39451400 E

CONTROL DATA CORPORATION

CDC® BASIC MICRO-PROGRAMMABLE PROCESSOR

GENERAL DESCRIPTION
OPERATION
INSTALLATION
THEORY OF OPERATION
DIAGRAM DESCRIPTIONS
MAINTENANCE

HARDWARE REFERENCE/MAINTENANCE MANUAL

39451400

CONTROL DATA CORPORATION

CDC® BASIC MICRO-PROGRAMMABLE PROCESSOR

GENERAL DESCRIPTION
OPERATION
INSTALLATION
THEORY OF OPERATION
DIAGRAM DESCRIPTIONS
MAINTENANCE

HARDWARE REFERENCE/MAINTENANCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
01 (8/75)	Preliminary edition released.
A (5/77)	Manual released.
B (10/78)	Manual revised; includes ECO DS21302, providing a new part number for the card extender.
C (7/81)	Manual revised; includes ECO DS022118, providing an FCC non-tested label to the preface.
D (11/82)	Manual revised to incorporate appropriate I/O-TTY and transform controller information. This edition obsoletes all previous editions of this manual and publication numbers 60475330, 96728700 and 96728900.
E (12/84)	Manual revised to make diagrams more legible.

Publication No.
39451400

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

by Control Data Corporation
Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Div.
2401 North Fairview Avenue
St. Paul, Minnesota 55113

or use Comment Sheet in the back of this manual.

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV
Cover	-				
Title Page	-				
ii thru iv	E				
v thru ix	D				
1-1 thru 1-7	D				
1-8	E				
1-9/1-10	D				
1-11	E				
1-12/1-13	D				
2-1 thru 2-11	D				
3-1	D				
4-1	E				
4-2/4-3	D				
4-4	E				
4-5 thru 4-34	D				
5-1 thru 5-77	D				
6-1	D				
A-1 thru A-12	D				
Comment Sheet	E				
Mailer	-				
Cover	-				

PREFACE

This manual describes the operation and maintenance of the basic Control Data® CYBER 18 Micro-Programmable (MP) Processor that consists of the control 1, control 2, status mode interrupt (SMI), arithmetic/logical unit (ALU), transform, and I/O-TTY controller modules. Although designed for use in the standard CYBER 18 product line, the equivalent components are used in various quoted special equipments (QSEs) and other non-standard basic micro-programmable processor configurations. Other building blocks that make up the total processor, such as the micro memory, macro memory, and breakpoint controller and breakpoint panel, are described in separate hardware maintenance manuals.

This manual contains physical and functional descriptions of the basic micro-programmable processor system, including the operating procedure, summary of micro code, simple checkout, theory of operation, and maintenance aids. Macro instruction information is described in the Micro-Processor hardware reference manual.

The theory of operation and diagram descriptions contained in this manual are intended for use in conjunction with the logic diagrams contained in the applicable processor field print package manual.

The diagram description section contains references to logic diagram sheet numbers and logic element location designations. Logic diagram sheet number references, appearing in the upper and lower corners of the block diagram functional blocks, indicate the sheet number of the logic diagram that contains the logic element(s) that perform the function. The logic element location designations are included in the text and contained in parenthesis or brackets.

The following documents may be useful to readers of this manual.

Publication	Publication Number
1700 Computer Systems MSMP Diagnostic Reference Manual	96700000
AA109, AA135 CYBER 18 Processor with MOS Memory Field Print Package Manual	60475150
AA132, AA133, AA153, AA155 CYBER 18 Processor with MOS Memory Field Print Package Manual	60475100
AA132, AA133, AA153, DT120, FC402 CYBER 18 Computer Systems Central Processor Field Repair Guide	60475001
AA145 CYBER 18 Batch Terminal Cabinet Field Print Package	96752052
AT241, AT275, BA212, DT223 MOS Memory Subsystem Hardware Reference/Maintenance Manual	96768600
CYBER 18 Computer Systems Overview Manual	60475000
CYBER 18 Computer Systems Site Planning Kit	96768510
CYBER 18 Computer Systems with MOS Memory Installation Manual	96768360
CYBER 18 Processor with MOS Memory (Macro Level) Hardware Reference Manual	96768300
CYBER 18-5M Batch Terminal Computer System Hardware Maintenance Manual Volume 1	96768110
CYBER 18-5M Batch Terminal Computer System Hardware Maintenance Manual Volume 2	96768111
DT120, DT195, FC402 Breakpoint Controller and Breakpoint Panel Hardware Reference/Maintenance Manual	96729000
Micro-Programmable Computer Family (Micro Processor) Hardware Reference Manual	88973400
Operational Diagnostic System (ODS) Version 2 Reference Manual	96768410

CONTENTS

APPENDIX

FIGURES

TABLES

GENERAL DESCRIPTION 1

SYSTEM CHARACTERISTICS

Table 1-1 lists the physical and functional characteristics of a typical micro-programmable processor system.

BASIC MP PROCESSOR SYSTEM DESCRIPTION

The basic microprocessor can be configured in many different forms with the same basic hardware. The basic processor combines with other standard modules such as breakpoint controller (panel interface), micro memory, and macro memory to form a complete processor. This processor controls and gathers data from devices in an application system that performs mathematical computations, logical analyses, and data management operations. Figure 1-1 is a simplified block diagram of the interface between the processor and an application system.

With a card reader, flexible disk drive or cassette transport subsystem installed, the subsystem controller installed in a deadstart controller slot can be used to load the macro memory and processor read/write micro memory. Refer to the MOS memory installation manual for deadstart slot capability.

