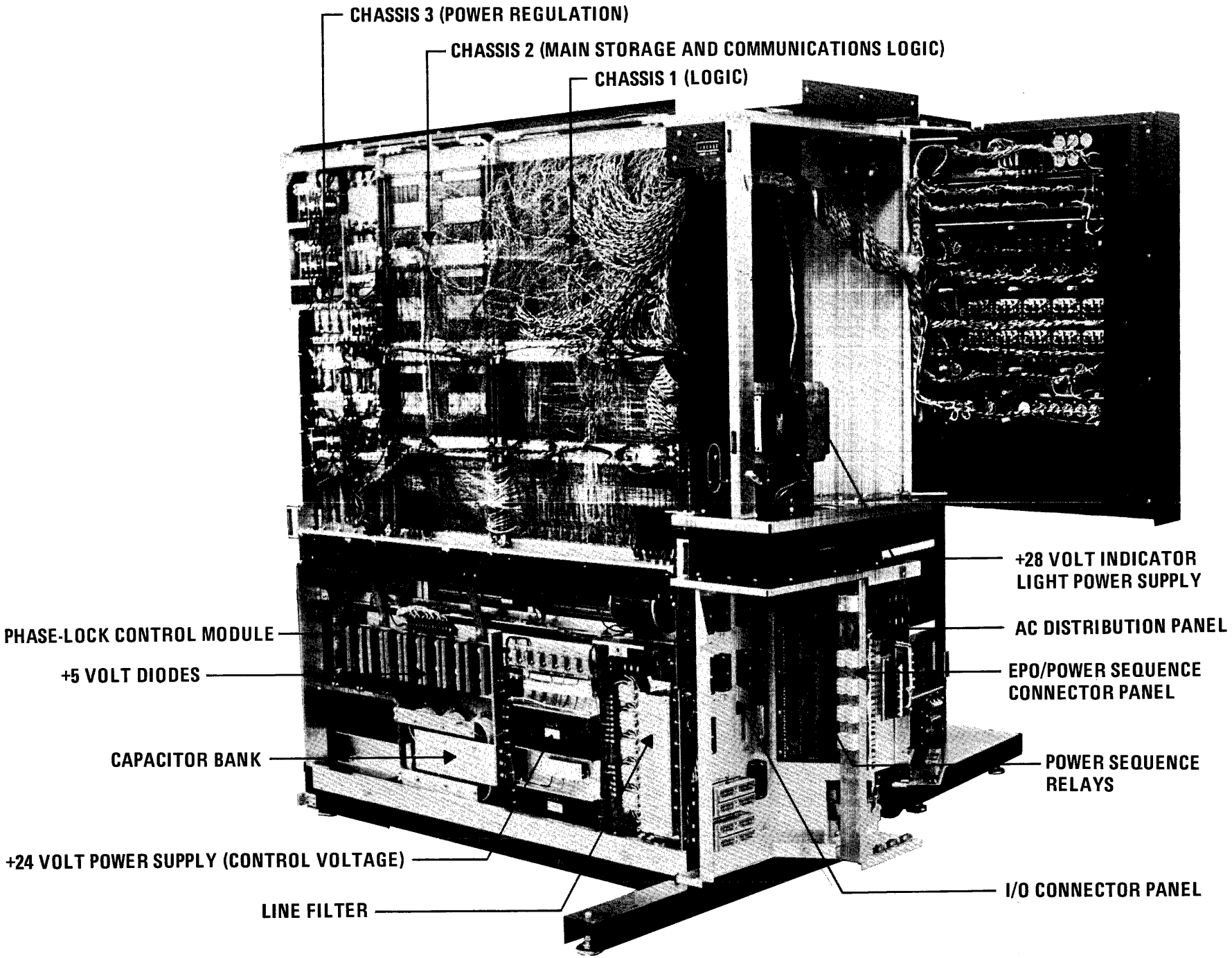



Figure 1-2. Computer Locations, Overall

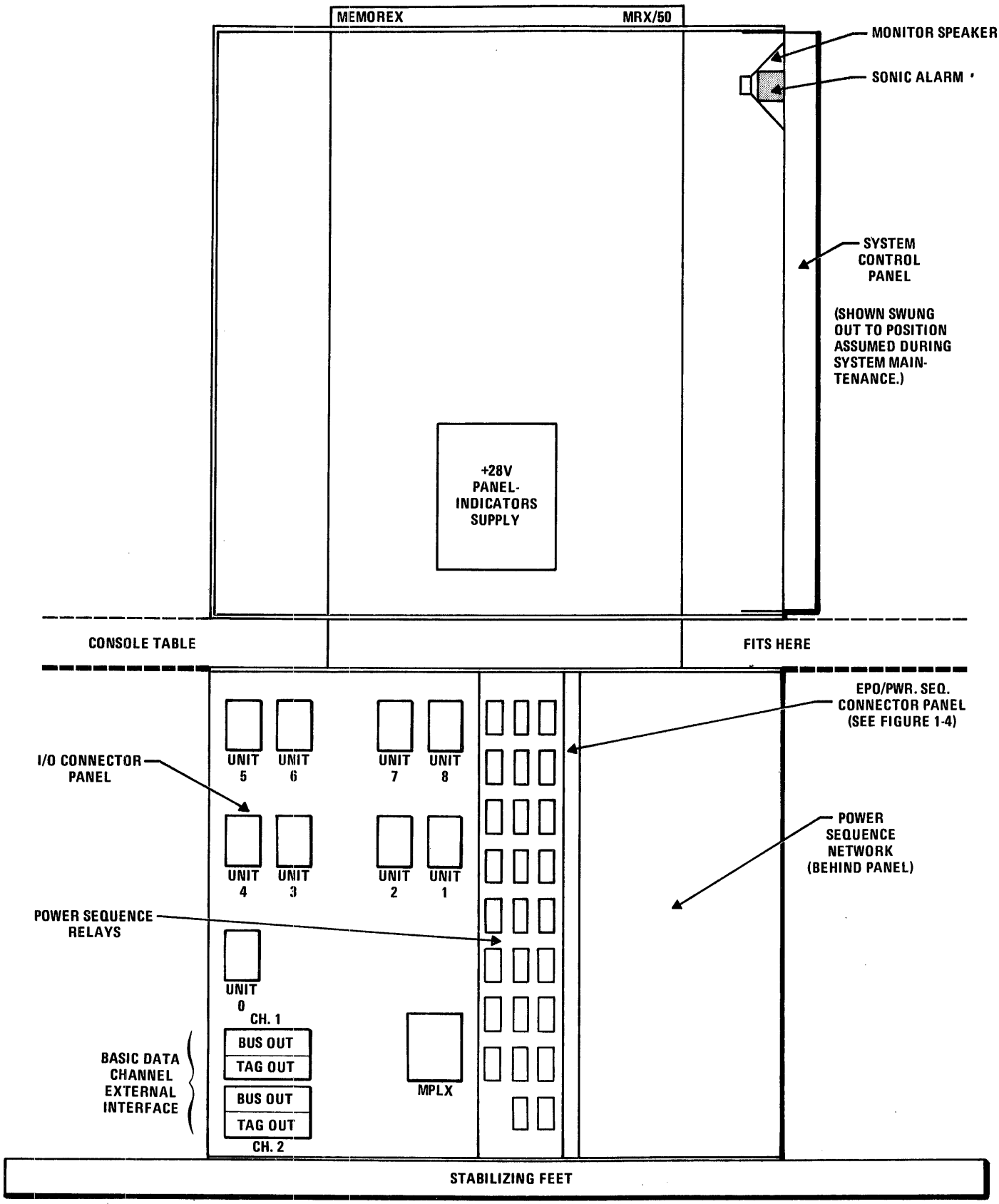



Figure 1-3. Computer Locations, Front

provide data to and from the respective drive units. The "Multiplex" connector routes control signals to each drive in a daisy-chain fashion. Additionally, the "Multiplex" connector returns disc-drive status information to the computer.

It should be pointed out that peripheral devices controlled by integrated adapters associated with processor states 0, 1, and 2 do not attach to the I/O panel. Rather, those connections are made directly to the associated adapter by means of a connector slipped over the wire-wrap pins on the logic chassis. These devices include card readers, card reader/punches, and all external communication-line modems and local terminals.

The power-sequence relays are part of the power sequence network and are used in conjunction with the EPO*/Power Sequence connector panel (shown in Figure 1-4) to sequence any subsystems attached to the Processing Unit.

Figure 1-5 shows a map of the two logic chassis, generalized by function.

