41618400

CONTROL DATA CORPORATION

BASIC SYSTEM CONTROLLER MODEL 65109 HARDWARE REFERENCE MANUAL

PRELIMINARY EDITION

CONTROL DATA®
CYBER 170 SYSTEMS
CYBER 70 COMPUTER SYSTEMS
MODELS 71, 72, 73, 74
6000 COMPUTER SYSTEMS

REVISION RECORD

REVISION	DESCRIPTION
01	Original release describing the characteristics of the CONTROL DATA 65109 (BSC) and (MPM) at the micro code level.
(76-05-15)	
02	Engineering technical changes have been made since original publication. This edition obsoletes all previous editions.
(77-03-11)	
Publication No. 41618400	

REVISION LETTERS I, O, Q AND X ARE NOT USED

Address comments concerning this manual to:

CONTROL DATA CORPORATION

215 MOFFETT PARK DRIVE
SUNNYVALE, CALIFORNIA 94086

or use Comment Sheet in the back of this manual

PREFACE

This manual describes the functional, operational and programming characteristics of the CONTROL DATA 65109 Basic System Controller (BSC) and associated Multi Port Memory (MPM) at the micro code level. Several products can be used with the BSC/MPM to expand its capability. These products are:

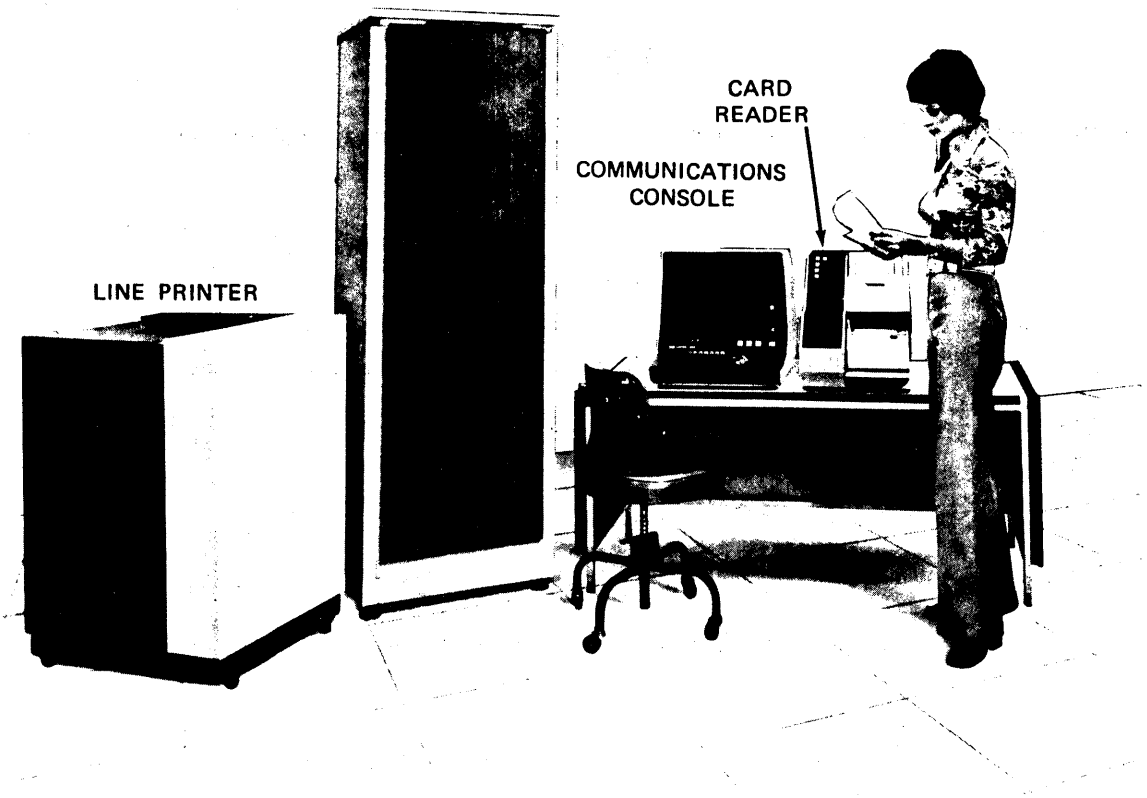
Micro Code Assembler/Simulator (executes on CYBER 170 Series System)

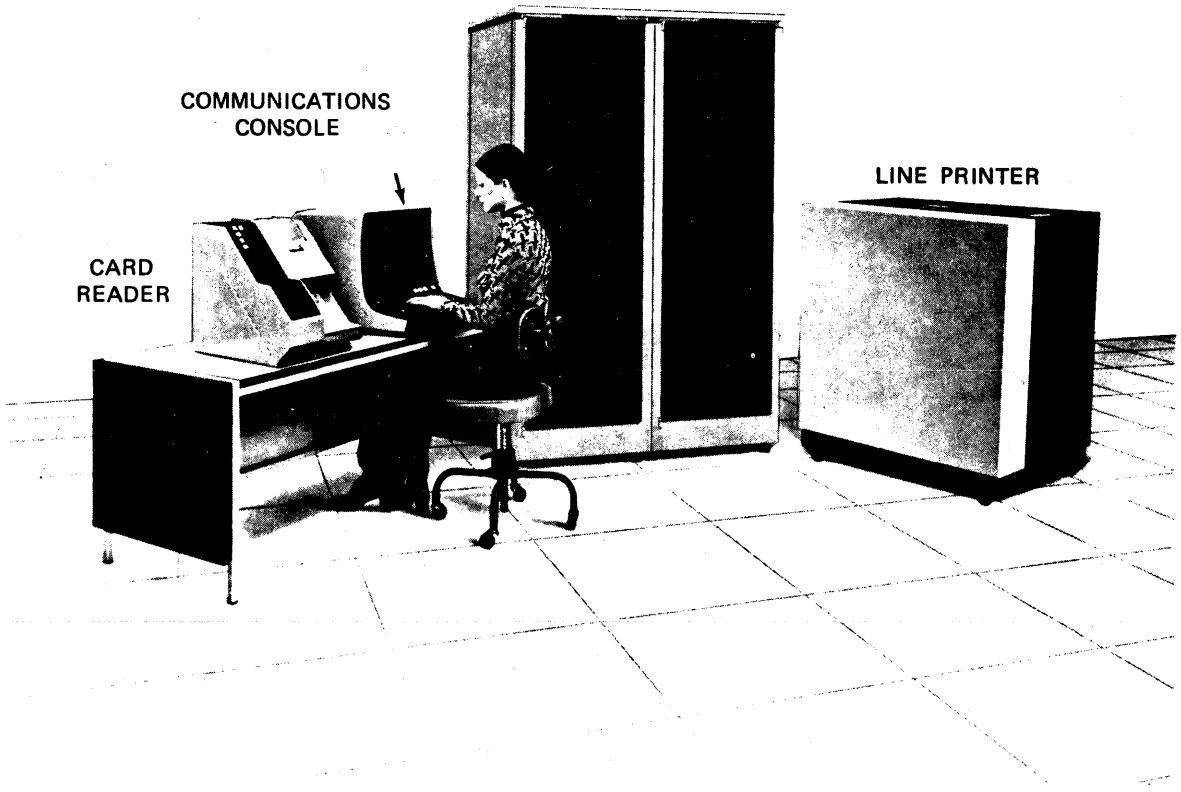
MP-60 Firmware Program (causes BSC/MPM to emulate the MP-60 computer)

MPX Operating System (executes on MP-60 computer)

- Executive O.S.
- Library Maintenance and Update Utilities
- COMPASS (MP-60 Macro-Assembler)
- FORTRAN
- Micro Code Assembler

This manual is supplemented by separate customer engineer manuals which describe installation, checkout, operation and maintenance of the system.





APPLICATIONS CONFIGURED BSCs SHOWN WITH SOME OPTIONAL PERIPHERALS

CONTENTS

CONTENTS (Cont'd.)

FIGURES

FIGURES (Cont'd.)

TABLES

TABLES (Cont'd.)

GLOSSARY

SYSTEM DESCRIPTION 1

INTRODUCTION

This specification describes the Basic System Controller (BSC), the Multi Port Memory (MPM)* and a number of associated interfaces and controllers. The BSC incorporates a 32-bit medium scale, stored program, parallel mode digital computer: the CDC Basic Microprogrammable Processor (MP-32). The BSC is a microprogrammable multilevel processor utilizing high-speed local micromemory for micro-program execution and the external MOS main memory for macro-program execution. The BSC/MPM with associated interfaces and controllers can be configured in a variety of ways to provide a powerful, flexible solution to a wide spectrum of applications.

*Eight Port Memory (8PM) is used synonymously in this manual.

TABLE 1-1. BSC/MPM GENERAL CHARACTERISTICS

BASIC CONFIGURATION	
Processor	
Type	General purpose, microprogrammable digital processor.
Organization	Register oriented or file oriented.
Word Length	32-bits.
Instruction Word	32-bit format, two microinstructions per micromemory word.
Micromemory Type	Semiconductor read/write memory (RAM).
Micromemory Size	512 words, 32-bit instructions; maximum of 8192 instructions.
Micromemory Cycle Time	168 nanoseconds cycle time; 70 nanoseconds access time.
Arithmetic	Binary with dynamic selection of 1s or 2s complement mode. Up to four parallel unrelated operations possible in one microinstruction.
Execution Time	Typical - 168 nanoseconds (excluding shifts). Maximum - 504 nanoseconds (excluding shifts). Shifts - 280 + 56 (n) nanoseconds (Where n = the number of shifts).
Input/Output (I/O)	
Interfaces	A/Q MO5 Four-Port Maintenance panel RS232 compatible console
Operator Input Devices	CDC Model 1811-1, 1811-2 Conversational Display

TABLE 1-1. BSC/MPM GENERAL CHARACTERISTICS (Cont'd)

Mechanical	
Hardware	Modular.
Construction	RETMA 19-inch, rack mountable.
Dimensions	Logic Chassis: Height - 18.5 inches. Width - 17.5 inches. Depth - 16.0 inches. Power Supply Chassis: Height - 8.75 inches. Width - 17.5 inches. Depth - 16.0 inches.
Weight	Logic Chassis - 40 pounds (approximately). Power Supply - 50 pounds (approximately). +5V supply only.
Input Power	110 Volts, 50/60 Hz, Single Phase.
Main Memory	
Type: NMOS	NMOS Memory - Available in 32K x 18 bit Boards. Chip Access Time - 250 ns Chip Cycle Time - 425 ns Maximum Cycle Times Measured Request to Request Cold Cycle - 900 ns --- Taking advantage of look-ahead (L.A.) capabibility. Warm Cycle - Read - 725 ns Write - 600 ns Taking advantage of a single L.A. register. Hot Cycle - 600 ns --- Taking advantage of multiple L.A. registers. Memory Management Interface - Interfaces to a maximum of 16 memory chassis containing 256K 32-bit words each.
Direct Memory Access	Eight 32-bit CPU/Device Ports per MPM chassis. Four 16-bit I/O Ports (through a CPU) per BSC.

FUNCTIONAL CHARACTERISTICS

The 65109 series performs as a multilevel processor. The control section consists of a micromemory, a transform function and a microlevel control logic. The BSC may be set up in a number of ways, providing the user with the ability to adapt the 65109 to meet a particular requirement. Figure 1-1. shows the basic differences in organization between the multilevel processor and the conventional processor.

This design technique provides the advantage of versatility by making a portion of the hardware variable. Each user can thus design his transform logic and micromemory program to perform his specialized task. In this way, by modifying a small portion of the hardware, a user can utilize BSC units for such functions as emulators, algorithm processors, peripheral controllers and preprocessors.

Typical applications include a complex algorithm processor, emulator or system controller. A typical multiprocessor application is a real-time telemetry pre-processor sub-system. A BSC system of this type is used as an input subsystem to a large mainframe system.

The following is a list of typical functions that the BSC and its associated peripheral controllers and special products are capable of performing:

BSC is a general-purpose digital processor:

- Supports program execution under a multiprogramming real-time monitor operating system of the MP-60 type.
- Controls external peripheral devices via plug-in peripheral controllers.
- Interface to large external high-speed memory.

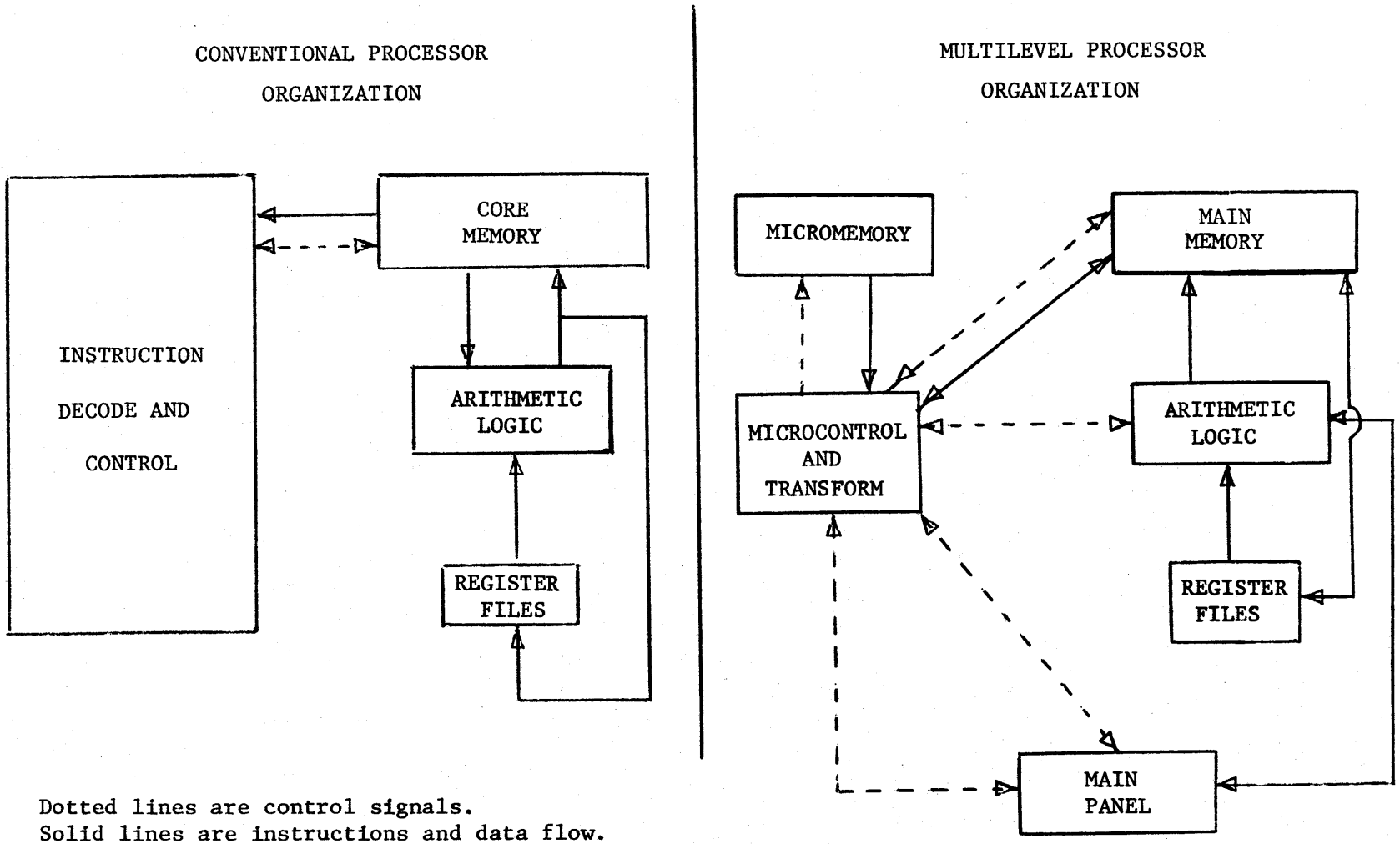


Figure 1-1. Digital Processor Organizations

BSC as a control processor:

- Accept real-time data inputs.
- Synchronize and time correlate real-time data inputs.
- Perform data buffering and blocking.
- Real-time source interrupt response.
- Subsystem control and status maintenance.
- Control special products and peripheral controllers.
- Provide data throughput strategies, including scheduling data output to CYBER level processor.
- Route data outputs from CYBER level processors to real-time environments.

BSC as an arithmetic and logical processor:

- Arithmetic and logical subroutines for real-time data
- synchronization strategy.
- Data sample processing in real-time.
- Logical decisions based on data sample information.
- Selection of desired data based on sample processing.
- Event detection based on cross-sample correlation or sample contained logical information.
- Application of calibration constants to data samples.
- Normalization and linearization of data samples.
- Data sample limit checking.
- Data parameter rate of change detection.
- Change format of data sample.
- Complex algorithms for analysis of individual data samples and sample arrays.

PHYSICAL CHARACTERISTICS

The equipment rack is shown in Figure 1-2. and encloses a power supply, maintenance panel, a BSC chassis and up to two MPM unit chassis. The BSC is modularly designed, with standard TTL MSI components and commercial construction.

The BSC power requirements vary with the particular user's application. CDC provides power supplies of +5 and +12 volts with input power requirements of 115 VAC, 50 or 60 HZ. Cooling fans require 110 VAC, 50 or 60 HZ. Power supplies should maintain stated values within +5% deviation.

The BSC chassis contains a pre-wired location for a panel interface card. The maintenancne panel is a 16-inch by 4.5-inch printed circuit board connected by flexible cable to the panel interface card. The panel contains 16 LED's for display, momentary pushbuttons and switches for data and control entry.

The standard BSC chassis is shown in Figures 1-3. and 1-4. and is 18.5 inches high by 17.5 inches wide by 16.0 inches deep. This chassis includes cooling fans sufficient for the fully populated unit. The back panel is available in two configurations, depending upon whether it is for the BSC or MPM. Each card slot is 0.625 inches wide. The front cover panel protects the circuit cards and improves chassis cooling.

The BSC operates in computer rooms and industrial environments where proper grid ground is provided. It operates at temperatures from 59°F to 104°F, withstanding a maximum temperature gradient of 20°F per hour or at a rate that precludes condensation and a relative humidity of 5 to 95 percent. Non-operating environments extend the temperature range from 20°F to 125°F and a

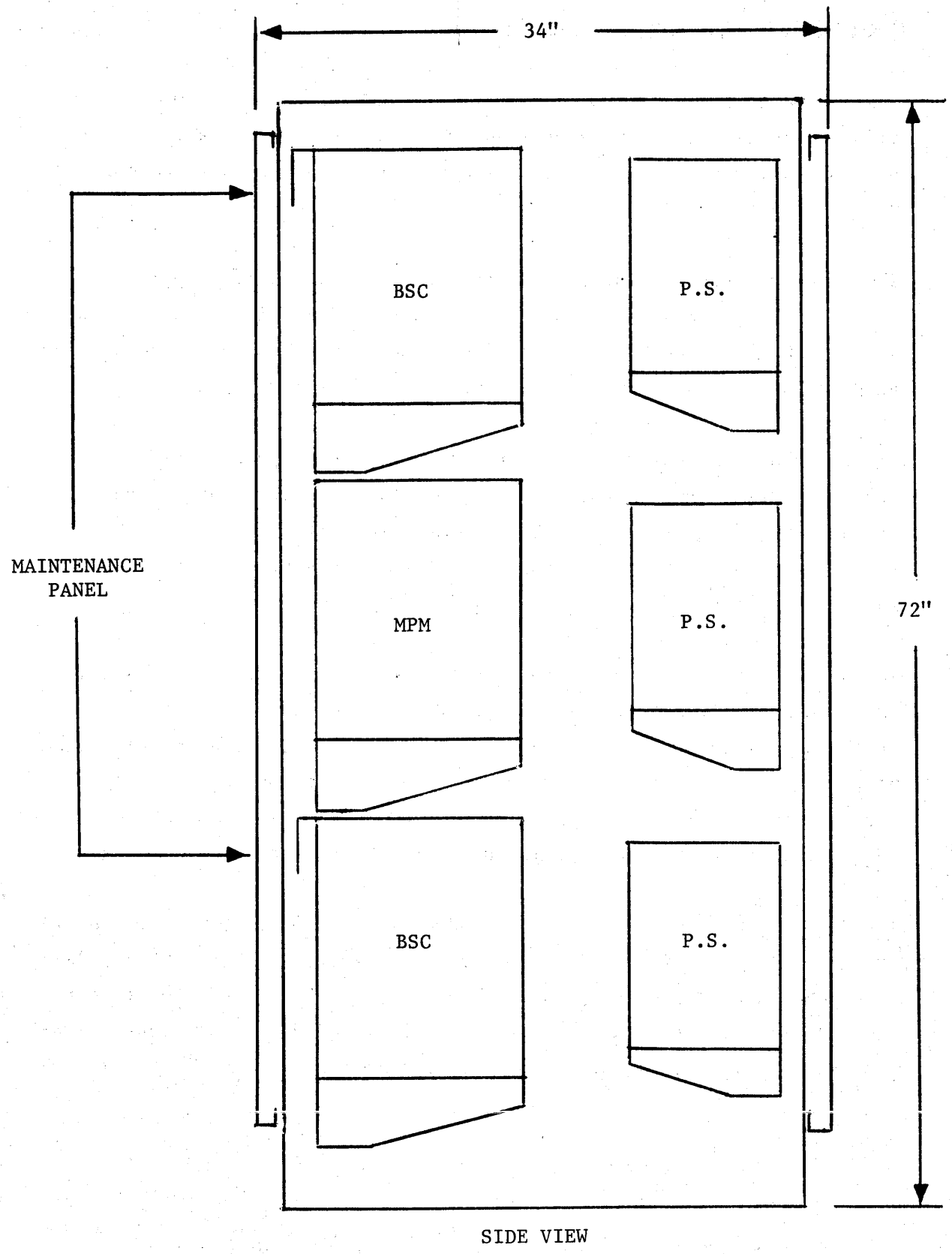


Figure 1-2. Equipment Rack

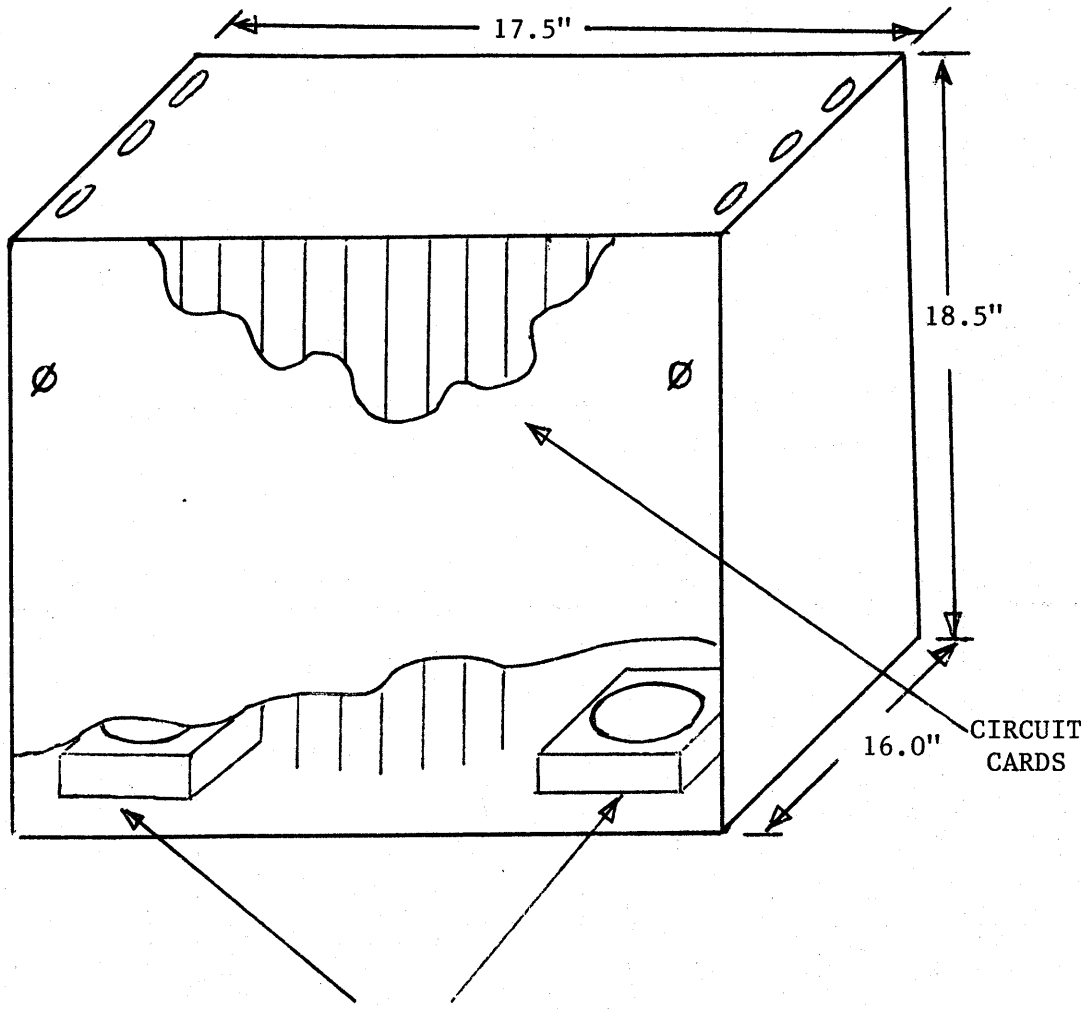
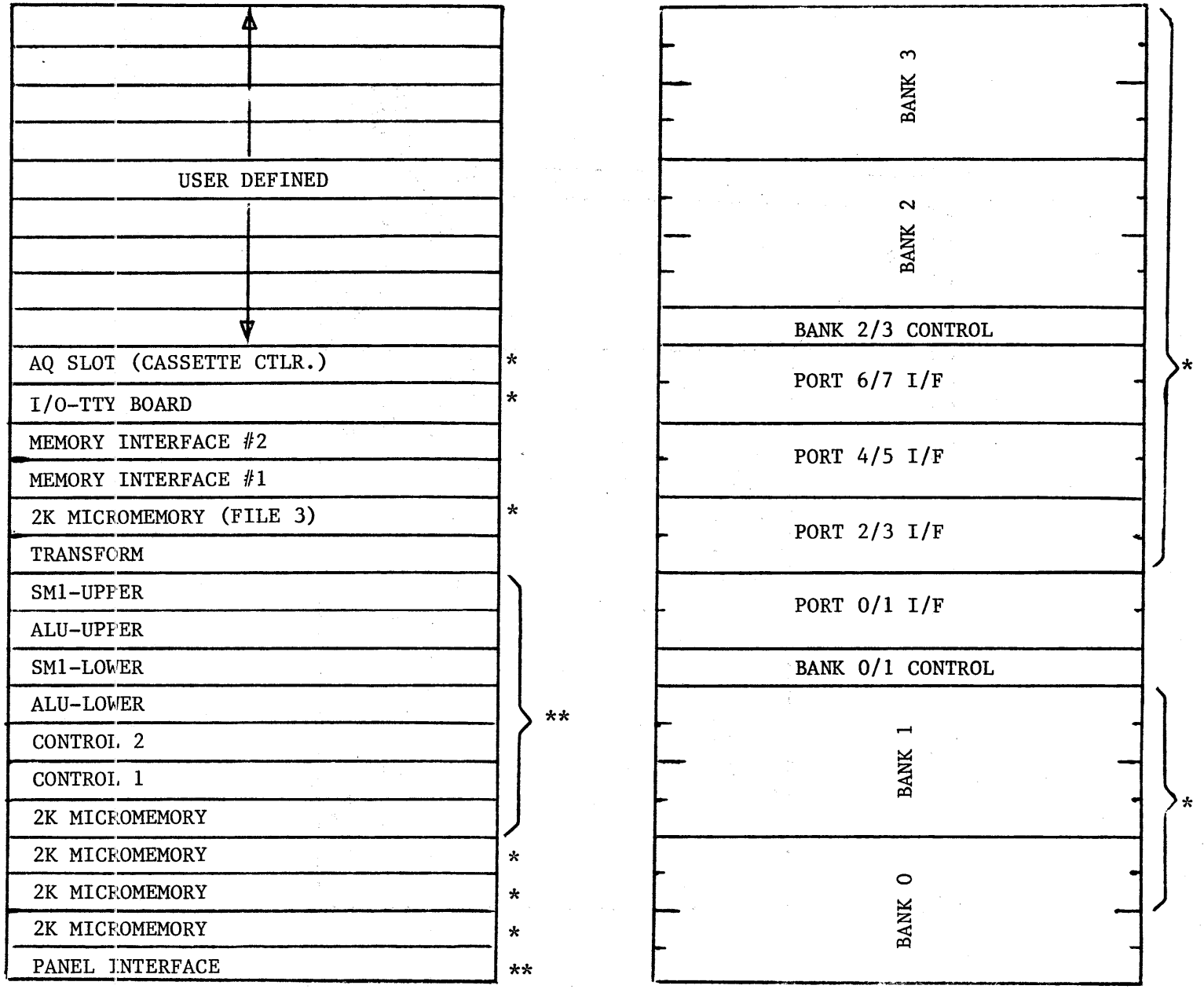


Figure 1-3. Standard Chassis



*OPTIONALS.
** COMMON BOARDS WITH MP-32 PRODUCT.

Figure 1-4. BSC and MPM Chassis Layout

gradient not to exceed 60°F per hour or at a rate that precludes condensation. Storage temperatures with proper packaging protection may be from -60F to 160°F and relative humidity from 2 to 95% with temperature cycles of not more than 60°F per hour at a rate that precludes condensation. The user should note that these ranges cover only the BSC; peripheral equipments may require more stringent environment control.

MAJOR SYSTEM COMPONENT DESCRIPTION

This section briefly discusses major components of a typical BSC system, as illustrated in Figure 1-5. The system is flexible and may be configured in many ways to fit the desired application. The system consists of three major components: Basic System Controller (BSC), Main Memory and BSC peripheral devices.

BASIC SYSTEM CONTROLLER

The Basic System Controller (BSC) is the microprogrammable component of the system and consists of the following four parts: Central Processing Unit, Memory Management Interface, Input Output Interface and BSC Panel Interface.

CENTRAL PROCESSING UNIT

The Central Processing Unit (CPU) is a 32-bit microprogrammable processor containing two arithmetic logic cards (ALU) and two status mode/interrupt (SMI) cards. Two control cards (CONT 1 and 2) provide timing and control signals. Special user required algorithm cards may also be included. The microprocessor cards are interconnected through the basic back panel layout; however, nonstandard options require additional wiring. The microprogram (firmware) instructions reside in micromemory which holds two 32-bit instructions per word. The micromemory boards are available with 2K instructions of RAM: writeable control store. The micromemory is interconnected to the microprocessor through the backpanel and is accessible only by the microprocessor. The transform module is an application

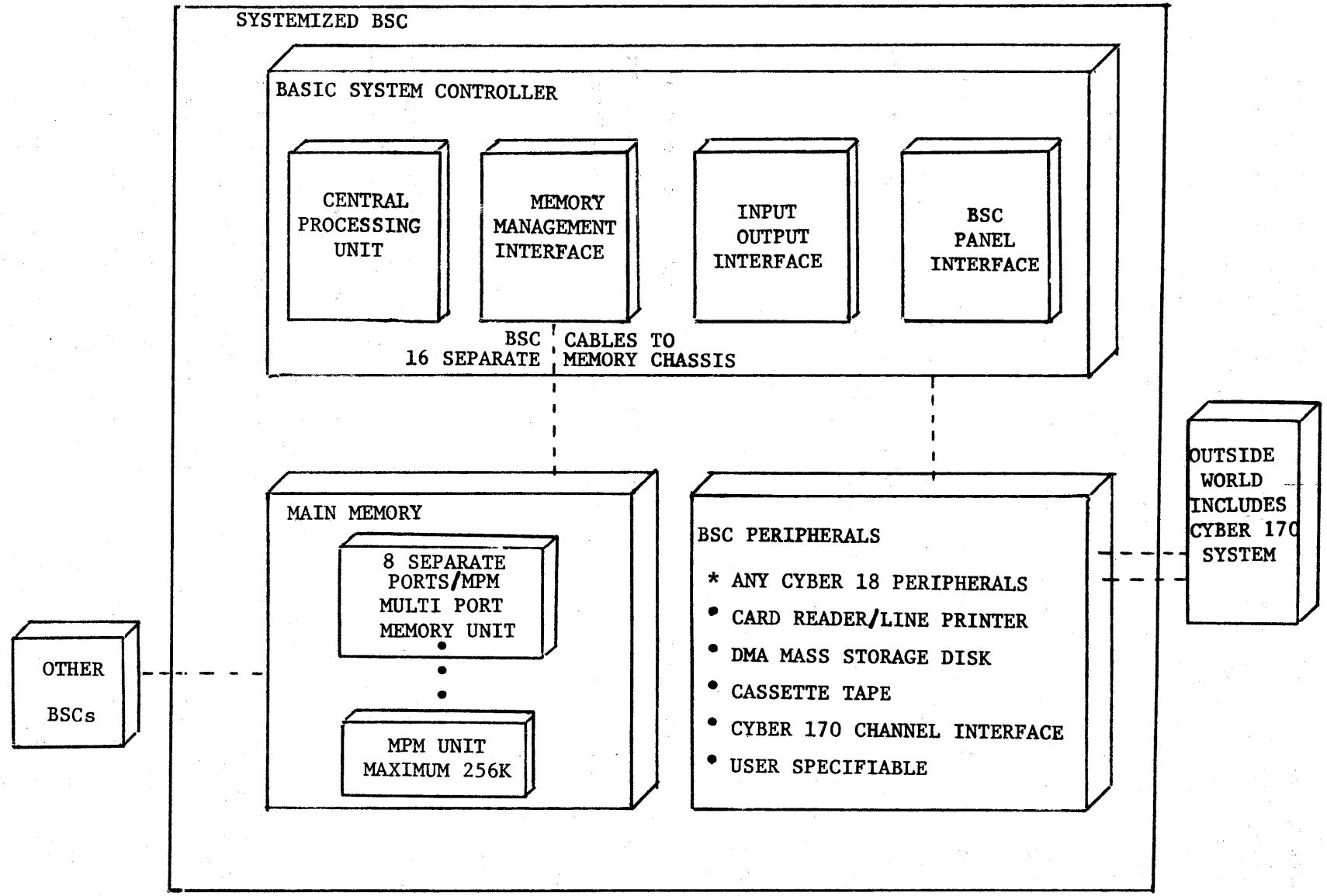


Figure 1-5. Major System Components of a "Systemized" BSC

dependent module with special hardware features to speed and streamline tasks such as the emulation of other processors. Typical transform features would include micromemory address, instruction and register manipulation by transform, instruction sequencing via a bit test and optional register file expansion.

MEMORY MANAGEMENT INTERFACE

The Memory Management Interface (MMI) contains the logic and control necessary to interface a remote main macro-memory to the basic processor. The interface provides paging logic for address generation and look-ahead logic for efficient read/write operations with minimum microprogram delay. The MMI also contains four I/O Direct Memory Access (DMA) channel ports giving peripheral controllers direct access to main memory. The DMA logic interfaces to and works with the main memory interface logic.

INPUT OUTPUT INTERFACE

The Input Output Interface (IOI) includes the IO-TTY module, adaptors to external devices that contain their own controllers, peripheral controllers and a Register I/O channel bus that handles both the CYBER 18 AQ and NCR M05 set/sample I/O protocols.

I/O TTY Module provides the basic microprocessor with input/output capability. Its basic features:

- Real-Time Clock
- I/O Controller for Teletype or Display
- Accepts Interrupts From Controllers
- Creates NCR M05 Set/Sample Channel
- Creates CDC CYBER 18 A/Q Channel

As required by system application, a variety of special interfaces and peripheral controllers may be integrated into the BSC system using the DMA, A/Q and M05 channels.