The input/output section of the processor consists of the program-controller internal CDC 1700 A/Q channel, NCR M05 set/sample I/O channel, CDC 1700 direct memory access (DMA) channel, interrupt logic, teletypewriter (TTY)/display controller, real-time clock, and breakpoint controller. The 1700 data/command (A/Q)channel and M05 set/sample channel use A (data) and Q (address) registers to communicate with the peripheral controllers via the RD/SD (data) and ADR (address) lines. These A/Q channels are single-word transfer channels that interface with low-speed equipments in the application system (for example, magnetic tape transport, card reader, line printer, and tape cassette transport). The 1700 DMA channel interfaces with medium- and high-speed equipments (flexible disk drive, storage module drive, etc.) in the application system and transfers data blocks directly to and from macro memory. The I/O-TTY (TTY/display) controller provides an interface between the processor and a CDC display console or teletypewriter (Model ASR/KSR 33/35). Interrupts indicate that an equipment requires servicing by the processor or that an operation has been completed. Interrupts may also indicate a specific condition within an equipment (such as a malfunction, manual intervention, end of operation, etc.). The real-time clock is used to determine elapsed time. The breakpoint controller (panel interface) provides for all common computer control panel functions (such as: enter display registers, read/write macro memory, read/write micro memory, breakpoint, start, stop, master clear, etc.). These functions are inserted from either the breakpoint panel (maintenance panel), an RS232-compatible console, or a teletypewriter.

The interface also provides a data path for a peripheral controller to transmit ASCII characters to the breakpoint controller. This feature allows macro or micro memory to be loaded from devices such as the card reader.

The auto-data transfer (ADT) mode is an optional feature of the processor. It requires the firmware that controls the ADT operations. This firmware provides for pseudo direct memory transfers of data blocks to and from the selected peripheral device. Once initiated, the ADT only requires micro-level interrupt response from the processor and is completely a function of firmware. Refer to the applicable processor (macro-level) reference manual.

The core macro memory is configured as a one-bank, two-port memory. The DMA port has higher priority than the CPU port. The CPU port transfers program instructions to the processor and interchanges data with the processor. The DMA port allows direct data transfer between macro memory and high-speed application system equipments.

The MOS memory is configured as a one-bank, two-port memory that may be expanded to create a two-bank, three-port memory system. Each MOS bank provides from 16K to 131K 18-bit words of storage without error correction, and from 16K to 98K of storage with error correction. The two banks are referred to as local and external banks. Each bank may have only one memory request active at one time. However, each bank operates independently, thus permitting concurrent access by different ports to each bank. Each of the three ports in each bank (local CPU, local DMA and external) has independent address and data control paths to and from memory and may request memory independently of any operation currently in progress on another port.

TABLE 1-1. PHYSICAL AND FUNCTIONAL CHARACTERISTICS

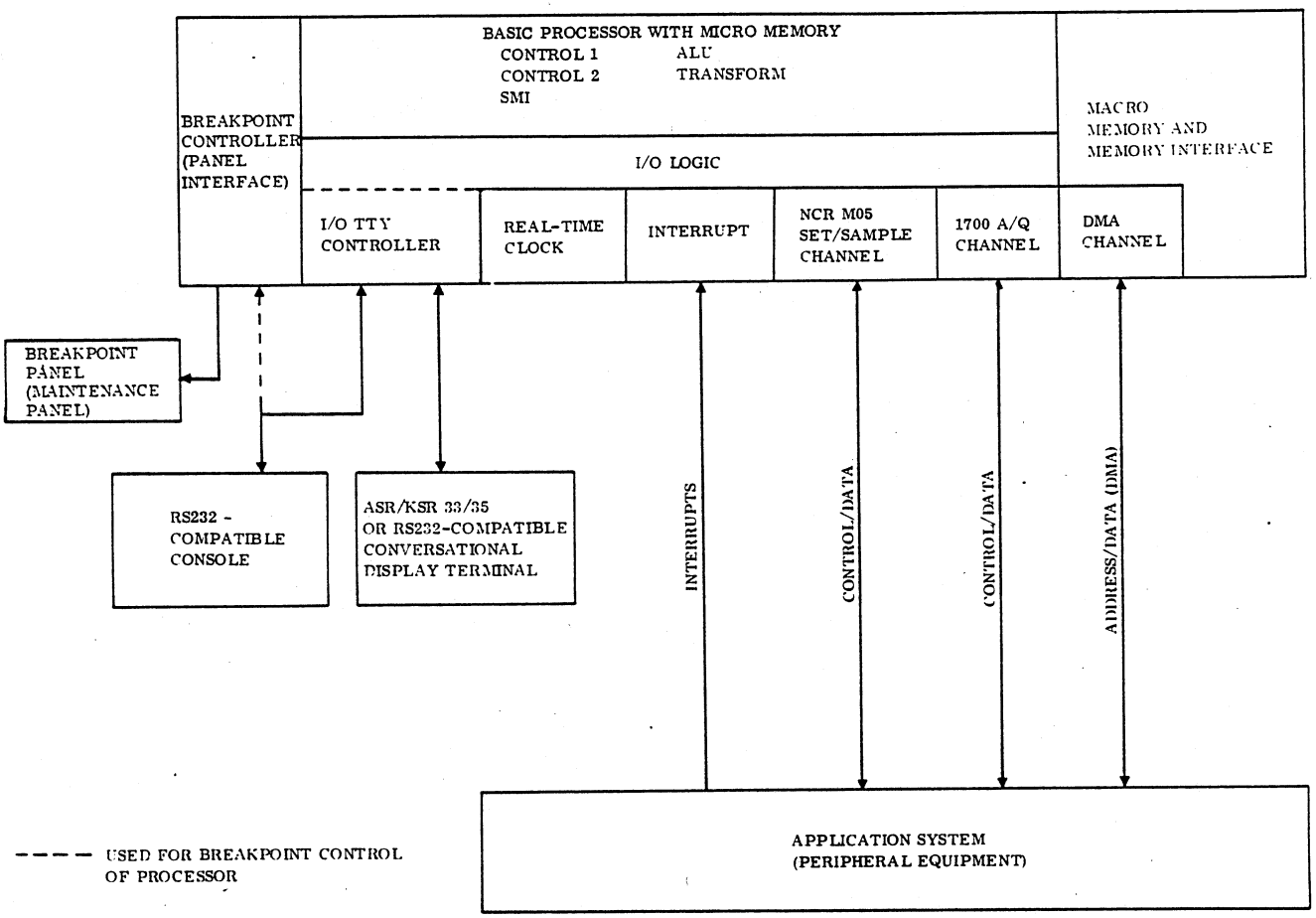
Feature	Description
SYSTEM CHARACTERISTICS	
Type	Stored program, general purpose, parallel mode, micro-programmable digital computer
Word length	16 bits
Organization	Register- or file-oriented
Arithmetic	Binary, with dynamic selection of ones or twos complement mode
Micro-memory type	Semiconductor read/write memory and/or read-only memory
Micro-memory cycle time	168 nanoseconds cycle time; 70 nanoseconds access time
Macro-memory type	Core memory - Available in 8K stacks of 20 bit words; maximum 32K (16 data bits, 1 parity bit, 1 protect bit, 1 protect parity bit, and 1 unused bit) MOS memory - Available in modules of 16K or 32K times 18 bits up to a maximum of 131K (16 data bits, 1 parity bit, and 1 protect bit)
Macro-memory addressing mode†	Absolute (direct), constant, indirect (multilevel), storage, relative, 16-bit relative, relative indirect
Macro-memory cycle time	Core memory - Read: 600 nanosecond minimum cycle time†† Write: 700 nanosecond minimum cycle time†† MOS memory - Read: 550 nanoseconds minimum cycle time Read/Modify/Write: 950 nanoseconds minimum cycle time
Macro-memory protection	Allows only protected instructions or protected input/output users to change contents of protected areas of memory.
Input/Output	CDC 1700 DMA/DSA I/O Bus TTL level CDC 1700 A/Q I/O Bus TTL level NCR M05 set/sample I/O bus Interface to Teletype ASR/KSR 33/35 or CDC RS232-compatible display terminal Interface to breakpoint panel Interface to RS232 compatible console
Real-time clock	Data interrupt generated every 3.33 milliseconds derived from crystal oscillator