*EPO: Emergency Power Off

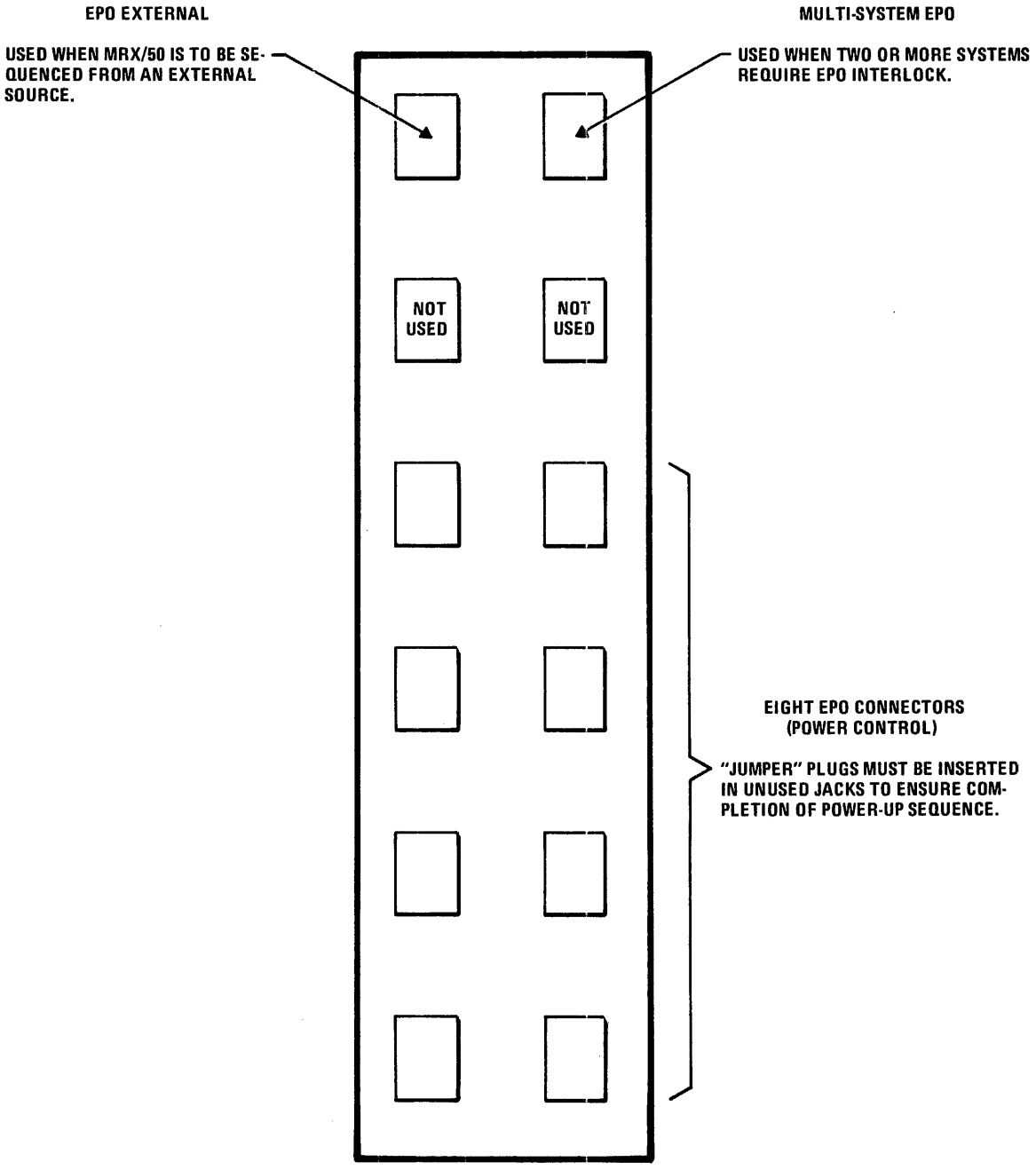



Figure 1-4. EPO/Power Sequence Connector Panel

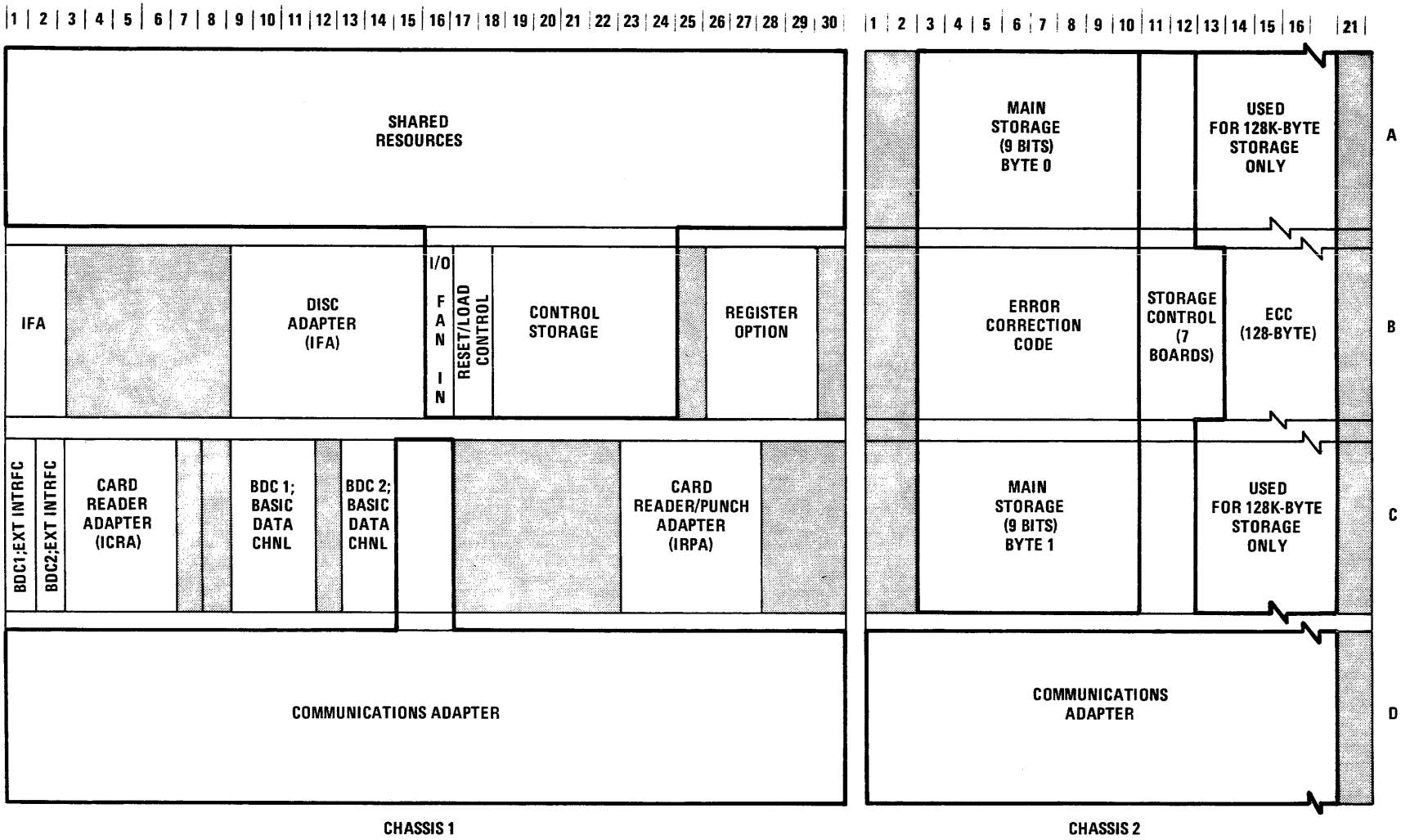



Figure 1-5. Logic Chassis Layout

## FUNCTIONAL DESCRIPTION

### SHARED RESOURCES

Shared resources are elements of the Processing Unit that are shared by all of the processor states. They consist of the resource allocation network (RAN), arithmetic-logical unit (ALU), main storage (MS), and control storage (CS) as described in the following paragraphs.

### Resource Allocation Network

The resource allocation network (RAN) equitably apportions time slices among the eight processor states and has the ability to override normal sequencing to forestall losing I/O data. However, two elements must be added to complete the picture.

1. When the manual entry or display of data — or any other operation involving the System Control Panel — is required, activating the CONSOLE RUN button invokes the Console Mode. This mode acts as a ninth processor state, and receives a time slice in sequence; that is, after the highest-numbered "busy" processor but before the RAN starts its next scanning cycle.

2. Each dynamic MOS storage element in main storage requires refreshing (charging the circuit capacitor) every two milliseconds. A major-cycle counter in the hardware issues a Refresh Request at such times, thereby provoking one of the following actions:

   - If the next time slice does not require a storage reference, the refresh operation occurs simultaneously with the execution of the micro instructions scheduled for that major cycle.

   - If the next time slice requires a storage reference, that storage reference is satisfied, and the very next major cycle is used to perform the refresh operation. Access to the shared resources by the processor state next in line is then deferred until the following time slice. The insertion of refresh cycles increases by less than 2% the total "run time" for a given job.

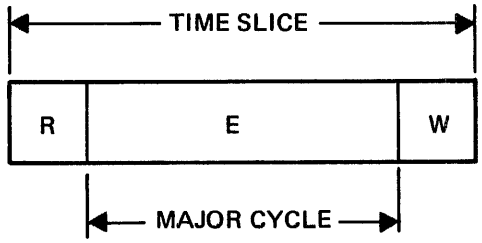
### Time Slicing

The MEMOREX 7300 Processing Unit is always in one or another of several "processor states". Each state is active for a period of time, then is dormant for a period (during which other states are active), then becomes active again. These active and dormant periods are determined by a continuous series of "time slices" generated by the shared resources.