BSC PANEL INTERFACE

The BSC Panel Interface card provides the interface for a maintenance panel or RS232 compatible consoles that have full duplex serial ASCII characteristics. The maintenance panel is a 16 by 4.5 inch P.C. board mounted directly above the BSC chassis. It connects to the panel interface module through a flexible cable. Power and ground connections are provided in the cable. This interface, in conjunction with the IO-TTY module, cassette controller and cassette drive, provides a deadstart capability whereby the BSC firmware can be loaded into micromemory from a cassette magnetic tape.

MAIN MEMORY

The main memory is remote from the BSC chassis and contains logic to allow connection of up to eight BSC/s per MPM chassis. The main memory for a BSC may consist of up to 16 Eight Port Memory chassis daisy chained together. Each chassis may contain up to 256K 32-bit words, plus 4 bits parity MOS memory, divided into four independent banks of 0, 32K or 65K words.

The BSC interface to memory (MMI) contains logic for the following:

- Character/Half-Word/Word/ addressing.
- Address Paging.
- Data and Address Buffering.
- Memory Protect Logic

The (MMI) interface to memory provides for the connection of up to 4 16-bit direct memory access (DMA) external peripherals.

BSC PERIPHERALS

Any of the CYBER 18 and CDC 2550 peripherals are connectable to the BSC and the corresponding controller/adapters are placed in the BSC chassis. The following list is representative of the types of available peripherals:

- Card Reader (also can function as a deadstart device).
- Line Printer (300LPM).
- Cassette Tape Drive (also can function as a deadstart device).
- AQ-DMA Disk Unit (25 and 50M bytes).
- CYBER 70/170 Communications Coupler (CCC) (provides CYBER PPU a DMA capability to main memory).
- Floppy Disk Unit.

EXAMPLE OF COMPONENT INTERACTION

A brief description of typical system operations illustrates the function of various system components. The system can be initiated from a peripheral device such as a cassette tape unit. The deadstart logic loads the emulator and/or other firmware into the RAM micromemory. Once micromemory loading is completed, software programs may be loaded into the main macromemory under firmware control. With system initiation complete, the operator may communicate with the system via a display terminal or other programmer's console. Control of system peripherals and low-speed data transfers would be handled by the NCR M05 channel and the CYBER 18 A/Q channel. Medium and high-speed peripherals may transfer data to and from main macro-memory via the DMA channel.

The multi-port, multi-bank architecture of the main macro-memory allows a variety of multiprocessor system configurations. Main memory can be used to transfer blocks of data between processors. In a multiprocessor system, several processors can manipulate a data sample via the systems ability to share main memory.

FUNCTIONAL DESCRIPTION 2

The major functional areas of the BSC/MPM described in this section are grouped as follows (see Figure 2-1.):

- CPU Basic Microprocessor
- CPU Micromemory
- CPU Transform
- Memory Management Interface
- Input Output Interface
- BSC Panel Interface
- Multi Port (Main) Memory

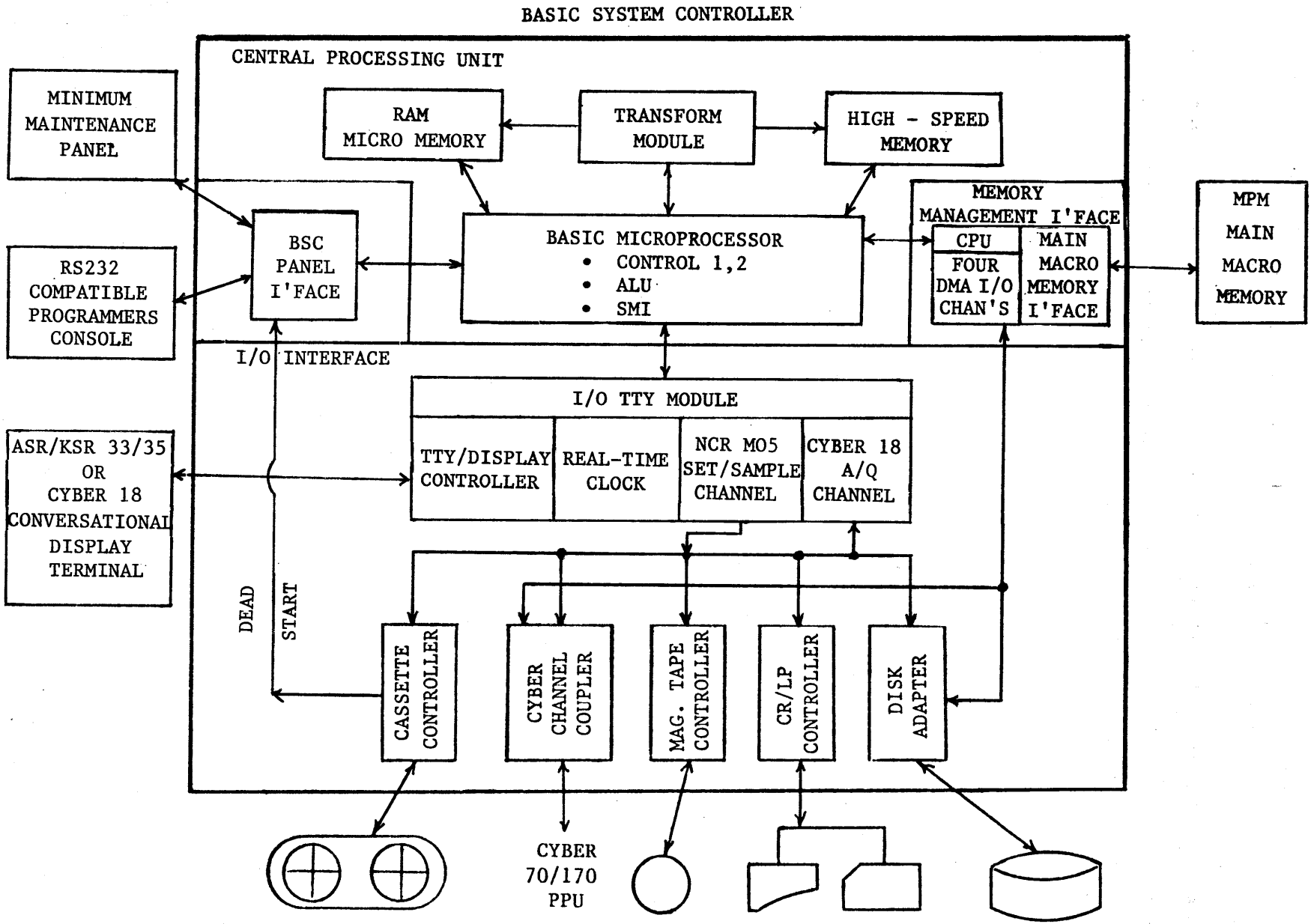


Figure 2-1. BSC Block Diagram

CPU BASIC MICROPROCESSOR

The Basic Microprocessor (MP) is a 32-bit parallel mode, microprogrammable digital processor. It consists of two arithmetic and logical unit (ALU) modules, two status/mode interrupt (SMI) modules and two control modules. The detailed MP organization is shown in Figure 2-2. It shows the ALU and working registers connected by selectors and a main 3-state bus. Discussion of MP organization and operation is broken down into the following areas:

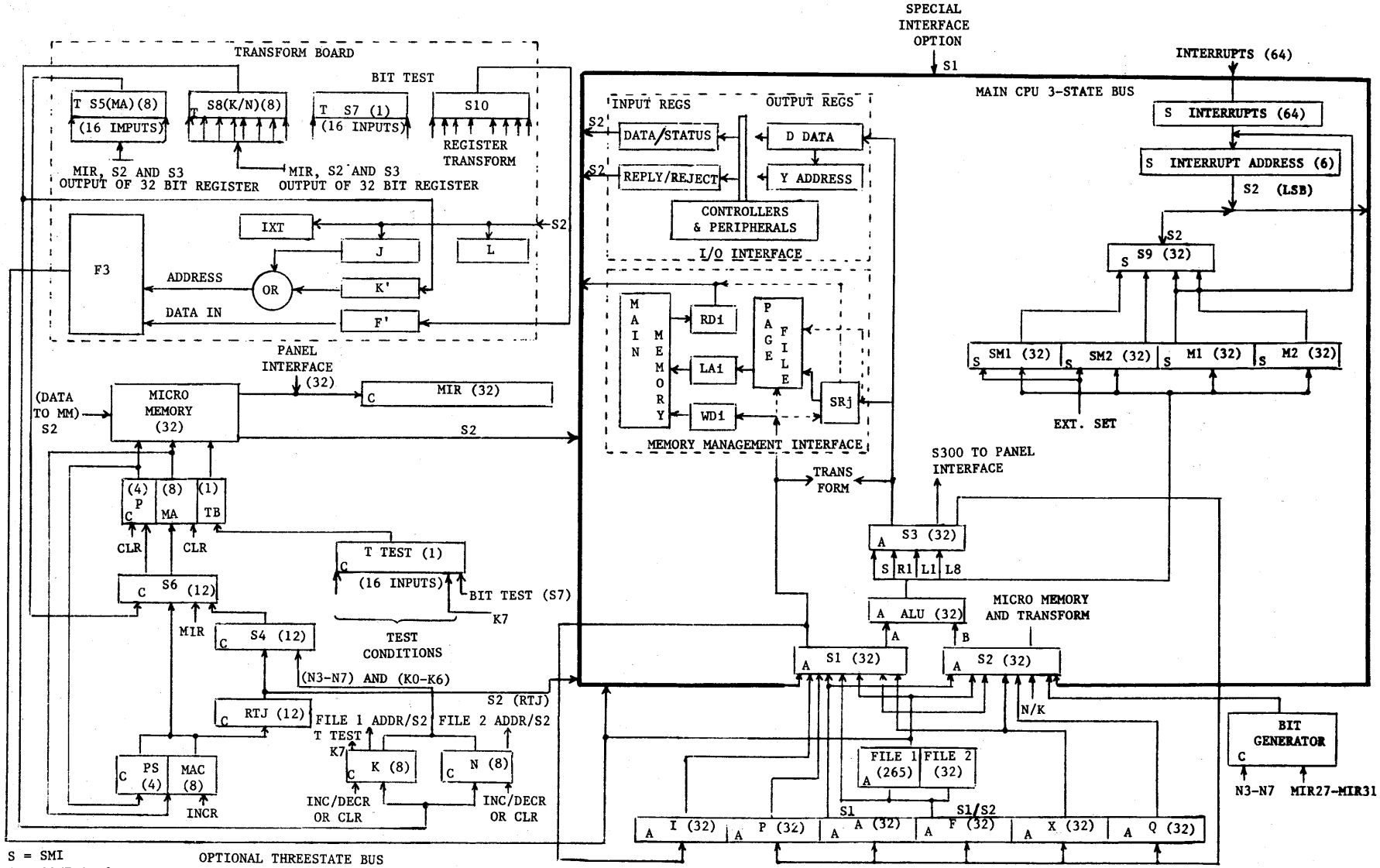
- Selectors
- Main CPU 3-State Bus
- Arithmetic Logical Unit
- Working Registers
- N/K Registers
- Register Files
- Bit Generator
- Status/Mode Operations
- Interrupts and Mask Register
- Options

SELECTORS

A selector is a multiplexer which transfers one of several inputs to an output. The selectors are numbered S1 through S10. Selectors S5, S7, S8 and S10 are physically located on the transform module. Specific details of their operation are found in the transform area. The selectors and the main 3-state bus provide the mechanism for routing data within the MP.

SELECTOR S1

S1 accepts input data from the working registers, the main 3-state bus and the register files. S1 selects one of these eight inputs as the A input to the ALU. The S1 output is also available to the I Register and to the main memory interface as a data source.



S = SMI
C = CONTROL 2
A = ALU
T = TRANSFORM

OPTIONAL THREESTATE BUS INPUT TO SELECTOR S1 AND S2 (VIA F1 INPUTS)

Figure 2-2. Central Processing Unit Detailed Block Diagram

* File 1 is furnished as an option.

SELECTOR S2

S2 accepts input data from the working registers, the main 3-state bus, the register files, the N/K Registers and the Bit Generator. S2 selects one of these eight inputs as the B input to the ALU. The S2 output is also available to the transform module and to micromemory.

SELECTOR S3

S3 accepts ALU output data and provides the capability of shifting data left or right one place, left eight places end-around or performing a straight transfer prior to making data available to the P, A, F, X or Q Registers. In addition, the output of S3 is routed to the main memory interface to provide addressing information to the transform module, the I/O Interface and the Panel Interface.

SELECTOR S4

S4 selects either the output of the RTJ Register or the Output of the K and N Registers as input to Selector S6.

SELECTOR S5

S5 selects one of eight inputs as the MA transform to become an input to Selector S6. Selector S5 is located on the transform module.

SELECTOR S6

S6 selects one of four sources as input for the P/MA Register. S6 inputs come from Selector S4, Selector S5, the PS/MAC Register and the MIR Register.

SELECTOR S7

S7 selects one of sixteen inputs for the T-Field Bit Test. Selector S7 is located on the transform module.

SELECTOR S8

S8 selects one of sixteen inputs for the K/N Transform. Selector S8 is located on the transform module.

SELECTOR S9

S9 selects one of four sources (SM1, SM2, M1 or M2) to become the A input to the ALU. The output of S9 is connected to S1 via the main CPU 3-state bus.

SELECTOR S10

S10 selects one of four inputs to perform the Register Transform function and is located on the transform module.

MAIN CPU 3—STATE BUS

This bus is a major data transfer path in the MP. The bus provides sub-multiplexing of data for input to Selectors S1 and S2. Inputs to the 3-state bus are:

- Main Memory Interface
- Micromemory
- Status Mode and Mask Register via Selector S9
- Interrupt Address
- Panel Interface Module
- I/O TTY Module
- RTJ Register
- Special Algorithm Option
- File 3
- Floating Point Hardware (Option)

ARITHMETIC LOGICAL UNIT

The ALU provides the arithmetic and logical capabilities of the MP. This unit combines two input words of the system word length; one from the A input, provided by S1, and the other from the B input, provided by S2.

These two inputs are combined according to the function code specified in the microinstruction. The result is available at the output of the ALU for possible shifting via S3 and delivery to a destination register, the main memory interface, the panel interface or to the I/O logic. The unshifted output of the ALU is delivered to the SM and mask registers. The ALU output can be ignored on an operation. The results of the ALU operation regarding sign, zero and magnitude (by means of carryout test) are available to the test bit logic for instruction sequencing.

WORKING REGISTERS

The I, P, A, F, X and Q registers provide temporary working storage for data being processed by the ALU. The working registers may be selected as ALU inputs. ALU outputs may then be destinated back to these registers (with the exception of I, which loads from S1).

A detailed description of these registers follows:

I REGISTER

This is a 32-bit register. It is available as an input to S1 and therefore to the A input of the ALU. Data is delivered to the I register from the output of S1.

P REGISTER

This is a 32-bit, general-purpose register that receives data from the ALU. Its output is provided to S1 and thus to the A input of the ALU.

A REGISTER

This is a 32-bit, general-purpose register that receives data from the ALU and provides output to S1 and, thus, to the A input of the ALU. The A register is mechanized as a shifting register and can be shifted left to right without using the ALU. The A register may also be combined with the Q register to form a double length shifting register that operates independently of the ALU.

F REGISTER

This is a 32-bit, general-purpose register that receives data from the ALU and provides output to S1 or S2 as ALU input. This register is also used as the file entry register and contains information written into file 1 or file 2 when it is used as the destination of an ALU operation.

X REGISTER

This is a 32-bit, general-purpose register that receives data from the ALU and provides output to S1 or S2 as ALU input.

Q REGISTER

This is a 32-bit, general-purpose register that receives data from the ALU. It provides output to S2 and, thus, to the B side of the ALU. The Q register is mechanized as a shifting register. It may be shifted left or right in conjunction with the A register without using the ALU.

N/K REGISTERS

K REGISTER

The K register is an 8-bit counter that can be set, cleared, incremented or decremented. It is used to address file 1 in addition to any program usage as a counter. The original value of K can be tested against zero by the microinstruction. K is selectable as an input to S2.

N REGISTER

The N register is an 8-bit counter which may be set, cleared, incremented or decremented. It is used to address file 2, control shifts, control the scale operations and may be used as an iteration counter which controls microinstruction execution for operations such as multiplication and division. It may also be used as a programmed counter, since the original value of N can be tested against zero by the microinstructions. N is selectable as an input to S2. The lower five bits may be selected to drive the Bit Generator.

N/K REGISTER

The N and K registers may be combined to provide micromemory operand addresses outside the current operating page. They are gated through S6 to P/MA for this operation. In this operation, the five least significant bits of N provide the upper portion of the 12-bit address. The seven most significant bits of K provide the lower portion of the 12-bit address. The least significant bit of K is used to determine whether to use the upper or lower microinstruction of the microinstruction pair.

REGISTER FILES

Additional temporary storage capability is provided by register files F1, F2 and F3.

FILE F1

File F1 is a 256-word by 32-bit RAM memory. This file receives data from the F-Register and outputs data to selectors S1 and S2. It is addressed by the K-Register.

FILE F2

File F2 is a 32-word by 32-bit RAM memory. F2 receives data from the F-Register and outputs data to Selectors S1 and S2. It is addressed by the least significant 5 bits of the N-Register.

FILE F3

File F3 is a 2K to 8K word by 32-bit RAM memory. This optional file receives data from the F'-Register on the transform module and outputs data to Selectors S1 and S2. It is addressed by the J and K' Registers on the transform module. (Refer to Transform Area.)

BIT GENERATOR

The BG circuit generates one bit at any position in a word as input to the B side of the ALU. Bits are numbered from left to right as bits 0 to 31. Control to drive the bit generator is derived from either the microinstruction (bits 27 to 31) or the lower five bits of the N Register. Control is usually obtained from the microinstruction. The input that drives the bit generator is established by a bit setting in a SM register.

STATUS/MODE OPERATIONS

Status/Mode Register (SM) - The SM register allows the microprogram to control the mode of operation and also allows the microprogram to examine the status of certain internal and external conditions. The MP can access either the SM1 or SM2 register. Each SM Register contains 32 bits. Each SMI module contains 16 bits of SM1 and 16 bits of SM2. All 32 bits of an SM module can be set or reset by the microprogram by transferring information to the SM register from the output of the ALU. Master Clear will clear SM1 and SM2. The SM register bits are numbered as follows:

SM1 bits (first module) numbered
S100 to S115 (most significant)
SM1 bits (second module) numbered

S116 to S131 (least significant)

SM2 bits (first module) numbered

S200 to S215 (most significant)

SM2 bits (second module) numbered

S216 to S231 (least significant)

The functional characteristics of the individual bits of the SM module are shown in Table 2-1. SM module bit assignments are depicted in Table 2-2.

TABLE 2-1. SM MODULE BIT CHARACTERISTICS

MODULE BITS	FUNCTIONAL CHARACTERISTICS
SM1 (0-3) SM1 (16-19) SM2 (0-3) SM2 (16-19)	These are flag bits which are set or reset by the SETF and CLRF commands in a microinstruction. These commands address up to 16 flags. 0-3 of SM1 are flags 0-3, bits 16-19 of SM1 are flags 4-7, bits 0-3 of SM2 are flags 8-11 and bits 16-19 of SM2 are flags 12-15. SM1 (0-3) and SM1 (16-19) are available as a true output from the SM module. SM2 (0-3) and SM2 (16-19) are available as a complement output from the SM module.
SM1 (4-7) SM1 (20-23) SM2 (4-7) SM2 (20-23)	These bits can be set by an external signal and they are available as a true and complement output from the SM module.
SM1 (08-11) SM1 (24-27) SM2 (08-15) SM2 (24-31)	These bits can be set by an external signal. SM2 (14-15) and SM2 (30-31) can also be cleared by an external signal. These bits are available only as true outputs from the SM module.
SM1 (12-15) SM1 (28-31)	These bits are set by an external signal. SM1 (14-15) and SM1 (30-31) can also be cleared by external signal. These bits are available only as complement outputs from the SM module.

The status and mode conditions of specific bits of the SM register are a design function. These conditions are shown in Table 2-2.

TABLE 2-2. BSC SM REGISTER BIT ASSIGNMENTS (1 of 2)

SM BIT	STATUS/ MODE	TYPE OF DEFINITION	FUNCTION
100	M	1 , 3	Double Precision
101	M	1 , 3	One's Complement
102	M	1 , 3	Bit Generator Input from N
103	M	1	Adder Split
104			OPEN
105			OPEN
106			OPEN
107	S	1 , 2 , 3	OVERFLOW
108	S	2 , 3	MP-60 Illegal Instruction
109	M	1	Enable Micro Halt
110			
111	M	1 , 3	File 1 Enable
112	S	1 , 2 , 3	Breakpoint Interrupt Capture
113	M	1	Select XT/MA (R/W MM)
114			OPEN
115			OPEN
116	M	1 , 3	Lock Memory
117	S	2 , 3	MP-60 Bit Register
118	M	2 , 3	Monitor/Program Mode
119			OPEN
120	S	2 , 3	Real Time Interrupt 07 Capture
121	S	2 , 3	Real Time Interrupt 06 Capture
122	S	2 , 3	Real Time Interrupt 05 Capture
123	S	2 , 3	Real Time Interrupt 04 Capture
124	M	1 , 3	Enable Interrupt System 0
125	M	1 , 3	Enable Interrupt System 1
126	M	1 , 3	Enable Interrupt System 2
127	M	1 , 3	Enable Interrupt System 3
128	S	2 , 3	Real Time Interrupt 03 Capture
129	S	2 , 3	Real Time Interrupt 02 Capture
130	S	2 , 3	Real Time Interrupt 01 Capture
131	S	2 , 3	Real Time Interrupt 00 Capture

S = STATUS
M = mode

1 = Specific BSC Defintiion
2 = General BSC Definition with a Specific MP-60 Application
3 = This bit is used by the MP-60 Emulator

TABLE 2-2. BSC SM REGISTER BIT ASSIGNMENTS (2 of 2)

SM BIT	STATUS/ MODE	TYPE OF DEFINITION	FUNCTION
200	M	1	Enable Auto Data Transfer
201	M	1	READ I/O Strobe
202	M	1	WRITE I/O Strobe
203	M	1	STERM I/O Strobe
204	M	2 , 3	Select Deadstart Interface
205			OPEN
206	S	1 , 2 , 3	One Millisecond Pulse
207	M	1 , 3	Enable S-Field Page Select
208	S	2 , 3	Real Time Clock Internal
209	S	2 , 3	Programmable Interrupts (M100 & M209)
210	S	2 , 3	Programmable Interrupts (M101 & M210)
211	S	2 , 3	Programmable Interrupts (M102 & M211)
212	S	2 , 3	Programmable Interrupts (M103 & M212)
213			OPEN
214	M	1	Enable Panel I/F to MIR
215	M	1 , 2 , 3	Macro Run/Stop
216	M	1	Test/XMC MPM
217	M	1	Isolate BSC
218			OPEN
219	M	1 , 3	Page Address from S3
220	S	2 , 3	Real Time Interrupt 11 Capture
221	S	2 , 3	Real Time Interrupt 10 Capture
222	S	2 , 3	Real Time Interrupt 09 Capture
223	S	2 , 3	Real Time Interrupt 08 Capture
224			OPEN
225	S	2 , 3	Function Fault
226	S	2 , 3	Exponent Fault
227	S	2 , 3	Divide Fault
228	S	2 , 3	IOC 3 End of Operation
229	S	2 , 3	IOC 2 End of Operation
230	S	2 , 3	IOC 1 End of Operation
231	S	2 , 3	IOC 0 End of Operation

S = STATUS
M = MODE

1 = Specific BSC Definition
2 = General BSC Definition with a Specific MP-60 Application
3 = This but is used by the MP-60 Emulator

MP STATUS

Status bit assignments are incorporated into the system during manufacture and these status bits are set by the condition detected. The clearing of the status bit must be performed by the microprogram with the exception of SM1 (bits 14, 15, 30, 31) and SM2 (bits 14, 15, 30, 31) which have an external clear input.

As indicated in Table 2-2, some status bits have specific definitions for all BSCs. Some have a general BSC definition with a specific definition for the case of a BSC configured for MP-60 emulation. Status bit definitions are discussed in the following paragraphs.

- SM107 - Overflow (M200)

  This status bit is set to 1 on detection of an arithmetic overflow. An arithmetic overflow is detected only if the microinstruction performing the addition is an add or subtract with overflow test and the arithmetic result is inconsistent with the sign of the operands and the arithmetic operation. In addition, if this status bit is set to 1 an interrupt associated with M200 is generated. UPON DETECTION BY THE FIRMWARE, THIS BIT IS CLEARED TO 0.

- SM108 - MP-60 Illegal Instruction

  If this MP-60 status bit is set to 1, no privileged or undefined instructions have been detected. If this status bit is cleared to 0, an attempt to execute a privileged or undefined instruction has been detected and an interrupt associated with M127 is generated. UPON DETECTION BY THE FIRMWARE THIS BIT IS SET TO 1 AGAIN.

  Privileged macro instructions may be executed in monitor, but not in program mode (see SM118).

- SM112 - Breakpoint Interrupt Capture

  This status bit is set to 1 when the panel interface detects a breakpoint address condition. In addition, an interrupt associated with M125 is generated.

  UPON DETECTION BY THE FIRMWARE, THIS BIT IS CLEARED TO 0.

- SM117/FLAG 05 - MP-60 Bit Register

  This status bit represents the MP-60 bit register. If this bit is set to 1, the bit register is 1. If this bit is cleared to 0, the bit register is 0.

  If this bit is set to 1 while a BTU op-code is executed, micro-execution will continue with the upper instruction of the next microinstruction pair. If this bit is cleared to 0 while a BTU op-code is executed, micro-execution will continue with the lower instruction of the next microinstruction pair.

- SM120 - Real Time Interrupt 07 Capture (M108)
- SM121 - Real Time Interrupt 06 Capture (M109)
- SM122 - Real Time Interrupt 05 Capture (M110)
- SM123 - Real Time Interrupt 04 Capture (M111)

  Setting of any of these status bits to 1 will generate an interrupt associated with M108, M109, M110, and M111, respectively. UPON DETECTION BY THE FIRMWARE, THESE BITS ARE CLEARED TO 0.

- SM128 - Real Time Interrupt 03 Capture (M112)
- SM129 - Real Time Interrupt 02 Capture (M113)
- SM130 - Real Time Interrupt 01 Capture (M114)
- SM131 - Real Time Interrupt 00 Capture (M115)

  Setting of any of these status bits to 1 will generate an interrupt associated with M112, M113, M114 and M115, respectively. UPON DETECTION BY THE FIRMWARE, THESE BITS ARE CLEARED TO 0.

- SM206 - One Millisecond Pulse

  This status bit is set to 1 by the one millisecond oscillator on the transform module. When set to 1, this bit will generate an interrupt associated with M128. UPON DETECTION OF THE INTERRUPT, THE FIRMWARE WILL CLEAR THIS STATUS BIT TO 0.

- SM208 - Real Time Clock Interval (M208)

  This status bit is cleared to 0 by the firmware when the Real Time Clock Interval has elapsed, causing an interrupt associated with M208. UPON DETECTION BY THE FIRMWARE, THIS BIT IS SET BACK TO 1.

- SM209 - Programmable Interrupts (M100 & M209)
- SM210 - Programmable Interrupts (M101 & M210)
- SM211 - Programmable Interrupts (M102 and M211)
- SM212 - Programmable Interrupts (M103 and M212)

  Programmatically clearing any of these status bits to 0 (using the SINT macro instruction) will generate an interrupt. UPON DETECTION BY THE FIRMWARE, THE BIT WILL BE SET BACK TO 1.

- SM220 - Real Time Interrupt 11 Capture (M104)
- SM221 - Real Time Interrupt 10 Capture (M105)
- SM222 - Real Time Interrupt 09 Capture (M106)
- SM223 - Real Time Interrupt 08 Capture (M107)

  Setting any of these status bits to 1 will generate an interrupt associated with M104, M105, M106 and M107, respectively. UPON DETECTION BY THE FIRMWARE, THESE BITS ARE CLEARED TO 0.

- SM225 - Function Fault (M201)

  Execution of a macro function instruction which results in an abnormal answer causes this status bit to be cleared to 0. This generates an interrupt associated with M201. UPON DETECTION BY THE FIRMWARE, THIS BIT IS SET BACK TO 1.

- SM226 - Exponent Fault (M202)

  Execution of a macro floating point instruction which results in an abnormal answer causes this status bit to be cleared to 0. This generates an interrupt associated with M202. UPON DETECTION BY THE FIRMWARE, THIS BIT IS SET BACK TO 1.

- SM227 - Divide Fault (M203)

  Exeuciton of a macro integer divide instruciton which results in an abnormal answer causes this status bit to be cleared to 0. This generates an interrupt associated with M203. UPON DETECTION BY THE FIRMWARE, THIS BIT IS SET BACK TO 1.

- SM228 - IOC 3 End of Operation (M204)
- SM229 - IOC 2 End of Operation (M205)
- SM230 - IOC 1 End of Operation (M206)
- SM231 - IOC 0 End of Operation (M207)

  At IOC end of operation these status bits are cleared to 0, causing an interrupt associated with M204, M205, M206 and M207, respectively. UPON DETECTION BY THE FIRMWARE, THESE BITS ARE BACK TO 1.

MP OPERATING MODES

Operating modes are incorporated into the system during manufacture and these modes are controlled by a bit in the SM register. However, if an option is not needed, the associated mode bit in the SM register may be reassigned as required. As indicated in Table 2-2, some mode bits are defined for all BSCs and some are more specifically defined for BSCs configured for MP-60 emulation. These pre-assigned modes are discussed in the following paragraphs.

- SM100/FLAG00 - Double Precison

  If this mode bit is set to 1 and if the double precision hardware is included in the BSC, the ALU and the double precision ALU* are combined to form a double word length ALU. This double-word length ALU operates for addition and subtraction operations, but not for logical operations. In addition, if this mode bit is set to 1, the MP-60 shift count K/N transform eleven (11) and MP-60 bit test one (1) are used to limit the shift count to 63.

  If this mode bit is cleared to 0, the double precision ALU* is disconnected from the ALU and no operations take place in ALU*.

- Two's/One's Complement (SM101)/FLAG01 - If this mode bit is set to 1, the ALU operates in one's complement arithmetic mode for addition and subtraction. If the mode bit is set to 0, operations are in two's complement arithmetic mode for addition and subtraction.

INTERRUPTS AND MASK REGISTER

The interrupt system is implemented as a sampled data system at the microprogram level instead of a true interrupt as used in conventional computers. That is, the interrupt system provides a sampling capability in which a microinstruction can sample the interrupt system to see if there is any interrupt present that has its corresponding mask register bit set to 1. This sample is taken by performing an INTU operation in the T field of a microinstruction. If there is an interrupt in the system whose mask register bit is 1 and the interrupt system is enabled, the next microinstruction is executed from the upper of the next microinstruction pair. If there is no such interrupt, the next microinstruction is executed from the lower of the next microinstruction pair.

When an interrupt is recognized, the microprogram samples the interrupt address encoder to identify the most significant (highest priority) interrupt. The interrupt address encoder must be read in the microinstruction following the interrupt test to be sure of a correct interrupt line address. If the interrupt address is read earlier or later, there is a possibility that the address encoder output is unstable due to a newly arrived interrupt. The interrupt address is read by performing an INTA operation in the B'field of any microinstruction.

Interrupts are identified by the corresponding mask bits which are assigned to control the interrupt recognition. The bits in the mask register are identified as follows:

a. Mask Register 1 (M1)

   M100 through M131

b. Mask Register 2 (M2)

   M200 through M231

Interrupt addresses are generated by the interrupt address encoder, according to the assignments shown in Tables 2-3 and 2-4. The interrupt priorities correspond to the interrupt address generated; that is, interrupt address 0 is associated with the highest priority interrupt line, and interrupt address 63 is associated with the lowest priority interrupt line. For example, an interrupt associated with M112 would have priority over an interrupt associated with M111 and an interrupt address of 35_{10} would be generated by the interrupt address encoder.

The output from the interrupt address encoder is the complement of the interrupt address for input to S2; thus, the transfer of the interrupt address to the X register, for example, would be coded using a -B code in the F field, INTA in the B' field and X in the D field. This results in the transfer of the correct interrupt address.

A design option in the interrupt system provides for activating interrupts in groups of eight interrupt lines. Therefore, any combination of eight groups can be controlled by SM bit(s). Groups not controlled by SM bit(s) can be allowed to remain active while the remaining interrupts may be enabled or disabled.

Interrupt signals must be steady-state when input to the interrupt system and indicate the presence of an interrupt when set to a 0. If a pulse type interrupt is required, the pulse interrupt signal is used to set a bit in the SM register; this SM bit is then wired to the interrupt system. On recognizing this interrupt, the microprogram is able to clear the interrupt condition by clearing the SM bit.

TABLE 2-3. INTERRUPT ADDRESSES (MASK REGISTER 1)

MASK BIT	ADDRESS/ PRIORITY	TYPE OF DEFINITION	FUNCTION
M100	47	2	Programmable 3
M101	46	2	Programmable 2
M102	45	2	Programmable 1
M103	44	2	Programmable 0
M104	43	2	Real Time 11
M105	42	2	Real Time 10
M106	41	2	Real Time 09
M107	40	2	Real Time 08
M108	39	2	Real Time 07
M109	38	2	Real Time 06
M110	37	2	Real Time 05
M111	36	2	Real Time 04
M112	35	2	Real Time 03
M113	34	2	Real Time 02
M114	33	2	Real Time 01
M115	32	2	Real Time 00
M116	15	3	Micro I/O Interrupt 0
M117	14	3	Micro I/O Interrupt 1
M118	13	3	Micro I/O Interrupt 2
M119	12	3	Micro I/O Interrupt 3
M120	11	3	Micro I/O Interrupt 4
M121	10	3	Micro I/O Interrupt 5
M122	9	3	Micro I/O Interrupt 6
M123	8	3	Micro I/O Interrupt 7
M124	7	1	Macro Run/Stop
M125	6	1	Breakpoint Interrupt
M126	5	1	Panel I/F Request
M127	4	1	Illegal Instruction
M128	3	1	1.0 Millisecond Interrupt
M129	2	1	Memory Error - DMA
M130	1	1	Memory Error - Macro
M131	0	1	Power Failure

* See Notes Following Tables

TABLE 2-4. MP INTERRUPT ADDRESSES (MASK REGISTER 2)

MASK BIT	ADDRESS/ PRIORITY	TYPE OF DEFINITION	FUNCTION
M200	63	1	Arithmetic Fault
M201	62	2	Function Fault
M202	61	2	Exponent Fault
M203	60	2	Divide Fault
M204	59	2	IOC 3 End of Operation
M205	58	2	IOC 2 End of Operation
M206	57	2	IOC 1 End of Operation
M207	56	2	IOC 0 End of Operation
M208	55	2	Real Time Clock Interval
M209	54	2	Programmable 7
M210	53	2	Programmable 6
M211	52	2	Programmable 5
M212	51	2	Programmable 4
M213	50	2	External 2
M214	49	2	External 1
M215	48	2	External 0
M216	31	3	Macro I/O Interrupt 0 LP Common
M217	30	3	Macro I/O Interrupt 1
M218	29	3	Macro I/O Interrupt 2
M219	28	3	Macro I/O Interrupt 3
M220	27	3	Macro I/O Interrupt 4
M221	26	3	Macro I/O Interrupt 5
M222	25	3	Macro I/O Interrupt 6
M223	24	3	Macro I/O Interrupt 7 CR Common
M224	23	3	Macro I/O Interrupt 8
M225	22	3	Macro I/O Interrupt 9
M226	21	3	Macro I/O Interrupt A
M227	20	3	Macro I/O Interrupt B CRT Common
M228	19	3	Macro I/O Interrupt C
M229	18	3	Macro I/O Interrupt D
M230	17	3	Macro I/P Interrupt E
M231	16	3	Macro I/O Interrupt F DA Common

Notes for Tables 2-3 and 2-4.