†Function of firmware

††The shortest possible time between successive operations

TABLE 1-1. PHYSICAL AND FUNCTIONAL CHARACTERISTICS (Contd)

Feature	Description
BASIC PROCESSOR PARAMETERS	
Word size	16 bits
Micro-memory word	64 bits; two micro instructions per micro-memory word
Micro instruction	32 bits
Micro-memory size	Read-only memory on transform module: 1024 micro instructions (512 words by 64 bits) Read/write micro memory 1. 512 micro-instruction increments (256 words of 64 bits; two micro-instructions per word) 2. 2048 micro-instruction increments, expandable up to 8192 micro instructions (1096 words of 64 bits; two micro-instructions per word)
Macro memory	Core memory - Available in 8K stacks of 20 bit words; maximum 32K (16 data bits, 1 parity bit, 1 protect bit, 1 protect parity bit, and 1 unused bit) MOS memory - Modules of 16K or 32K times 18 bits; expandable to 131K in CPU
Scale bits	Scale on A register MSB, but can be reconfigured to specify different bits of A register
Adder split	Forms two 8-bit adders
File 1	256 x 16-bit register, addressed by K register
File 2	32 x 16-bit register, addressed by N register
DIMENSIONS	Processor chassis Height - 470 mm (18.5 in) Width - 445 mm (17.5 in) Depth - 406.5 mm (16.0 in)
WEIGHT	Processor chassis - approximately 18 kg (40 lbs)



031-2

Figure 1-1. Simplified MP Processor System Interface

PHYSICAL CHARACTERISTICS

The processor is modularly designed with standard TTL MSI components and commercial construction.

The standard chassis, shown in figure 1-2, is 470 mm (18.5 in.) high by 448 mm (17.5 in.) wide by 305 mm (12 in.) deep. The chassis includes cooling fans and a front cover panel. The standard chassis back panel has the input/output wiring for the 1700 A/Q and 1700 A/Q-DMA. However, it may also contain specialized input/output for the user. Wiring details are included in the system wire list provided with the unit if specialized wiring is required.

Power requirements for the processor vary with the user's application. Power supplies are included in a separate chassis. Physical dimensions for the power supply chassis are 127 mm (5 in) high by 178 mm (7 in) wide by 380 mm (15 in) deep.