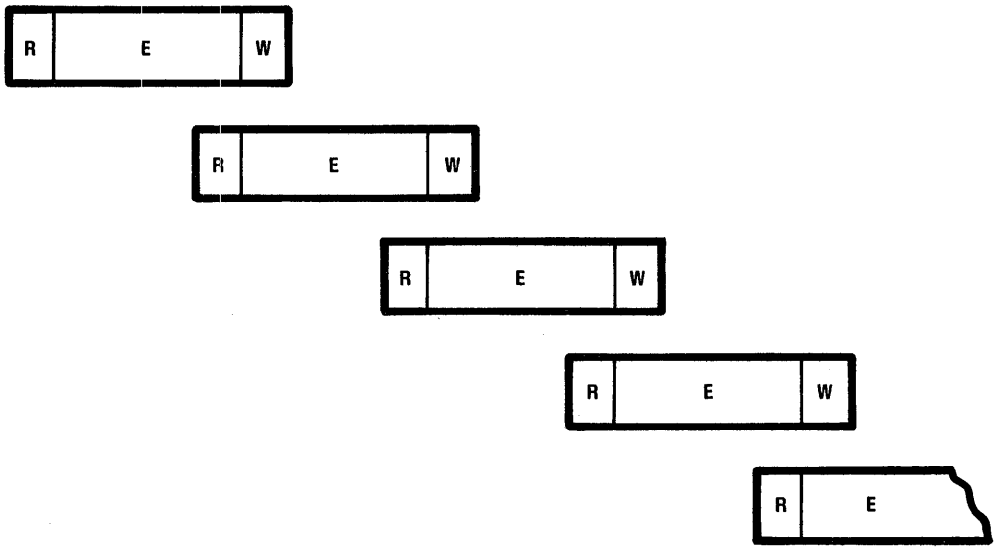
During its assigned time slice, a processor state uses the shared resources to perform a portion of its designated task by executing a series of micro instructions held in Control Storage. To perform a complete task, a state uses many time slices, each of which is separated by time slices that are being used by other states to perform segments of their tasks.

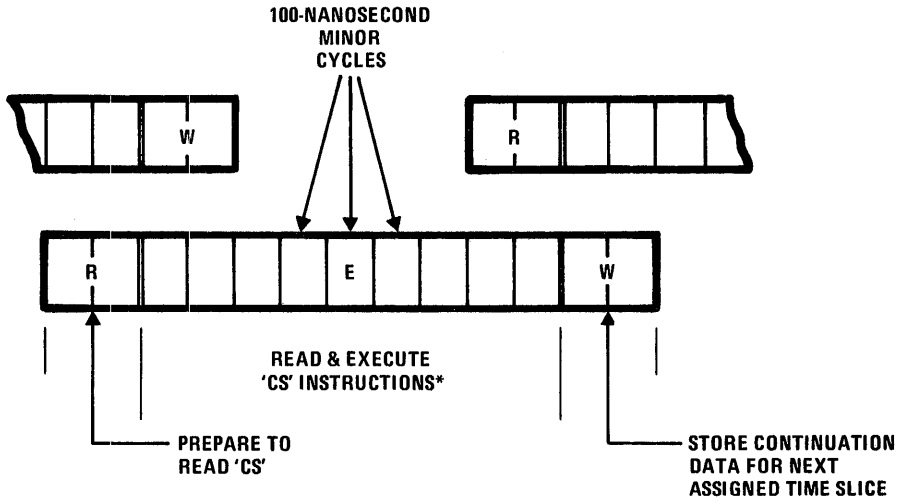
This means that at the beginning of each time slice the processor state must obtain "housekeeping" information pertaining to what is to be accomplished during the current time slice; that is, what micro instructions are to be executed and where those $\mu$I's are located in Control Storage. Moreover, before the end of the time slice the processor state must store this same kind of information so as to have it available at the start of its next assigned time slice.

It is convenient to divide each time slice into three parts: the time used to retrieve (or read) the housekeeping information at the start of the slice is designated 'R'; and the time used at the end of the time slice to store (or write) this information is called 'W'. What's left is the time available to execute ('E') micro instructions. This 'E' portion is called a Major Cycle. The relationship is shown below.




The housekeeping information is held in certain file registers that are addressable by micro instructions. During the housekeeping portions of a time slice the $\mu$I's are not permitted to address these registers; rather, that right is delegated solely to the hardware. Therefore, the hardware-implemented housekeeping functions of one time slice can occur at the same time as the $\mu$I-implemented Execute functions of the adjoining time slices. This decreases the elapsed time for a given series of contiguous time slices by a value equal to the sum of the times for their housekeeping functions. Figure 1-6 shows a series of time slices arranged in such a manner. Note that the time slices overlap, but major cycles do not. Also, the R and W portions coincide with minor cycles for the adjoining major cycles; this provides the timing needed to gate the housekeeping data to and from the register file.




* THE NUMBER OF MINOR CYCLES IN A GIVEN MAJOR CYCLE VARIES AS FOLLOWS:

| | |
|---|---|
| NO STORAGE REFERENCE | 8 |
| STORAGE REFERENCE | 9 |
| STORAGE REFERENCE WITH ECC | 10 |

Figure 1-6. Time Slicing

## Arithmetic – Logical Unit

The ALU configuration for the MRX 7300 is shown in Figure 1-7. A brief description of the components follows.

| | |
|---|---|
| A$\mu$ , B$\mu$ | 16-bit feeder registers |
| Adder | 16-bits; accommodates binary or decimal operations. Double-word (32 bits) quantities are handled via $\mu$I's |
| Compare & Bit Sense Network | 16 bits; performs single-word Compares, as well as individual bit sensing for "0" or "1". Double-word Compares are $\mu$I implemented. |
| Forced Carry Register (FCR) | This single FF, feeding the least significant bit (LSB) of the Adder, has two purposes:<br>1. Provides the correct 2's-complement result for arithmetic operations<br>2. On two-word arithmetic operations, the Link function propagates the carry out of the most significant bit (MSB) of the first word to the LSB of the second word. |
| Inner Carry Register (ICR) | Used primarily for decimal arithmetic, each stage in this 4-bit register represents one 4-bit decimal digit position in the Adder. The ICR bits are used in conjunction with excess-3 operations to provide the correct decimal value for each digit position in the final answer. If an arithmetic operation generated a carry out of a digit group (ICR bit set), the value '3' is added to that group to derive the correctly encoded answer. If no carry were generated, the hexadecimal value 'D' (binary 1101) is added. |
| Constants Generator | The constants generator introduces values into B$\mu$ as directed by certain micro instructions. |
| Shift Network | 32-bits. A$\mu$ provides and receives the upper 16 bits; B$\mu$ provides and receives the lower 16 bits. Shifts are left end-off, with zeros inserted on the right (in B$\mu$). As many as 15 positions can be shifted in a single pass. |

In Figure 1-7, the following symbols are used to represent the sources (or destination) of data to/from the ALU:

| | |
|---|---|
| X, (X) | A file register, or the contents thereof. |
| (MS) | The contents of a main storage address. |
| (D) | The contents of the storage data register (provides input only to main storage). For the ALU, this input indicates that D is being used as a holding (or "transient") register; it does NOT indicate a data source directly from main storage. |

The operands used during ALU operations can come from either a file register or from main storage. The A$\mu$ and B$\mu$ inputs will directly accommodate any operand combination except the case where both operands are obtained from main storage. For that situation, the first operand must be read out of MS and transferred to a transient register. Then, as the second operand from MS is being loaded into A$\mu$, the first operand, (X), may be loaded into B$\mu$.

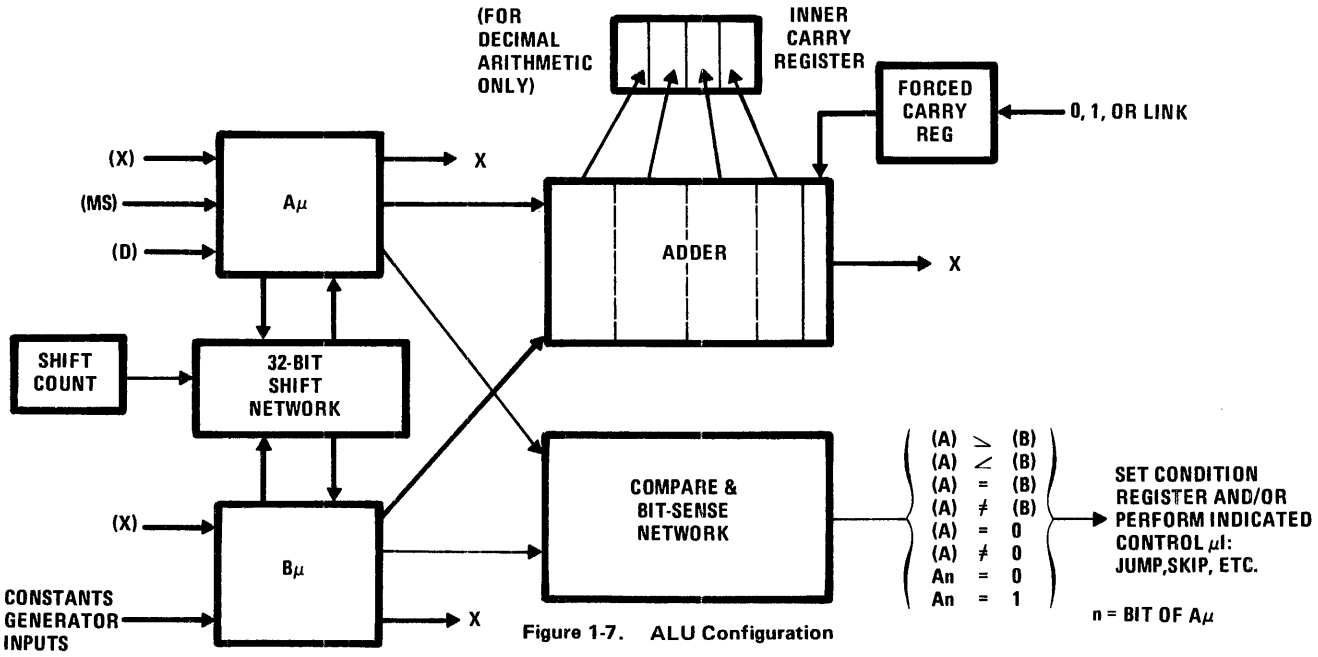



Figure 1-7. ALU Configuration

## Main Storage

Main storage provides high-volume, random-access storage for Machine Language Instruction programs and data. The storage medium consists of MOS integrated circuits arranged on PC boards so that one board provides 4096 (4K) storage bytes – eight bits of data, plus one parity bit. Two such PC boards constitute a "storage unit" of 4K words (8K bytes), which is the minimum size by which the capacity of main storage may be increased.