- Priority 0 = highest priority
- Priority 63 = lowest priority
- The interrupt address generated is the same as its priority level, i.e., the lowest priority interrupt generates a 63_{10} interrupt address.
- 1 = Secific BSC Definition
- 2 = General BSC Definition with specific MP-60 application
- 3 = Beneral MP-60 Definition with specific definition determined by system configuration.

Tables 2-3 and 2-4 list all interrupt mask bit assignments. As shown, some bits have specific definitions for all BSCs, some have general BSC definitions with specific MP-60 definitions, while others have general MP-60 I/O definitions with configuration dependent specific assignments.

OPTIONS

- A third file (file 3) is available as an extension to file 1. In addition, file 3 may also be used as a high-speed local memory. File 3 has 2K to 8K general purpose, word-length registers, which are addressed by the contents of the J or K' registers on the transform module. The output of file 3 is delivered to S1 and S2 on the file 1 bus when file 1 is not active. It is gated to the main CPU 3-state bus when the bus is not otherwise being used. (Refer to Transform area.)

- Normally, the scale point is between bits 0 and 1 of the A register. It is possible through a hardware change to set the scale point between different bits in the A register when necessary for efficient floating point emulation.

- The split adder option allows the main ALU to be split into independent adders. This split is acivated by setting the adder split flag in the SM register. The split blocks the carry between the two portions of the adder. The upper portion functions as a two's complement or as a two's complement adder, depending upon the state of one's complement SM register flag. In one's complement mode, the carryout of the lower portion is used as the end-around carry bit. In two's complement mode, both portions act as independent two's complement adders. The split adder has no effect on logical operations because no carry is involved in these operations. Carryout and overflow are generated by the most significant portion at the ALU whether or not split adder operation has been selected.

CPU MICROMEMORY

The BSC micromemory is a high-speed, solid-state, bi-polar RAM memory normally used to store the firmware that controls the operation of the microprocessor. The micromemory is physically located in the BSC chassis and is connected to the microprocessor via the backpanel. Four locations in the BSC chassis are reserved for micromemory boards. Each board contains 2K of RAM micromemory, making possible a maximum configuration of 8K of micromemory.

The addressing structure of the micromemory divides each 2K RAM board into four equal pages. Within a page of 512 microinstructions, an address specifies one of 256 microinstruction pairs. Within a pair, the upper or lower microinstruction is selected by the T-Field portion of the address. (See Figure 2-3.)

Micromemory may be loaded from an external device such as a terminal keyboard, card reader or cassette tape or may be written into under control of the microprogram. The micromemory gets its input data from selector S2. The output of micromemory is connected to the microinstruction register (MIR) and to the main CPU 3-state bus. The micromemory is accessible only by the microprocessor.

The following are associated with the micromemory and microcontrol section:

- Page/Micromemory Address Register
- Memory Address Counter
- Page Storage Register
- Microinstruction Register
- Return Jump Register

Brief descriptions of these sections are as follows:

PAGE/MICROMEMORY ADDRESS (P/MA) REGISTER

The micromemory is addressed by the P/MA register. The P portion of the register is a page address and consists of four bits that specify one of 16 pages of micromemory. A page consists of 256 microinstruction pairs of 512 microinstructions. The MA register is an 8-bit register that specifies one of 256 microinstruction pairs within the page that is to be the source of the next microinstruction. Microinstruction sequencing is designed so that no automatic overflow of addressing from the MA register to the P register occurs. Any control transfer between pages is initiated by a page jump operation, MA transform operation or clear page operation.

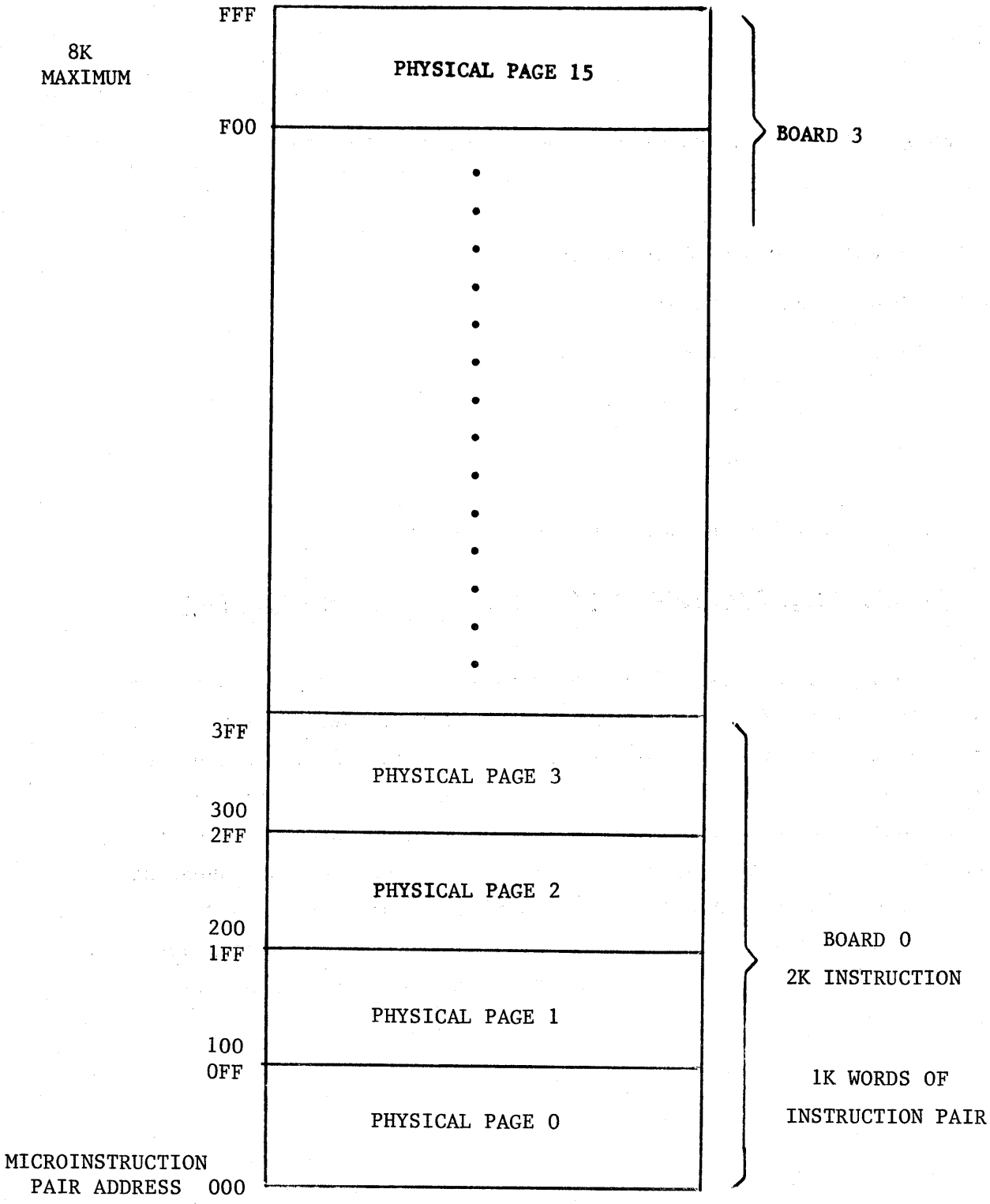


Figure 2-3. Micromemory Page Structure

MEMORY ADDRESS COUNTER (MAC)

The MAC is a counter used to determine the next location within a page following the current location specified in the MA register. In operation, the contents of the MA register are transferred to the MAC at each micromemory reference, then the contents of the MAC are incremented by one to point to the next location. Depending on the sequencing operation specified in the microinstruction, the MAC may or may not be used to obtain the next microinstruction. Sequencing of the MAC is such that location 0 within a page follows location 255 of that page.

PAGE STORAGE (PS) REGISTER

The PS register is a four-bit holding register used to determine the page for the instruction. In operation, the contents of the page register are transferred to the PS at each micromemory reference. Depending on the sequencing operation specified in the microinstruction, the PS may or may not be used to obtain the next microinstruction.

MICROINSTRUCTION (MIR) REGISTER

The MIR is a 32-bit register used to hold the microinstruction during execution. Microinstructions are normally entered into the MIR from the micromemory; either the upper or lower 32 bits of the micromemory location are gated to the MIR based on the value of the test bit determined during the preceding microinstruction.

Microinstructions may also be transferred into the MIR from the maintenance panel interface. This is done as a result of an operator request.

RETURN JUMP (RTJ) REGISTER

When specified by a microinstruction, the RTJ register captures the location of the next microinstruction pair. When this capture is specified, the contents of MAC are incremented, then PS and MAC are stored in the RTJ

register. The contents of the RTJ register remain unchanged until the next command is given to save a new address. This saving of the next instruction pair location is independent of any actual transfer of control. The output of the RTJ register can be gated to the P/MA register to perform operations or may be read into the MP through Selector S2.

CPU TRANSFORM

The transform feature of the BSC is packaged on a single module and enables the microprogram to select (tranform) various patterns of bits from the BSC data transmission paths which, in turn, forms micromemory addresses, sequences the microprogram and sets the contents of K and N registers.

A standard BSC transform module is delivered with the BSC and provides some space for the user to adapt it to his application. This module is replaceable with any other transforms that are specially designed to meet user requirements.

The standard BSC/MPM transform module* provides a facility for emulation of the CDC MP-60 Computer System running under the MPX-RT Operating System. In addition, the SPC provides the BSC with tools necessary for data processing applications. SPC also provides a facility to define several flexible transforma via wirewrap techniques. To allow for unique system requirements, about 10% of the I/O pins and chip locations are reserved for application dependent logic.

The SPC is a single circuit board physically housed in the BSC chassis. The transform feature provides a means of forming micromemory addresses from any pattern of bits from the selectors, register or data paths of the microprocessor. The addresses may be used to sequence the microprogram

---

*Special Processor Control module (SPC) and transform module designators are used interchangeably as names for the standard BSC/MPM transform board.

or to set the contents of the N and K registers. In addition, a bit test feature allows selection of upper or lower instructions of a micro-instruction pair. A register transform is also included. The ability to manipulate addresses and registers in this manner gives the BSC the capability to emulate other processors. The details of the transform vary with each specific BSC application.

In addition to transform functions, the SPC module contains other special logic. An IXT register captures Selector S2 outputs. Special logic is included to address and supply data to up to 8K of RAM memory. This memory shall be dual purpose in that it may be accessed as an extension of the File F1 or as a high-speed local memory. This dual purpose memory is called File F3.

Figure 2-4 is a block diagram of the SPC transform module. The transforms are implemented on four selectors:

- Selector S5 - MA Transforms
- Selector S7 - Bit Test
- Selector S8 - N/K Transforms
- Selector S10 - Register Transforms

Data Capture and file management are implemented on five registers:

- IXT Register - Captures Selector S2 outputs
- F' Register - Acts as an input data holding register for File F3. Captures Selector S10 outputs.
- K' Register - Provides addressing input to File F3.
- J Register - Provides addressing input to File F3.
- L Register - Provides address increment used in modifying Register J.

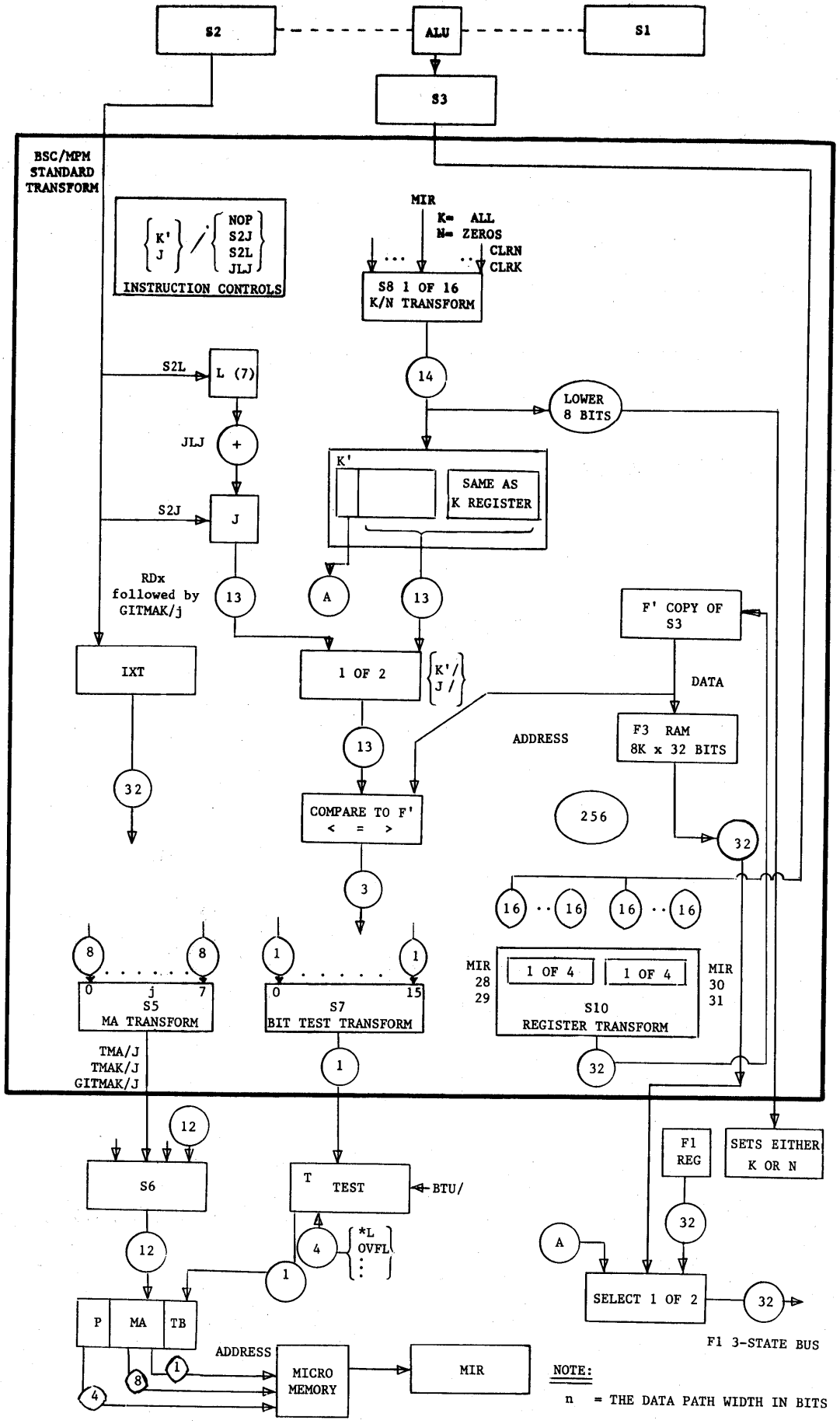


Figure 2-4. BSC TRANSFORM MODULE BLOCK DIAGRAM

MA TRANSFORM

The MA Transform (Selector S5) is an eight-bit wide, one of eight selector that is used to choose between a maximum of eight different sources for forming micromemory addresses. The output of Selector S5 becomes an input to Selector S6 and controls the sequencing of the MP. The eight-bit transformed MA specifies one of the 256 microinstruction pairs within a micromemory page.

The selector position is determined by the letter "j" of TMA/j, TMAK/j or GITMAK/j microinstruction sub-field specification. The selected position j directly corresponds to microinstruction register bits MIR29 through MIR31.

BIT TEST

The Bit Test (Selector S7) is a one-bit wide, one of sixteen selector that allows up to sixteen different external and/or internal conditions to be tested to determine which (upper or lower) microinstruction to execute from the next microinstruction pair. The test bit is selected by the least significant four bits of the microinstruction register (MIR 28 through MIR 31). The output of the selector is sent to the T-field test multiplexer 2 module; the selected bit is tested if the T-field of the microinstruction contains a BTU command.

N/K TRANSFORM

The N/K (Selector S8) is a 14-bit wide, one of sixteen selector that is used to choose between a maximum of sixteen different sources for loading the N, K and K' Registers. The least significant eight bits of S8 are selectively routed to provide data inputs for either the N or K Register. All fourteen bits of S8 become data inputs for the K' Register located on the transform module.

Control of Selector S8 is generated by a combination of MIR bits 28-31 and MIR instruction decoding. The N= and K= instructions force S8 to position 6. The C' codes of CLRK and CLRN force S8 to position 15 (outputs are logical zero) in order to clear the N and K registers.

REGISTER TRANSFORM

The Register Transform (Selector S10) is a 32-bit register transform. It is divided into two 16-bit wide, one of four selectors. The upper 16-bits are controlled by MIR bits 28 and 29. The lower 16-bits are controlled by MIR bits 30 and 31. By splitting the selector into two independent sections, sixteen register transform combinations are possible.

The output of Selector S10 is gated directly to the F' register at T4 on every micro intruction.

IXT REGISTER

This 32-bit register can capture the output of Selector S2. This could be used to hold the macroinstruction currently under emulation. The IXT Register is loaded by executing a microinstruction with GITMAK/j in the C-Field or a GATEIXT in the C" Field.

FILE F3 MANAGEMENT

File F3 is implemented with 2K to 8K of RAM micromemory modules. File F3 serves two purposes. It can be used as an extension of File F1 or as a high-speed local memory.

FILE F1 EXTENSION

Either F1 or F3 is gated to the F1 3-state bus at all times based on the most significant bit of the K' register:

1 = F1
0 = F3

F1 can be written only if the K' MSB = 1.

HIGH-SPEED LOCAL MEMORY

File F3 is gated to the main CPU bus whenever the bus is not being used for another purpose (e.g., A', B' or memory read). It is accessed by a code of 7 in either the S1 or S2 fields (A or B mnemonic of MEM).

F3 DATA

The F' Register is a 32-bit register that holds data to be stored into File F3. The F' Register is an inverted copy of Selector S10 input. F' Register outputs provide data inputs to File F3. Data is transferred from F' to F3 on the instruction following decode of a C" code of 5j (TR/x) or D code of 4 (F1) and the K' MSB = 0.

F3 ADDRESS

Addressing information for File F3 originates on the transform module. Either of the two addressing sources, the K' and J Register, is selected with the corresponding K'/ or J/ transform. The specified register (J/K') is compared with the F' register, providing J/K'<F, J/K' = F', and J/K'>F' indications.

- K' Register - This 14-bit register has three sections. Least significant bits 0-7 are identical copies of bits in the K register on the ALU module. They load, clear, increment and decrement with the K register. The any instance in which the lower K' Register would differ from the K Register would be if IXT was selected as the K/N transform source during a GITMAK. In this case the resulting contents of K' would be undefined. Bits 8-12 load, but do not increment, decrement, clear or interchange carry information with the lower eight bits. Bit 13 selects either F1 or F3 for gating onto the F1 bus and also determines whether F1 can be written. All bits load from the N/K transform.

- J Register - This 13-bit register can be loaded from Selector S2 or perform a two's complement J+L→J operation with C" 6x codes.

- L Register - This 7-bit register loads from S2 with a C" 6x code. It provides a 7-bit (6 bit + sign) two's complement constant for the J+L→J operation.

- For Example: To increment J by 3, Load L with 0/000011 and do a J+L→J. To decrement J by 3, load L with 1/111101 and do a J+L→J.

MISCELLANEOUS LOGIC

The transform module provides a means to implement special logic features. These features include:

- Inverters to complement incoming signals prior to transform and test.

- Clocked register chips to capture incoming signals for test on the following instruction.

- Logic to set and clear status bits (see Status/Mode description.)

MEMORY MANAGEMENT INTERFACE

The memory management interface consists of logic cards in the BSC chassis. The main functions of the memory interface include the following:

- Provides all logic and control necessary to interface the BSC to a large multiport, multibank external main memory system.
- Provides capability to address 4-million words of memory in 2K pages.
- Decouples memory references from CPU operations with random access, firmware supported lookahead logic. This reduces effective memory cycle time by doing address generation in a background mode (overlapped with other CPU activity).
- Supports memory architecture that reduces average memory cycle times by making decisions on address parity and priority resolution in a background mode.
- Four-port interface giving CYBER 18 compatible DMA devices access to the address paging capability and main memory.
- Provides parity, protect and other fault detection and reporting logic.
- Supports full-word, half-word and character operations.

The following topics are discussed in this section:

- MMI Information Flow
- MMI External Data and Address Formats
- Memory Address Generation
- MMI Registers and File

MMI INFORMATION FLOW

The block diagram in Figure 2-5. gives a greatly simplified view of the address generating and data handling logic of the memory management interface of the BSC. The MMI accepts main memory read/write requests from the CPU and DMA devices. On a write request, the output data is formatted (as required by main memory) and saved in a write data holding register. The corresponding virtual memory address is mapped into a physical memory address and saved in a lookahead address register. The MMI then writes the data into main memory. On an initiate main memory read request, the virtual memory address is mapped into a physical memory address and saved in a lookahead address register. The MMI then reads the corresponding main memory data into a read data holding register after formatting the data (as expected by the requestor). The requestor completes its read memory request by telling MMI to transfer the read data holding register content to it (the requestor). Since the DMA devices only handle 16-bit data words and the main memory is 32-bits per word, the DMA interface handles the assembly/disassembly of data between the two formats (for efficiency reasons). The DMA interface also contains data holding registers for each of the four channels.

The mapping of a virtual address to a physical address is performed by the MMI utilizing the virtual address, a state register or S3 and a word of data from the page file. CPU address mappings use the CPU state register or S3. DMA address mappings use the DMA state registers; a separate DMA state register is dedicated to each channel.

The content of the mapping (paging) registers is specifiable by special CPU microinstructions that can read/write any particular state register or page file word. Thus, the CPU is the ultimate controller of the virtual to physical address mapping function.

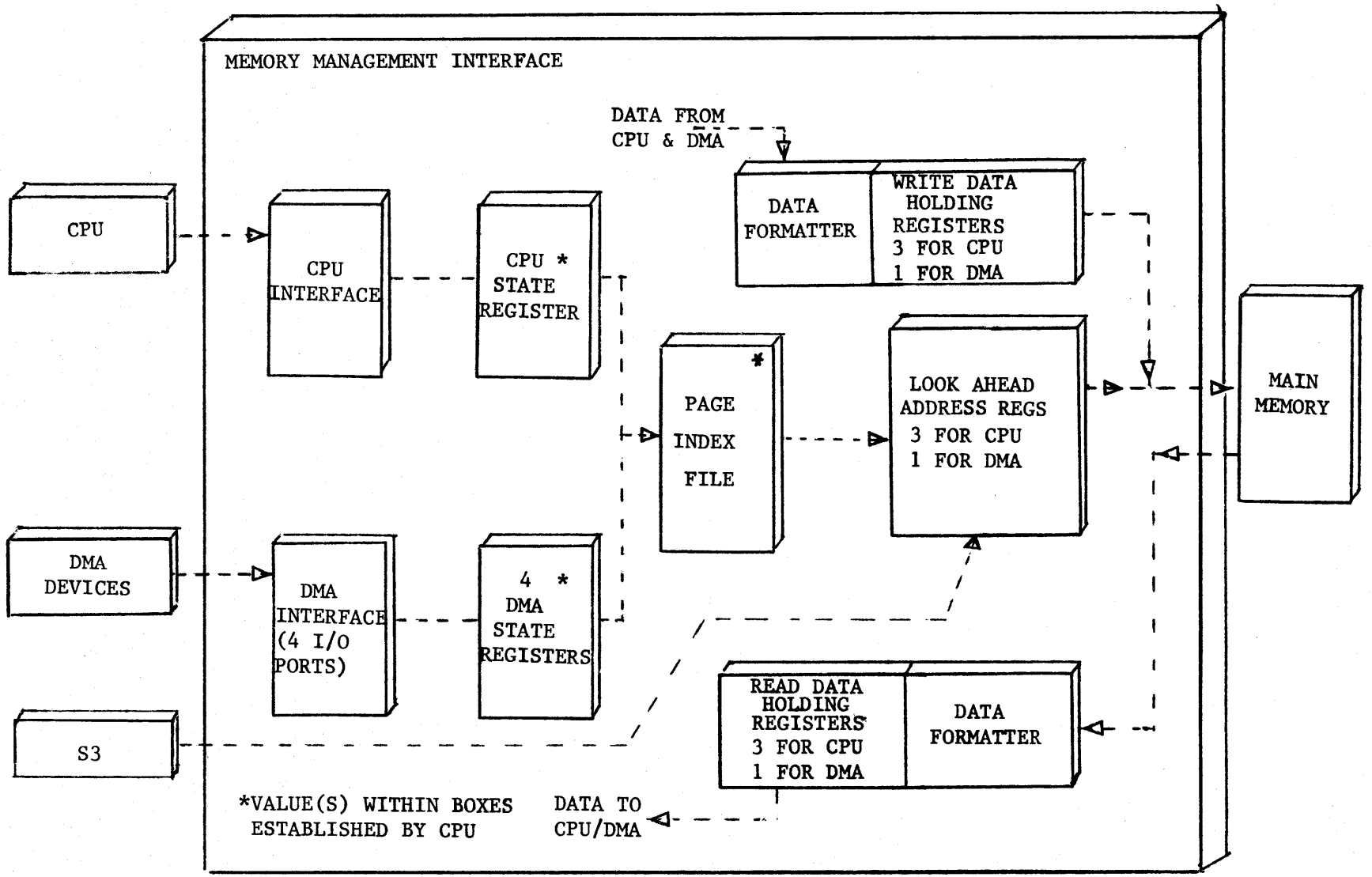


Figure 2-5. Memory Management Interface Block Diagram

MMI EXTERNAL ADDRESS AND DATA FORMATS

Figure 2-6. summarizes pictorially the seven address and data formats, along with their corresponding microinstructions that the CPU can utilize to reference either main memory or the MMI paging registers. The CYBER 18 compatible devices only use the half-word address/data mode and the DMA interface uses either assembly/disassembly logic and full-word addressing and data mode, or half-word address and data mode (selectable on a channel basis) to interface with main memory.

For data output operations, Selector S1 contains the data and Selector S3 specifies the address. The appropriate microinstruction C' field specifies the addressing mode. For main memory writes, the i-portion of the micro-instruction mnemonic specifies which of the CPU's lookahead address and output data holding registers to use, and a bit in the CPU state register specifies that the operation is an output (write).

For data reads (inputs) of main memory, three distinct operations are required: initiating the read, gating the read data onto the CPU 3-state bus and then gating the 3-state bus to a destination register. A read is initiated by specifying the appropriate C' field instruction (LWAi, LHAi or LCAi) and the corresponding Selector S3 main memory address; a bit in the CPU state register specifies that the operation is a read. The data read from memory is gated from the appropriate input data holding register (i) to the main 3-state CPU bus with S field instruction RDi. The MEM instruction described in the instruction detail section is one way of gating the 3-state CPU bus to a destination register.

A state register or word of the page file is gated to the 3-state CPU bus by specifying either C' field microinstruction of RSR or RPF. Selector S3 addresses the particular state register or page file word that is to be read.

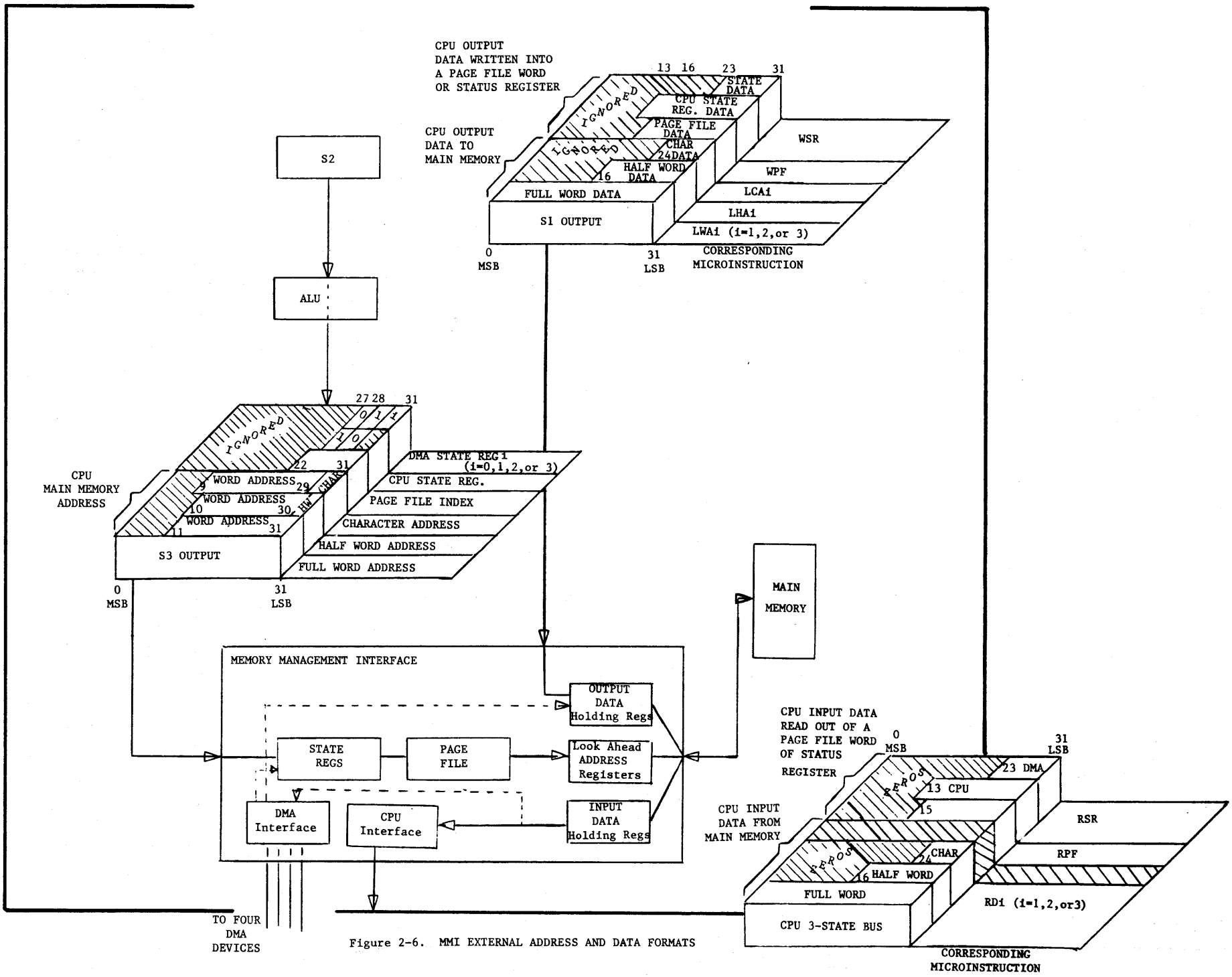


Figure 2-6. MMI EXTERNAL ADDRESS AND DATA FORMATS

Figure 2-7. shows the format of an address that MMI sends to main memory. The 22-bit word address is generated by mapping a virtual address to a physical word address and provides the capability to address 4-million words of memory organized in 2K pages. The MPM uses the selected addressing mode and address to generate the 4-bit control word to specify which characters within the word are being addressed. Figure 2-8. shows the correspondence half-word and character addresses and the associated data formats in main memory.

MEMORY ADDRESS GENERATION

Memory address generation converts a virtual address to a main physical address which consists of a 22-bit physical word address and a 4-bit character mask (previously explained). The virtual address originates from Selector S3 for CPU address and the DMA address bus within the MMI for DMA interface addresses. Note: DMA devices send an address to their DMA interface channel (contained on MMI) and the interface generates word or half-word address on the DMA bus.

There is a possibility of three addressing modes and associated address formats, as depicted at the top of Figure 2-9. The low order 11 bits of word address portion within the virtual address are identically the 11 low order (least significant) bits of the physical word address (i.e., address within a 2K size page). The next 5 higher bits in the virtual address form the lower half of the page file index; the least significant 5 bits from the proper state register (CPU or DMAi) form the upper half of the page file. If SM204 bit is a 1 (set), then deadstart mode is selected and page file index is directly specified by the upper 10 bits from S3 of word address field within the virtual address. Having a complete page file index, this item specifies 1 of the 1024 words contained in the page file. The lower 11 bits of this word are the upper 11 bits of the physical word address. This completes the generation of a physical word address from a virtual address.