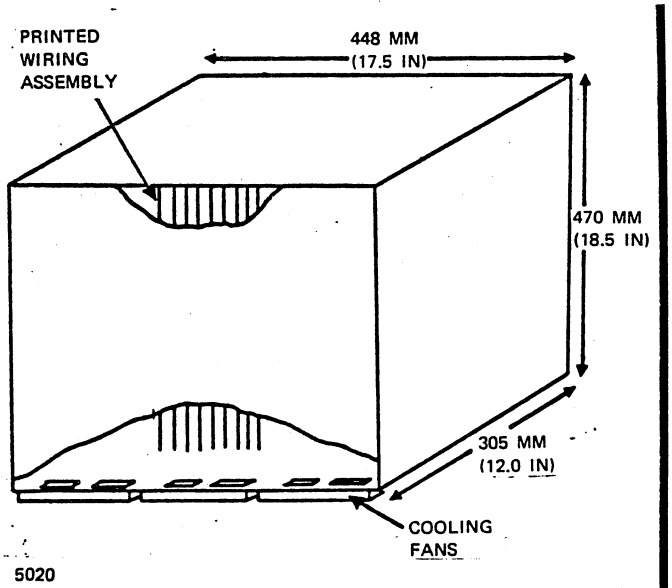
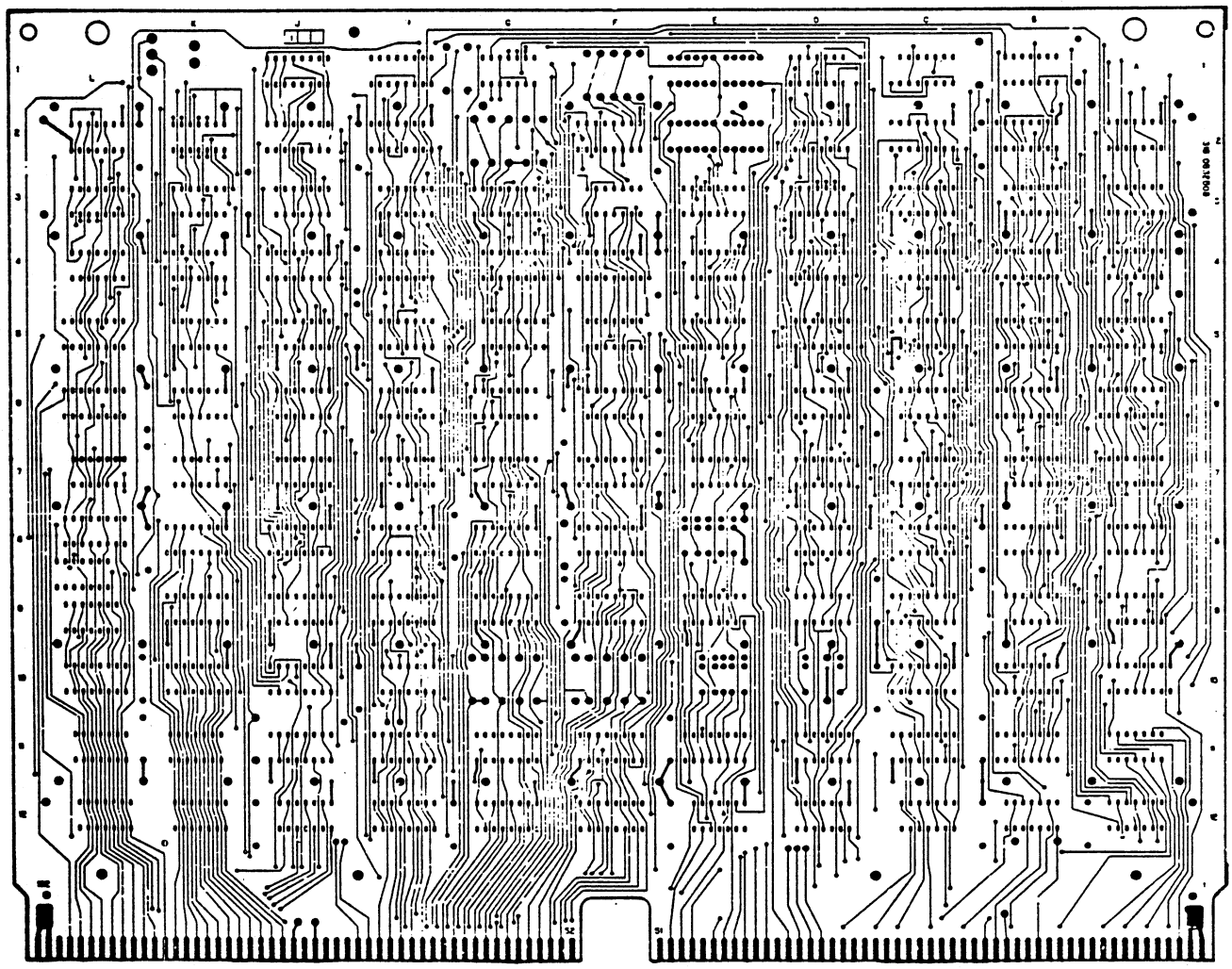


Figure 1-2. Standard Processor Chassis

A typical processor printed wiring assembly, shown in figure 1-3, is 279.4 by 355.6 mm (11 by 14 in) and has 204 input/output contacts.

The processor chassis has a prewired location for an optional breakpoint panel interface card. The breakpoint panel is a 406.4 mm (16 in) by 114.3 mm (4.5 in) printed circuit board, connected by a flexible cable to the breakpoint controller printed wiring assembly. The panel contains controls and light emitting diode indicators for manually controlling the processor at the micro level. The breakpoint controller printed wiring assembly also provides an interface to ASCII RS232-compatible consoles (full-duplex interface) for control of the processor. The console display normally attaches to the RS232 serial interface on the I/O-TTY module. A teletypewriter-compatible current loop interface is also available.

The processor operates in computer rooms, general offices, and industrial environments. It operates at temperatures of 4.5°C to 48.8°C (40°F to 120°F) withstands a maximum temperature gradient of 10°C (18°F) per hour at a rate that precludes condensation, and a relative humidity of 10 to 90 percent. Nonoperating environment extends the temperature range from -35°C to 65°C (-30°F to 150°F) and a maximum thermal gradient not to exceed 11°C (20°F) per hour or at a rate that precludes condensation. Storage temperatures with proper packaging protection may range from -51°C to 71°C (-60°F to 160°F) and relative humidity from 2 to 98 percent with temperature cycles of not more than 33°C (60°F) per hour or at a range that precludes condensation. The user should note that these ranges cover only the microprocessor; peripheral equipments may require more stringent environmental controls.



5021

Figure 1-3. Typical Processor Printed Wiring Assembly

MICROPROCESSOR

The enhanced processor consists of an arithmetic logical unit (ALU) printed wiring assembly, a status mode interrupt (SMI) printed wiring assembly, control 1 and 2 printed wiring assemblies an I/O TTY controller printed wiring assembly, and a 1700 transform printed wiring assembly. The microprocessor cards are interconnected through the basic backpanel wiring. Special user options require additional wiring. Figure 1-4 shows the chassis layout for the standard processor equipment.

TRANSFORM

The transform hardware is packaged as a separate printed wiring assembly and is specially designed for processor application. The processor has a 512-word, 64-bit read-only micro memory on the transform module.