The ECC option, which increases the length of the storage word to 26 bits, requires three PC boards per storage unit. In addition to the storage units, seven PC boards are provided for control. This number is constant, regardless of storage size or the presence (or absence) of the ECC option.

The storage reference cycle time is nominally 900 nanoseconds. When the ECC option (which requires additional storage cycle time to correct for errors) is present, major cycles used to address main storage will be 1000 nanoseconds long. Non-storage-reference cycles always require 800 nanoseconds.

Figure 1-8 shows the main storage configuration. Two registers, S and D, hold the storage address and the data to be written into that address, respectively. The lower output line from the D register is used on the occasions when the microprogram uses D as a holding register for non-storage operations. The input from D to the Register Option (RO) is used to write information in the RO registers; similarly, the storage data fan-in receives data from the RO. The S register is used to address the registers in the Register Option, as was explained in the MRX/50 **Reference Manual**. Whenever the Relocation and Protection feature is utilized to make a storage reference, the appropriate 4-bit segment tag is appended to the S-register address, providing a 20-bit address to main storage.

## Control Storage

The MEMOREX 7300 micro instruction repertoire consists of 65 micro instructions arranged in microprograms which implement all computer functions, including the execution of machine-language instructions. These microprograms are held in control storage.

Most micro instructions either enable inter-register transfers or perform control functions such as skips, branches, compares, etc. The computer decides the specific function to be performed by examining the contents of shared-resources register $F\mu$, which holds the contents of each micro instruction as it is read out of control storage. The simplified diagram of the control storage network, Figure 1-9, shows these decisions as dashed lines from $F\mu$ to the elements at the right of the figure.

The shared-resources register, $S\mu$, holds the address of the micro instructions to be read out of control storage during the major cycle. An incremental adder, $S\mu + 1$, increases the contents of $S\mu$ by 1 during each minor cycle of the EXECUTE (E) portion of the time slice. If the $\mu$I translated in the $F\mu$ register is a Blockpoint instruction, $S\mu + 1$ is also entered into the shared-resources $P\mu$ Pointer (Pp) register.

Blockpoint instructions mark convenient break-off points in microprograms, most of which take several major cycles to complete. The shared resources, including the microprograms in control storage, are not processor-related. This means that values in the shared resources must be stored – usually in file registers – so as to be available when the microprogram execution continues during the next time slice assigned to that processor state.

The relationship between these "register-file write" – Blockpoint – $\mu$I's and the beginning or end of a major cycle is not fixed; jumps and skips within the microprogram may cause a given $\mu$I in a sequence to be executed at any point in a major cycle. Then, if a Blockpoint did not occur before the end of the major cycle assigned to processor A, the values in the shared-resource registers (all but $P\mu$ in Figure 1-9) could be changed by $\mu$I's executed during the time slice assigned to processor B. That would mean, of course, that continuing processor A's microprogram during its succeeding time slices would produce meaningless results.

This situation is prevented by ensuring that in any $\mu$I sequence in control storage, Blockpoint instructions occur frequently enough to provide a point to which the microprogram can return in the event that the major cycle ended before the results of a microprogram segment could be saved in the executing processor's file registers. In this manner, a continual advance through the microprogram is ensured, even though it may mean repeating some previously executed $\mu$I's (those between Blockpoints) during the next assigned time slice.

But to return to a discussion of Figure 1-9. During the W (write) portion of the time slice, the address of the $\mu$I following the last Blockpoint instruction to be executed is transferred from Pp to $P\mu$. This provides the processor state with the address from which it will read out the first $\mu$I during its next time slice. Then, during the R (read) portion of that time slice, the new starting address is transferred from the processor's $P\mu$ register to $S\mu$ and the microprogram execution continues.