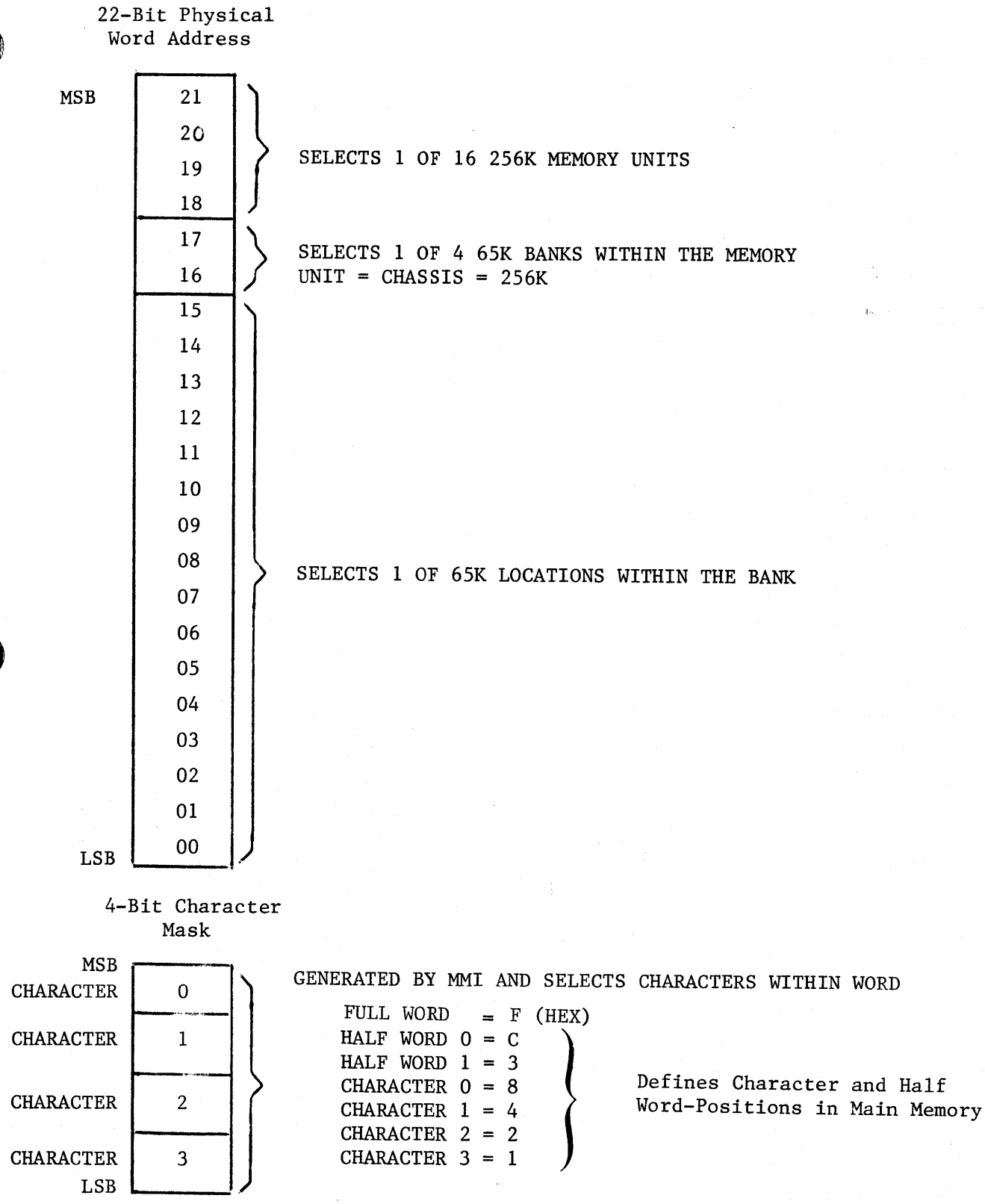


Figure 2-7. Memory Address Structure Between MMI and Main Memory

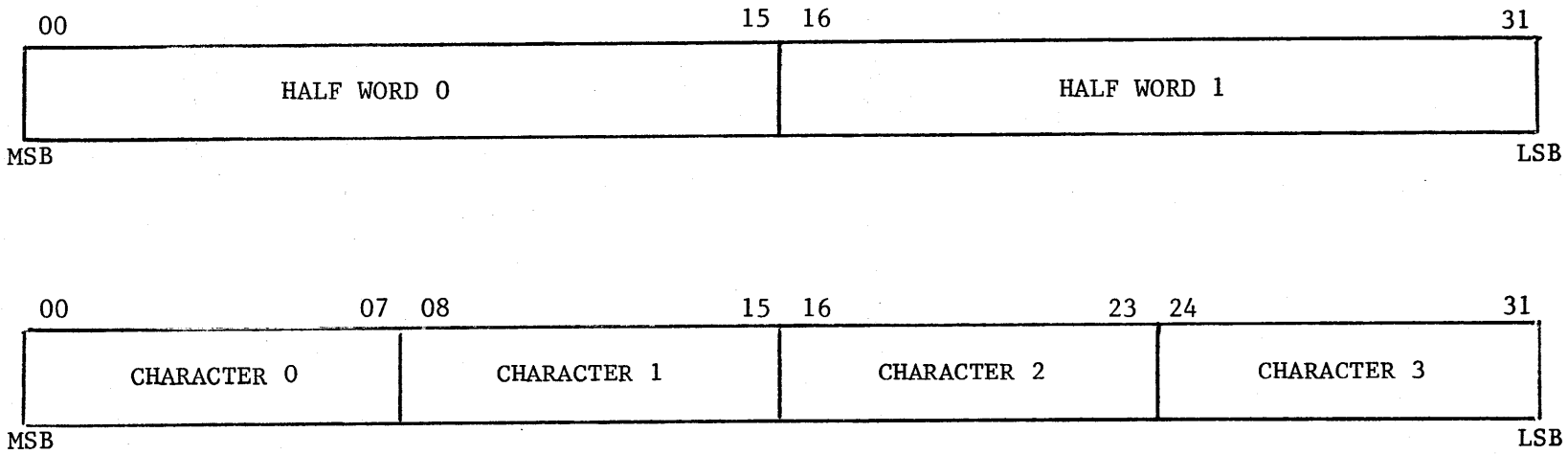
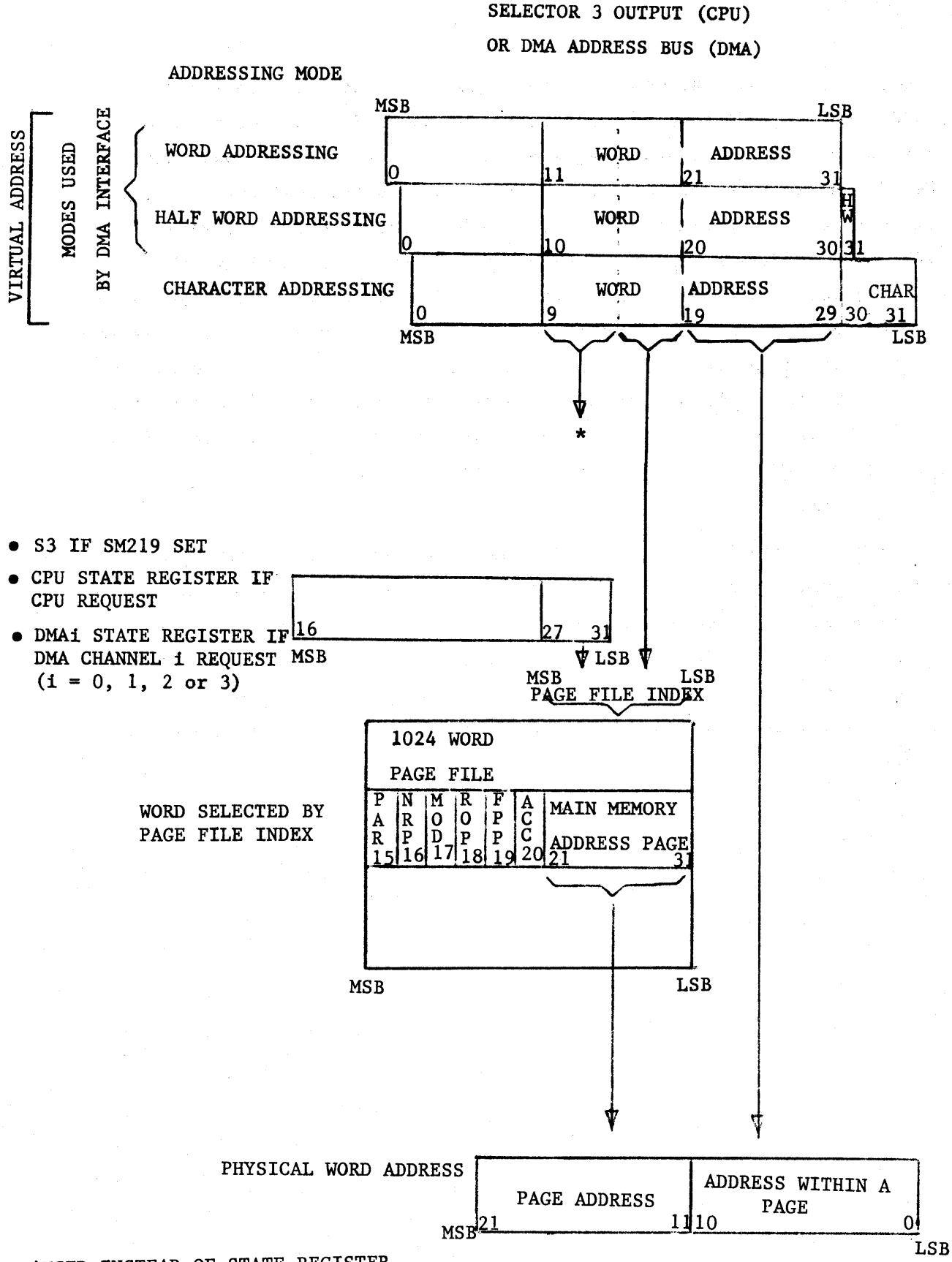


Figure 2-8. Data Formats in Main Memory



*USED INSTEAD OF STATE REGISTER OUTPUT IF SM219 IS SET.

Figure 2-9. Illustrated Mapping of Virtual Addresses to Physical Addresses

From a virtual/physical memory paging viewpoint, the state register divides the page file into 32 groups, each having 32 consecutive page file words. Thus, the state register selects 1 of 32 page groups and the CPU and each of the DMA channels (via their state register) are each set up to use a particular page set. Each of the 32-page file words within a page file group can specify any 2K physical page within this BSC's address space (4 million words or 2048 pages). In fact, they could all specify the same page if desired. In summary, a state register selects a page file group (32 words), the page file group selects a set of 32 2K size pages within the main memory architecture, the virtual address selects one of the 32 pages and specifies a particular word within a page. Figure 2-10 illustrates the page structure of main memory.

The address space is actually richer than described here because each memory chassis, via a four-bit dip switch, can be set to correspond to one of the possible 16 addresses specified by the upper four bits of the 22-bit physical word address. This feature facilitates the rich address space needed in multi BSC/multi port memory configurations used in special applications. See Multi Port Memory section for additional details.

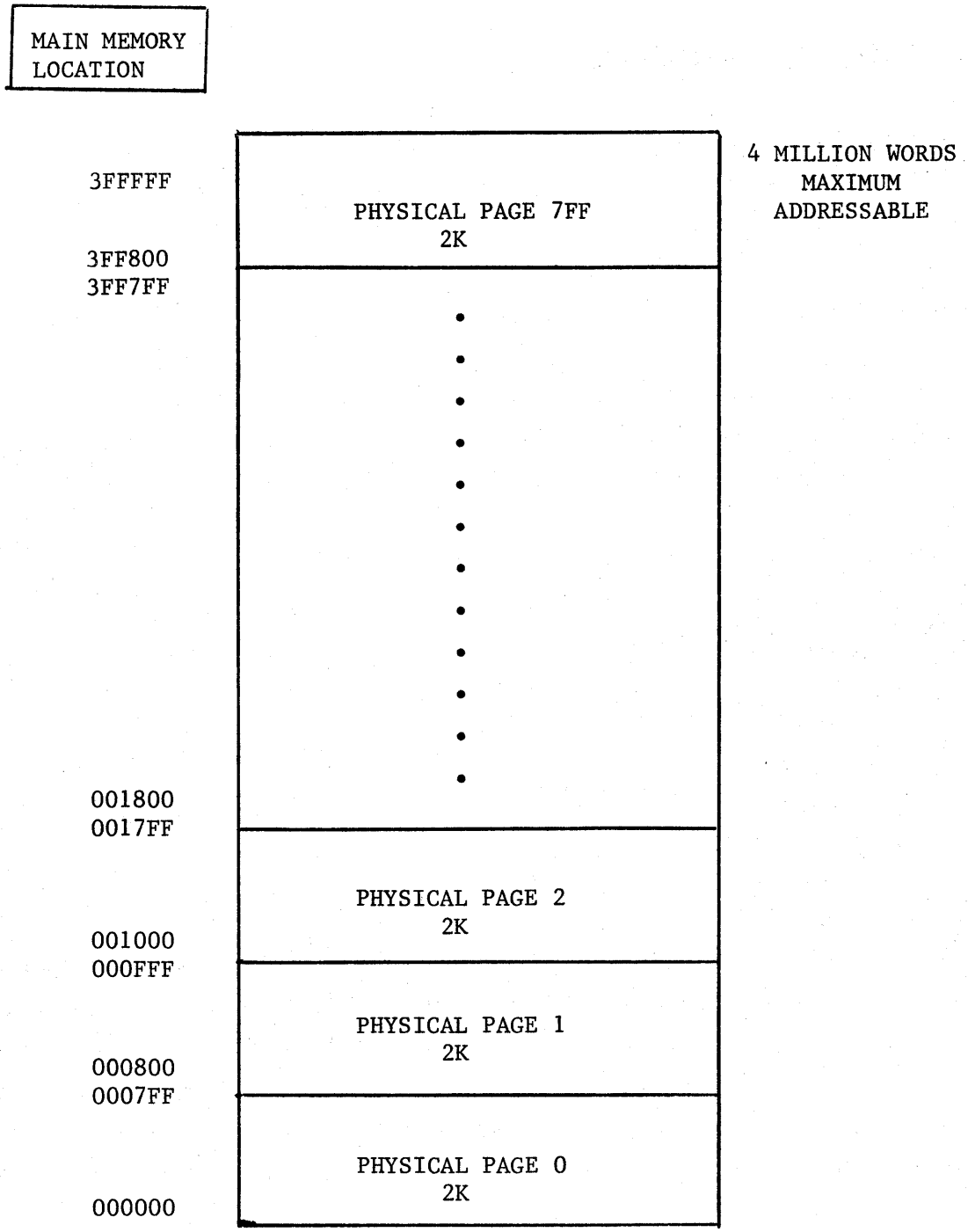


Figure 2-10. Page Structure of Main Memory

MMI REGISTERS AND FILE

The Memory Management Interface uses the content of state registers, a page file and lookahead registers to convert CPU/DMA requests into read/write requests that are sent to main memory.

STATE REGISTERS

The memory interface contains one CPU state register and four DMA state registers, one per DMA port. As illustrated in Figures 2-11. and 2-12, the state registers perform several functions for their associated I/O logic areas:

- Provide upper half of page file address for memory allocation.
- Provide mode control and set up parameters.
- Report fault conditions.

CPU STATE REGISTER

The CPU State Register performs several functions for the memory management logic. It provides page file address information to the address generation logic, provides mode control the lookahead logic and works with the address snapshot registers to provide error reporting. The CPU State Register is loaded by the CPU via S1 and by the memory management error reporting logic. Bit assignments are shown in Figure 2-11. The register may be read by the CPU via the main CPU bus.

Addressing - Bits 27-31 form the upper page file address, selecting one of 32 groups of 32-page file locations. Bits 27-31 shall be routed to the backpanel so that they are available for use by the transform module and the Function Control Register. These bits are undefined after master clear.

CPU STATE REGISTER

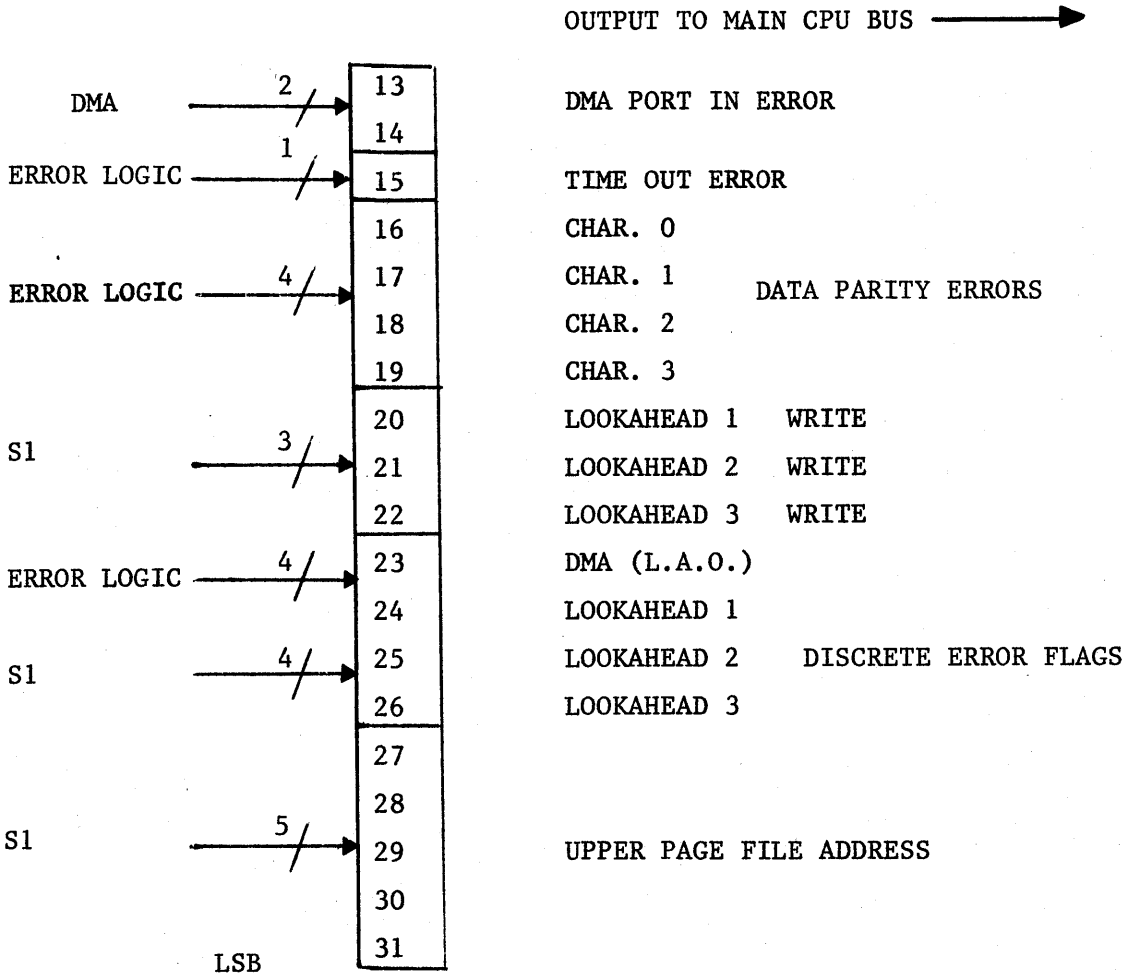


Figure 2-11. CPU State Register

TYPICAL DMA STATE REGISTER

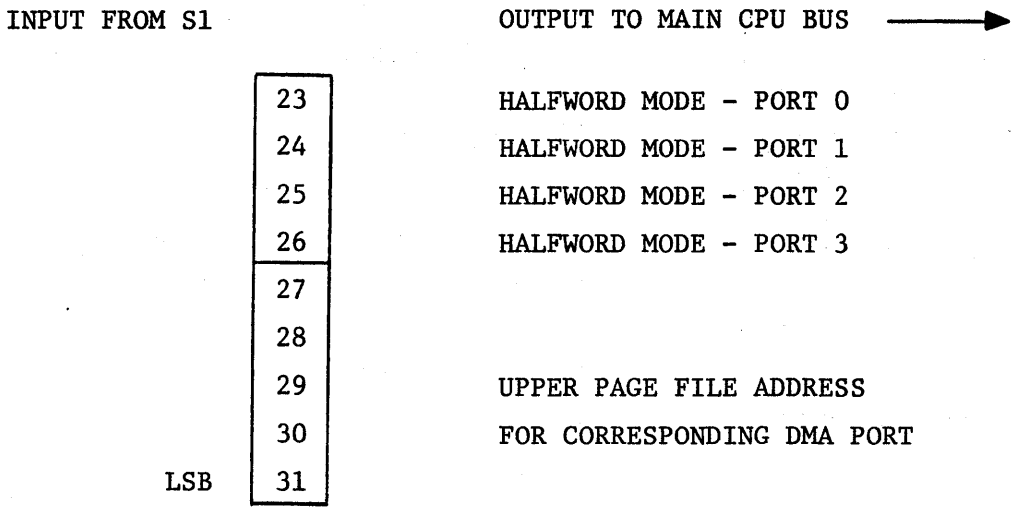


Figure 2-12. Typical DMA State Register

Mode Control - Bits 20-22 are mode control bits for the three lookahead address/data register sets associated with the CPU. Setting a bit puts the corresponding register set in the write mode. Clearing a bit puts the corresponding register set in the read mode. These bits are cleared by Master Clear.

ERROR REPORTING CAPABILITIES

- Bits 13/14 - These bits are updated continuously and have no significance until a DMA error occurs and bit 23 is set. At this time they contain the encoded port number of the DMA port which caused the error. Bit 23 setting blocks any further updating of these bits until a WSR (Write State Register) function is issued to the CPU State Register. These bits are undefined after Master Clear.

- Bit 15 - This bit will be set when a timeout error occurs. One of bits 23/26 will also set at this time indicating which lookahead register set generated the timeout error. This bit clears on a CPU WSR command or on Master Clear.

- Bits 16/19 - These bits will set when a data parity error occurs. A bit set to a one indicates a data parity error on the corresponding data character. One to four bits could be set on a single error. One of bits 23/26 would set at the same time, indicating which lookahead register set caused the error and at the same time blocking any further setting of data parity error flags by subsequent errors. Master Clear on a CPU WSR command will clear these bits and remove the inhibit update signal.

- Bits 23/26 - These discrete error flags set on any error to indicate which lookahead register set generated the error. In addition, these bits provide inhibits to prevent further updating of other error information until the CPU clears these bits with a CPU WSR command (loading the state register with S1) or with Master Clear.
    - Each bit inhibits updating of the corresponding address snapshot register.

- Any bit will inhibit updating of state register bits 16/19.
- Bit 23 will inhibit updating of state register bits 13/14.

These inhibit flags cannot prevent multiple errors, but rather assign some significance to the error information. In the event of an error, bits 13/14 would represent the first DMA port to fail.

Bits 16/19 would report the first data parity error. The four address snapshot registers would hold information representing the first failure for each of the four lookahead register sets. More than one of bits 23/26 could be set in the case of multiple errors.

These discrete error bits must be available to the CPU to signal that an error has occurred.

- Bit 24 (Lookahead 1 Error) shall be routed to a backpanel pin for routing to the T'-Field Test logic. T'=Test 5 (ERL) shall go to the lower microinstruction on a fault.
- Bits 24/26 - The logical OR of these bits is routed to a backpanel pin. This signal may then be routed to a BSC interrupt and/or to the Function Control Register as a CPU error indication.
- Bit 23 (DMA error) shall be routed to a backpanel pin. This DMA error signal may then be routed to a BSC interrupt, bit test and/or Function Control Register status bit.

DMA STATE REGISTERS

The DMA State Registers perform two functions for the BSC. They provide page file address information to the address generation logic and provide mode control information to the DMA port logic. There is a separate set of addressing bits corresponding to each DMA port and a common set of mode control bits servicing all ports. All mode bits and one set of address bits are loaded by the CPU via S1 with each DMA Write State Register (WSR) command. All mode bits and one set of address bits are read back to the CPU via the main CPU bus with each Read State register (RSR) command. Bit assignments are illustrated in Figure 2-12.

Mode Control - Bits 23/26 select halfword (1) or fullword (0) operation for the corresponding DMA port.

Addressing - When a DMA port has use of the address generation logic, the corresponding DMA State Register bits 27/31 form the most significant portion of the page file address, selecting one of 32 groups of 32 page file locations.

PAGE FILE

The page file is a 1024 location by 17-bit bipolar RAM memory. As discussed previously, the page file address originates from a combination of the state registers, S3 and DMA, depending on the operation to be performed. The page file may be loaded by the CPU via 1] and read back to the CPU via the main CPU bus. The page file also accepts inputs from the page status logic and the page address parity generation logic. In addition to providing page address outputs, the page file supplies information to the memory protect logic. Page file functions may be divided into three areas:

Memory Page Addressing
Page Status
Memory Protect Logic

- Memory Page Addressing - As shown in Figure 2-13, each page file location holds the 11-bit address of a 2K page of main memory. The page file can hold only 1024, or 1/2 of the 2048 possible page addresses at any given time. This paging capability gives the BSC the ability to address 4 million words of main memory. It also provides the system software a powerful tool for memory allocation. Memory address generation was previously discussed.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK

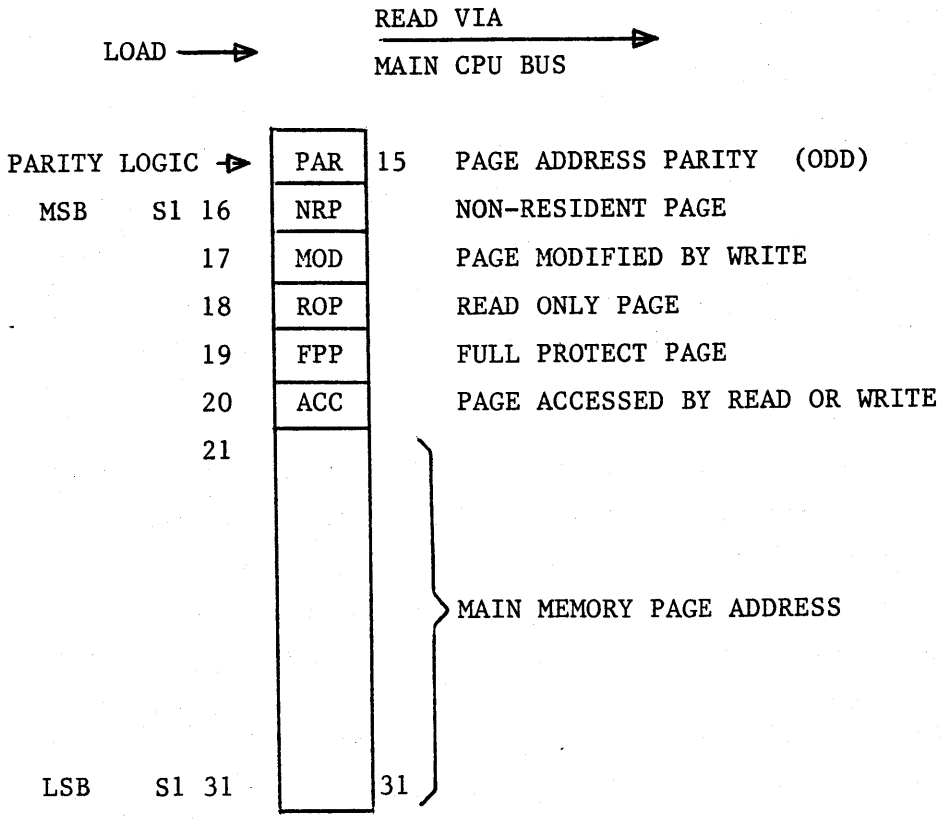


Figure 2-13. Page File Bit Assignments

The page address parity logic generates an odd parity bit on the 11-bit address as it is loaded from S1. This bit is stored with the page address in the page file at bit position 15. When this page file location is accessed, the parity bit is sent with the page address to the address holding register.

- Page Status - Bit positions 17 and 20 hold the MODIFIED and ACCESSED status bits respectively. The page status logic monitors activity of the page file. When a page file location is accessed for a read operation, the page status logic sets the ACCESSED bit in that location. On a write operation, both the ACCESSED and MODIFIED bits are set in the corresponding page file location.

- Memory Protect Logic - Bits 16, 18 and 19 specify the protect state associated with each page. These bits are monitored by the protect logic to ensure that no illegal operations are allowed to occur. The resulting fault conditions are discussed in the Faults, Interrupts and Error Conditions section.

- Non-Resident Page - This bit indicates that the page is physically or logically not present in the system and cannot be accessed.

- Read Only Page - This bit indicates that the page may only be accessed by read instructions.

- Full Protect Page - This bit indicates that the page is not available for any access.

LOOKAHEAD REGISTERS

The purpose of lookahead is to decouple CPU and memory operations and to aid in producing the minimum average memory cycle times. This is not a classical P + 1 lookahead, but a random access scheme with addresses supplied by the firmware. To read data from memory, the CPU presents the read address to the memory interface logic. The CPU is then free to continue firmware operations. The memory interface then generates a read cycle and stores the requested data word in a read data holding register. When the CPU needs the data word, it gates the data from the holding register to the main CPU bus.

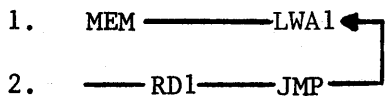
As illustrated in Figure 2-14, lookahead consists of four pairs of read/write data and address output registers, four data input registers and associated control logic to manage register and data availability.

Basic lookahead features include:

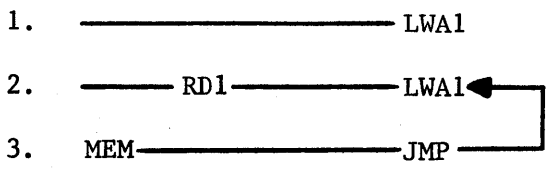
- Decouple CPU and memory operation to avoid stopping CPU operation during memory references.
- Minimize average memory cycle times by having the ability to generate upcoming addresses in a background mode, buffer the addresses and make them available to the memory bus as needed.
- Minimize average memory cycle times by having the ability to supply addresses in an overlapped cycle mode.
- Have the ability to handle one DMA and three CPU operations simultaneously.
- Have the ability to do multiple reads to the CPU of a data sample in a CPU read data holding register.

The following examples define and illustrate the various levels of parallelism available in referencing main memory.

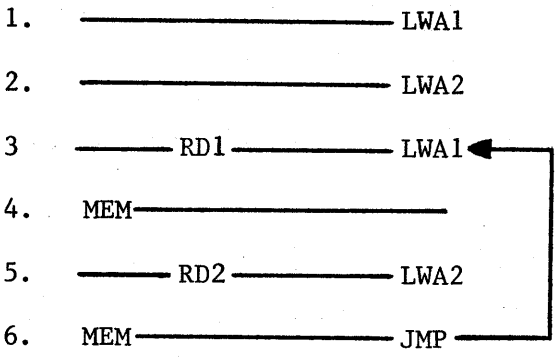
COLD CYCLE - The CPU is not taking advantage of the lookahead capabilities of the memory interface. The following program illustrates a cold cycle:



WARM CYCLE - The CPU is taking advantage of lookahead capabilities, but utilizing only one set of lookahead registers. An address is constantly available for loading into the lookahead address register. The following program illustrates a warm cycle:



HOT CYCLE - Maximum CPU speed, utilizing full lookahead features and more than one pair of lookahead registers. An address is always available to the memory address bus. The following program illustrates a hot cycle:



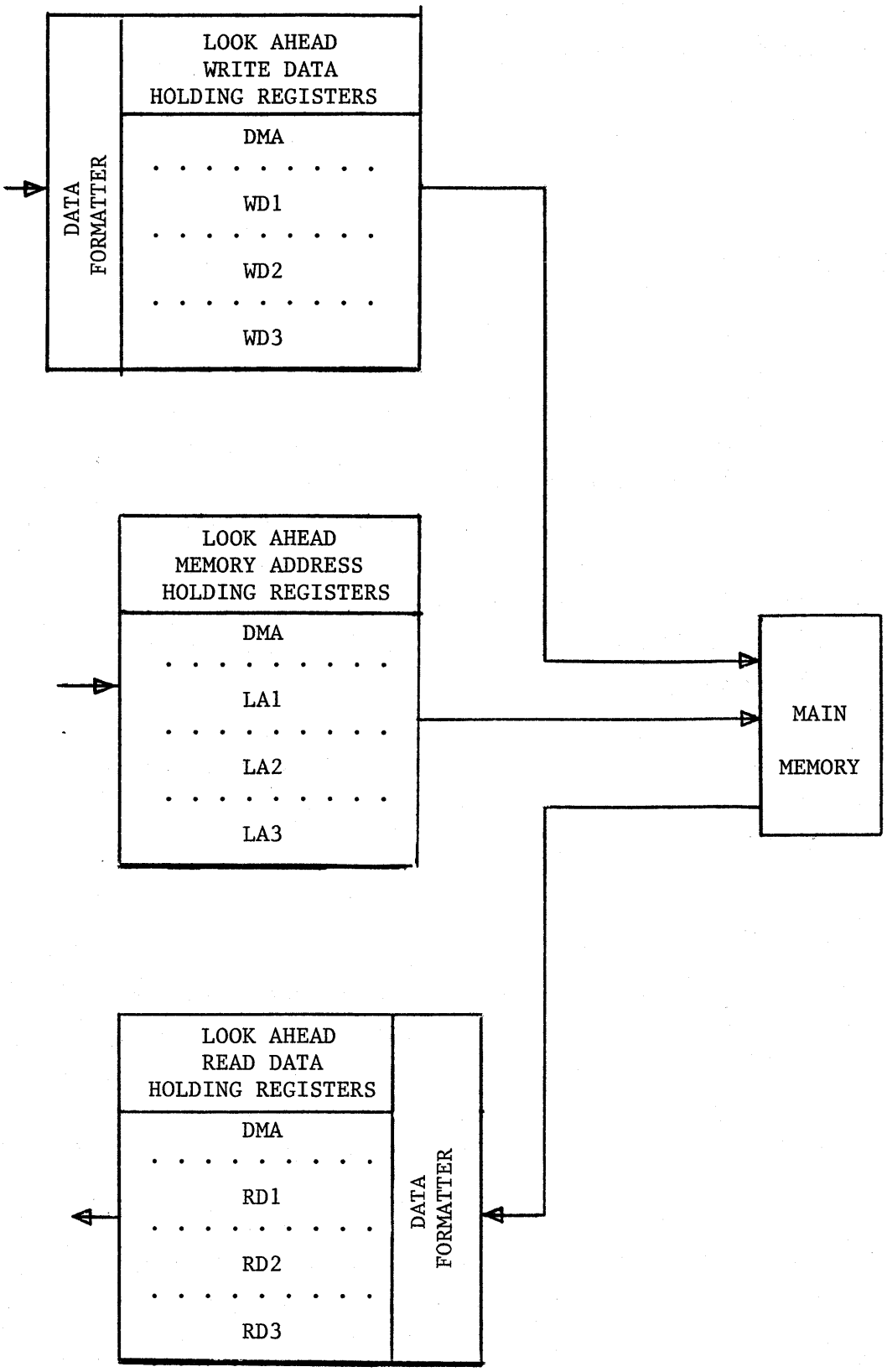


Figure 2-14. Look Ahead Registers

FAULTS, INTERRUPTS AND ERROR CONDITIONS

Errors associated with the memory interface fall into three categories:

- Protect Violations
- Addressing Faults
- Parity Errors

Errors associated with the four sets of lookahead registers are reported to the CPU State Register, as shown in Figure 2-11. Faults are combined by logical OR into one backplane signal for CPU faults and one backplane signal for DMA faults. These signals are routed to BSC interrupts and to the T-Field test logic. DMA devices are not informed of error conditions such as protect fault, parity error and addressing error. These faults are reported to the CPU and must be handled by the firmware.

Instruction cycles during which an error is detected end with the associated lookahead logic in a non-busy state. Neither read nor write cycles are left pending.

PROTECT VIOLATIONS

- Full Protect Page Fault - This bit indicates that an attempt was made to access a fully protected page (FPP bit set in page file). Bits 00 and 03 are set in the associated address snapshot register, memory cycle is initiated, and the respective discrete error flag is set in the CPU State Register.

- Read Only Page Fault - This bit indicates that an attempt was made into a read only page (ROP bit set in page file). Bits 00 and 02 are set in the associated snapshot register, no memory cycle is initiated, and the respective discrete error flag is set in the CPU State Register.

CONFLICTS AND PRIORITY RESOLUTION

There are three areas that require priority resolution:

- Between the four DMA ports.
- Between DMA and CPU for use of the memory management logic.
- Between DMA and CPU lookahead logic for use of main memory.

- First, one of the DMA ports is selected on a fixed priority basis:

    1. DMA I/O Channel (Port) 0
    2. DMA I/O Channel (Port) 1
    3. DMA I/O Channel (Port) 2
    4. DMA I/O Channel (Port) 3

- The selected DMA port must then obtain use of the paging logic, but the CPU has priority in this case.

- After forming the main memory address and loading the lookahead register, the DMA has priority over the CPU for use of main memory. The lookahead registers obtain memory based on the following priority:

    1. DMA
    2. Lookahead 1
    3. Lookahead 2
    4. Lookahead 3

  The DMA interface is configured in logically independent blocks to allow time savings via background assembly/disassembly operations.

ADDRESSING ERRORS

- Non-Resident Page Fault - This bit indicates that an attempt was made to access a page that is logically not currently part of the memory system (NRP bit set in page file). Bits 00 and 01 are set in the associated address snapshot register, no memory cycle is initiated, and the respective discrete error flag is set in the CPU State Register.

- Time-Out Error - If an attempt is made to access a memory location that is physically not present in the memory system, the memory unit does not respond and a time-out error occurs.

PARITY ERRORS

- Data Parity Errors - Parity is generated in the BSC memory interface on each 8-bit data character. It is checked again in the memory interface on data characters being read from memory.

- Time-Out Error - Address parity is generated on the page address before it enters the page file and on the lower 11 bits prior to loading into the address holding registers. If an address parity error is detected by the memory, no response is made to the CPU request. The meory interface logic times out giving a time-out error.

INPUT OUTPUT INTERFACE

As illustrated in Figure 2-15, the Input/Output Interface consists of an I/O-TTY module and peripheral controllers/adaptors that communicate with the I/O-TTY module via the register I/O bus. Controllers/adaptors are also connected to their associated peripheral devices. If a controller/adaptor has a CYBER 18 compatible DMA channel, then the channel is connected to one of the four DMA I/O ports of the Memory Management Interface.