Fuctioning as the hardware portion of the macro-instruction decode process, the transform causes the micro program to form program branches, set various parameters, and perform arithmetic or logical operations. It provides the micro program with the capability of selecting patterns of bits from the data transmission paths to form the micro-memory addresses that sequence the micro program.

MICRO MEMORY

The processor contains a 512-word micro memory on the 1700 transform board. It also has two printed wiring assembly slots for optional micro-memory or special algorithms if required by the user. The slots are interconnected to the microprocessor through the backpanel and are accessible only by the microprocessor.

MAIN MEMORY (MOS) AND MEMORY INTERFACE

The MOS main memory consists of MOS memory array modules and two interface modules. The memory array modules are configured in 16K or 32K increments of 18 bits: 1 parity, 1 protect, and 16 data bits. Data flow is in 16-bit word format, with a maximum of 131K words possible in the basic chassis. A direct memory access (DMA) channel is included in the memory interface as well as the parity and program protect generation and checking. The DMA for the processor can provide access for four external DMA devices through a port to main memory.

I/O-TTY CONTROLLER

The standard operator interface to the processor is through the I/O-TTY controller. It can interface with a Teletype Corporation Model ASR/KSR 33/35 Teletype or the CONTROL DATA RS232-C compatible console display. A TTL bus is available in the I/O-TTY module for interfacing the controllers in the main chassis to the microprocessor.

16K OR 32K MOS MEMORY ARRAY OR ERROR CHECKING AND CORRECTION ARRAY†	AC
16K OR 32K MOS MEMORY ARRAY†	Z
16K OR 32K MOS MEMORY ARRAY†	Y
16K OR 32K MOS MEMORY ARRAY†	X
MOS MEMORY ADDRESS/CONTROL INTERFACE†	W
MOS MEMORY DATA INTERFACE†	V
BREAKPOINT CONTROLLER (PANEL INTERFACE)†	U
2K MICRO MEMORY†	T
2K MICRO MEMORY†	S
TRANSFORM	R
CONTROL 1	P
CONTROL 2	N
ARITHMETIC/LOGICAL UNIT	M
STATUS/MODE AND INTERRUPT	L
I/O-TTY CONTROLLER	K
(A/Q) CARD READER/LINE PRINTER CONTROLLER†	J
(A/Q-DMA) SMD OR CDD CONTROLLER†	H
(A/Q-DMA) DUAL MODE MAG. TAPE CONTROLLER†	G
(A/Q) 8-CHANNEL OR 2-CHANNEL CLA†	F
(A/Q) TAPE CASSETTE OR FDD CONTROLLER†	E
(A/Q-DMA)†	D
(A/Q)†	C
(OPEN)†† SECONDARY BCLA†	B
(A/Q-DMA) PRIMARY BCLA†	A
(A/Q) PAPER TAPE READER/PUNCH CONTROLLER†	AA
(A/Q SPECIAL)††† NRZI MAG. TAPE CONTROLLER†	AB

†OPTIONAL PWA OR MODULE.
††NO I/O BUS CONNECTIONS. POWER AND GROUND ONLY.
†††SET/SAMPLE I/O CONNECTIONS ONLY.

0581-1

Figure 1-4. Standard Chassis Layout (Printed Wiring Assembly Placement)

EXTERNAL I/O INTERFACE

The main chassis for the processor includes 11 slots for external input/output devices (in addition to the input/output capability of the I/O-TTY module). Printed wiring assembly slot assignments in the processor chassis are shown in figure 1-4. Four slots are prewired for 1700 A/Q-DMA channels, and five slots are prewired for 1700 A/Q channels:

1700 A/Q-DMA channels: Slots A, D, G, H

1700 A/Q channels: Slots AA, C, E, F, J

These may be used with standard CDC equipment or for special user applications. Note that slot B is wired with power and ground only.

BREAKPOINT PANEL/BREAKPOINT CONTROLLER

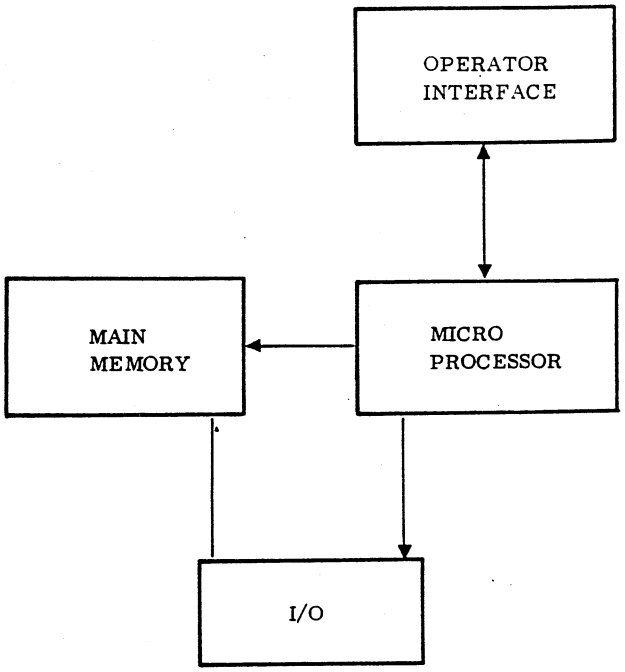
The breakpoint controller (panel interface) is an optional circuit module available for manual interface to the processor. The controller provides interfaces for a breakpoint panel (maintenance panel) or for an RS232-C compatible console that has full-duplex serial ASCII characteristics. A slot is prewired in the processor chassis for the breakpoint controller. Control and data lines tie directly into the control logic of the controller and to the arithmetic logical unit printed wiring assembly within the processor.

FUNCTIONAL DESCRIPTION

The micro-programmable processor emulates a CDC 1700 computer system. It can perform all 1700 functions, utilizing an expanded instruction set with interfacing capabilities to 1700 Series peripherals. Figure 1-5 shows a block diagram of the processor system. The basic processor configuration includes the microprocessor, main memory, input/output interface, and operator's interface. The flexible design of the system permits the user to incorporate his own equipment or to upgrade the processor with additional micro memory, the input/output capability, or a special hardware algorithm module.