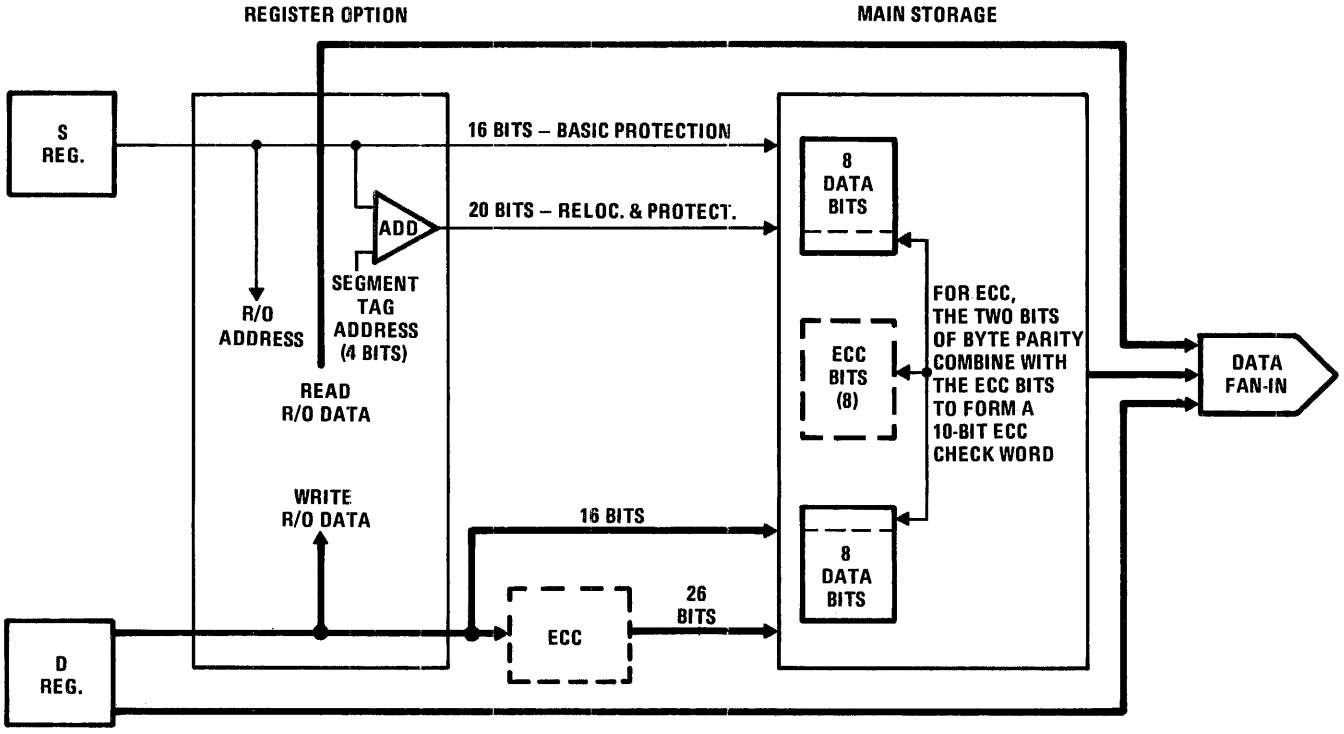



Figure 1-8. Main Storage Configuration

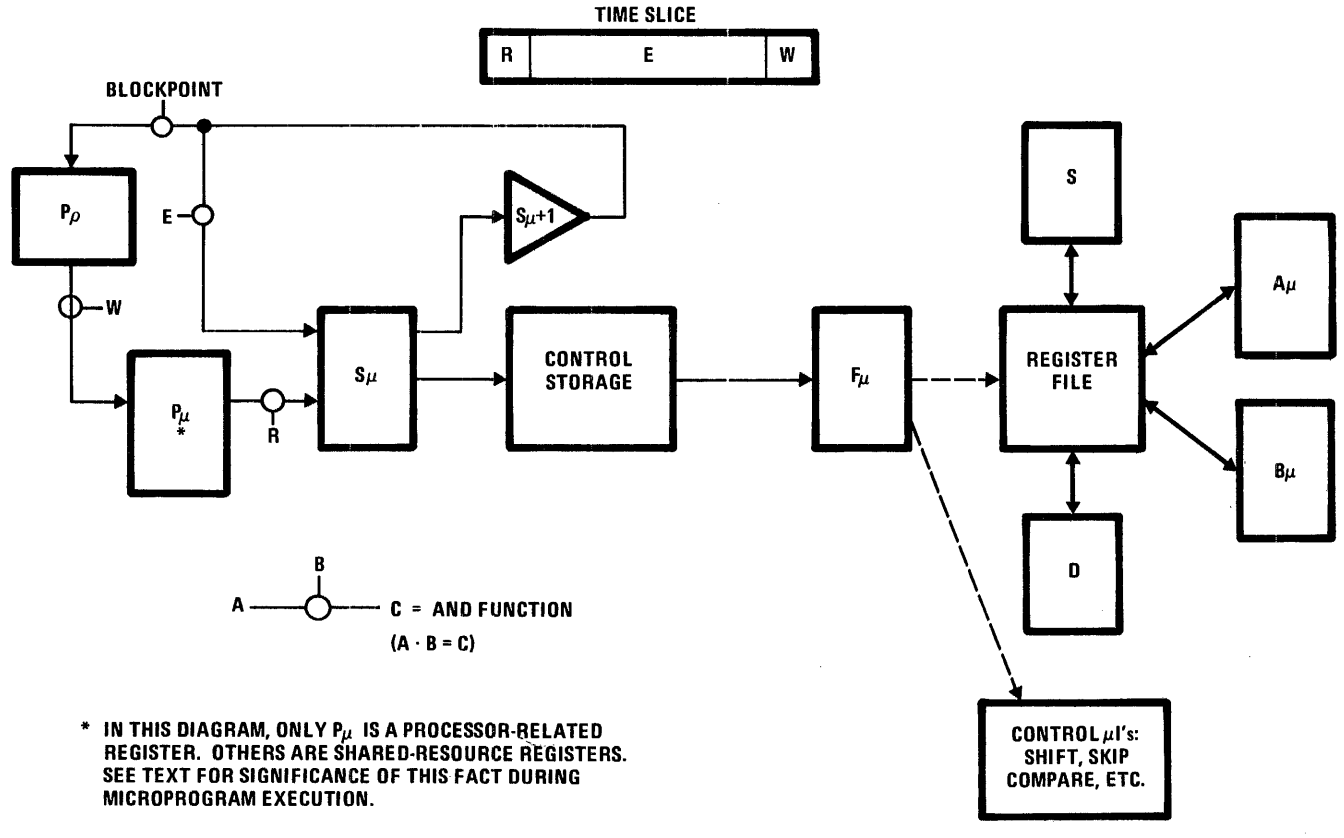



Figure 1-9. Control Storage Network, Simplified

## DEDICATED RESOURCES

The dedicated resources of the Processing Unit comprise eight processor states and the register file. Each processor state has assigned to it a unique set of file registers to which, except in extraordinary circumstances, no other state may gain access. Thus, the register file combines with the various states to provide "dedicated" resources for up to eight discrete operations, as described in the following paragraphs.

### Register File

All registers in the basic and extended file are addressable via micro instructions ($\mu$I's). User programs may use these $\mu$I's to gain direct access to a limited number of registers; certain other registers can be directly addressed only by system programs operating through a processor state that is empowered to execute "restricted" or "privileged" instructions. Finally, the 22 transient registers in the basic file (0A-1F) are directly accessible only via microprograms that have been written to implement the execution of processing-unit instructions. For example, a microprogram may use a transient register to store the partial results of a Multiply instruction that could not be completed in one major cycle. But the programmer who includes that Multiply instruction as part of his User or System program need not be concerned about (and indeed, may be oblivious of) the existence of the transient registers.

The file registers are divided into four groups, as shown in Figure 1-10, and each group is designed for optimum efficiency within its particular usage. Table 1-2, describes the characteristics of the integrated circuit (IC) types used in the groups, even the registers within a group may employ different IC's. The constraints upon each register group are discussed below.

#### Basic Register File

Generally, these registers are addressable only via micro instructions. Also, they are set or cleared as entities; that is, they are not bit-oriented. The exception to these rules is the Condition register. The leftmost eight bits of this register are affected simultaneously during a Compare $\mu$I. Reading any of the registers in the basic file is solely a $\mu$I function.

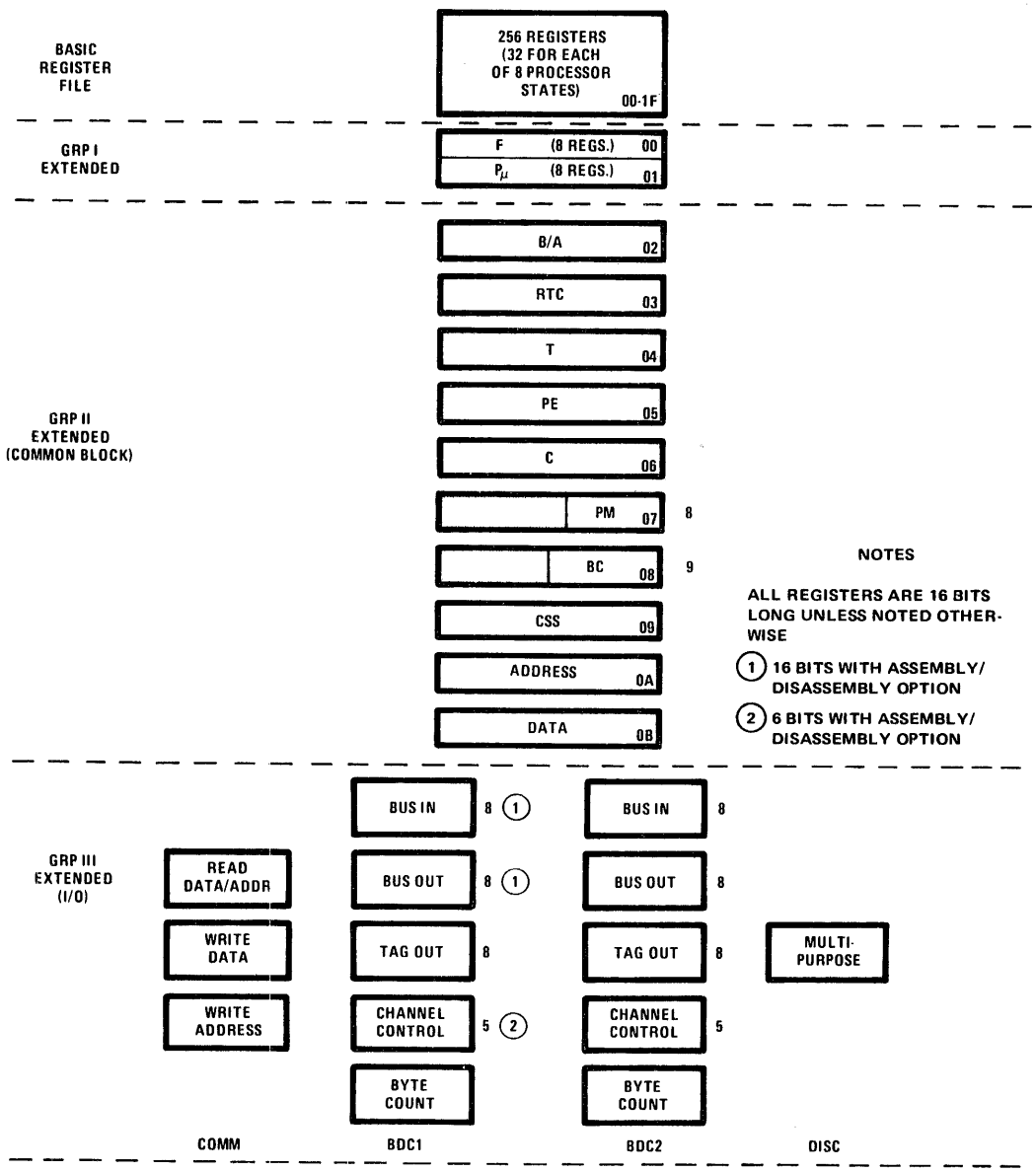



Figure 1-10. Register File Grouping

### Group I, Extended

The two registers in this group, F and P$\mu$, are addressed by the hardware during the R and W portions of the time slice. As described previously, these "housekeeping" functions overlap the E portions of adjoining time slices. The adjoining time slice may be addressing other file registers via minor-cycle-oriented $\mu$I's at the same time that the hardware needs to have access to F and P$\mu$. Because of this, F and P$\mu$ are set apart in this special group. Of course, F and P$\mu$ may also be addressed by $\mu$I's, but NOT during the minor cycles that coincide with the overlapping R and W housekeeping functions.

### Group II, Extended

These "common block" registers serve all processor states and encompass divergent functions. The Console Address, Console Data, RTC, and PE registers are set asynchronously by the hardware; that is, their setting is not related to major-cycle timing. Console Address and Console Data, for example, are set manually from the System Control Panel — definitely an asynchronous function. The setting of BUSY-ACTIVE by the hardware (via I/O Request or Attention) is also asynchronous, although the $\mu$I-implemented set/reset is synchronous. The CSS register, on the other hand, is set only by the hardware, but such setting is always related to minor-cycle timing.

A second important distinction in this register group is that, of the ten registers, four are definitely bit-oriented with respect to setting and clearing: B/A, T, C, and PM. All of the registers, however, are read synchronously. Even Console Address and Console Data, the reading of which is initiated by pressing the CONSOLE RUN button, are read during the major cycle assigned to the Console state.