PERIPHERAL CONTROLLERS AND ADAPTERS

A peripheral controller/adapter is a single P.C. card which is housed in the BSC chassis. Typically 8 card slots are available for this purpose. The controller/adapter has either a CYBER 18 compatible or NCR M05 compatible interface to communicate with the I/O-TTY module register I/O bus. The CPU uses this interface to communicate with the controller*: direct the controller, obtain the current status of the controller and exchange peripheral device data if the controller is not utilizing a CYBER 18 compatible channel to exchange data with memory. Although not explicitly shown in Figure 2-15, status/mode register bits and interrupt lines are also used for communication between the CPU and controllers.

As options, several standard controllers and peripherals are available. Additionally, since the BSC/MPM was specifically designed to handle special applications, CDC can (upon mutual agreement with a customer) design controllers to satisfy a customer's special requirements.

I/O-TTY MODULE

The I/O-TTY Module is a single P.C. card which is physically housed in the BSC chassis. (See Figure 2-16.)

---

*For convenience of discussion, controller and controller/adaptor are used interchangeably throughout the rest of this manual.

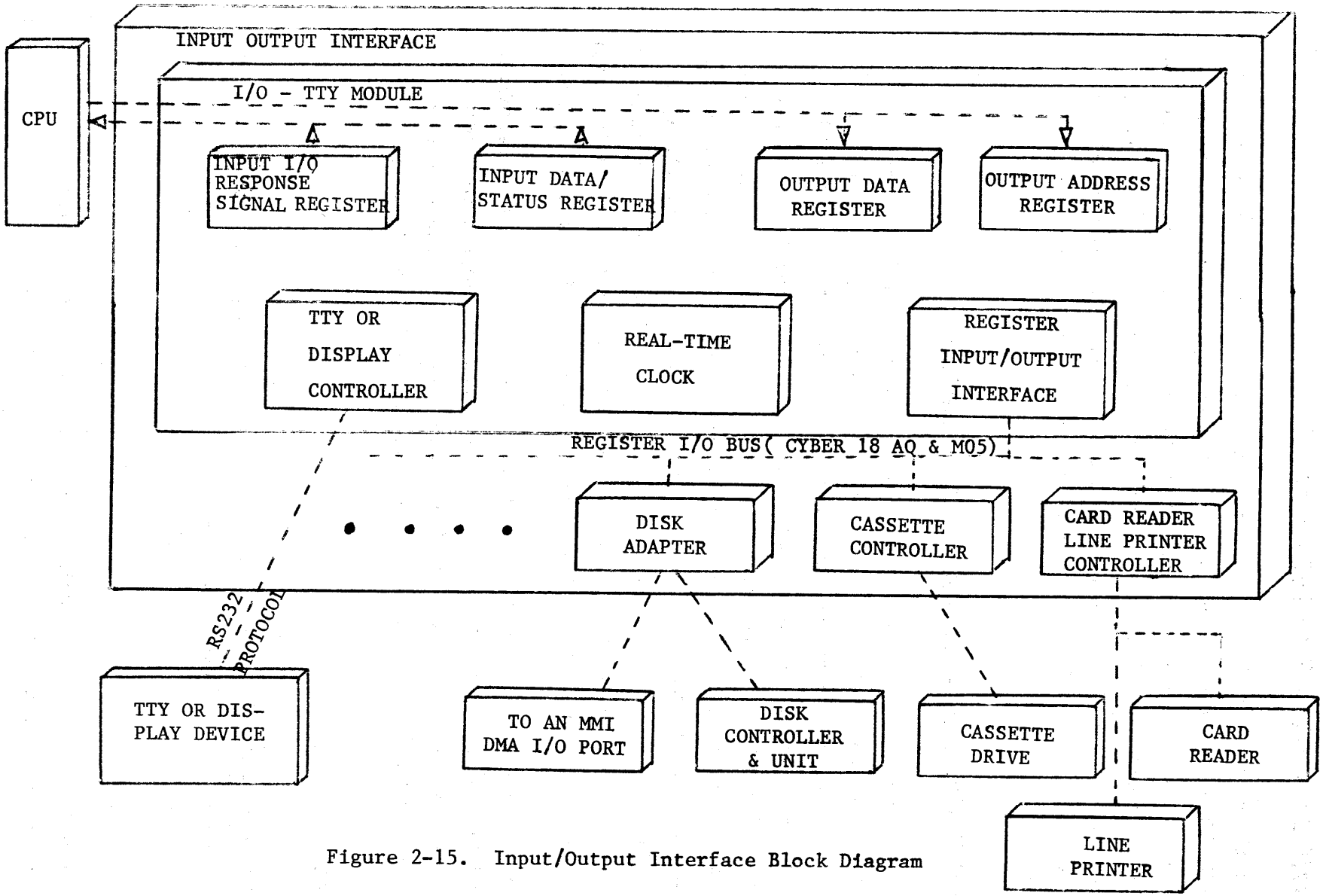


Figure 2-15. Input/Output Interface Block Diagram

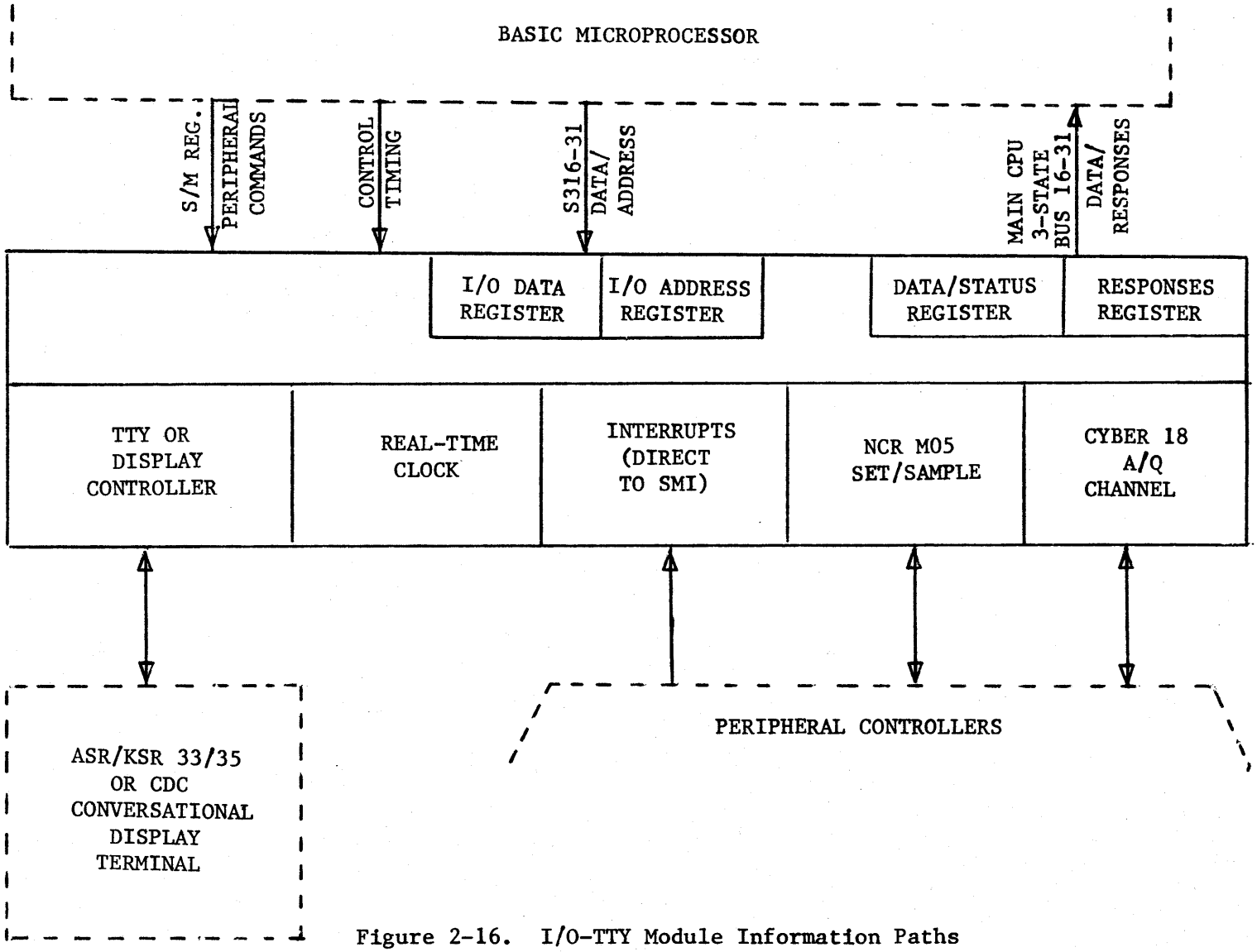


Figure 2-16. I/O-TTY Module Information Paths

BASIC FUNCTIONS

- The Real-Time Clock, in conjunction with the micro code, appears as a CYBER 18-type peripheral to the macro-level programmer. This clock is mainly used to measure elapsed time.

- The I/O Controller for Teletypewriter or Display acts as a controller for Teletype Model ASR/KSR 33/35 teletypewriter or asynchronous RS232 type Conversational Display Terminal.

- The Internal Peripheral Controller Bus provides all I/O data lines and control signals necessary to generate, in conjunction with firmware, both an internal CYBER 18 A/Q bus and an NCR M05 set/sample I/O bus. Any controller cards housed in the BSC requiring the use of one of these busses (e.g., the CR/LP Controller) is connected to the I/O-TTY Module via special backpanel wiring.

MICROPROCESSOR INTERFACE

The I/O-TTY Module is interfaced to the Basic Microprocessor as follows:

- ALU Output - All output data and address information is provided from the output of the ALU, via S3.

- SM Register - All commands to peripheral controllers are generated by micro code manipulation of the MP status mode register. Refer to the descriptions of SM200,SM201,SM202 and SM203.

- MP Control - Timing and control information for controlling internal I/O module data gating is provided from the MP control signals.

- Interrupts - Interrupts from peripheral controllers (within the basic chassis) are wired directly from the peripheral controller module to the MP SMI modules.

- Input Data and Peripheral Response Signals - All of these are provided to the MP on the main CPU 3-state bus.

- Clock - The clock circuit is an integral part of the I/O-TTY module and is designed to appear as a CYBER 18 peripheral to the macro-level software.

GENERAL INFORMATION FLOW

Figure 2-16. illustrates two I/O output registers, two I/O input registers, the A/Q M05 interface to controllers and two controllers/devices (TTY/CDT and RTC) that are local to the I/O-TTY module.

The CPU sends information to the I/O-TTY module by placing data into the I/O output registers Y and D located on the module and also by setting and clearing SMI bits. This information selects a particular controller and sends data/control information to it. The selected controller will perform an operation such as output data to a peripheral, input data from a peripheral for the CPU, interpret the data as control information (director function) or make its status (director status) available to the CPU. Information from the selected controller is input by the CPU from input registers located on the I/O-TTY module. The I/O Input Data/Status register contains either input data or director status. The I/O Input Reply/Reject register contains reply/reject information about the operation performed by the controller.

The I/O-TTY module, in conjunction with the BSC, firmware and peripheral I/O controllers, provides an I/O technique that is fully compatible with the CYBER 18 computer A/Q scheme and NCR M05 scheme of I/O systems. This A/Q scheme utilizes the CDC data/address (director function and status/W-E-S/D field) word and replay/reject (handshake) conventions for processing and addressing I/O data within the BSC. The NCR M05 I/O scheme employs a set/sample (write/read) technique whereby the data is placed on the input/output line (direction is determined by set or sample request).

I/O OUTPUT REGISTERS

The I/O-TTY I/O output registers receive data and address words from the CPU via selector 3 (bits 16 through 31) of the ALU for use in the controllers. These registers are designated D (data) and Y (address) and are

used functionally like the CYBER 18 I/O registers A and Q, respectively. The Y register is loaded by gating the data from selector S3 with a D' = 001 (mnemonic IOA). Since the selector S3 data passes through the D register to get to the Y register, the contents of the D register are always destroyed when the Y register is loaded. The D register is loaded by gating the selector S3 output with a D' = 000 (mnemonic IOD). The D register contains either director functions or data for the I/O controllers. The format of a director function word is unique to each peripheral controller; the director code field in the Y register is used to specify which format is being used when a particular controller has several different director function words (see Figure 2-17 for a typical director function word).

For notational convenience, D register bits are labeled either D01 through D16 or SD01 through SD16, and Y register bits either Y01 through Y16 or ADR01 through ADR16. The data bits of the address word impose the address word contents upon the peripheral I/O controllers and I/O-TTY controller functions. Address word contents are decoded in accordance with the CDC CYBER 18 WES/D convention as indicated in Figure 2-18. The I/O-TTY module forces the W field equal to zero during an ADT operation.

The WES field of the Y register selects a particular controller and the director code (D field) of the Y register, in conjunction with a read or write, determines the controller operation performed (see Table 2-5).

When M05 set/sample controllers are used, output lines ADR02 through ADR04 and ADR05 through ADR07 are designated SMB07 through SMB09 and SPOS01 through SPOS03, respectively. These SMB and SPOS lines control the M05 operating mode and the port selection of M05 peripheral devices. When SMB09 is a 1, the 16 bits of information contained in the data register are placed on output lines SD01 through SD16. When SMB09 is a 0, the 16 bits of information on input lines RD01 through RD16 are transferred via the I/O-TTY controller to the I/O Data/Input Registers. The SMB07 and SMB08 bits are not defined and may be employed at the user's discretion. SPOS01 through SPOS03 designate the port selected by placing a discrete signal on the port line (SPT) in accordance with the octal code designation. The M05 set/sample is detailed later in this section.

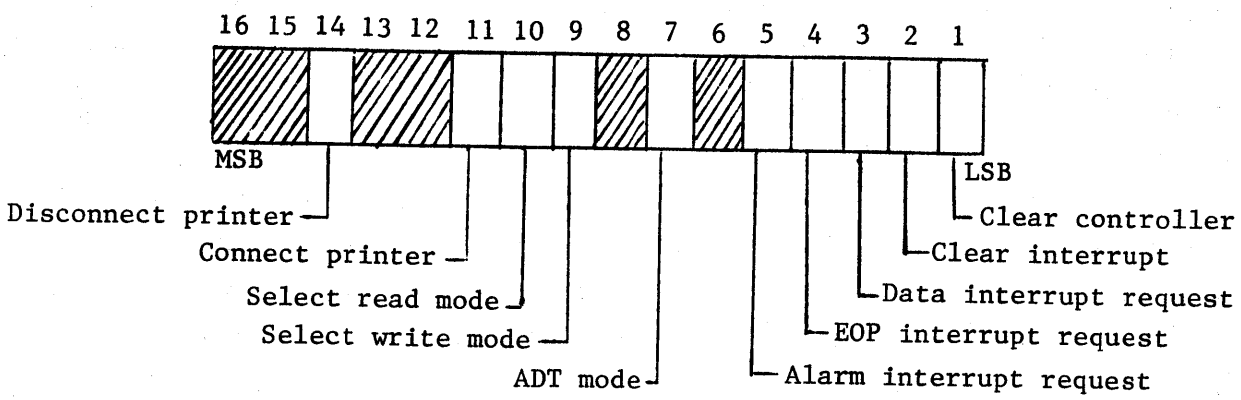
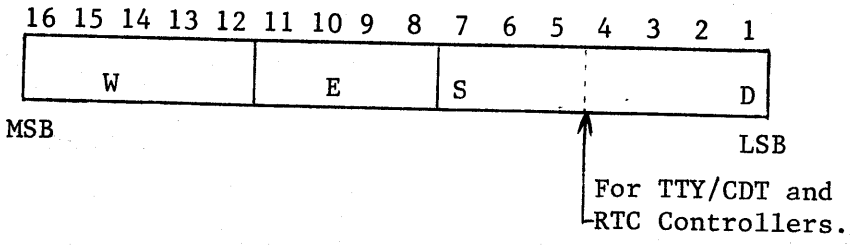


Figure 2-17. Typical D Register Bit Definitions, I/O-TTY Controller Director Function Word



W = Converter code (in this configuration, W is always equal to 0)

E = Equipment number (depends upon I/O configuration)

S = Station code (I/O controller functions)

D = Director code (status and transfer commands)

NOTE: S and D fields have no bound within the S/D zone, except that they may not overlap.

Figure 2-18. Y Register Field Designation for WES/D Convention

TABLE 2-5. EXAMPLE OF CONTROLLER/CARD READER OPERATIONS

D Field Bits in Y Register			
Y02	Y01	Read (SM201=1)	Write (SM202=1)
0	0	Data Transfer	Illegal
0	1	Director Status 1	Director Function 1
1	0	Illegal	Test Mode
1	1	Director Status 2	Director Function 2
All illegal commands will be rejected.			

I/O INPUT REGISTERS

The data read function provides for the transfer of I/O controller data (character data, TTY/CDT status, RTC status, I/O control responses and I/O data from A/Q M05 controller) to the CPU 3-state bus in bit positions 16-31. The data to be gated onto the 3-state bus is determined by a combination of the contents in the microinstruction B' field and the Y register at the time the microinstruction is executed. The Y register selection is decoded in accordance with the CDC CYBER 18 WES/D convention (see Figure 2-18. and Table 2-6.) when the microinstruction B' field equals 2 (B' = 010) to designate INRD, input data/status register (see Figure 2-19). Which data is selected for transfer to the CPU 3-state bus is indicated in Table 2-6. The I/O control response selection is determined when the microinstruction B' field equals 3_8 (B' = 011) to designate INRS (input response signals from I/O Input Reply/Register) and the data format is shown in Table 2-7.

TABLE 2-6. Y REGISTER SELECTION OF READ DATA

W	E	S	D	Data Gated to CPU 3-State Bus from I/O Input Data/Status Register
0	1	1	0	TTY/CDT Data
0	1	1	1	TTY/CDT Status
0	1	7	-	Real-Time Clock Status
-	-	-	-	Data or Status from Peripheral Controllers

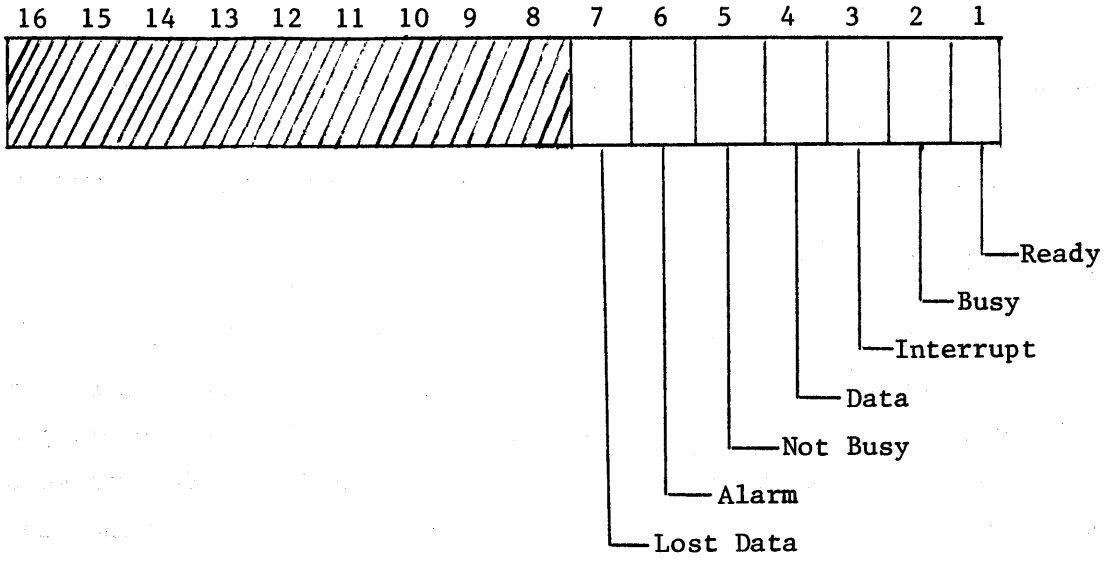


Figure 2-19. Typical I/O Input Register Director Status Word Format

TABLE 2-7. I/O INPUT REJECT/REPLY REGISTER FORMAT

Description		I/O Input Reply/Reject Register Bit Positions
M05 Peripheral	CYBER 18 Peripheral	
Undefined	Undefined	00 and 01
Position 00	Recover terminate (R-TERM)	02
Position 01	Reply	03
Position 02	Reject	04
Direction	Character input	05
Undefined	Undefined	06 through 15

The reject and reply signals are described below.

- REJECT

  The reject signal originates from the I/O-TTY module for the TTY/CDT and RTC controllers and from the A/Q M05 I/O peripheral controllers. If the specified operation cannot be performed by the I/O, at the time the read or write signal appears, a reject signal response is sent to the CPU within 10 µsec. If no I/O response (reply or reject) occurs within 13 µsec after the read or write request, the emulator generates an internal reject.

- REPLY

  The reply signal originates from the same sources as the reject signals. Reply is a signal which indicates to the CPU that the associated I/O device is available to send or receive data.

  Write event sequence:

  1. The CPU bus (internal A/Q channel) transfers data to the appropriate register of the peripheral device.
  2. The peripheral device sends a reply signal to the A/Q channel a maximum of 10 µsec later.
  3. The A/Q channel drops the write signal when the reply is received.
  4. Absence of a write signal for 100 nsec drops the reply.

  Read event sequence:

  1. Available data is gated onto the input bus.
  2. Reply is executed a minimum of 200 nsec and a maximum of 10 µsec.
  3. Reply causes the read signal to drop.
  4. Absence of a read signal for 100 nsec drops the reply signal. The data line signal drops when the reply signal drops.

NCR M05 SET/SAMPLE SCHEME

The NCR M05 set/sample scheme employs a program controlled selection technique which selects the active I/O port, line device and mode. These selections are deciphered from the contents of the send port (SPT00 through SPT07), select send line (SSEL00 through SSEL07), select position (SPOS01 through SPOS03) and select mode (SMB07 through SMB09) lines (see Table 2-8). The send port selection is designated by the octal code value of Y register bits Y08 through Y10. The line selection is designated by the octal code value of bits Y08 through Y10 when Y11 is high. The select position (device) is designated by the octal code value of bits Y05 through Y07. The select mode is determined by the state of Y04. When bit Y04 is high, the set mode is selected which places a data word (16 bits) contained in the D register on the send lines (SD01 through SD16). When bit Y04 is low, the sample mode is selected; the data word on the read data (RD01 through RD16) lines is transferred to the I/O Input Data/Status register. This register can be read by the CPU with an INRD mnemonic instruction and the data is transferred to the CPU 3-state bus.

The Y register bits Y05 through Y16 are deciphered to determine that the NCR M05 scheme has been selected. This is accomplished by the M05 Set/Sample Function detecting that the address word W field bits (Y12 through Y16) are all low (W = 0), the E field (Y08 through Y11) bits hexadecimal value is greater (8_{16}) and the S field bits (Y05 through Y07) are selecting a device.

AUTO DATA TRANSFER (ADT) MODE

Auto data transfer is a pseudo buffered I/O scheme within the BSC. From a macro programming level, I/O operations are performed similarly to a CYBER 18 System with a DMA buffered data channel. ADT mode may be used with controllers that support ADT.

TABLE 2-8. M05 SET/SAMPLE SELECTION

Direct I/O Connection		Multiplexed I/O Connections							
		SPOS01, 02 and 03 Octal Code Equals							
Bit	Port	0	1	2	3	4	5	6	7
SPT00	0	1	9	17	25	33	41	49	57
SPT01	1	2	10	18	26	34	42	50	58
SPT02	2	3	11	19	27	35	43	51	59
SPT03	3	4	12	20	28	36	44	52	60
SPT04	4	5	13	21	29	37	45	53	61
SPT05	5	6	14	22	30	38	46	54	62
SPT06	6	7	15	23	31	39	47	55	63
SPT07	7	8	16	24	32	40	48	56	64

Within either the line printer or card reader controller, ADT mode is enabled by a director function command with bit A06=1. For proper operation, bit A02 must also be set to select the data interrupt. When operating in the ADT mode, the controller will simultaneously generate a data interrupt and a program interrupt when the data interrupt condition occurs. If either or both of the other interrupts are enabled (EOP or alarm), their existence will cause a program interrupt only.

When the simultaneous program and data interrupt is recognized by the CPU, the micro code causes data to be transferred to or from main memory. This I/O operation is transparent to the macro code. At the controllers, ADT I/O functions identically to standard I/O operations.

REAL-TIME CLOCK

The real-time clock receives Y register (Y12 through Y16 and Y08 through Y10) to determine that the RTC function has been selected. The actual elapse time pulses (3.3 milliseconds) are processed by the ADT facilities and only the status signals are coupled to the CPU.

The clock can be used and controlled at the micro level in the following manner:

1. Micro code enables the clock circuit, which causes the clock circuit to generate an interrupt every 3.33 milliseconds (after the first occurrence of the clock interrupt), and enables the detection of a Lost Count condition:

   Set ADT line (SM200)
   Set address lines to $00F3_{16}$
   Set WRITE line (SM202)
   Clear WRITE line
   Clear ADT line

2. When micro code detects the clock interrupt (whenever the clock is enabled and corresponding mask bit is set), it clears the clock interrupt:

   Set ADT line
   Set address lines to either $00F0_{16}$ or $00F3_{16}$
   Set Read line (SM201)
   Clear READ line
   Clear ADT line

3. Another clock interrupt will occur in 3.33 ms and Step 2 is repeated.

4. Lost Count occurs whenever the micro code fails to clear the existing clock interrupt prior to the occurrence of the next clock interrupt in 3.33 ms. Whenever a Lost Count condition occurs, the most significant bit of the main 3-state bus (B' input to the ALU) will be 1 when the clock is selected. If it is desired to check for a Lost Count condition, the micro code must:

   Set address lines to either $00F0_{16}$ or $00F3_{16}$.

   Gate B' input from I/O-TTY board onto the 3-state bus using an INRD B-source code.

   Check if the most significant bit of I/O-TTY data just gated off the 3-state bus is 1. Note: The word gated from the I/O-TTY board is only 16 bits wide whereas the main 3-state bus is 32 bits wide (BUS00-31). The I/O-TTY board is gated onto the lower 16 bits (BUS 16-31) of the main 3-state bus. Thus, whenever a Lost Count condition is present, the bit will be a 1.

5. Whenever the micro code wants to clear the Lost Count status, it must repeat the WRITE operation as outlined in Step 1.

6. Whenever the micro code wants to disable the clock circuit so that no further interrupts will be generated, the WRITE operation, as outlined in Step 1, is repeated except that the address lines are set to $00F0_{16}$.

TELETYPEWRITER/CONVERSATIONAL DISPLAY TERMINAL

The teletypewriter/CDT controller contained on the I/O-TTY module provides a communication link for the transfer of character code data (teletypewriter, RS232-C, or TTL level) to and from the CPU. The communication link supports four baud rates (110, 300, 1200 and 9600) and provides three transmission interface methods:

- Teletype (20 ma current loop)
- CDT or panel interface (RS232-C level)
- M05 or panel interface (TTL level)

All external character code data transmission between this controller and the teletype/terminal is serial, asynchronous, with a program-selectable data format of seven or eight bits with or without parity; parity is always even when selected. Each character data word format consists of a start bit, seven or eight data bits, a parity bit (if selected) and one or two stop bits (two stop bits when 110 baud is selected). All character data transferred within the CPU and I/O-TTY module is parallel format of seven or eight character code bits.

This controller also provides character echo response and sharing of the teletypewriter or CDT between the I/O-TTY module during program mode and the panel interface controller during panel mode. This sharing is accomplished by processing particular special character codes as control characters. These special codes (ESCape ($1B_{16}$), @ (40_{16}), and BELL (07_{16})) are reserved as control codes and cannot be accepted as message data.

RESERVE STATUS - This ESCape ($1B_{16}$) code causes the TTY/CDT to be transferred from the I/O-TTY controller to the panel interface controller.

RELEASE RESERVE - This @ (40_{16}) code causes the TTY/CDT to be transferred from the panel interface controller to the I/O-TTY controller.

MANUAL INTERRUPT - This BELL (07_{16}) code can be used to impose a manual interrupt from the input device instead of a console manual interrupt switch.

BSC PANEL INTERFACE

The Maintenance/Programmer Panel Interface is contained on a single module which plugs into a dedicated card slot in the basic microprocessor. The panel interface provides two basic functions. (See Figure 2-20.)

The first function is to provide a basic man/machine interface. This is accomplished via the Minimum Maintenance Panel and/or an RS232 compatible programmer's console. The Minimum Maintenance Panel is a 16 x 4.5 inch P.C. board which can be mounted directly above the BSC chassis (see Figure 1-2). The Panel connects to the panel interface module through a flexible ribbon cable. Power and ground coneections are provided in the cable. The programmer's console may be a teletype, a CRT terminal or any other RS232 compatible console that has full duplex serial ASCII characteristics. This basic man/machine interface provides for all common computer control panel functions such as start, stop, step, master clear, enter/display register and enter/display memory.

The second function of the panel interface is to provide the basic processor with a deadstart capability: an automated method of loading data into micro or macro memory. A data path is provided for peripheral

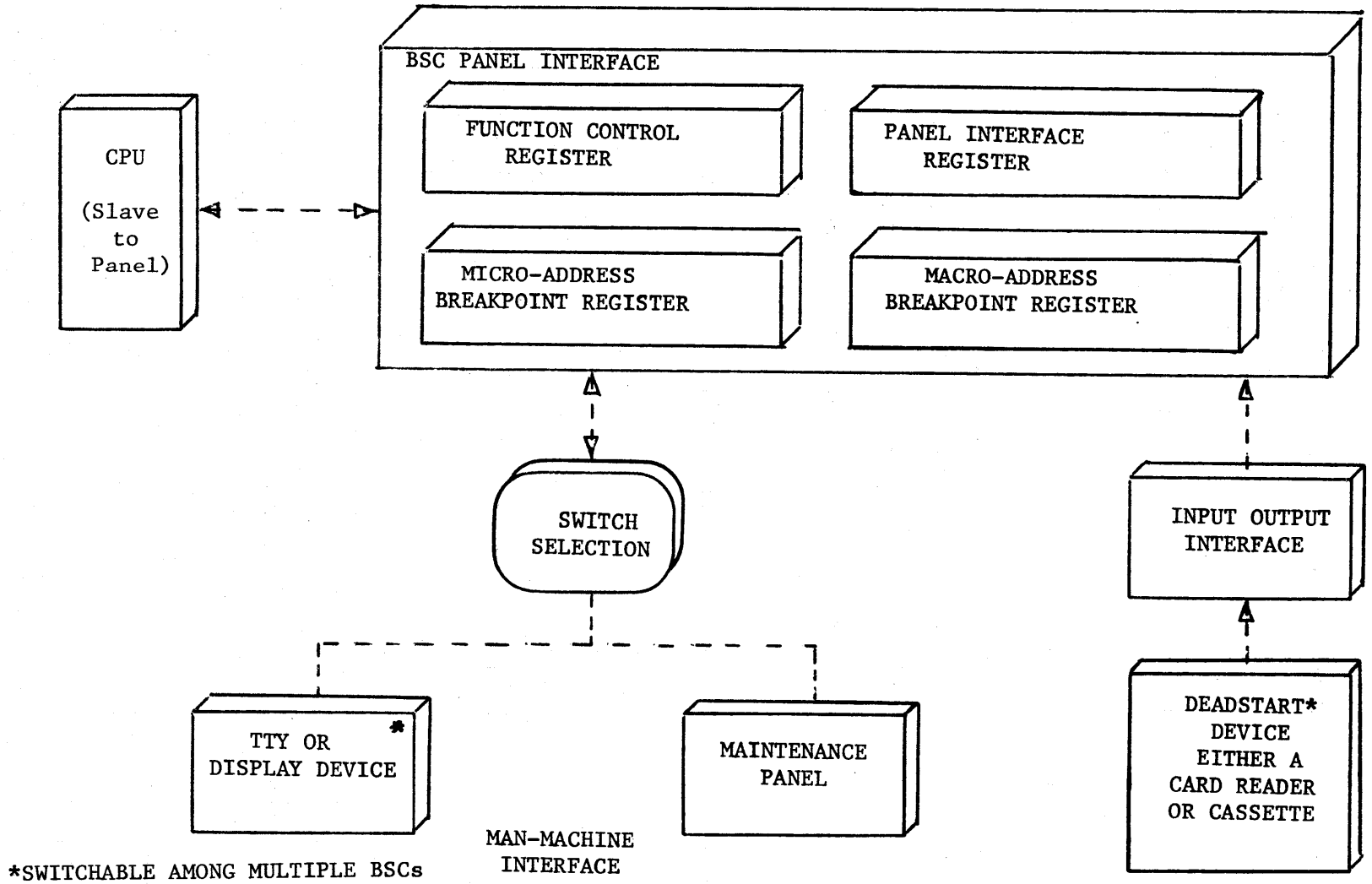


Figure 2-20. BSC Panel Interface Block Diagram

THIS PAGE HAS INTENTIONALLY BEEN LEFT BLANK

controllers to transmit ASCII characters to the processor to perform most operations that can be performed from the maintenance panel or programmer's console. This feature is normally used to load memory from a device such as a card reader or cassette tape unit.

During deadstart operation, the panel I/F executes the panel interface control commands (see Section 3) as they are continuously input from a deadstart device. The deadstart program must be written in terms of the panel interface control commands which select the desired micro or macro memory, set up the initial addresses, load the data and automatically generate a go signal at the end if desired.

PANEL INTERFACE AND MP COMMUNICATION

During the panel I/F operation, the MP is normally required by the panel interface to perform part of the operation. The panel I/F operation can be divided into three basic operations.

- Input operation - During this operation, the panel I/F receives the control command from the maintenance panel or remote devices.

- Change/fetch operation - During this operation, the control command operation is carried out by both panel I/F and MP (except when selected register is either MIR, FCR or BP, the MP is not required). The change operation changes the content of selected register with new entered data. The fetch operation reads out the content of selected register for display.

- Transmit operation - During this operation, the response of control command is transmitted out to maintenance panel or remote device.

The input operation and transmit operation are performed by panel I/F itself. To perform the change/fetch operation, the panel I/F must generate 3 basic sequences:

- Request control of the CPU. The panel I/F must first take over the control of MP processing element. The MP must also prepare itself before giving up the control to panel I/F.

- Generate micro instructions to be executed by MP for the desired change/fetch operation.

- Exit the MP to give back the control. Figure 2-21. and Figure 2-22. show the panel I/F and MP communication flow diagrams depending upon the MP micro or macro mode of operation.