MICRO PROCESSOR

The central processing unit (CPU) is a special configuration that consists of an arithmetic logical unit (ALU) module, a status mode interrupt (SMI) module, two control modules, and the standard processor transform module. Detailed processor organization is shown in figure 1-6. This diagram shows processor registers interconnected primarily by sectors. A sector is a multiplexer that transfers one of several inputs to an output. They are either 1, 8, 12, 16, or 32 bits wide.



0142

Figure 1-5. System Block Diagram

TRANSFORM

Transforms enable quick and efficient decoding of an emulated instruction. A transform can be designed to extract bits from a register or registers, shift the bits to the required position, and add a base address or constant bits. This result can then be transferred to the micro-memory address register (transform jump) or to the K or N register (transform register load). For example, when a 1700 instruction is read from main memory, one micro-instruction transform jump transfers control to one of 108 micro-memory locations. Without the transform features, the above operation requires many micro instructions.

The transform hardware is packaged in a separate module and is implemented using three selectors. The transform module includes 1,024 micro instructions (512 words) in read-only memory. The majority of these instructions are used to execute the 1700 emulator. The read-only memory also contains instructions for the panel interface simulation via the I/O-TTY printed wiring assembly.

ALU AND DATA TRANSFER ORGANIZATION

The arithmetic logical unit provides the arithmetic and logical capabilities of the processor. This unit combines two input words of the system word length. These two inputs are combined according to

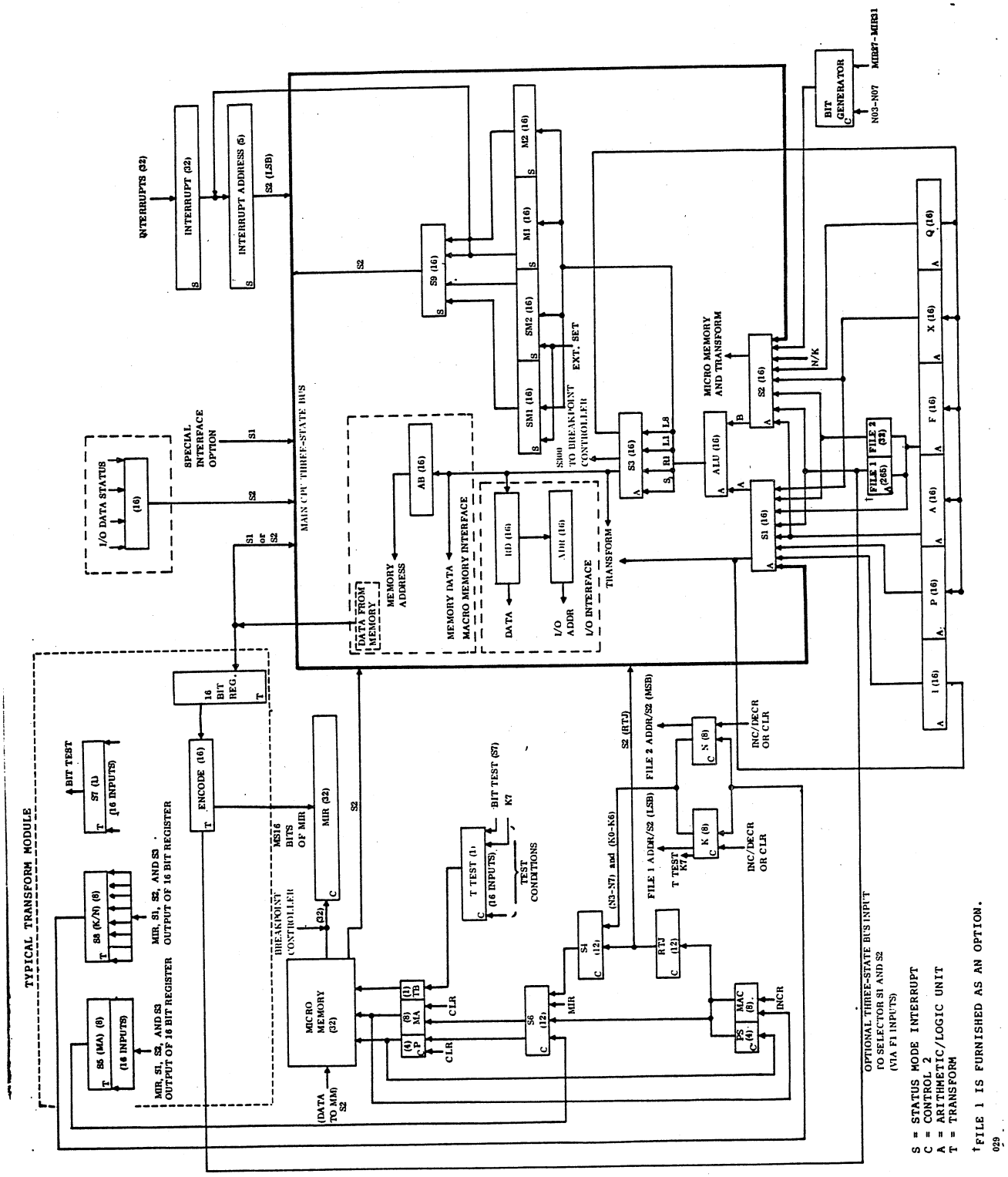


Figure 1-6. Detailed Block Diagram of Enhanced Processor

the function code specified in the micro instruction. The result is immediately available at the output of the arithmetic logical unit for possible shifting via selector S3 and delivery to the destination register, memory interface, panel interface, and input/output. The unshifted output of the arithmetic logical unit is delivered to the SM and mask registers. The arithmetic logical unit operation regarding sign, zero, and magnitude (by means of carryout test) are avilable to the test bit logic for instruction sequencing.