### Group III, Extended

The registers in this group are contained on the PC boards for the associated I/O processor state. The inbound (read) registers are set asynchronously by the I/O device and read synchronously by the processing unit. The outbound (write) registers are set synchronously by the processing unit and read asynchronously by the I/O device. (The device, of course, acts through an adapter or controller, but this does not alter the asynchronous character of the operation.)

**Table 1-2. Register File IC's**

| Register Group | Register Name or Number | Description | Quantity | Comments |
|---|---|---|---|---|
| Basic File | 0-32 | 1 x 256-bit addressable array; 1 bit of 256 registers (32 x 8) per IC | 16 | 8 Register Select Lines:<br>3 - Processor (1 of 8)<br>5 - Reg. No. (1 of 32) |
| Grp I Extended | F(0),P(1) | 4 x 16-bit addressable array; 4 bits of Reg 0, 1 for 8 processors per IC | 4 | 4 Register Select Lines:<br>3 - Processor (1 of 8)<br>1 - Reg. No. (1 of 2) |
| Grp II Extended | Console Data (N), Console Address (M) | Dual D-type FF's, edge-triggered; preset and preclear inputs | 8 each | 2 bits per IC. |
| | Busy/Active (B/A) | Same as above | 8 | |
| | Real-Time Clock (RTC) | 4-bit binary counter | 4 | Overflow every 1092 seconds. |
| | Others | 4 FF's per IC; no preset or preclear inputs | 4 each | Exceptions: PM - 2 IC's (8 bits)<br>BC - 3 IC's (9 bits) |
| Grp III Extended | Comm Processor | | | |
| | Read Data/Address | Quad 2-input mplxr | 4 | 4 bits of Data or Addr per IC |
| | Write Data | 4 FF's per IC; no preset or pre-clear inputs | 4 | |
| | Write Address | | 4 | |
| | Basic Data Channels 1 & 2: | | | |
| | Bus-Out | 4-bit bistable latch | 2 each | 8-bit, double-rank registers; quantity doubled to allow 16-bit transfers with presence of BDC1 |
| | Bus-In | See "Write Data" | 2 each | |
| | Tag-Out | See "Write Data" | 2 | |
| | Channel Control | See "Console Data" | 3 | 4 with Assy/Disassy option. |
| | Byte Count | 4-bit binary counter | 4 | |
| | Disc Processor: | See "Write Data" | 4 | Multi-purpose register; function depends upon reg. addr. selected. |

## Communications Processor (COMM)

The integrated communications adapter (ICA) in the Comm processor, Figure 1-11, consists of four Common Control PC boards that provide the interface between individual communication-line adapters and the processing unit. The I/O registers for state 0 are contained on these Common Control boards. The two Read registers are combined in dual-input multiplexer IC's, as shown in Table 1-2 of the Register File discussion. Write Address and Write Data are each discrete registers. In addition to the functions implied in Figure 1-11, the common control boards perform the recognition of certain message-framing control characters during the receipt of both synchronous and asynchronous transmissions.

Each communication line requires a Line Adapter (LA) and a Signal Conditioner (SC). The latter converts internal logic signals to those required by external equipment; this SC may be a level translator, used for local terminals and external modems, or it may be an internal modem. Present internal modems are the MRX 1220 (asynchronous mode only), equivalent to a Western Electric* 103A data set, and the MRX 1222 (synchronous or asynchronous modes), equivalent to a Western Electric 202C.

The Line Adapter comprises an Outbound Buffer and an Inbound Multiplexer, housed on separate PC boards.

*Tradename of American Telephone and Telegraph Corp.

Five combinations of Line Adapters are available:

- Asynchronous eight-level
- Synchronous EBCDIC, basic
- Synchronous ASCII, basic
- Synchronous EBCDIC, code transparent
- Synchronous ASCII, code transparent

Some logic for the code transparent modes is contained on a special Signal Conditioner board. This means that code transparent transmissions must use an external modem. With this limitation, the Line Adapters are interchangeable (that is, the 2-board LA sets are plug compatible), as are the Signal Conditioners.

A direct data path is provided between the Outbound Buffer and the Input Multiplexer for diagnostic testing. During such tests, data from the Write registers is sent to the Signal Conditioner, but is also routed through the Input Multiplexer to the 16-character queue, and thence to the Read registers. This "Loop Test" mode may be applied selectively to any Line Adapter, permitting diagnostic testing of the selected line without disturbing the normal transmit-receive capability of the other lines.

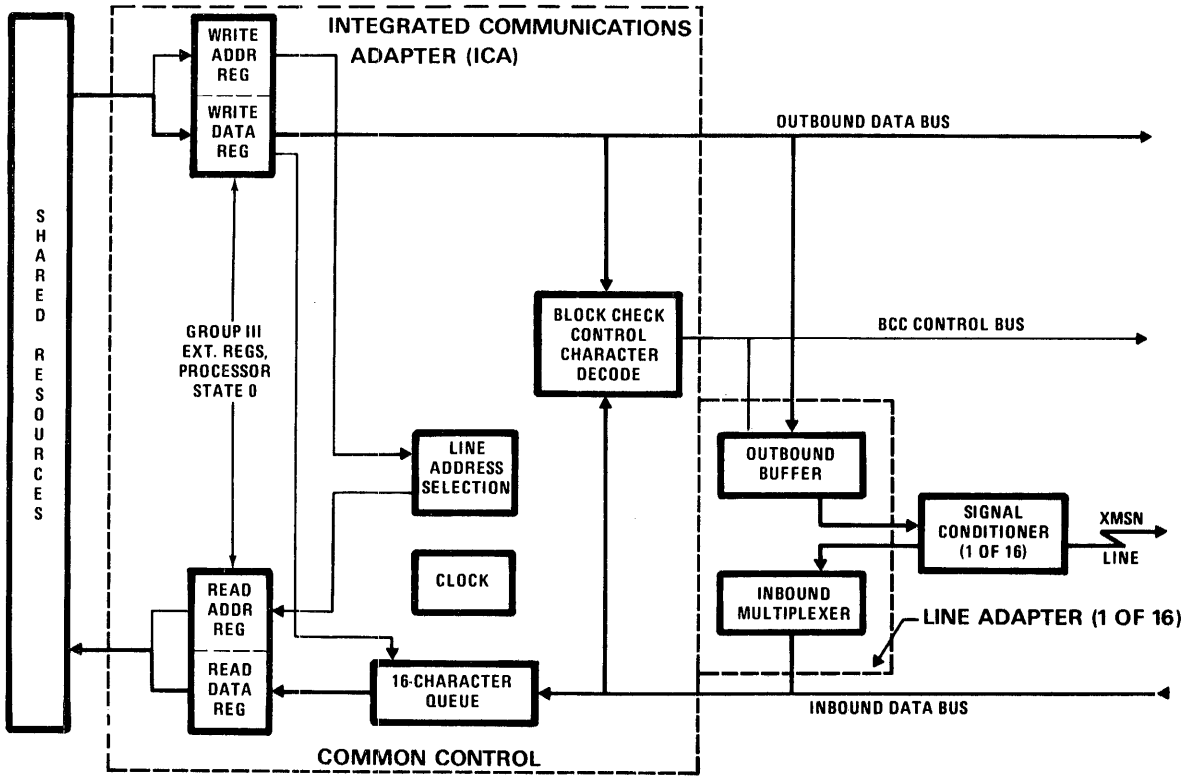



Figure 1-11. Communications Processor

## Basic Data Channel 2 (BDC2)

Processor state 2 and its associated basic data channel logic constitute BDC2. The I/O registers for state 2 are part of the basic data channel logic contained on two PC boards, as shown in Figure 1-12. The Bus In, Bus Out, and Tag Out registers, along with the Tag In lines, perform data and control signal interface transfer functions. In addition, the Channel Control and Byte Count registers with their associated logic enable BDC2 to perform the following:

- Terminate data transfers by use of the Byte Count register.
- Indicate to processor state 2 incorrect control signal progression and the status of the channel.
- Generate and check parity on all data transferred to or from, respectively, the external control units.
- Perform turn-around data transfer for diagnostic testing.
- Avoid "lost data" from non-buffered devices on the external channel by initiating a priority condition whenever software has instituted one of the Priority modes for processor state 2.

Each integrated adapter contains the logic necessary to control its particular I/O device and to adapt the device characteristics to the BDC2 format. In addition to controlling the I/O device, each adapter contains a buffer register for intermediate storage of data between the device and BDC2. The buffer register feature effects data transfers with BDC2 at electronic speeds rather than at the I/O device speed and eliminates the possibility of losing data.

The card reader adapter (ICRA) resides on four PC boards, with the card reader/punch adapter (IRPA) requiring five PC boards.

The external interface board contains the level-shifting transmitters and receivers required to communicate with external control units. A maximum of eight control units meeting the BDC2 interface requirements – this includes IBM* 360/370 devices – may be connected to the external channel.

*Trademark of International Buisness Machine Corp.