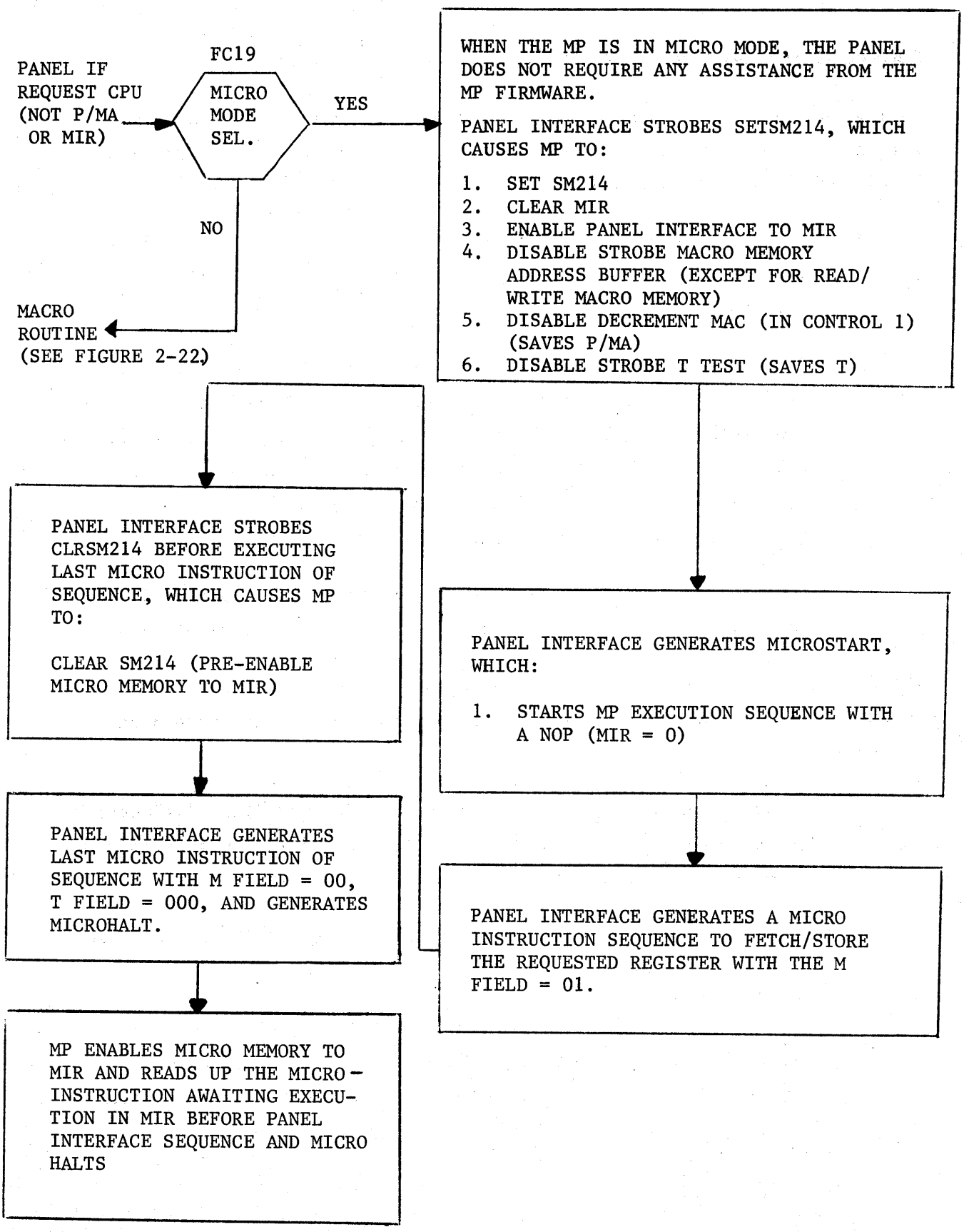


Figure 2-21. Micro Mode Flow Diagram

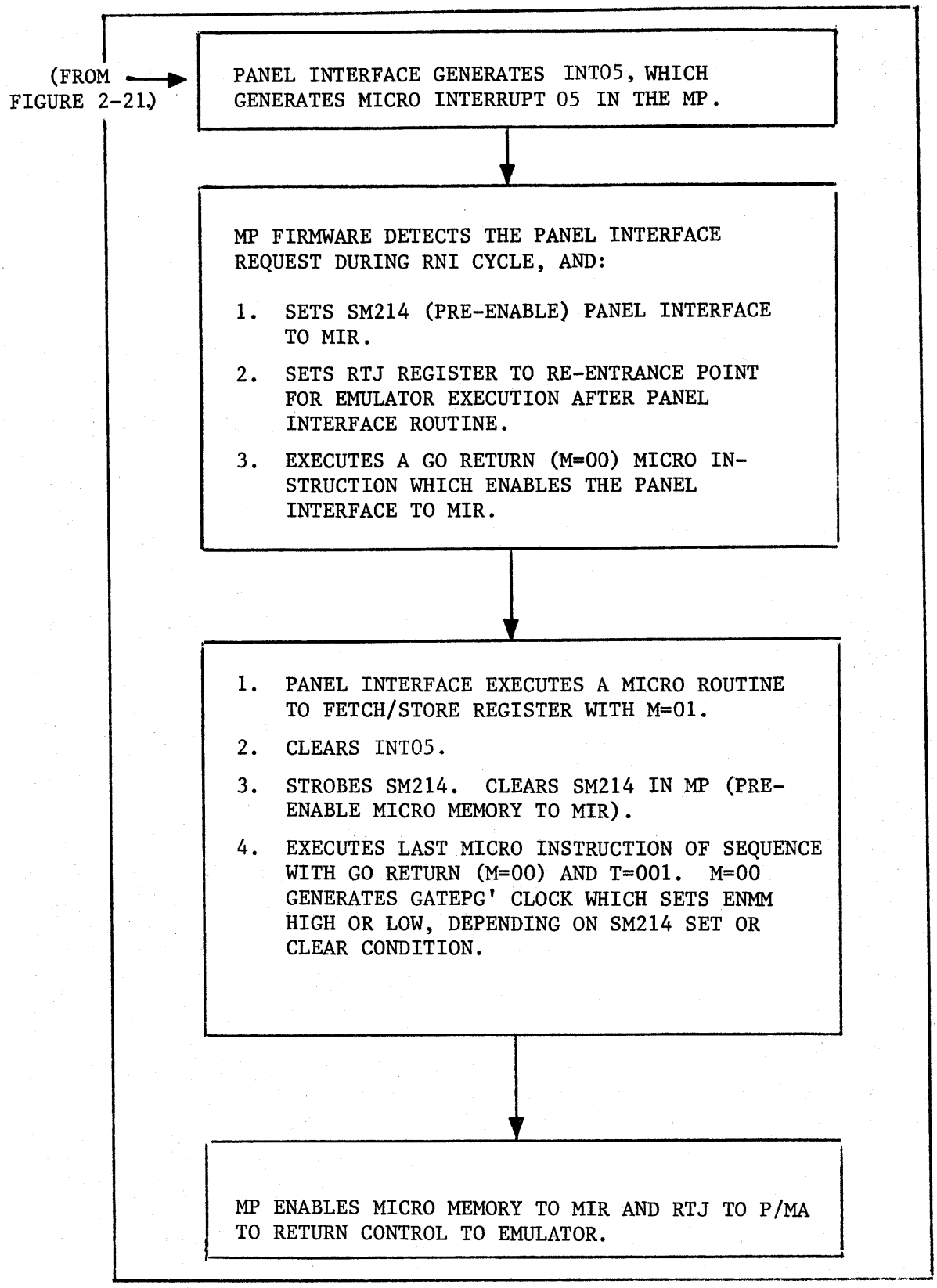


Figure 2-22. Macro Mode Flow Diagram

First, the panel I/F must determine if it needs the CPU. The MP is normally required except when MIR, FCR or BP/MA is the selected register. If the MP is needed, the panel I/F generates SETRCPU and, depending on whether the Function Control Register bit FC19 is set or clear, the micro mode or macro mode sequence is generated.

MICRO MODE

When the MP is in micro mode, the panel I/F does not require any assistance from the MP emulator (firmware), but it is required that the MP must initially be stopped before the panel I/F can take over the control of the MP. To take control of the MP, the panel I/F first generates the SET SM214 signal which causes MP to:

- Set the status mode register bit SM214.
- Clear the microinstruction register.
- Disable the micromemory and enable the Panel I/F microinstructions to the micromemory 3-state bus.
- Disable the macromemory address buffer register except for Read/Write Macromemory operation.
- Save the micromemory address.
- Save test bit TB.

The panel I/F now causes MP to begin running and be in micro mode. The NOP microinstruction, MIR = 0, is executed. The panel I/F generates a micro-instruction sequence with M field = 01, which is executed by the MP to change/fetch the selected register. Prior to execution of last panel I/F microinstruction, the panel I/F clears the status mode bit SM214. The panel I/F now generates last microinstruction with M field = 00 and T field = 000 to exit the CPU. The M field = 00 causes the re-enabling of micromemory and the disabling of the panel I/F to the microinstruction register. The MP now gets back the control and reads up the microinstruc-tion awaiting in MIR prior to panel I/F sequence. The panel I/F also stops the MP with a micro halt.

MACRO MODE

When the MP is in macromode, the firmware is required to prepare the MP before giving up the control to panel I/F. When the panel I/F determines that it needs the CPU, it generates micro interrupt INT05 which is panel request to CPU interrupt. Once the interrupt 05 is recognized by the firmware during RNI cycle, the firmware performs the following:

- Sets the Status Mode Interrupt bit SM214 to pre-enable panel I/F to the microinstruction register (MIR).
- Set RTJ register to re-entrance address for firmware execution after panel I/F sequence.
- Execute a GO Return (M field = 00) microinstruction which disables microinstructions from being loaded out of micromemory.

This disabling of micromemory enables the panel I/F to micromemory 3-state bus. The panel I/F now has the control of the CPU. The panel I/F generates a microinstruction sequence with M = 01 to be executed by the CPU. After the CPU completes the required operation, the panel I/F exits the CPU by clearing the INT05 and SM214 in MP. This pre-enables the micromemory to the micromemory bus. The last panel I/F microinstruction is executed with M field = 00 and T field = 001 to exit the CPU. The M = 00 causes enabling the micromemory and disabling the panel I/F to micromemory 3-state bus. The M field = 00 also selects the content of RTJ register as the address of the next microinstruction pair. Note that the RTJ contained the re-entrance address for the emulator. The T field = 001 selects the upper microinstruction at the address specified by RTJ register. The firmware now has its control back and continues at the point that it left off prior to panel I/F sequence.

MULTI PORT (MAIN) MEMORY

This section describes the external main memory system used with the BSC which consists of one or more Multi Port Memory (MPM) units. Each multi port memory unit is contained in the standard chassis, which is designed for 19-inch RETMA rack mounting. The MPM contains up to 9,437,184 bits of dynamic MOS Memory, configured as 262,144 36-bit words. In a multi-processor multi-memory unit configuration, each BSC may be connected in daisy chain fashion to any sixteen MPM chassis for a maximum of 4,194,304 words of memory per BSC. Each MPM chassis contains up to 8 independent ports and up to 4 independent banks of 32K or 65K words each. Each of the eight ports has access to all four memory banks and all four banks may be active simultaneously if they are accessed by different ports. Figure 2-23. shows the relationship of the ports and banks.

Three board types are used in the MPM:

- Port Interface (up to 8 used)
- Bank Control (1 or 2 used)
- MOS Memory Array (up to 16 used)

The physical location of the boards in the chassis is shown in Figure 1-4.

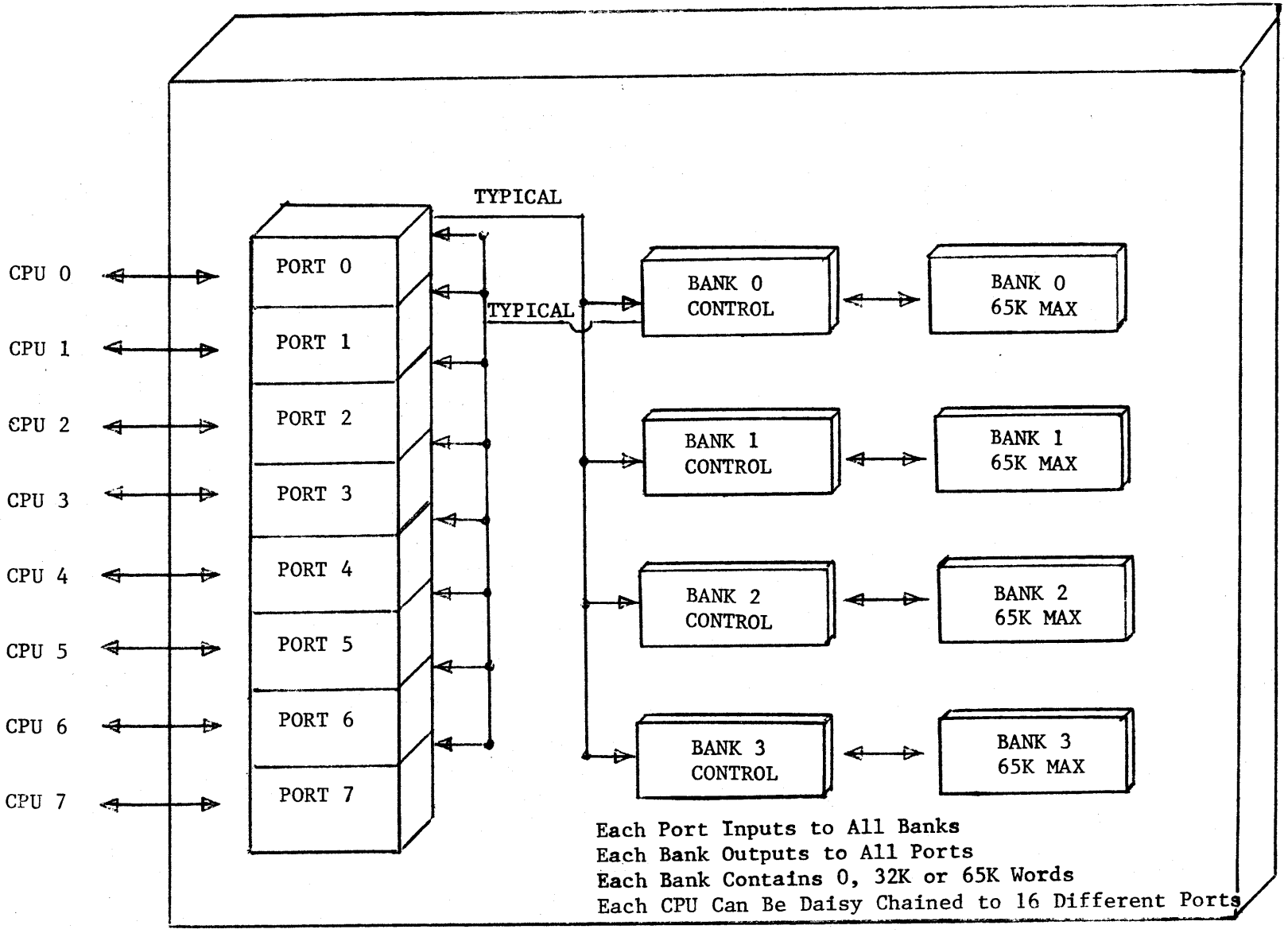


Figure 2-23. Eight Port Memory Unit Block Diagram

MODULARITY

An MPM chassis may contain 1, 2, 3 or 4 banks of memory. However, each bank control board contains logic for two banks, so there will be control for either 2 or 4 banks. If either the bank control or the array boards for a bank are not present, the bank will not respond to a request. Each bank may contain 0, 32K or 65K of memory.

Port I/F boards are configured in pairs and each pair provides two ports. Thus, a chassis may contain 2, 4, 6 or 8 ports.

Any bank may be made to appear absent to any port by means of a switch on the Bank Control Board.

MEMORY ADDRESSING

Twenty-two bits are required to address an MPM system. Refer to Figure 2-24 for the following discussion.

The most significant four bits (18-21) are compared with the setting of DIP switches on the port I/F board to determine the unit selection

The next two bits (16 and 17) select 1 of 4 65K banks within the unit. They are translated by the port I/F to determine which bank is requested.

The least significant 16 bits (00-15) select 1 of 65K words within the selected bank. They are translated by the Bank Control and the MOS Array logic.

One bit of odd parity accompanies the address and is checked in the Port I/F.

PORT INTERFACE

The Port I/F provides the interface between the CPU and the banks of memory. Although there is one Port I/F board per port, the boards must

be installed in pairs, with each board handling half of the data bits for both ports. All of the address and control lines for a given port are on one board.

ADDRESS AND CONTROL LINES

Figure 2-24. lists some of the port I/F signals. The complete set of signals may be broken into groups as follows:

Address Bits (XMA00-XMA21) - These 22 lines go to the MPM units to select the desired memory word.

Address Parity (XMAPAR) - This is one line to the MPM to provide odd address parity.

Request (XCPURQ) - This is one line to the MPM. When active, it indicates that a read or write is desired at the specified address.

Lock Bank (XLOCK) - This line, when accompanied by a CPU request, indicates that the BSC desires to lock the addressed bank. The CPU selects this condition with a status mode bit. A bank will be unlocked when the bank is accessed with the status mode bit clear, or after 5.4 μsecs. (±20%).

CPU CYCLE (XCPUCY) - This line is used in conjunction with the XLOCK line and indicates the bank accessed during the present CPU request is to be locked.

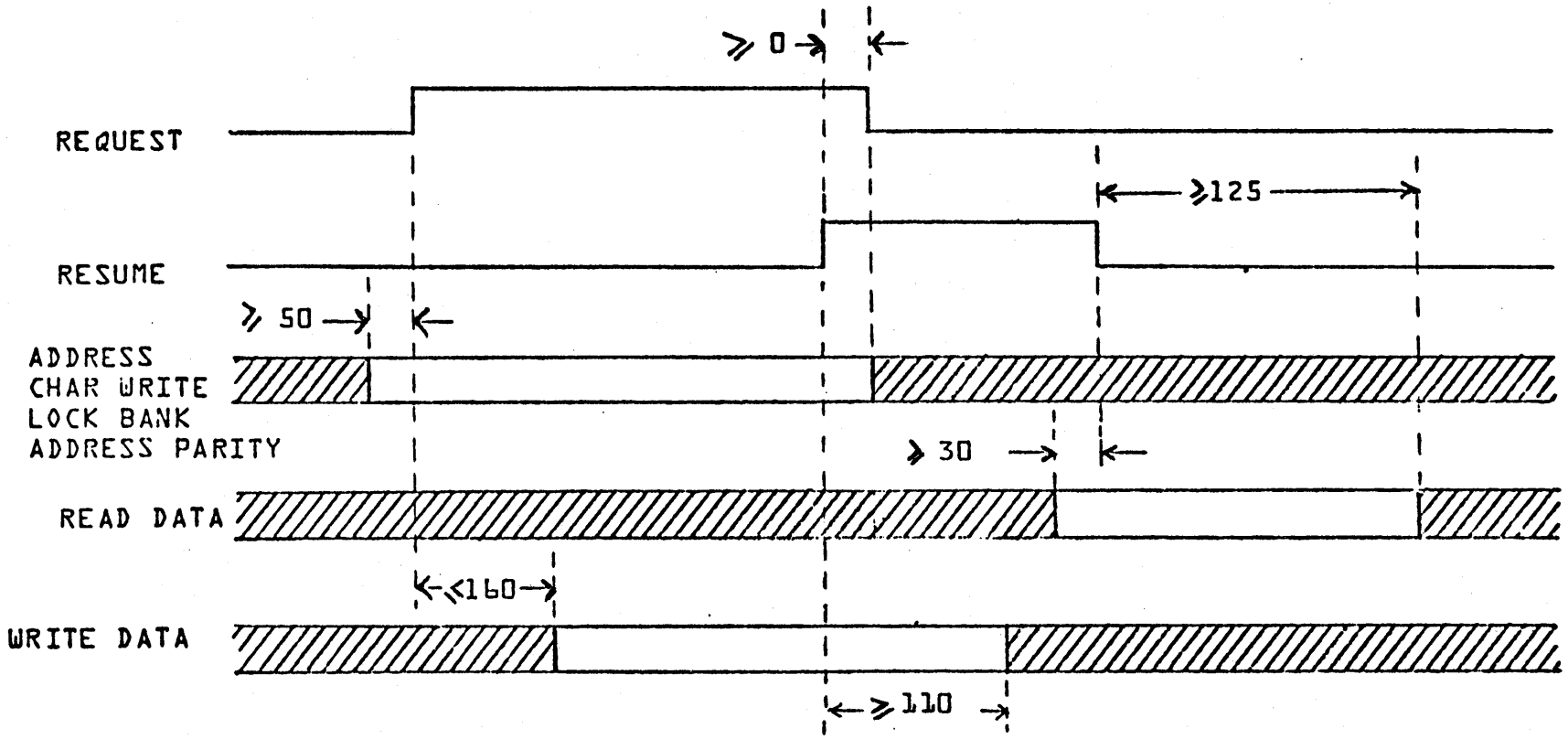
Resume (XRESUME) - This line from the MPM is the reply to a request. The leading edge indicates that control, address and write data may be removed. The trailing edge indicates that read data is available from the MPM (see Figure 2-25).

Character Write Lines (XCHAR0-XCHAR2) - These 3 lines to the MPM cause a write to the coded character (9 bits). If no character write lines are active, a read is specified.

Data Lines (XMD00-XMD31) and Parity (XPFD00-XPFD03) - These 36 bi-directional lines transmit data and parity to and from the MPM.

SIGNAL	
XMA00	SELECTS 1 OF 65K LOCATIONS WITHIN THE BANK
XMA01	
XMA02	
XMA03	
XMA04	
XMA05	
XMA06	
XMA07	
XMA08	
XMA09	
XMA10	
XMA11	
XMA12	
XMA13	
XMA14	
XMA15	
XMA16	SELECTS 1 OF 4 65K BANKS WITHIN THE MEMORY UNIT
XMA17	
XMA18	SELECTS 1 OF 16 256K MEMORY UNITS
XMA19	
XMA20	
XMA21	
XMAPAR	
XCPURQ	
XLOCK	
XUNLK	
XCHAR0	
XCHAR1	
XCHAR2	
XCPUCY	
XRESUME	

Figure 2-24. Address and Control Lines to MPM



All Times Are In Nanoseconds

Figure 2-25. Memory Port Timing

ADDRESS PARITY

One bit of odd parity accompanies the address to the MPM. Parity checking is done in the Port I/F. If a parity error occurs, it does not prevent a memory cycle from taking place, but it does block writes to all characters. It also blocks the Resume signal to the port (causing time out) and the lock line is ignored.

BANK CONTROL

Each memory chassis contains four independent sets of bank control logic, one for each of the 65K memory banks. Two banks are controlled by each Bank Control board. Since each set of bank control logic is independent, all four banks can run simultaneously if each is accessed by a different port.

The bank logic resolves priority between requests from the eight ports and refresh. Once resolved, the bank logic gates address, control and data to/from the selected port to the memory bank.

The bank logic puts the address into a form that is acceptable to the MOS Array and generates the necessary signals and timing.

REFRESH

The dynamic 4K MOS Memory used on the array board requires a memory cycle on each of the 64 row addresses every 2 milliseconds or less. This is accomplished with a timer that requests the bank in much the same way as a port request. The timer requests the bank at evenly spaced time intervals to meet the chip refresh requirements. Many chips may be refreshed at the same time to reduce overhead.

The Bank Control logic provides two backplane selectable refresh rates. The slow rate selects the entire 65K bank on each refresh cycle. This results in a refresh request rate of 2Msec/64 = 31.25 μsecond. The fast rate refreshes half of each array board on each cycle, which

results in a refresh rate of 15.625 μsecond. The MPM is designed such that only one bank is refreshing at any instant, which reduces the instantaneous load on the power supplies.

PRIORITY

Requests from the eight ports and refresh are resolved in the Bank Control priority logic. If a bank is not busy, the first request presented is accepted. When a bank is about to go not busy, the eight ports and refresh are scanned in order, starting one higher than the current active port. If a request is active, it is accepted. Thus, a port will never wait longer than eight cycles to be accepted unless the port is locked out.

LOCK/UNLOCK

The Bank Control logic provides a method of reading and changing cells in memory with the assurance that no other port has read or changed the cell(s) between reading and writing. Thus, words in memory may be used as flag words to communicate between processors without the danger of logical discontinuities. This is done by locking a bank to a port, which locks out all other ports (but not refresh). When a bank accepts a request from a port with the XLOCK and XCPUCY lines active, the bank becomes locked to that port. Requests from other ports are ignored. The bank stays locked to the port until a request is accepted from the CPU with the XLOCK line inactive. Then the bank is unlocked, pending requests from other ports are accepted in the normal order. A bank automatically unlocks if no XLOCK signal has been received for 5.4 μseconds (±20%). This timeout is significantly greater than the no-response timeout done by the memory interface logic in the CPU.

MASTER CLEAR

Because of the nature of multi port memories, and dynamic MOS memories in particular, a master clear signal from the CPU's cannot be utilized. Because of this, the bank control logic is designed to be self-clearing; i.e., if no activity (including refresh requests) takes place for 40 milliseconds, a time-out clears the logic.

MOS MEMORY ARRAY

The MOS Array Board contains 144 4096x1 N-Channel MOS Memory chips. The Array Board is arranged in a 32K by 18-bit pattern and nine bits can be written at a time to allow character writes. Two or four Array Boards are needed to make up a bank.

MEMORY TIMING

A memory cycle in the MPM is divided into two parts:

Addressing Time - This is the time required to resolve the requests from the ports and enable the correct address to the MOS chips. It also includes circuit delays in the request and address enables, etc. Addressing Time is approximately 200 nanoseconds.

Chip Cycle Time - This is the time required to cycle the MOS chip. Chip Cycle Time is approximately 400 nanoseconds. If a request is active when the bank is busy, all or part of the Addressing Time may take place concurrently with the Chip Cycle Time. Therefore, the Memory Cycle Time can equal the Chip Cycle Time if a request is active early enough.

Addressing Time delay is completely asynchronous to Chip Cycle timing, so a portion of the Addressing Time can be hidden behind the Chip Cycle Time even if a request becomes active late in a memory cycle.

Figure 2-25. shows the signal timing at the Memory ports. The MPM is designed such that the address, write data and control lines may be changed relatively early in the cycle to set up for the next request. The leading edge of the Resume signal is the time that this change can take place. The trailing edge of the Resume signal is the memory data strobe; i.e., read data is stable at this time. The address is stable 50 nanoseconds before the memory request signal.

The request signal drops immediately when the leading edge of Resume is received. It may be raised after the new address has been stable for 50 nanoseconds.

SWITCHES AND LIGHTS

Four pins are provided on the backplane at each port I/F for unit select. Those pins may be individually wired to SPST switches to selectively ground each pin. Figure 2-26 shows the bit/pin relationship. Ground level on the pin corresponds to a logical "1". All port I/F slots may be wired to one set of four switches, or each Port I/F can have its own set of switches.

Two sets of eight port enable DIP switches are included on the Bank Control Board. The switch block for the even bank is at location L12, and the switch block for the odd bank is at location A12.

When a switch is in the "ON" position, the bank/port operate normally. When the switch is in the "OFF" position, the request line from the specified port is broken, so it is locked out from that bank. Figure 2-27 shows the switch/port relationship.

A Form C switch may be wired to the backplane at the Bank Control slots to allow master clearing the Bank Control. This switch must be used with discretion because master clearing the Bank Control causes refresh to be interrupted momentarily, which causes the contents of memory to be lost.

Each port I/F board has a toggle switch on it. When this switch is up, the port which has control logic on the board is isolated; i.e., all transmitters and the request receiver are disabled. A LED directly above the switch lights when the switch is in the up position to indicate the port is isolated. This switch may be used to isolate a chassis while maintenance is being performed without disturbing other chassis.

A light emitting diode is provided on each port I/F to indicate that an Address Parity Error has occurred. The LED will stay lit until a Master Clear is received from the BSC.

Two light emitting diodes are located on each Bank Control board. These LED's indicate that an automatic unlock has occurred. The light will stay lit until a bank control clear or a power on master clear occurs.

ADDRESS BIT	PIN
18	239
19	294
20	99
21	98

Figure 2-26. Port Select Pins

PORT	SWITCH
1	1
2	2
3	3
4	4
5	5
6	6
7	7
0	8

Figure 2-27. Port Enable Switch Block

OPERATING INSTRUCTIONS 3

INTRODUCTION

This section addresses the operating procedure for BSC. Since each user will have a different setup for the application of the equipment, the procedure will not be detailed. It is recommended that each user evaluate and develop his own operating procedure. A detailed description of the panel interface and maintenance panel are included in this section.

START UP

The following start-up sequence is common for all installations:

- Pre-Power-On sequence. Turn on all peripherals and auxiliary power units (as required on user equipment).
- POWER switch. Turn the POWER switch to the on position.
- Check adjustment and indicators on user equipment. Verify that all indicators are in proper position for start-up. Verify that all adjustments have been made for start-up.
- Turn-on procedure for maintenance panel or console. The maintenance panel or console is turned on when the POWER switch for the BSC is turned on.
- Operator controls for interfacing to the MP via the maintenance panel are described in the following paragraph.
- Performance of Load Routine. A load may be accomplished via a disk, a paper tape, a magnetic tape or a bootstrap loader, depending on the application.

POWER-OFF SEQUENCE

Position the POWER switch to the off position and de-energize all peripherals.

SYSTEM FAILURE

After a system failure, follow the start-up procedure for restart.

PANEL INTERFACE AND MAINTENANCE PANEL

The panel interface card provides manual operator control of the BSC. A maintenance panel and/or programmer's console may be connected to this card. The following paragraphs will discuss the controls necessary at either the maintenance panel or programmer's console. This discussion will be followed by a description of the functional features of the maintenance panel.

PANEL INTERFACE

Basic to the panel interface operation is the Function Control Register (FCR). This 32-bit register is accessible to the BSC. The function control register bit assignments are listed in Tables 3-1. and 3-2. The unused FCR display codes in Table 3-2. are undefined.

AUTO DISPLAY

The auto-display feature is selected by setting bit 15_{16} in the FCR. It is removed by clearing this bit. When auto-display is selected, the panel interface continuously displays the register determined by the last control code and the display 0 or display 1 function. No line feeds are issued in this mode, so the display stays at the same line on the screen. Note that this precludes using auto-display with a teletypewriter.

TABLE 3-1. FUNCTION CONTROL REGISTER (FCR)

BIT	DIGIT	BIT DEFINITION	
31 1F	(LSB)	CPU State Reg 31	Status Only
30 1E	7	CPU State Reg 30	
29 1D		CPU State Reg 29	
28 1C		Monitor/Program	
27 1B		CPU MEM I/F Error	
26 1A	6	DMA MEM I/F Error	
25 19		Micro Running	
24 18		Macro Running	
23 17			Modes
22 16	5		
21 15		Enable Auto Display	
20 14		Enable Console Echo	
19 13		Enable Micromemory Write	
18 12	4		
17 11			
16 10		Suppress Console Transmit	
15 OF		Refer to BP Description in	
14 OE	3	Control Functions Section	
13 OD		BP Int. (BP Stop if Clr)	
12 OC		Micro BP, Step, Go, Stop (Macro if Clr)	
11 OB		Step	
10 OA	2	Selective Stop	
09 09		Selective Skip	
08 08			
07 07			Display Selection
06 06	1		
05 05		DISPLAY 1	
04 04			
03 03			
02 02	0	DISPLAY 0	
01 01			
00 00	(MSB)		

TABLE 3-2. DISPLAY CODE DEFINITIONS

CODE	SELECT CODE	DISPLAY 1 "K FUNCTION"	SELECT CODE	DISPLAY 0 "L FUNCTION"
0 0 0 0 0	J10:	FCR	J00:	F2
1 0 0 0 1	J11:	P	J01:	N
2 0 0 1 0	J12:	I	J02:	K
3 0 0 1 1		-	J03:	X
4 0 1 0 0	J14:	A	J04:	Q
5 0 1 0 1	J15:	MIR	J05:	F
6 0 1 1 0	J16:	BP/P-MA	J06:	F1 (F3, providing setup of the addressing method is fulfilled.)
7 0 1 1 1	J17:	BP/P-MA (Display only)		-
8 1 0 0 1	J18:	SM1		-
9 1 0 0 1	J19:	M1	J09:	RTJ
A 1 0 1 0	J1A:	SM2		-
B 1 0 1 1	J1B:	M2		-
C 1 1 0 0		-	J0C:	MM
D 1 1 0 1		-		-
E 1 1 1 0		-		-
F 1 1 1 1		-		- (dash specifies an undetermined result)

If a terminator (colon, G or @) is pushed with characters preceding it, a Go signal is generated. This feature is convenient for stepping through either macro or microprograms.

FUNCTIONAL OPERATION

The interface accepts nine different control characters: H, I, J, K, L, @, G, : and ?. G and @ are functionally identical to a colon; however, in this manual all operator functions use the colon (:). H through L identify the type of data or operation entered or returned. The character colon (:) terminates all entries except Master Clear.

A question mark will generate a Master Clear to the computer, memory and peripherals. There is no response to this entry.

A normal entry consists of one control character - H through L; zero, two, four or eight hexadecimal digits 0 through F; and the colon (:), in that order. If a transmission or operator error occurs on the entry, the control character is preceded by an asterisk (*) and the Function Control Register is displayed. All entries except ? cause a response unless bit 10_{16} of the FCR is set (suppress console transmit).

CONTROL FUNCTIONS

The control functions are used as follows:

H CONTROL FUNCTION

This function is used to clear a specific bit in the FCR.

Example: H14:

This would clear bit 14_{16} in the FCR and the response would be a display of the updated FCR.

I CONTROL FUNCTION

This function is identical to H except it sets a bit in the FCR. H and I are also used for Stop/Run control.

J CONTROL FUNCTION

The J control function is used to replace the contents of the function control register in a digit mode. While it may be used to change the value of any FCR digit, it is generally used to change digits 0 and 1. The value of display 0 and display 1 specifies which MP register is displayed on display requests or entered on enter requests. J functions always consist of J followed by two hexadecimal digits and the :. The first hexadecimal digit specifies the FCR digit 0 through 5 and the second hexadecimal digit specifies the value the digit is to assume, 0 through F.

Example: J14:

This would set FCR digit 1 to 4 (select the A register) and the response would be a display of the updated FCR.

The J code is also used to alternately display the upper and lower 16 bits of a 32-bit register on the maintenance panel.

Example: J:

This would cause display of the other 16 bits and would complement the U/L indicator on the maintenance panel.

K CONTROL FUNCTION

The K control function is used to display or enter data into the parameter specified by Display 1. The K functions use two formats. The first format is a request to display the parameter specified by Display 1.

Example: K:

The second format is an enter data request. The data is entered into the register specified by Display 1. It consists of K followed by four or eight hexadecimal digits, terminated by the :. The hexadecimal digits are the data to be entered. Some examples are:

- To display the P register, perform the following:

  J11: Set Display 1 to P register (1).
  K: Display register selected in Display 1.

- To enter $14FE_{16}$ into the breakpoint register, perform the following:

  J16: Set Display 1 to BP register (6).
  K16FE: Enter into register selected in Display 1.

L CONTROL FUNCTION

Operationally, the L function is the same as the K function, except it is associated with Display 0.

When micromemory is displayed or entered, the K register is the least significant eight bits of the address, and the N register provides the remaining bits. The K register is incremented by 1 after the display.

STOP/GO CONTROL

The following entry will cause a go:

Entry: I:

This is a micro go when bit 12 of FCR is set. It is both a micro and macro go when bit 12 of FCR is clear.

The following entry will cause a stop:

Entry: H:

This is a micro stop when bit 12 of FCR is set. It is a macro stop when bit 12 of FCR is Clear. The response to a start or stop entry is a display of the FCR.

MASTER CLEAR

A master clear can be generated in several ways:

- A question mark from remote console.
- The MC button on the maintenance panel.
- A signal from a peripheral controller.
- A power on Master Clear.

BREAKPOINT (BP)

There are two types of breakpoint: micro and macro. When bit 12 of the FCR is set, micro BP is selected. Use of the micro and macro BP is described below:

- Micro BP - FCR bits 14 and 15 are used to select two types of micro breakpoint.

Bit 14	Bit 15	
0	0	BP not selected
0	1	Upper/lower micro instruction BP
1	0	Micro-word BP
1	1	Micro-word BP

The upper/lower micro instruction breakpoint requires that the micro memory address P/MA and upper/lower micro instruction selections are equal to the lower 13 bits of the breakpoint register to cause a micro stop. The micro-word breakpoint only requires that the 12 bits of micro-memory address register P/MA are equal to the lower 12 bits of the breakpoint register.

- Macro BP - FCR bits 14 and 15 are used to select three types of macro breakpoint.