The data transfer organization of the processor provides for storing data in one of six working registers, and two files and for selecting data for processing through the arithmetic logical unit. Arithmetic logical unit results are transferred back to one of the registers or out of the organization to control external equipment.

The primary data registers are I, P, A, F, X, and Q.

The following are brief descriptions of the primary registers.

I Register - A word-length register whose only input and output is the selector S1. This register should not be confused with the 1700 I register (location $00FF_{16}$).

P Register† - A word-length, general-purpose register that receives data from the arithmetic logical unit and provides output to S1. Normally it is used to hold the software instruction counter.

A Register - A word-length, general-purpose register that receives data from the arithmetic logical unit and provides output to S1. The A register is mechanized as a shifting register and can be shifted left or right without using the arithmetic logical unit. The A register may also be combined with the Q register to form a double-length shifting register that operates independently of the arithmetic logical unit.

F Register - A word-length, general-purpose register that receives data from the arithmetic logical unit and provides data to S1 or S2 as arithmetic logical unit input. This register is also used as the file entry register and contains information written into the files when they are used as the destination of an arithmetic logical unit operation.

X Register - A word-length, general-purpose register that receives data from the arithmetic logical unit and provides data to S1 or S2.

Q Register† - A word-length, general-purpose register that receives data from the arithmetic logical unit and provides output to S2. 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 arithmetic logical unit.

Other major portions of the standard processor are:

File 2 - A 32-word scratchpad file that may be used as a general-purpose, word-sized register. It delivers its output to S1 and S2; data input is provided by the F register. File 2 is reserved for the emulator, except for registers R1, R2, R3, and R4, which are available to the 1700 programmer through enhanced instructions.

Bit Generator (BG) - The BG circuit generates one bit at any position in a word as input to the B side of the arithmetic logical unit. Control to drive the bit generator is derived from either the micro instruction (bits 27 to 31) or the lower five bits of the N register. Control is usually obtained from the micro instruction. A bit setting in an SM register determines the input that drives the bit generator.

Status/Mode Register (SM) - The SM register allows the micro program to control the mode of operation and allows the micro program to examine the status of certain internal and external conditions. The processor can access one of two SM registers, SM1 and SM2.

The SM register 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 micro program by transferring information to the SM register from the output of the arithmetic logical unit. Master clear also clears SM1 and SM2.

Interrupts and Mask Register - The interrupt system is implemented as a sampled data system at the micro-program level, instead of a true vectored interrupt system as used in conventional computers.

The mask register enables the processor to disable/enable interrupts. The processor can access two mask registers, M1 (micro interrupt) or M2 (macro interrupt). For each mask bit there is a corresponding bit in the interrupt register.

M1 is available to the 1700 programmer through the DMI instruction (micro level), while M2 (macro level, referred to as M) is available through the basic inter-register instruction (see section 4).

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

- Mask Register 1 (M1): M100 through M115

- Mask Register 2 (M2): M200 through M215

Interrupt addresses are generated by the interrupt address encoder, according to the assignments given in table 1-2.

†Available to the 1700 programmer.

TABLE 1-2. MASK REGISTER/INTERRUPT ADDRESS

Mask Bit	Interrupt Address Mask Register 1
M100	15 Lowest Priority (M1)
M101	14
M102	13
M103	12
M104	11
M105	10
M106	09
M107	08
M108	07
M109	06
M110	05
M111	04
M112	03
M113	02
M114	01
M115	00 Highest Priority (M1)
	Interrupt Address Mask Register 2
M200	31 Lowest Priority (M2)
M201	30
M202	29
M203	28
M204	27
M205	26
M206	25
M207	24
M208	23
M209	22
M210	21
M211	20
M212	19
M213	18
M214	17
M215	16 Highest Priority (M2)

Note: The interrupt address generated is the same as its priority level; i.e., the highest priority interrupt generates a 0 interrupt address and the lowest priority interrupt generates a 31 interrupt address.

The interrupt priorities correspond to the interrupt address generated; that is, interrupt address 00 is associated with the highest priority interrupt line, and interrupt address 31 is associated with the lowest priority linterrupt line. For example, an interrupt associated with M112 has priority over an interrupt associated with M111, and an interrupt address of 3 is developed by the interrupt address encoder.

K Register - An 8-bit counter that may be cleared, incremented, or decremented. It is used to address file 1 in addition to any program usage as a counter.

N Register - An 8-bit counter that may be 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 that controls micro-instruction execution.

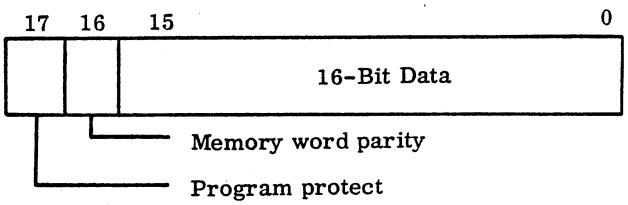
N/K Register - The N and K registers may be combined to provide operand addresses outside the current operating micro page.

File 1 - A file of 256 general-purpose, word-sized registers that are addressed by the contents of the K register. The output of the addressed file is delivered to S1 and S2 and thus to the A and B side of the arithmetic logical unit on demand. The file 1 input to selectors S1 and S2 is a submultiplexed input to the arithmetic logical unit. Thus, depending on the state of status mode bit (SM111), either file 1 or transform data can be selected as either an A or B input to the arithmetic logical unit.

MAIN MEMORY

Main memory for the processor consists of 16K or 32K MOS semiconductor memory array modules, a data interface module, and an address and control interface module. The two interface modules provide the control and interfacing required for the processor/memory function and peripheral (DMA) equipment/memory/functions.

The MOS memory words are in 18-bit format:



0575

The parity and program protect bits are generated and tested in the interface modules. One set of interface modules can handle up to four 32K MOS memory array modules for a total of 131K words in the main processor chassis. In addition, an external memory bank, located in a second processor chassis, may be accessed by the processor for a total of 262K words of addressable memory.