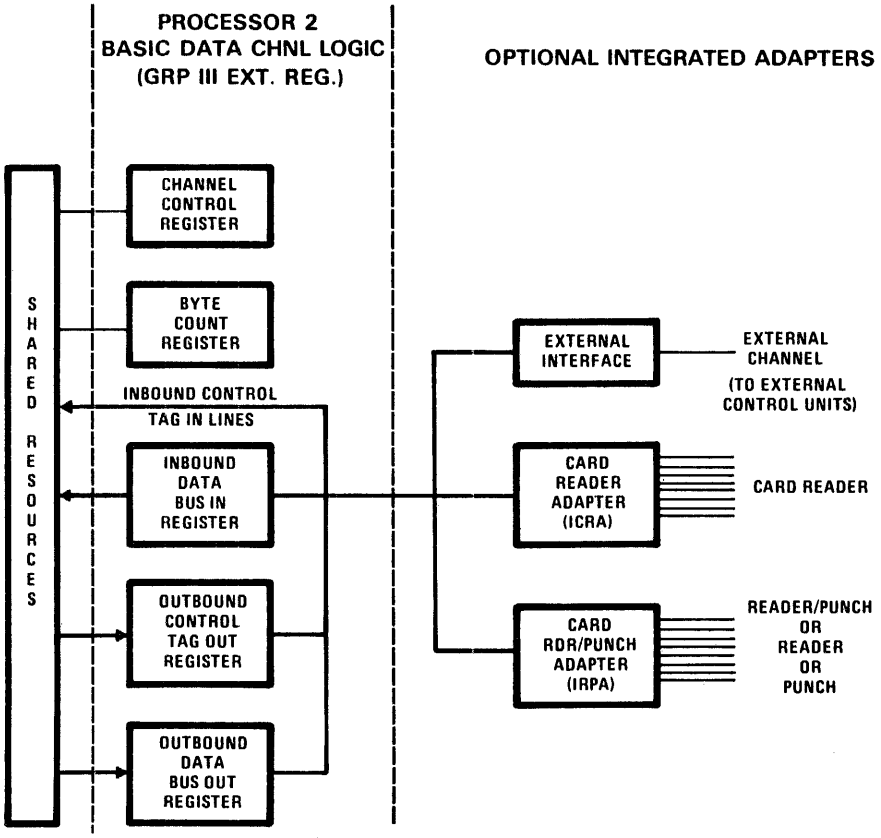



Figure 1-12. Basic Data Channel 2

## Basic Data Channel 1 (BDC1)

Processor state 1 and its associated basic data channel logic constitute BDC1. The I/O registers for state 1 are part of the basic data channel logic the same as for BDC2, as shown in Figure 1-13. The basic data channel registers and logic provide BDC1 with the same performance features as those listed for BDC2, except that there are no integrated adapters for BDC1.

Additionally, the assembly/disassembly logic contained on another PC board enables BDC1 to perform 16-bit data transfers with main storage, by doubling the Bus In and Bus Out register capacity. Although the data transfers with external control units are still 8 bits, this logic enables BDC1 to increase its data rate by only requiring one main storage transfer for every two data transfers with an external controller.

The external interface board performs the same function as on BDC2. Here again, up to eight control units meeting BDC1 interface requirements, which again are IBM 360/370 compatible, may be connected to the external channel.

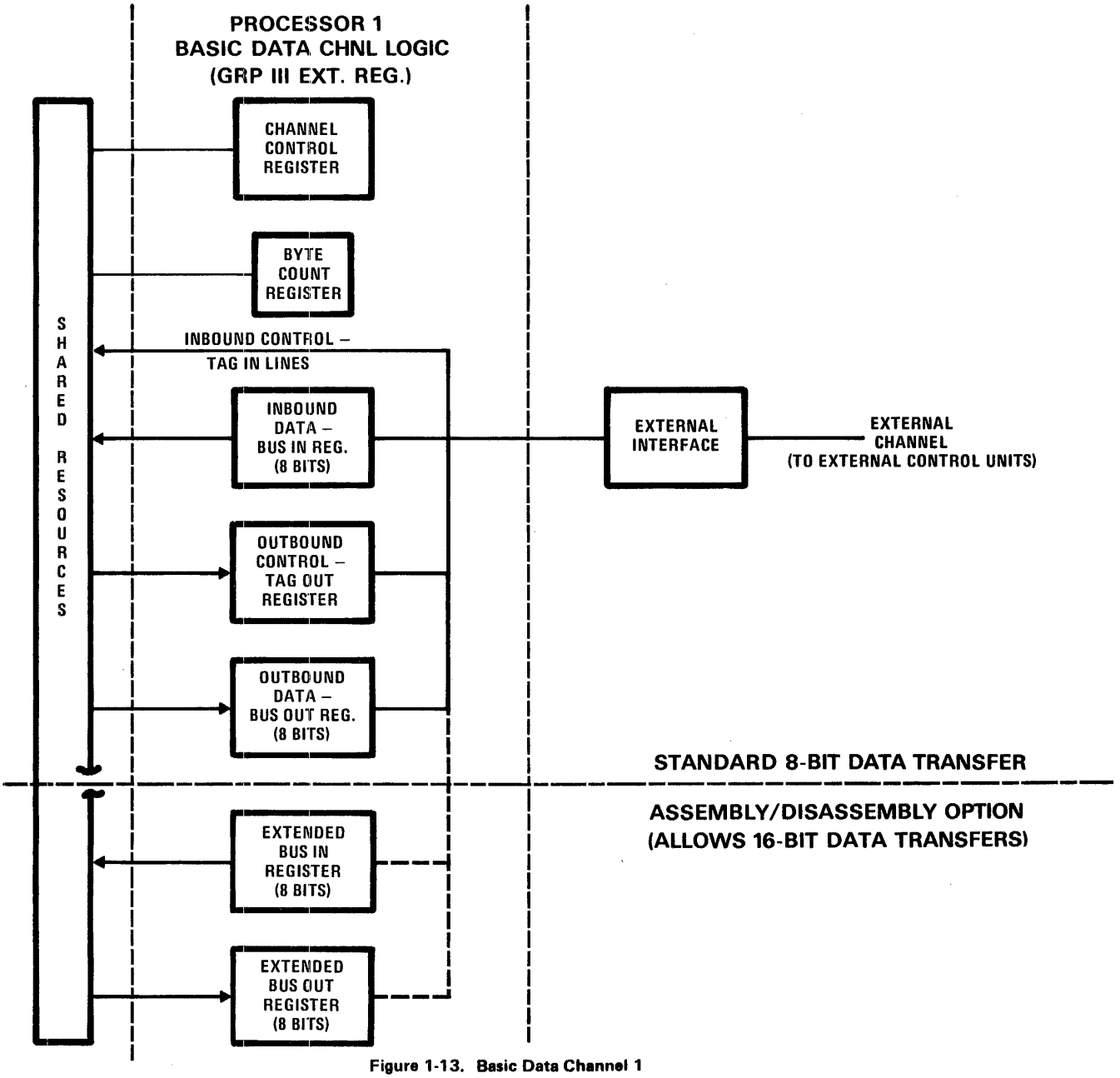



**Figure 1-13. Basic Data Channel 1**

## Disc Processor

Processor state 3 contains the integrated file adapter (IFA) that receives instructions and data from the shared resources, and transmits bi-directional information to and from as many as eight disc drives. A ninth drive, connected to the IFA electrically but not logically, may be used as a spare. A single Group III extended register, located within the IFA, operates as a combined data and instruction register. The nature of the information this register will contain is determined by the register address used to select it. The IFA is shown in Figure 1-14.

Information to be written on the disc is clocked from the extended register into the Assembly/Disassembly register, where it is converted to serial format. the Read/Write logic then intersperses the serialized information bits with clock pulses such that, within a given time frame, a "1" is represented by two pulses (a clock pulse followed by an information pulse) and a "0" is represented by only one pulse (clock pulse alone).

During Read operations, data from the selected drive is sent to the Variable Frequency Oscillator (VFO) prior to being transmitted to the Assembly/Disassembly register. The VFO establishes the Read time frame by "locking in" on the clock pulses received from the disc. Thus, the VFO permits a certain degree of speed variation or data migration during the write-read operation with no fear of losing data through loss of synchronization.

The VFO also strips the clock pulses from the disc pattern so that the information passed to the Assembly/Disassembly register is pure data. There, the data is converted from serial to parallel, and sent to shared resources via the Group III register.

Special logic, not shown in Figure 1-14, enables the IFA to perform a hardware-controlled read operation in response to switch settings on the System Control Panel. This logic is used to load control storage and/or main storage during Reset/Load and Autoload sequences.