Bit 14	Bit 15	
0	0	BP not selected
0	1	BP on Lookahead Register 3
1	0	BP on Lookahead Register 2
1	1	BP on all Lookahead Registers (LA1, LA2, LA3)

A macro breakpoint stop occurs when the breakpoint register is equal to the least significant 16 bits of the main memory address and the select conditions are met. If FCR bit 13 is set, an interrupt, rather than a stop, occurs when the breakpoint conditions are met.

SPECIAL CONSIDERATIONS

- The Enable Auto Display and Enable Console Echo bits of the FCR (15_{16} and 14_{16}) are mutually exclusive; that is, the user may select either one or the other, but not both at the same time.
- Display 1 selection of BP, P/MA permits changing contents of BP only. P/MA cannot be modified by the FCR.
- Display 0 for N or K will cause both N and K to be displayed. However, for code 0001, only N can be modified and for code 0010, only K can be modified.

MAINTENANCE PANEL

Figure 3-1. illustrates the available functional features of the maintenance panel. These features include:

- Data Display - This 16-bit panel displays the data. Data display is determined by the functions previously described. Display indicators are LEDs.
- CONTROL CODE - This three-bit LED display indicates the last control character entered. It is interpreted as an octal number. The equivalence table follows:

0 - H	4 - L
1 - I	5 - UNDEFINED
2 - J	6 - UNDEFINED
3 - K	7 - ERROR

- MASTER CLEAR - This momentary contact switch causes a master clear to the BSC, memory and peripheral controllers.

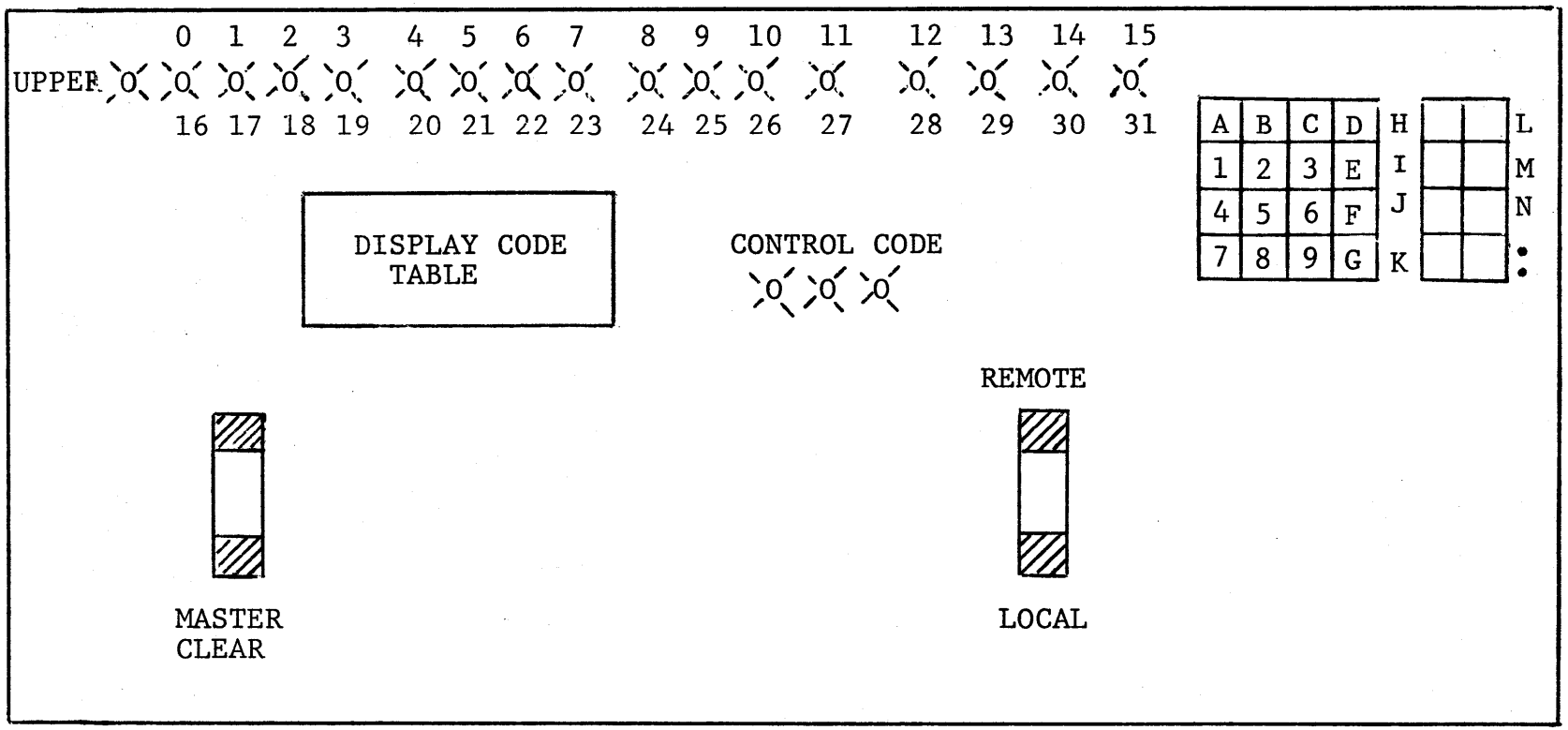


Figure 3-1. Functional Features of Minimum Maintenance Panel

- REMOTE - This two-position switch enables the panel or the remote programmer's console. In the local position, the panel is enabled; in the remote position, the programmer's console is enabled. If the panel is physically removed, remote is selected.

- Function Definition Table - This table consists of operator assistance information such as FCR definition, control character definition, etc.

- Data Entry - These 16 momentary pushbuttons are used to enter hexadecimal data.

- Control Characters - These eight momentary pushbuttons are used to enter control characters.

MICROINSTRUCTION FORMAT

Each micromemory address specifies the location of two microinstructions (upper instruction and lower instruction). Each 32-bit microinstruction is divided into five main sections and bits in the instruction are numbered from left to right as bits 0 to 31. Figure 4-1. illustrates the basic BSC instruction format.

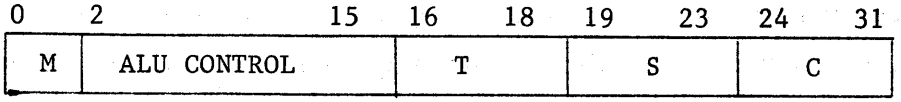


Figure 4-1. Basic BSC Instruction Format

The five basic fields of a BSC instruction are defined as follows:

Bit Positions	Field Definition
0, 1	Mode (M) field specifies the addressing used in obtaining the next microinstruction pair from micromemory, and specifies the format used in interpreting bits 19 through 31 (the S and C fields) of the microinstruction.
2 through 15	ALU Control field specifies the arithmetic and logical unit (ALU) operation code, sources of operands and destination of the operation result.
16 through 18	Test (T) field specifies the method of selecting which microinstruction (upper or lower) of the next microinstruction pair to execute.

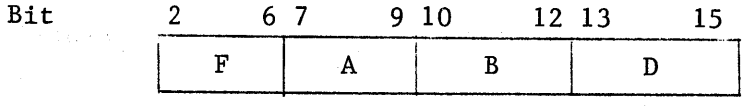
Bit Positions	Field Definition
19 through 23	Special (S) field specifies subformat selection (bit 19) and special operation codes (bits 20 through 23).
24 through 31	Constant (C) or suboperation field (bit 24 specifies constants or other codes.

The BSC instruction field definitions, including the various subfield definitions, are shown in Figure 4-2.

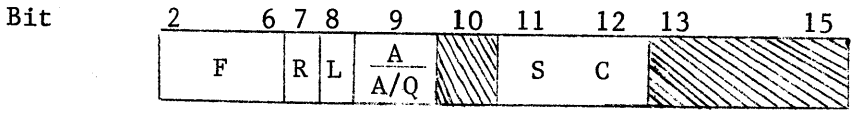
The M field specifies one of three addressing modes used to obtain the next microinstruction pair from micromemory, and specifies the format for interpreting bits 19 to 31 of the microinstruction, as follows:

M	Addressing Mode	Format For Bits 19 To 31
00	Return	Format 1
01	Sequential	Format 1
10	Jump	Format 2
11	Sequential	Format 3

The ALU control fields specify the sources of the two operands on which arithmetic, logical, shift or scale operation is to be performed, and specify the destination of the operation result. For arithmetic and logical operations, the ALU control fields consist of the ALU function (F), A source (A), B source (B) and destination (D) fields as follows:



For shift and scale operations, the A and B fields are interpreted as follows:



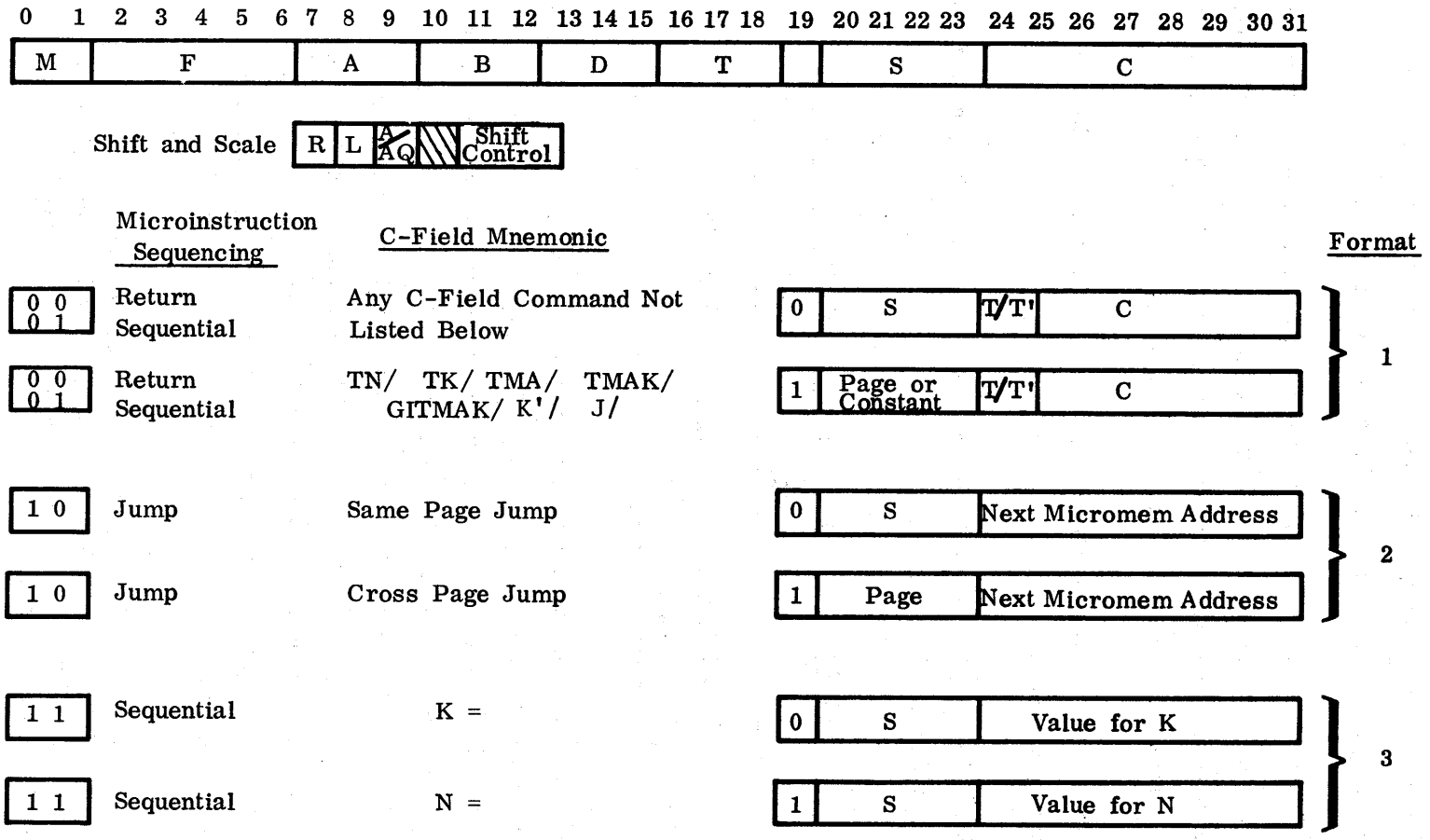


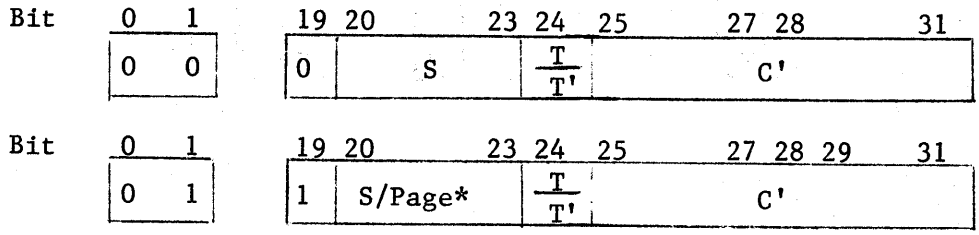
Fig. 4-2. Microinstruction Formats

The F field specifies shift or scale operation. Bits 7 and 8 specify right or left shifting. Bit 9 specifies whether the A register alone or the A and Q registers together are to be shifted or scaled. Bit 10 is not used, and bits 11 and 12 specify the shift control code. The D field contains a no-operation code for shift and scale operations.

The T field is the conditional branch of the microinstruction, and specifies which microinstruction, upper or lower, of the next microinstruction pair to execute. The test may be based on the result of the ALU operation of the current microinstruction or on some other condition.

The codings in the S and C fields depend upon the contents of the M field. The S and C fields are coded in three formats.

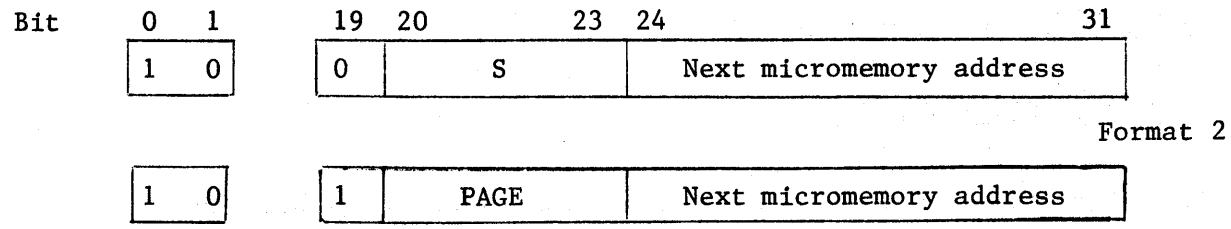
Format 1. Format 1 is specified when the M field contains 00 (return mode) or 01 (sequential mode) as follows:



The S field specifies such operations as main memory read, alternate codings to be used in the A, B and D fields, etc. The T/T' bit specifies that the code in the T field is to be interpreted as the normal T code (T/T'=0) or as the alternate T' code (T/T'=1). The subformat select bit, bit 19, determines whether bits 25 through 31 are to be interpreted as C' codes or as C" codes. The C' code can contain a constant for driving the bit generator, additional information to control the main memory read or write, or another operation. The C" codes are associated with transforms.

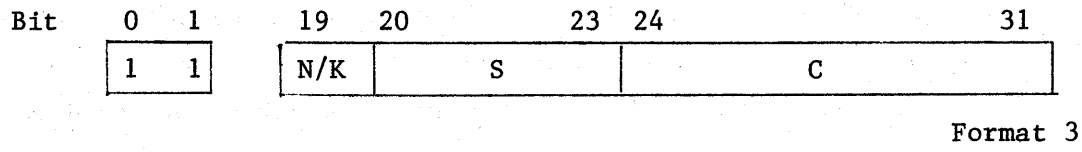
*If SM207 is set, normal S field commands are disabled and the S field then specifies a micromemory page for the TAM/j, TMAK/j, and GITMAK/j commands. In addition, normal S field decoding is inhibited for TMA, TMAK, GITMAK, and TN/K transforms.

Format 2. Format 2 is specified when the M field contains 10 (jump mode) as follows:



If a jump is specified to a microinstruction pair within the same micromemory page, the subformat select bit, bit 19, is 0 and bits 20 to 23 contain a special operation code as in format 1. Bits 24 through 31 contain the micromemory address of the next microinstruction pair. The subformat select bit is 1 when a jump is specified to a different micromemory page; bits 20 through 31 contain the complete micromemory address of the next microinstruction pair.

Format 3. Format 3 is specified when the M field contains 11 (sequential mode), as follows:



This format allows one special operation to be performed, as specified by the S code, and also causes the eight bits of the C field to be transferred to the N register (bit 19 = 1) or to the K register (bit 19 = 0).

DESCRIPTION OF MICROINSTRUCTIONS

The detailed coding for each field of the microinstruction is described in the following paragraphs.

ALU CONTROL FIELDS

The ALU control fields are described in the following paragraphs.

F FIELD

The F (bits 2 through 6) field specifies the logical or arithmetic operation performed by the ALU or the shift or scale operations performed with the A and Q registers.

- Logical Operations - The logical operations are described in Table 4-1.

- Arithmetic Operations - The arithmetic operations are listed in Table 4-2. Two options are provided and coded in the arithmetic function code. The first option is a carry input to the adder (indicated by a plus sign in the microinstruction mnemonic). The T-field code checks for a carry-out of the ALU to determine whether a carry into the ALU should be used on the next arithmetic operation.

The second option allows capture of the overflow condition in the SM register (indicated by a T in the instruction mnemonic). When this is indicated, the overflow test is performed by comparing the sign of the two inputs with the results of the ALU and setting a status/mode bit when the result is inconsistent. The SM overflow bit is set to 1 when the overflow occurs; it must be set to 0 by a microinstruction that sets the SM bit to 0.

- Shift Operations - The shift operations in the F field (11110) specify a shift of the A register or the AQ register of the main BSC organization only. The ALU is not used to perform the shift, but will perform some operation based on its decoding of the F field (which should be considered as unknown). The destination will receive this meaningless output unless an NOP is chosen for the destination of the D field.

  The type of shift is determined by the coding in bits 7 to 12 of the microinstruction, and the amount of the shift is determined by the number contained in the N register. The operation examines the N register and, if it is zero, the shift operation is terminated and the next microinstruction is executed. The T field codes that can be used with a shift are U, L, KZU, INTU, OVFL, K7L and BTU. Other T codes cannot be used.

  If the N register is not zero, a shift of one bit position is taken as specified, N is decremented by one and the test for zero is repeated as above.

  The shift conditions are as follows:

  - Shift A - Shift the A register only.
  - Shift AQ - Shift the combined A and Q register. The Q register is considered the least significant bits of the combined AQ register.
  - Shift left or right.
  - Enter 0 - Enter a 0 in the vacated bit position at the end of the register.

  NOTE - Upon completion of a shift operation the N register is equal to FE_{16}.

- Enter 1 - Enter a 1 in the vacated bit position at the end of the register.
- Extend sign - Extend the sign (for a right shift only).
- End-Around Carry - Enter the bit coming off the end of the register into the vacated on position at the other end.

All shifts are performed with an F code of 11110. The type of shift is determined by bits 7 through 12 of the microinstruction. The shifts are defined in Table 4-3.

- Scale Operations - The scale operations are similar to the shift operations, but stopping the shift is conditioned on bits 0 and 1 of A not being equal. (The scale point is normally between bits 0 and 1 of the A register. A hardware design option allows the scale point to be specified between different bits in the A register when necessary for efficient floating-point emulation.) The maximum number of bits to scale is contained in the N register and, on completion of the scale, N is decremented by the number of shifts necessary to scale the number. Table 4-4. illustrates scale operations.

The scale operation is performed as follows:

- Examine N; if it is zero, exit the microinstruction.
- Examine bits 0 and 1 of the A register; if they differ, exit the microinstruction.
- Shift the A or AQ register left by one bit position as specified in the instruction.
- Decrement the N register by one count and go to Step 2*.

*If the number being scaled is all 0's or all 1's (i.e., the number cannot be scaled), then when N = $ FE (after passing through N = 0), the scale operation is terminated. To avoid exiting the microinstruction before the scale operation is actually completed, (N) should be at least equal to the number of bits in the word to be scaled.

TABLE 4-1. LOGICAL OPERATIONS

F CODE	MNEMONIC	OPERATION	BIT RESULT FOR: A=0011 B=0101
01100	ZERO	Zeros output regardless of A and B.	0000
01110	A•B	Logical product of A AND B.	0001
01101	A•-B	Logical product of A AND NOT B.	0010
01111	A	Transfer A regardless of B.	0011
01000	-A•B	Logical product of NOT A AND B.	0100
01010	B	Transfer B regardless of A.	0101
01001	EOR	EXCLUSIVE OR of A and B.	0110
01011	A+B	INCLUSIVE OR of A and B.	0111
00100	-A•-B	Logical product of NOT A AND NOT B.	1000
00110	-EOR	Negation of EXCLUSIVE OR of A and B.	1001
00101	-B	Transfer one's complement of B regardless of A.	1010
00111	A+-B	INCLUSIVE OR of A and NOT B.	1011
00000	-A	Transfer one's complement of A regardless of B.	1100
00010	-A+B	INCLUSIVE OR of NOT A and B.	1101
00001	-A+-B	INCLUSIVE OR of NOT A and NOT B.	1110
00011	ONE	One's output regardless of A and B.	1111

TABLE 4-2. ARITHMETIC OPERATIONS

F CODE	MNEMONIC	OPERATION
10000	A-	Subtract 1 from A input.*
10001	A-T	Subtract 1 from A input with overflow test.*
10100	SUB	Subtract B input from A input.
10101	SUBT	SUB with an overflow test.
10110	SUB-	Perform A-B-1 when in two's complement mode. Perform A-B with forced carry-in when in one's complement mode. A-B is performed on the upper adder when split.
10111	SUB-T	Perform SUB- with an overflow test.
11000	ADD	ADD A and B inputs.
11001	ADDT	ADD with an overflow test.
11010	ADD+	Perform A+B with force carry-in. A+B is performed on the upper adder when split. (A+B+1 if either 2's complement mode or 1's complement mode with both A and B positive; otherwise, A+B).
11011	ADD+T	Perform ADD+ with an overflow test.
11100	A+	Add 1 to A input.*
11101	A+T	Add 1 to A input with overflow test.*

*These instructions are only valid in two's complement mode. With the adder split in one's complement mode, they will operate only on the upper 16 bits - which are still in two's complement mode.

TABLE 4-3. SHIFT OPERATIONS (F CODE = 11110)

BIT CODE		MNEMONIC	OPERATION
7 8 9	11 12		
1 0 0	0 0	AR0E	A is right-shifted (N) bits, with 0 entered as most significant bit.
1 0 0	0 1	ARSE	A is right-shifted (N) bits, with sign extension.
1 0 0	1 0	AREA	A is right-shifted (N) bits, with end-around carry.
0 1 0	0 0	AL0E	A is left-shifted (N) bits, with 0 entered as least significant bit.
0 1 0	0 1	AL1E	A is left-shifted (N) bits, with 1 entered as least significant bit.
0 1 0	1 0	ALEA	A is left-shifted (N) bits, with end-around carry.
1 0 1	0 0	AQR0E	AQ is right-shifted (N) bits, with 0 entered as most significant bit in A.
1 0 1	0 1	AQRSE	AQ is right-shifted (N) bits, with sign extension.
1 0 1	1 0	AQREA	AQ is right-shifted (N) bits, with end-around carry.
0 1 1	0 0	AQL0E	AQ is left-shifted (N) bits, with 0 entered as least significant bit in Q.
0 1 1	1 0	AQLEA	AQ is left-shifted (N) bits, with end-around carry.

TABLE 4-4. SCALE OPERATIONS (F CODE = 11111)

BIT CODE		MNEMONIC	OPERATION
7 8 9	11 12		
0 1 0	0 0	SL0E	A is scaled left, with 0 entered as least significant bit.
0 1 0	0 1	SL1E	A is scaled left, with 1 entered as least significant bit.
0 1 0	1 0	SLEA	A is scaled left, with end-around carry.
0 1 1	0 0	SDL0E	AQ is scaled left, with 0 entered as least significant bit in Q.
0 1 1	1 0	SDLEA	AQ is scaled left, with end-around carry.

The scale operation is coded the same in bits 7 through 12 of the micro-instruction and allows the same left shift options as the shift command. The same comments on exiting the shift and the usable T field codes apply to the scale operation.

All scale operations are performed with an F code of 11111. The type of shift for the scale is determined by bits 7 through 12 of the instruction. The scales are given in Table 4-4.

A FIELD

The A field (bits 7 through 9) specifies the input to S1 and thus to the A side of the ALU. The eight A codes are expanded by eight A' codes by placing 1010 or 0111 in the S field.

The eight A inputs to S1 and to the A side of the ALU are indicated when the S field is not 0111 or 1010; the eight A codes specify inputs from the files, the registers or main CPU 3-state bus according to Table 4-5.

The eight A' inputs to S1 and to the A side of the ALU are indicated if the S field is 0111 or 1010. The A' codes specify input from the SM registers or mask registers via the CPU 3-state bus. The A' codes are given in Table 4-6.

Because A' inputs use the main CPU 3-state bus for their inputs, the bus is not available during instruction for other operations (reading main memory, the state registers, file 3 via the MEM bus, etc.).

B FIELD

The B field (bits 10 through 12) specifies the input to S2 and thus to the B side of the ALU. The 11 possible B codes are expanded by the seven B' codes when the S field contains 1000. The N and K codes are controlled by bits 28 and 29 from the C field.

The eleven B inputs to S2 and thus to the B side of the ALU are indicated if the S field is not 1000. Code 001 of the B field is expanded by the use of bits 28 and 29 of the microinstruction for enabling the N or K register to S2. This use of bits 28 and 29 is independent of the other use of the C field and, thus, by judicious use of commands in the C field, the input for the N and K may be used in conjunction with commands or constants in the C field. The codes for B inputs are given in Table 4-7.

The B' inputs are specified in the B field when the S field contains 1000. The codes and actions are given in Table 4-8.

The B' inputs use the main CPU 3-state bus for their inputs, so the bus is not available during the same instruction for other operations (reading main memory, the state registers, file 3 via the MEM bus, etc.).

TABLE 4-5. A INPUT OPERATIONS

A CODE	MNEMONIC	OPERATION
000	F2[1]	Use contents of file 2 register as A source input. Current value of N register is used to address register file 2. If value of N is changed in current microinstruction, its initial value is used to reference the file register.
001	P	Use contents of P register as A source.
010	I	Use contents of I register as A source.
011	X	Use contents of X register as A source.
100	A	Use contents of A register as A source.
101	F	Use contents of F register as A source.
110	F1[1]	Use contents of file 1 register as A source. Current value of K register is used to address register file 1. If value of K is changed in current microinstruction, initial value of K is used to reference file register. With the standard BSC transform, bit 13 of the K' register selects either F1 (Bit = 1) or F3 (Bit = 0) for gating to the F1 bus.
111	MEM	Obtains the selector 1 destined information from the main CPU 3-state bus as an A source. The data may be from either the main memory interface, output from a transform, I/O input, register file 3 or some other source.

[1]RESTRICTION: Value of addressing register (N or K) cannot have been modified by a C' increment or decrement command in preceding microinstruction. In addition, whichever file is read during the current microinstruction, this same file cannot have been written in the preceding microinstruction.

TABLE 4-6. A' INPUT OPERATIONS

A' CODE	MNEMONIC	OPERATION
000	SM1	Use contents of SM register 1 as A source.
001	M1	Use contents of interrupt mask register 1 as A source.
010	SM2	Use contents of SM register 2 as A source.
011	M2	Use contents of interrupt mask register 2 as A source input.
100	Open	Unspecified.
101	Open	Unspecified.
110	Open	Unspecified.
111	Open	Unspecified

NOTE: The A' codes are specified by S field equal to 0111 or 1010.

TABLE 4-7. B CODES

B CODE	MNEMONIC	MIR 28 29	OPERATION
000	F2[1]		Use contents of file register 2 as B source. Value of N register, before instruction is executed, is used to address register file 2.
001	ZERO	11	B source is all zeros.
001	N	10	Use contents of N register as B source. Since N is an 8-bit register, this source uses N as bits 16-23 and zeros as remaining bits.
001	K	01	Use contents of K register as B source. Since K is an 8-bit register, this source uses K as lower bits and zeros as remaining bits.
001	N,K	00	Use contents of K and N registers as B source. These registers are combined with K as bits 24-31, N as bits 16-23 and zeros in bits 0-15.

TABLE 4-7. B CODES (Cont'd.)

B CODE	MNEMONIC	OPERATION
010	BG	Use contents of the bit generator as B source. This source has only one bit set to 1, and position of the bit is specified by either the value of the N register or by a number in the lower 5 bits of the C field, depending upon the state specified by the controlling SM register bit.
011	X	Use contents of X register as B source.
100	Q	Use contents of Q register as B source.
101	F	Use contents of F register as B source.
110	F1[1]	Use contents of file register 1 or file register 3 as B source. Value of K register, before instruction is executed, is used to address register file 1. With the standard BSC transform, bit 13 of the K' register selects either F1 (bit=1) or F3 (bit=0) for gating to the F1 bus.
111	MEM	Enables the main CPU 3-state bus to selector S2 as B source.

[1]RESTRICTION: Value of addressing register (N or K) cannot have been modified by a C increment of decrement command in the preceding microinstruction. In addition, whichever file is read during the current microinstruction, this same file cannot have been written in the preceding microinstruction.

TABLE 4-8. B' CODES

B' CODE	MNEMONIC	OPERATION
001	CRTJ	Transfer the complement of the RTJ register to the 12 LSBs of S2. Transfer 1's to the twenty most significant bits of S2.
010	INRD	Input data/status from AQ/M05 I/O channel contained on the I/O-TTY module.
011	INRS	Input I/O response signals AQ/M05 I/O channel.
100	MMU	Input a 32-bit microinstruction. F field must make a reference to B source. Address is specified by transform or NK. D field must be a NOP. (See MICROMEMORY OPERAND REFERENCES Section for further detail.)
110 or	INTA	Use contents of interrupt address encoder as B source. The output of this encoder represents the complement of the interrupt address of the highest priority interrupt line active having its corresponding mask bit set. Restriction: An INTU test command must be given in the preceding microinstruction.

NOTE: The B' codes are specified by S field equal to 1000.

D FIELD

The D field (bits 13 through 15) specifies the destination of information from the main organization of the MP. There are four sources of this information as follows:

- An optionally shifted ALU output. This shifting occurs in the S3 shift network that connects the ALU output to the P, A, F, X or Q registers.
- The output of the ALU.
- The A source.
- The B source.

All D destinations except the I register are optionally shiftable by S3 when specified by a code in the C field or by the L8EA command in the S field, if the alternate codings D' are not specified. The I destination differs from the others in that the output of the A source is the input to the I register. The codes and their operations are given in Table 4-9.

The D' destinations are specified by the D field if the S field is set to 1001 or 1010. The codes and actions are given in Table 4-10.

The D" destinations are specified by the D field if the S field is set to 1101. The codes and actions are given in Table 4-11.

TABLE 4-9. D CODE TRANSFERS

D CODE	MNEMONIC	OPERATION
000	NOP	Does not transfer data to any destination.
001	P	Transfer output of S3 to P.
010	I	Transfer output of S1 to I.
011	Q	Transfer output of S3 to Q.
100	F1[2]	If MSB of K' = 1, transfer output of S3 to F register, and write this data in file 1 at address specified by K at completion of this instruction. If MSB of K' = 0, transfer output of S3 to F' register, and write this data in file 3 at address specified by K' or J at completion of this instruction.
101	A	Transfer output of S3 to A.
110	X	Transfer output of S3 to X.
111	F	Transfer output of S3 to F.

[2]The writing of data into the file 1 register takes place during the first part of the next microinstruction and takes advantage of the updated value of K from this microinstruction. Also, the next microinstruction must not specify a read of file 1.

TABLE 4-10. D' CODE TRANSFERS

D' CODE	MNEMONIC	OPERATION
000	IOD	Transfer output of S3 to I/O data input register on the I/O-TTY module. (See Input Output for additional information on I/O.)
001	IOA	Transfer output of S3 via the I/O data register to I/O data input address register on the I/O-TTY module. Destroys contents of I/O data register.
010 or 011	MMU	Transfer output of S2 to 32-bit word in micro-memory. (See MICROMEMORY OPERAND REFERENCES section for further detail.)
100	M1	Transfer output of ALU to mask register 1.
101	SM1	Transfer output of ALU to SM register 1.
110	M2	Transfer output of ALU to mask register 2.
111	SM2	Transfer output of ALU to SM register 2.

NOTE: Outputs to the mask and SM registers are direct from the ALU and are not shiftable.

THIS PAGE HAS INTENTIONALLY BEEN LEFT BLANK

T FIELD

The purpose of the T field (bits 16 through 18) is to select the upper or lower microinstruction of the next microinstruction pair to execute. The selection of the next microinstruction may be a conditional selection or an unconditional selection, depending on the T field code. This field is available to all addressing modes in addition to I/O operations. A conditional selection may test the ALU output, the value of certain registers, certain internal conditions such as interrupts and particular bits wired to the transform board. The only exception to these uses of the T field is when micromemory data is being read or written; in these cases, the T field is used as part of the micromemory data reference address and the upper instruction in the next sequential microinstruction pair is always selected.

The T field codes consist of two groups: T codes and T' codes. Similarly, as the A, B and D fields are extended by using the S field, the T field is always extended in the following sense. Bit 24, or format 1 micro-instruction, is either 0 or 1 if T or T', respectively, is specified. The T' codes are available only for the return and sequential addressing modes (format 1). The T/T' codes select the upper or lower portion of the next microinstruction pair as the next microinstruction to execute. The T codes are listed in Table 4-12. and the T' codes are listed in Table 4-13.

TABLE 4-12. T ADDRESSING MODES

T CODE	MNEMONIC	OPERATION
000	*L	Execute the lower microinstruction of this microinstruction pair as next microinstruction. This operation overrides M field addressing mode.
001	U	Execute the upper microinstruction of the next microinstruction pair.
010	L	Execute the lower microinstruction of the next microinstruction pair.
011	[1]KZU	If the initial contents of K register are zero, execute the upper microinstruction of the next microinstruction pair; otherwise, execute the lower microinstruction of the next microinstruction pair. If decrement K command is included in the same microinstruction, K will contain all ones on satisfying the zero test.
100	[1]NZU	If the initial contents of N register are zero, execute the upper microinstruction of the next microinstruction pair; otherwise, execute the lower microinstruction of the next microinstruction pair. If decrement N command is included in the same microinstruction, N will contain all ones on satisfying the zero test.
101	INTU	If there is an interrupt and its corresponding interrupt mask bit is set, execute the upper microinstruction of the next microinstruction pair; otherwise, execute the lower microinstruction of the next microinstruction pair.
110	[2]NU	If the sign bit of ALU output is negative on completion of this microinstruction, execute the upper microinstruction of the next microinstruction pair; otherwise, execute the lower microinstruction of the next microinstruction pair.
111	[2]ZL	If output of ALU is zero on completion of this instruction, execute the lower microinstruction of the next microinstruction pair; otherwise, execute the upper microinstruction of the next microinstruction pair.

[1]RESTRICTION: These T-field commands may not be used in a microinstruction with a C-field INCK or INCN command.

[2]RESTRICTION: These T-field commands may not be used in a microinstruction with either an A' or B' code.