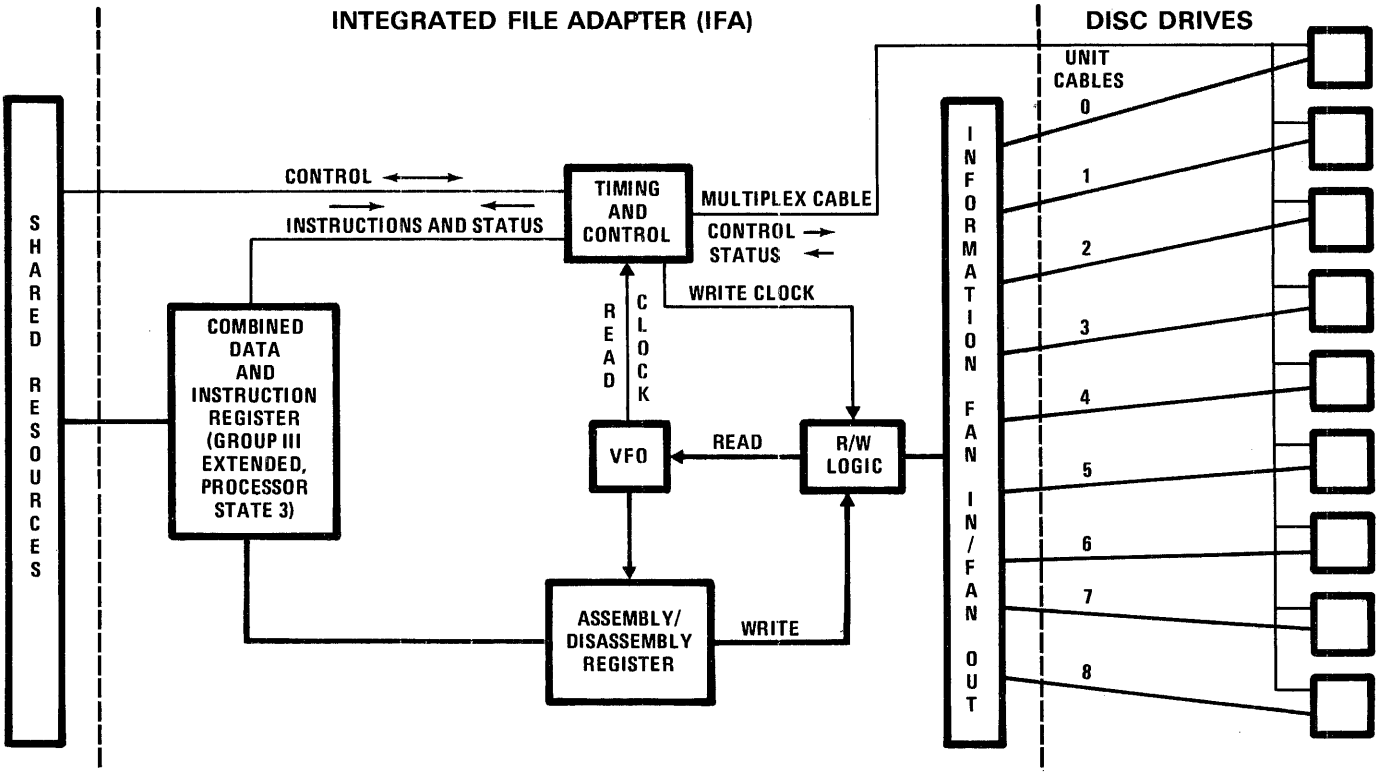



**Figure 1-14. Disc Processor**

## Executive Processor (EXEC)

Figure 1-15 shows how the Exec, processor state 4, interacts with other processor states to monitor system operations. With the exception of the hard-wired I/O REQUEST and ATTENTION signals to the four I/O processors, all lines in the figure represent μl-implemented requests. Although these requests are not strictly hardware considerations, a brief explanation of each may help to under stand the manner in which Exec functions.

As an example, consider the case where state 6 is performing a User program and that, at some point in its execution, the program calls for data from a card reader attached to BDC2. At this time the User program will provide a Service Request (SR) MLI. As state 6 reads this SR instruction, it sets the SR bit in its Condition register, turns on the Exec (via Exec's B/A bits), and turns itself off.

The Exec then scans the Condition registers for all eight processor states; in this example, it sees that state 6 issued a Service Request. Thereupon, Exec goes to the Program Address register for state 6 to locate the SR instruction that state 6 had just executed, and from it determines the class of request – in this case, a request for I/O.

From the User program, Exec obtains the necessary parameters (device type, byte count, where data will be stored, etc.), deposits this information in the work queue for state 2, and issues an "Executive Request" to BDC2, which then executes the I/O Driver indicated by its work queue.

After the required data has been transferred, the I/O driver in state 2 issues a "Return Control" request to Exec. Exec, in turn, passes this "I/O completed" information, via another Executive Request, to state 6. State 6 then resumes its execution of the User program.

The question may arise as to why, since *Service Request* is a general-purpose MLI equally applicable to any processor. Figure 1-15 shows this request as emanating from only states 5, 6, and 7.

Answer: The basic operating system was designed that way. However, if a particular customer does not use state 1 (for example) for I/O processing, the operating system which is always tailored to customer requirements, may very well use state 1 as another General-Purpose processor. In this instance, it could be advantageous for state 1 to utilize the SR instruction. The fact remains that the I/O-oriented "return control" request is implemented as part of the specialized I/O driver routines, and does not require SR to provide the desired functions.

During discussions in the Reference Manual, it was convenient to consider hardware and microprogramming as one and the same, inasmuch as both contributed to the end result; namely, the successful completion of a machine-language instruction. From now on, we will be considering these elements – hardware, microprograms (firmware), and MLI's (software) – as separate entities.

We have seen, for example, that the four I/O processors contain specialized hardware. With some minor exceptions, this does not apply to the four non-I/O states; they are essentially similar from a hardware standpoint, and it is the microprograms that primarily determine the role of these processors in the scheme of things.

A case in point: apart from the unique RTC (hardware) input to Exec's Busy flip-flop, as shown in the 7300 Processing Unit Reference Manual, the Exec derives its system-monitoring capability from the microprograms. A μl sequence, not hardware, determines whether or not a given MLI is restricted to processor state 4. And it is not the hardware, but the microprogram (via implementation of the SR instruction) that turns program control over to Exec before each I/O operation.

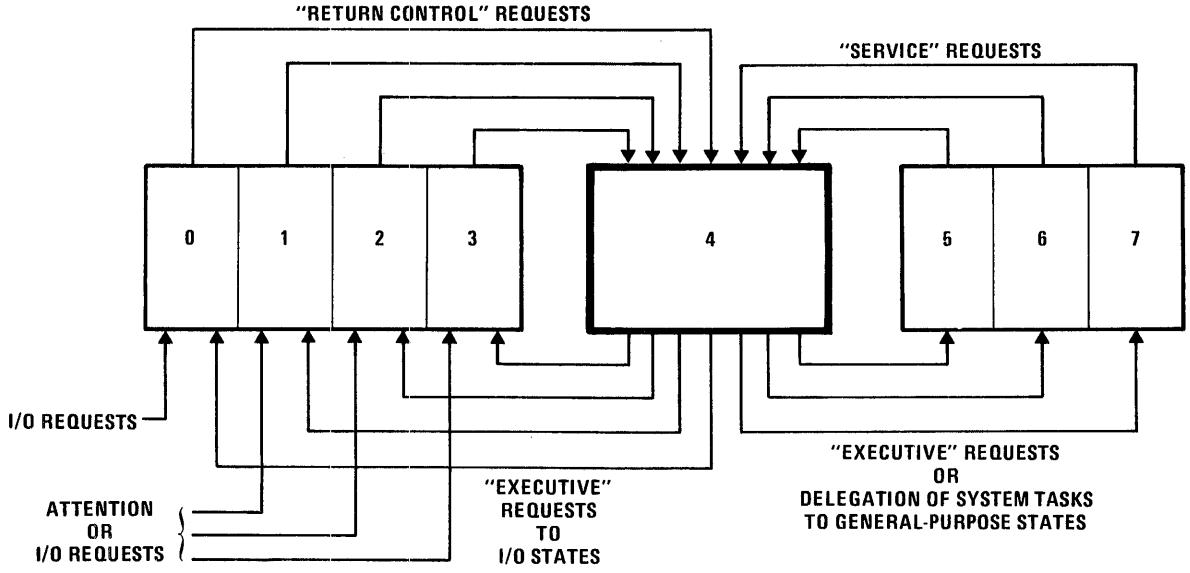



Figure 1-15. Interaction of Exec with Other Processor States

# COMMENTS FORM

**7300 Processing Unit Design Description Manual Volume I: Overview (2501.001)**

Please send us your comments, to help us produce better publications. Use the space below to qualify your responses to the following questions, if you wish, or to comment on other aspects of the publication. Please use specific page and paragraph/line references where appropriate. All comments become the property of the Memorex Corporation.

| | Yes | No |
|---|---|---|
| • Is the material: | | |
| Easy to understand? . . . . . | ☐ | ☐ |
| Conveniently organized? . . . . . | ☐ | ☐ |
| Complete? . . . . . | ☐ | ☐ |
| Well illustrated? . . . . . | ☐ | ☐ |
| Accurate? . . . . . | ☐ | ☐ |
| Suitable for its intended audience? . . . . . | ☐ | ☐ |
| Adequately indexed? . . . . . | ☐ | ☐ |

- For what purpose did you use this publication? (reference, general interest, etc.)

  _____

- Please state your department's function: _____

  _____

- Please check specific criticism(s), give page number(s), and explain below:

  - ☐ Clarification on page(s) _____
  - ☐ Addition on page(s) _____
  - ☐ Deletion on page(s) _____
  - ☐ Error on page(s) _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

MEMOREX

First Class

Permit No. 250
Santa Clara
California 95050

**Business Reply Mail**

No Postage Necessary if Mailed in the United States

Postage Will be Paid By

**Memorex Corporation**

Santa Clara Publications
Mail Stop 00-21
1200 Memorex Drive
Santa Clara, California 95052

Thank you for your information.....

Our goal is to provide better, more useful manuals, and your comments will help us to do so.

.....Memorex Publications