TABLE 4-13. T' ADDRESSING MODES

T' CODE	MNEMONIC	OPERATION
000	*L	Execute lower microinstruction of this microinstruction pair. This operation overrides M field addressing mode.
001	LQL	If at start of this microinstruction's least significant bit of Q is 1, execute the lower microinstruction of the next microinstruction pair. Otherwise, execute the upper microinstruction of the next microinstruction pair.
010	K7L	If the LSB of K register is set, execute the lower of the next microinstruction pair. If clear, execute the upper microinstruction of the next microinstruction pair.
011	OVFL	If overflow exists, execute the lower microinstruction of the next microinstruction pair; if not, execute the upper microinstruction of the next microinstruction pair.
100	BTU	Bit test. Lower order four bits in the C field of this microinstruction specify a setting of the bit test selector. If the bit at this position is 1, execute the upper microinstruction of the next microinstruction pair; otherwise, execute the lower microinstruction of the next microinstruction pair. Bit test is the general-purpose testing facility that allows wiring any bit of organization available on the machine's backpanel to bit test the selector. This wiring is defined on transform board.
101	ERL	Test for any memory fault and if a fault occurred (error lower), execute the lower microinstruction of the next microinstruction pair; otherwise, execute the upper microinstruction of the next microinstruction pair.
110	[2]COL	Carry-out lower. If the carry-out of ALU results from arithmetic operation, execute the lower microinstruction of the next microinstruction pair. Otherwise, execute the upper microinstruction of the next pair. This instruction allows a test for carry-out of ALU during multiple-operation arithmetic.
111	Open	Unspecified.

[2]RESTRICTION: These T-fields commonly may be used in an instruction with either an A' or B' code.

FORMAT MODES

The M field (bits 0 and 1) defines the major operation taking place in the microinstruction and specifies the type of sequencing which will be used to obtain the next microinstruction pair. The operations specified in the M field are listed in Table 4-14. Subformat (SF) select bit 19 is used to select either variation in Format 1 and Format 2 decoding, or the choice of loading the N or K register in Format 3.

TABLE 4-14. M FIELD OPERATIONS

M CODE	MNEMONIC	OPERATION
00	R	Select next microinstruction pair from address contained in RTJ register. Use Format 1 for special operations.
01	S	Select the next microinstruction pair in the current page from the address contained in MAC (normally the next sequential pair, unless suppressed by T field coding). Use Format 1 for special operations.
10	J	A jump or page jump. Select the next microinstruction pair from the address specified by bits 24 to 31 of this microinstruction. Address is in the current MM page if bit 19 is 0, or from the page specified in bits 20 through 23 if bit 19 is 1. Use Format 2 for special operations.
11	S	Transfer bits 24 through 31 of this microinstruction to N or K register as specified by bit 19 of this microinstruction. N register is specified if bit 19 is 1, and K register is specified if bit 19 is 0. Select the next microinstruction pair in current page from address contained in MAC (normally next sequential microinstruction pair). Use Format 3 for special operations.

S FIELD

The S field (bits 20 through 23) of the microinstruction is used to specify a special command (including alternate codings in the A, B and D fields), in addition to page or constant information (as required by the code in the C field). The S codes specify actions which take place at the same time as the ALU operation specified in the F, A, B and D fields. The codes and operations are given in Table 4-15.

C FIELD

The C field (bits 24 through 31) is used to specify an additional special operation, an address for a jump or a constant for setting the K or N register. Bit 24 in Format 1 specifies the T field interpretation.

The codes for this field are listed in Table 4-16.

TABLE 4-15. S FIELD CODES

S CODE	MNEMONIC	OPERATION
0000	NOP	No operation for S field.
0010	RPT	If the N register is not equal to zero, selection of the next microinstruction pair is inhibited and the current microinstruction pair is the next microinstruction pair. N register is decremented by one. Normal T field selection applies. If N register is equal to zero, the normal next microinstruction pair is used. Upon completion of the microinstruction, N will equal FF_{16}.
0101	L8EA	Output of ALU via S3 is shifted left eight bits, end-around.
0110	F2WR	Write data contained in F register into file 2 at the address specified by the contents of the N register at the beginning of the current microinstruction. Actual writing takes place during the first part of the instruction.
0111	AP	Alternate A field coding, A'.
1000	BP	Alternate B field coding, B'.
1001	DP	Alternate D field coding, D'.
1010	APDP	Alternate A and D field coding, A'D'.
1011	DPP	Alternate D field coding, D".
1100	GATEI	Gate output of S1 to I register.

TABLE 4-15. S FIELD CODES (Cont'd.)

S CODE	MNEMONIC	OPERATION
1101	HALT	If halt bit of SM register is 1, stop operation of the MP on completion of this microinstruction. When the start signal is received, continue with the microinstruction specified by the addressing mode and the T field. If halt bit is 0, continue with the microinstruction sequencing.
1110	RTJ	Transfer the address of the next sequential microinstruction pair to the RTJ register. This is done regardless of the actual addressing mode used in this instruction.
1111	CLRNP	Clear the N register and page register in P-MA register so that the next instruction is executed from page zero.
0001 0011 0100	RD2 RD3 RD1	These commands are used to gate their corresponding data register onto the main CPU 3-state bus for use by the next instruction.

TABLE 4-16. C CODES

C CODE	MNEMONIC	OPERATION
The following special operations, for Format 1, have a zero in bit 19 (SF) to specify C' codes (bits 25 through 31). Bit 24 specifies the T field interpetation.		
00xxxxx		xxxxx is a constant for use in driving the bit generator or in any other commands using lower five (5) bits of instruction for control.
0100100 0101000 0101100	LWA1 LWA2 LWA3	LWAx - This group of instructions loads one of the lookahead address registers with a full-word address from the page file and S3. If write mode is selected, a full data word is also loaded into the associated write data holding register from S1. The lookahead logic notes that an address has been loaded and generates a read or write cycle when priority allows. (Priority is DMA lookahead (LA) register, then LA1, LA2 and finally LA3 register.) See Memory Management Interface for additional details.
0100101 0101001 0101101	LHA1 LHA2 LHA3	LHAx - This group of instructions performs the half-word addressing function. The shifted full-word address is loaded into the appropriate address holding register. On a write, data is shifted through the data formatter and stored in the proper half-word of the corresponding write data holding register. The correct pair of write enable lines will be activated during the write operation. On a read, half-word information is sent to the data formatter for input data control.
0100110 0101010 0101110	LCA1 LCA2 LCA3	LCAx - This group of instructions performs the character addressing function. The shifted full-word address is loaded into the appropriate address holding register. On a write, data is shifted through the data formatter and stored in the proper character position of the corresponding write data holding register. The correct write line is activated during the write cycle. On a read, character information is sent to the data formatter for positioning of the character prior to storage in the input data holding register.

TABLE 4-16. C CODES (Cont'd.)

C CODE	MNEMONIC	OPERATION
0100000	WPF	Loads the page file location specified by S3 with the data word from S1.
0100001	WSR	Loads the state register with data from S1. S3 specifies the particular state register.
0100010	RPF	Gates the contents of the page file location specified by S3 to the main CPU 3-state bus during the following instruction cycle.
0100011	RSR	Gates the contents of the S3 specified state register or snapshot register onto the main CPU 3-state bus during the following instruction cycle.
0110000 through 0111111	Open	Unspecified
1000101	INCK	Increment number contained in K register by one.
1001101	INCN	Increment number contained in N register by one.
1000100	DECK	Decrement number contained in K register by one.
1001100	DECN	Decrement number contained in N register by one.
1000000	CLRK	Clear K register.
1001000	CLRN	Clear N register.
101xxxx	xxx	xxx is value of j, from 0 to 15. Set SM register flag j to 1.

TABLE 4-16. C CODES (Cont'd.)

C CODE	MNEMONIC	OPERATION
110xxxx	CLRF/j	xxxx is value of j, from 0 to 15. Clear SM register flag j to 0.
1110000	[1]RQLXN	Destination register (P, A, F or X) and Q register are considered as one double-length register with Q register as lower order bits. Combined register is shifted left one bit position with complement of ALU sign bit entered into lowest bit position of Q register.
1110010	[1]RQR0E	Shift combined destination and Q register right one bit, and enter 0 in sign position of destination register. This command is used in multiply iteration.
1110011	[1]RQR1E	Shift combined destination and Q register right one bit, and enter 1 in sign position of destination register. This command is used in multiply iteration.
1110100	[1]RL0E	Shift destination register left one bit, entering 0 in lowest bit position of the register. This operation cannot be performed when Q is destination register.
1110101	[1]RL1E	Shift destination register left one bit, entering 1 in lowest bit position. This operation cannot be performed when Q is destination register.
1110110	[1]RR0E	Shift destination register right one bit, entering 0 in sign position of register. This operation cannot be performed when Q is destination register.
1110111	[1]RR1E	Shift destination register right one bit, entering 1 in sign position. This operation cannot be performed when Q is destination register.

[1]RESTRICTION: This operation cannot be performed when S field command (L8EA) is selected.

TABLE 4-16. C CODES (Cont'd.)

C CODE	MENMONIC	OPERATION
The following special operations, for Format 1, have a one in bit 19 (SF) to specify the C" codes (bits 25 through 31). Bit 24 specifies the T field interpretation. The S field decode is inhibited for all operations.		
See discussion of transforms and transform module.		
000xxxx	TMA/j	xxxx = j, with values from 0 to 15. Obtain next microinstruction pair from address specified by MA transform selector setting j.
001xxxx	TMAK/j	xxxx = j, from 0 to 15. Obtain next instruction pair from address specified by MA transform selector setting j. Also, set K register to value specified by K transform selector setting j.
010xxxx	GITMAK/j[1]	Gate output of main memory to IXT register (on transform module) and perform TMAK/j operation. Note that j = xxxx with values of 0 to 15.
S field decoding is inhibited for the above instructions. If SM207 = 1, microaddress transforms are performed with the S field containing the page address.		
011xxxx	TK/j	xxxx = j, with values from 0 to 15. Set K register to value specified by K transform selector setting j.
100xxxx	TN/j	xxxx = j, with values from 0 to 15. Set N register to value specified by N transform.
101xxxx	TR/j	xxxx = j, with values from 0 to 15. Set the transform register to value specified by R transform selector setting j.

[1]GITMAK to K' with IXT as S8 source leaves K' in undetermined state.

TABLE 4-16. C CODES (Cont'd.)

C CODE	MNEMONIC	OPERATION
The following instructions are used to manage the File F3 and registers J, K' and L which are located on the standard BSC transform module.		
1100000	K'/NOP	No other effect. Note: K' selects K' register for file 3 address for all K'/ instructions.
1100001	K'/S2J	Gates S2 output (least significant 13 bits) to the J register.
1100010	K'/S2L	Gates S2 output (least significant 7 bits) to the L register.
1100011	K'/JLJ	Add L (sign extended) to J.
1100100	J/NOP	No other effect. Note: J selects J register for file 3 address for all J/ instructions.
1100101	J/S2J	Gates S2 output (least significant 13 bits) to the J register.
1100110	J/S2L	Gates S2 output (least significant 7 bits) to the L register.
1100111	J/JLJ	Add L (sign extended) to J.
1101000	GATEIXT	Gates S2 to the IXT register.
111xxxx	Open	Unspecified.
The following Format 3 codings are for setting the K and N register; this format has the M field bits 0 and 1 set to 11, while bit 19 selects the register.		
Value	K = value	When microinstruction bit 19 = 0, transfer C field value (bits 24 to 31) to K register and execute the next sequential microinstruction pair.
Value	N = value	When microinstruction bit 19 = 1, transfer C field value (bits 24 to 31) to N register and execute the next sequential microinstruction pair.
The following Format 2 codings in the C field are used to perform a jump, specified if the M field (bits 0 and 1) is 10.		
Number	Number	Number is the address of the next instruction pair. If page jump is required (bit 19 = 1), S field contains page setting instead of S field code.

MICROMEMORY OPERAND REFERENCES

The BSC has the capability of transferring information between the micromemory and the registers of the processor. The transfer from the register to micromemory is possible only if the micromemory is a read/write micromemory.

Micromemory is addressed as one to sixteen pages of 256 words each, where each word is 64 bits and is divided into upper and lower 32-bit words. The BSC processor can reference 32-bit micromemory words by specifying page, address and upper or lower microinstruction.

Microinstruction for micromemory operand references may be an upper or lower microinstruction; the next microinstruction executed following the referencing microinstruction is always the upper microinstruction of the next sequential location. Microinstructions referencing a micromemory operand do not have to reside in the same page as the micromemory operand being referenced, when the combined N/K register is used to reference the operands.

The command for reading an operand from micromemory is coded in the B' field as MMU; the command for writing an operand into micromemory is coded in the D' field as MMU. The data read from micromemory is destinated to the X register and the D field should be a NOP. The data to write into micromemory comes from the B field.

MICROINSTRUCTION TIMING

The basic CPU microinstruction execution time is 168 nanoseconds. Some microinstructions have longer execution times to allow certain operations to be completed. The microinstructions have been grouped according to execution times as requiring 3, 4, 5, 6, 5 + n and 9 56-nanosecond cycles.

The classification of microinstructions is shown in Figure 4-3. This figure provides execution time by types for all legal combinations of microcommands which extend the basic cycle time.

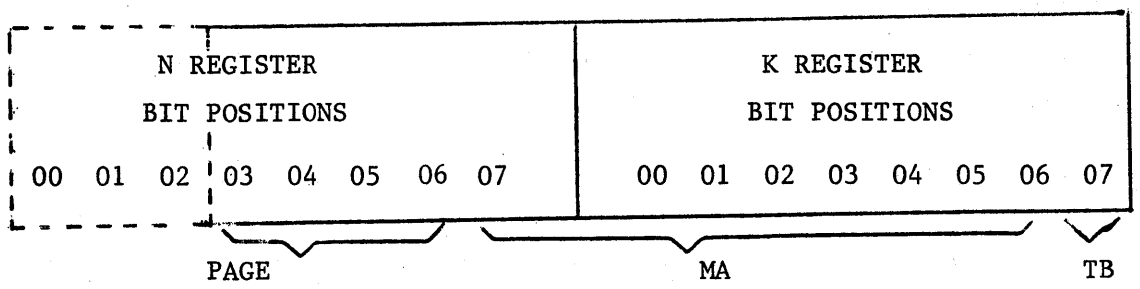
Exceptions to the execution times, as listed in Figure 4-3, are:

- The combination of one's complement arithmetic with ADD+ or ADD+T arithmetic operation requires four 56-nanosecond cycles.
- The MMU command is included under read/write micromemory operand commands; therefore, it requires nine, rather than four, cycles.

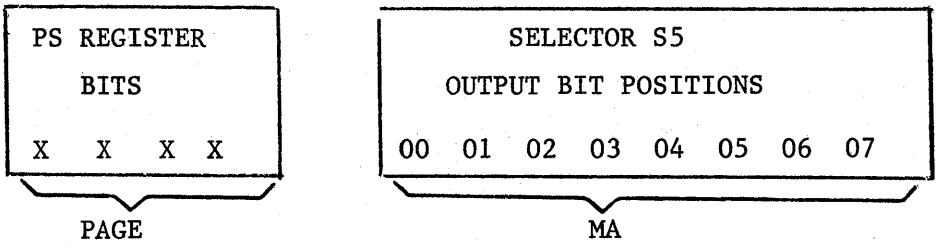
Analysis of a microprogram for execution time starts by classifying each of the microinstructions. This is done by using the microinstruction classification table or by examining the assembler output lising.

Two addressing modes are available for operand references via status mode bit 113 (SM113) as follows:

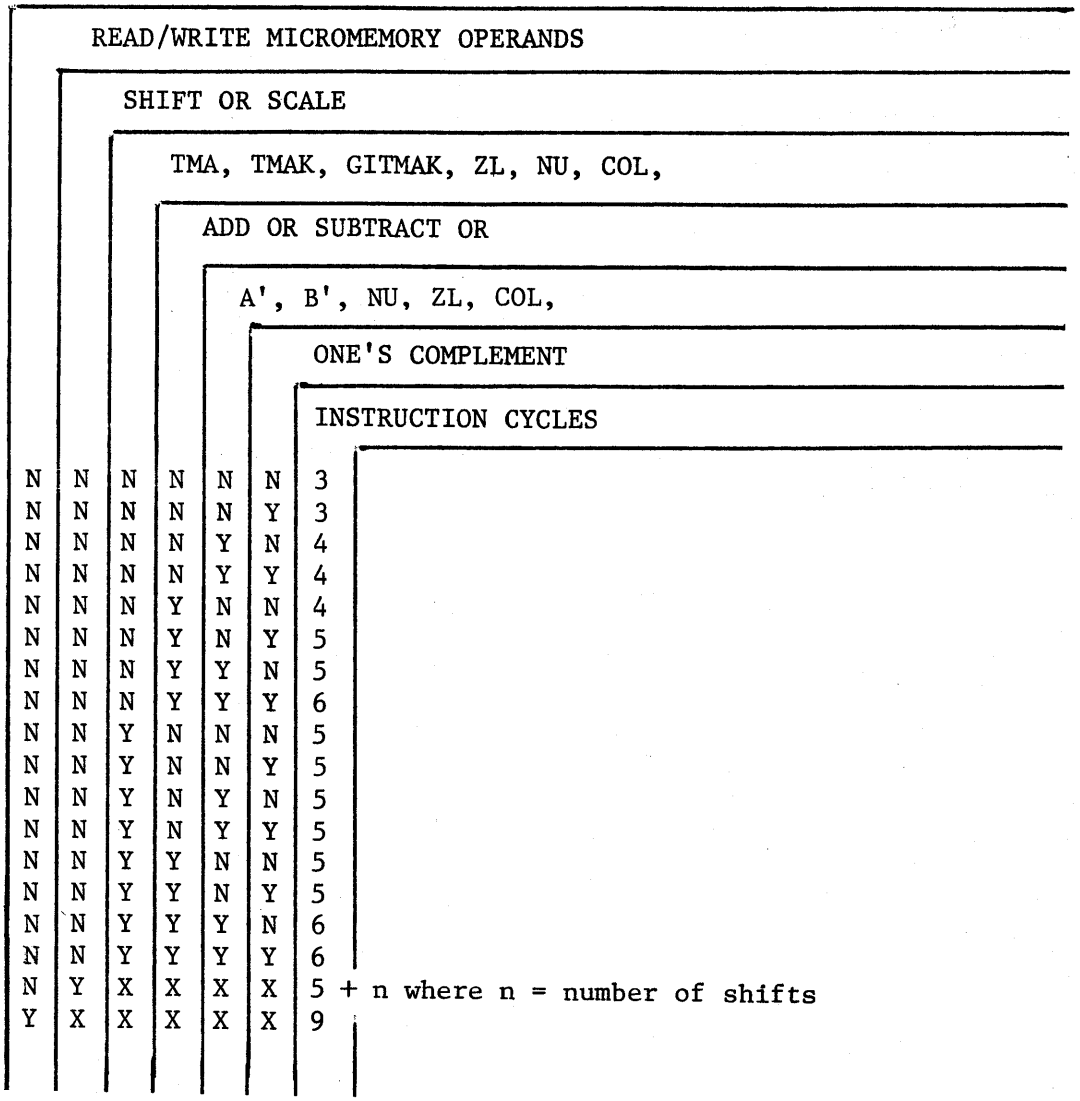
- SM113 = 0 - The contents of the combined N/K registers are used to reference operands as indicated. The least significant bit of K (bit 07) determines which 32-bit half-word is referenced via T' code 010.



- SM113 = 1 - An MA transform via S5 determines the 64-bit word. The 32-bit half-word to be referenced must be selected by a T field test.



XXXX = Page in which referencing microinstruction resides.



N = No
Y = Yes
X = Don't Care

Figure 4-3. Microinstruction Classification

INSTRUCTION SUMMARIES

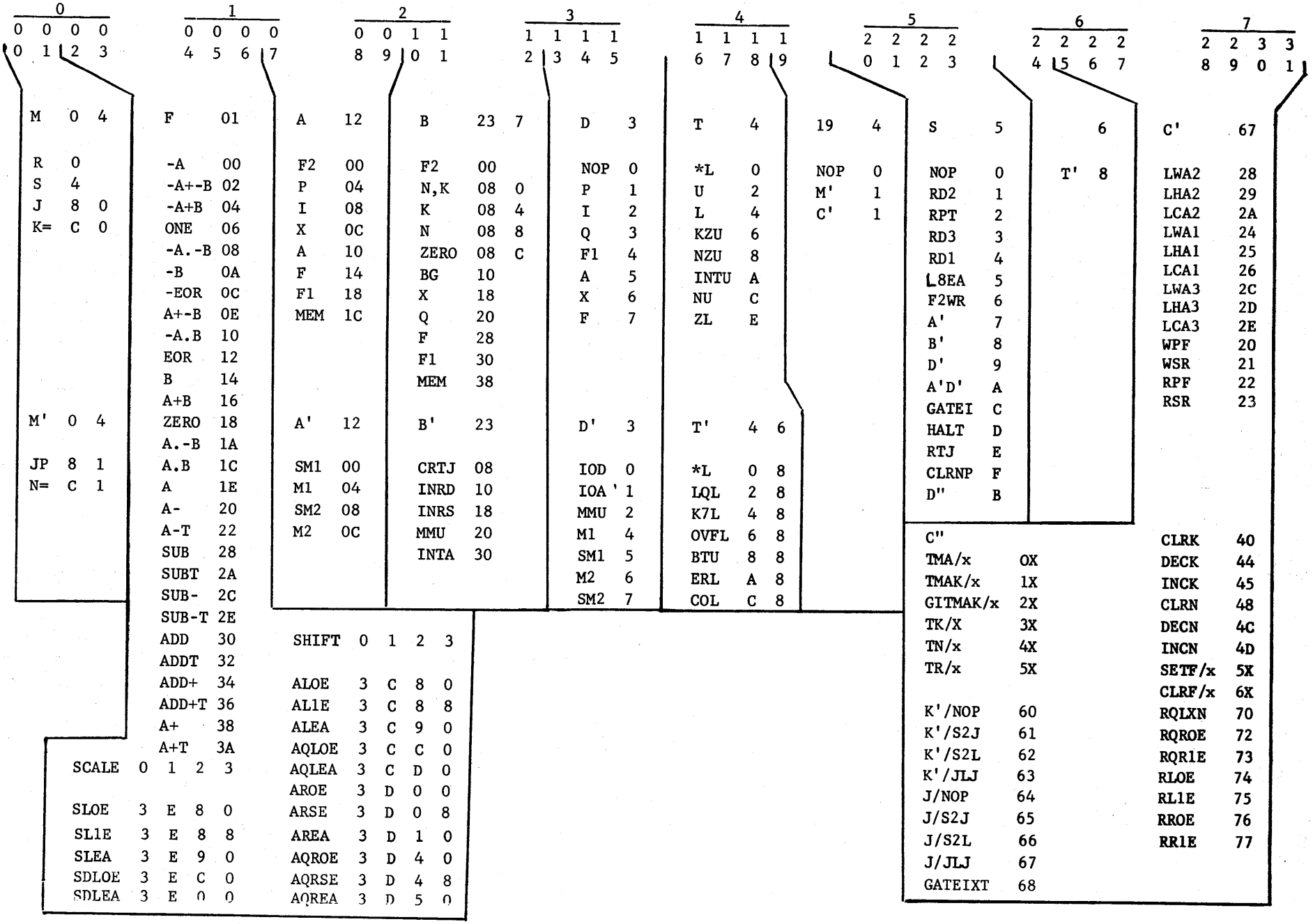
Figures 4-4 and 4-5 summarize the BSC mnemonic instructions and their corresponding bit pattern in the instruction.

In Figure 4-4, the hexadecimal mask at the top of each column defines the instruction subfield and the mnemonic/hexadecimal values in the column 1 specify the subfield values.

Figure 4-5 illustrates the same information, but on subfield boundaries rather than on hexadecimal digit boundaries.

M	C...1...	F	3E 3FFF....	A	.1C..F..	B	..38.F.C	D	...7.F..	T	....E.8.	S	.....F..	C	....1.7F
R	0.....	-A	00.....	F2	.00.....	F2	..00....	NOP	...0....	*L	....0.0.	NOP	.....0..	**LWA2**	**....0.28**
S	4.....	-A+-B	02.....	P	.04.....	N,K	..08...0	P	...1....	U	....2.0.	RD2	.....1..	**LHA2**	**....0.29**
J	8...0...	-A+B	04.....	I	.08.....	K	..08...4	I	...2....	L	....4.0.	RPT	.....2..	**LCA2**	**....0.2A**
JP	8...1...	ONE	06.....	X	.0C.....	N	..08...8	Q	...3....	KZU	....6.0.	RD3	.....3..	**LWA1**	**....0.24**
K=	C...0.nn	-A.-B	08.....	A	.10.....	ZERO	..08...C	F1	...4....	NZU	....8.0.	RD1	.....4..	**LHA1**	**....0.25**
N=	C...1.nn	-B	0A.....	F	.14.....	BG	..10..nn	A	...5....	INTU	....A.0.	L8EA	.....5..	**LCA1**	**....0.26**
		-EOR	0C.....	F1	.18.....	X	..18....	X	...6....	NU	....C.0.	F2WR	.....6..	**LWA3**	**....0.2C**
		A+-B	0E.....	MEM	.1C.....	Q	..20....	F	...7....	ZL	....E.0.	A'	.....7..	**LHA3**	**....0.2D**
		-A.B	10	SM1	.00..7..	F	..28....	IOD	...0.9..	LQL	....2.8.	B'	.....8..	**LCA3**	**....0.2E**
		EOR	12	M1	.04..7..	F1	..30....	IOA	...1.9..	K7L	....4.8.	D'	.....9..	**WPF**	**....0.20**
		B	14	SM2	.08..7..	MEM	..38....	MMU	...2.9..	OVFL	....6.8.	A'D'	.....A..	**WSR**	**....0.21**
		A+B	16	M2	.0C..7..	CRTJ	..08.8..	M1	...4.9..	BTU	....8.8n	GATEI	.....C..	**RPF**	**....0.22**
		ZERO	18			INRD	..10.8..	SM1	...5.9..	ERL	....A.8.	HALT	.....D..	**RSR**	**....0.23**
		A.-B	1A			INRS	..18.8..	M2	...6.9..	COL	....C.8.	RTJ	.....E..	CLRK	....0.40
		A.B	1C			MMU	..20.8..	SM2	...7.9..			CLRNP	.....F..	DECK	....0.44
		A	1E			INTA	..30.8..							INCK	....0.45
		A-	20											CLRN	....0.48
		A-T	22											DECN	....0.4C
		SUB	28											INCN	....0.4D
		SUBT	2A											SETF/n	....0.5n
		SUB-	2C											CLRF/n	....0.6n
		SUB-T	2E											RQLXN	....0.70
		ADD	30											RQR0E	....0.72
		ADDT	32											RQR1E	....0.73
		ADD+	34											RL0E	....0.74
		ADD+T	36											RL1E	....0.75
		A+	38											RR0E	....0.76
		A+T	3A											RR1E	....0.77
		AL0E	3C80											TMA/n	....1.0n
		AL1E	3C88											TMAK/n	....1.1n
		ALEA	3C90											GITMAK/n	....1.2n
		AQL0E	3CC0											TK/n	....1.3n
		AQLEA	3CD0											TN/n	....1.4n
		AR0E	3D00											TR/n	....1.5n
		ARSE	3D08											K'/NOP	....1.60
		AREA	3D10											K'/S2J	....1.61
		AQR0E	3D40											K'/S2L	....1.62
		AQRSE	3D48											K'/JLJ	....1.63
		AQREA	3D50											J/NOP	....1.64
		SL0E	3E80											J/S2J	....1.65
		SL1E	3E88											J/S2L	....1.66
		SLEA	3E90											J/JLJ	....1.67
		SDL0E	3EC0											**GATEIXT**	**....1.68**
		SDLEA	3ED0												

Figure 4-4. Summary of BSC Instruction and Fields



0				1				2				3				4				5				6				7			
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1

M	0	4	F	01	A	12	B	23	7	D	3	T	4	19	4	S	5		6	C'	67
R	0		-A	00	F2	00	F2	00		NOP	0	*L	0	NOP	0	NOP	0			LWA2	28
S	4		-A+-B	02	P	04	N,K	08	0	P	1	U	2	M'	1	RD2	1	T'	8	LHA2	29
J	8	0	-A+B	04	I	08	K	08	4	I	2	L	4	C'	1	RPT	2			LCA2	2A
K=	C	0	ONE	06	X	0C	N	08	8	Q	3	KZU	6			RD3	3			LWA1	24
			-A.-B	08	A	10	ZERO	08	C	F1	4	NZU	8			RD1	4			LHA1	25
			-B	0A	F	14	BG	10		A	5	INTU	A			L8EA	5			LCA1	26
			-EOR	0C	F1	18	X	18		X	6	NU	C			F2WR	6			LWA3	2C
			A+-B	0E	MEM	1C	Q	20		F	7	ZL	E			A'	7			LHA3	2D
			-A.B	10			F	28								B'	8			LCA3	2E
			EOR	12			F1	30								D'	9			WPF	20
			B	14			MEM	38								A'D'	A			WSR	21
			A+B	16												GATEI	C			RPF	22
M'	0	4	ZERO	18	A'	12	B'	23		D'	3	T'	4 6			HALT	D			RSR	23
			A.-B	1A												RTJ	E				
JP	8	1	A.B	1C	SM1	00	CRTJ	08		IOD	0	*L	0 8			CLRNP	F				
N=	C	1	A	1E	M1	04	INRD	10		IOA '	1	LQL	2 8			D"	B				
			A-	20	SM2	08	INRS	18		MMU	2	K7L	4 8								
			A-T	22	M2	0C	MMU	20		M1	4	OVFL	6 8								
			SUB	28			INTA	30		SM1	5	BTU	8 8								
			SUBT	2A						M2	6	ERL	A 8								
			SUB-	2C						SM2	7	COL	C 8								
			SUB-T	2E																	
			ADD	30	SHIFT	0	1	2	3							C"				CLRK	40
			ADDT	32												TMA/x	0X			DECK	44
			ADD+	34	ALOE	3	C	8	0							TMAK/x	1X			INCK	45
			ADD+T	36	ALlE	3	C	8	8							GITMAK/x	2X			CLRN	48
			A+	38	ALEA	3	C	9	0							TK/X	3X			DECN	4C
			A+T	3A	AQLOE	3	C	C	0							TN/x	4X			INCN	4D
			SCALE	0	1	2	3	AQLEA	3	C	D	0				TR/x	5X			SETF/x	5X
								AROE	3	D	0	0								CLRF/x	6X
			SLOE	3	E	8	0	ARSE	3	D	0	8				K'/NOP	60			RQLXN	70
			SLlE	3	E	8	8	AREA	3	D	1	0				K'/S2J	61			RQROE	72
			SLEA	3	E	9	0	AQROE	3	D	4	0				K'/S2L	62			RQRlE	73
			SDLOE	3	E	C	0	AQRSE	3	D	4	8				K'/JLJ	63			RLOE	74
			SDLEA	3	E	0	0	AQREA	3	D	5	0				J/NOP	64			RL1E	75
																J/S2J	65			RROE	76
																J/S2L	66			RR1E	77
																J/JLJ	67				
																GATEIXT	68				

Figure 4-5. Alternate Instruction Summary

GLOSSARY

A Field	In microinstruction, A field specifies source of operand to be sent to ALU from selector 1.
A Register	General-purpose register.
ALU	Arithmetic and logical unit. Performs arithmetic and logical operations on two operands sent to it from two selectors.
B Field	In microinstruction, B field specifies source of operand to be sent to ALU from selector 2.
BSC	Basic System Controller.
BG	Bit Generator. Allows a word to be sent to ALU with all zeros except one bit set at any bit position; used for masking or arithmetic operations.
C Field	Constant field. May contain constants, micro memory addresses, or other codes, depending on microinstruction format.
CPU	Central processing unit. Consists of micromemory, control section, arithmetic section and SMI.
D Field	Destination field. Specifies destination for results of operation performed by ALU.
Deadstart	Optional logic that allows loading of read/write micromemory from external input device.
DMA	Direct memory access.
Emulation	Process combining hardware and firmware design, by which one processor (emulator) executes programs designed for a different processor, even though one-to-one hardware correspondence does not exist.

F Field	Function field. Specifies operation to be performed by ALU or shift or scale of A or AQ registers.
F Register	General-purpose register.
F' Register	Holding register on transform board for data being written into F3.
File 1	Register file addressed by contents of K register.
File 2	Register file addressed by contents of N register.
File 3	Register file on the standard BSC transform board and addressed by the content of either the J or K' register.
Firmware	General term for combination of microinstructions used in microprogram to perform a certain operation.
I Register	General-purpose register.
IC	Integrated circuit.
IXT	Data holding register on the transform.
I/O	Input/output.
J Register	Used to address file F3 and located on the BSC transform board.
K Register	Eight-bit counter that can be cleared, incremented or decremented under microinstruction control. Also used to address file 1.
K' Register	Used to address file 3. Lower 8 bits identical to the K register.
L Register	Used to increment J register and contained on the BSC transform board.

M Field	Mode field. Specifies addressing mode for use in obtaining next microinstruction pair from micromemory.
MA Register	Micromemory address register. Holds micromemory address of current microinstruction pair.
MA Transform	Micromemory address transform.
MAC	Memory address counter. Holds address of the next sequential microinstruction pair.
Main Memory	MOS memory used by MP for storage of operands, macro instructions, etc.
Mask Registers (M1, M2)	Used to control processing of internal and external interrupts.
Microinstruction	32-bit instruction from micromemory that controls all operations through MP system.
Micromemory	High-speed semiconductor memory, which contains microprograms.
Microprogram	Set of microinstructions stored in micromemory.
MIR	Microinstruction register. Holds microinstruction being executed.
MM	Micromemory.
MMI	Memory Management Interface of BSC that interfaces the CPU and 4 DMA I/O channels to main memory.
MPM	Multi Port memory.
MP	Microprocessor. The basic microprogrammable processor, which can be configured in many forms/applications.
MP 60	AN MP 32 with a specific emulation firmware set.

MP 32	A 32-bit MP processing element with an application defined microprogram
N Register	Eight-bit counter that can be cleared, incremented or decremented under microinstruction control. Also used to address file 2.
P Register	General-purpose register used to hold main memory address of software instruction being executed when MP is configured as emulator.
P.S.	Page storage register used with RTJ.
Q Register	General-purpose register used in multiply and divide operations.
RTJ Register	Return jump register. Holds micromemory address to which control returns at completion of a subroutine.
S Field	Special field. Specifies operation to be performed in parallel with ALU operation.
S1, S2, etc.	Selector 1, selector 2, etc.
Selector	Multiplexer that allows the selection of one of several sources of data for transfer from one location in MP organization to another, under microinstruction control.
Status/Mode Register	Contains flag bits and status/mode bits. Flag bits are set under microinstruction control to enable certain internal MP operations. Status/Mode bits indicate internal or external conditions (e.g., memory parity error).
T Field	Test field. Specifies whether upper or lower microinstruction of the next microinstruction pair is to be executed.

Transform Matrix	Selects bits from various sources in MP organization and translates them into micromemory address in MA register, or transfers them to K or N register. Used in converting user application to MP microcode.
X Register	General-purpose processor register.

COMMENT SHEET

CONTROL DATA CORPORATION

TITLE: BASIC SYSTEM CONTROLLER MODEL 65109 HARDWARE REFERENCE MANUAL

PUBLICATION NO. 41618400 REVISION 02

This form is not intended to be used as an order blank. Control Data Corporation solicits your comments about this manual with a view to improving its usefulness in later editions.

Applications for which you use this manual.

Do you find it adequate for your purpose?

What improvements to this manual do you recommend to better serve your purpose?

Note specific errors discovered (please include page number reference).

CUT ON THIS LINE

General comments:

FROM NAME: _____ POSITION: _____

COMPANY NAME: _____

ADDRESS: _____

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Publications and Graphics Division
215 Moffett Park Drive
Sunnyvale, California 94086

CUT ON THIS LINE

FOLD

FOLD

STAPLE

STAPLE