If the error checking and correction option is installed, an additional 5 bits per word are stored. These ECC bits are stored on a separate memory array module. With ECC, memory is limited to 98K words in the main processor chassis.

The mimimum processor memory cycle time is 600 nanoseconds, which is defined as the shortest possible time between successive read operations in main memory. The minimum processor main memory cycle time is 700 nanoseconds for write operations.

The main memory configuration is shown in figure 1-7.

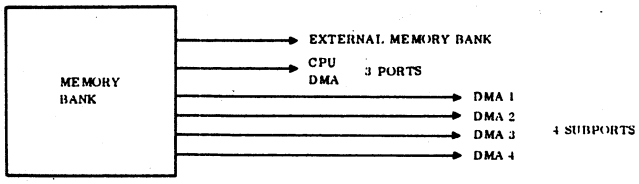
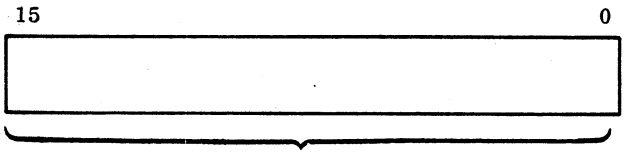


Figure 1-7. Main Memory Configuration

The MOS memory configuration (for 16K to 131K) is a one-bank, three-port memory. One bank signifies that only one reference may take place at one time. Three ports provide three independent data and control paths to the memory; any port may request memory independent of any operation underway on the other ports. The ports are CPU, DMA (direct memory access), and the external memory bank port.

Main memory is addressed in 16-bit format, as shown in the following. Only the first 65K words are addressable in absolute mode. In page mode, all 131K words of MOS memory in the main processor chassis plus all 131K words of the external bank are addressable.

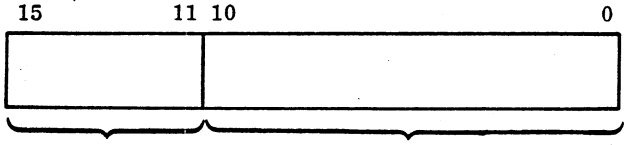
Absolute Mode Addressing



16-bit address

0000_{16} to $FFFF_{16}$ (First 65K words only)

Page Mode Addressing



5-bit page address (32 pages)

11-bit word address (2048 words/page)

0574

I/O-TTY CONTROLLER

Figure 1-8 illustrates major signal flow paths to and from the I/O-TTY controller.

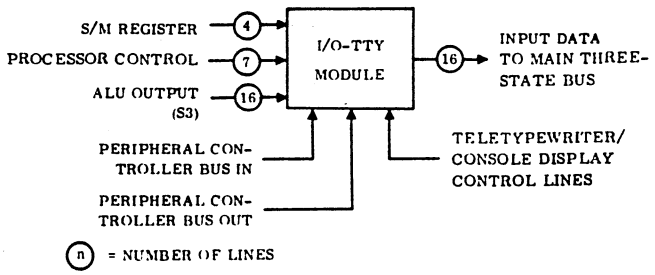


Figure 1-8. Major I/O-TTY Signal Flow Paths

This module provides the following functions:

Real-time clock - In conjunction with the micro code, it appears as a 1700 peripheral to the macro-level programmer.

I/O teletypewriter/display control - This controller is an integral part of the module. It interfaces to Teletype Corporation ASR/KSR 33/35 Teletypes and to the CONTROL DATA RS232-C compatible conversational display terminals.

Internal peripheral controller bus - Provides all input/output data lines, interrupts, and control signals necessary to generate, in conjunction with the micro code, an internal CDC 1700 A/Q (input/output) bus. This TTL-level bus is intended to interface with controllers located in the basic processor chassis.

Panel interface simulation - A logic section that is required when a panel/program device is used for operator input in the panel mode.

The processor is interfaced to the input/output module as follows:

ALU output - All output data and address information is provided from the output of the arithmetic logical unit via S3.

SM register - All commands to peripheral controllers are generated by micro code manipulation of the processor status mode register.

Processor control - Timing and control information for controlling internal input/output module data gating is provided from the processor control signals.

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

Input data and peripheral response signals - All of these are provided to the processor on the main processor three-state bus.

Real-time clock - An integral part of the input/output module, the real-time clock appears as a 1700 peripheral to the macro-level software. Two functions are available to the macro-level program: enable limit interrupt and disable limit interrupt. Two status bits are also available to the macro-level program: limit interrupt and lost count.

Figure 1-9 is a functional block diagram of the I/O-TTY controller. Inputs to the I/O-TTY controller consist of address and data words, micro and macro code interrupts from the CPU, keyboard codes from the TTY/CD, and data/status from the peripheral I/O controllers. The controller either outputs a data/function word to the appropriate peripheral I/O controller or processes the inputs and sends data, status, and real-time clock interrupts, or the peripheral I/O controller data/status to the CPU via the three-state bus. Display data is also sent to the TTY/CD via the associated transmission line.

Address and data words are received by the input registers from the CPU arithmetic/logical unit (ALU) (selector S3) under control of the control gate signal. All words from the CPU A and Q registers are gated into the D register of the I/O-TTY controller by the gate I/O data signal. If a word contains address information, it is gated from the D register into the Y register under control of the gate I/O address signal. The data and status (A register) words carry data and director function information to the I/O controllers and data and director status from the I/O controllers. The address (Q register) words utilize the WES/D convention for addressing input/output data between the CPU and the peripheral I/O controllers.

The NCR M05 set/sample scheme provides eight-port addressing and eight-level priority assignment. Each port, designated 1 through 8, can communicate with an associated peripheral device. It may also be multiplexed to communicate with up to eight peripheral devices, so that up to 64 peripheral devices can be controlled. Control of NCR equipment is implemented with the M05 set/sample I/O instructions that are included in the enhanced instruction set of the microprocessor.

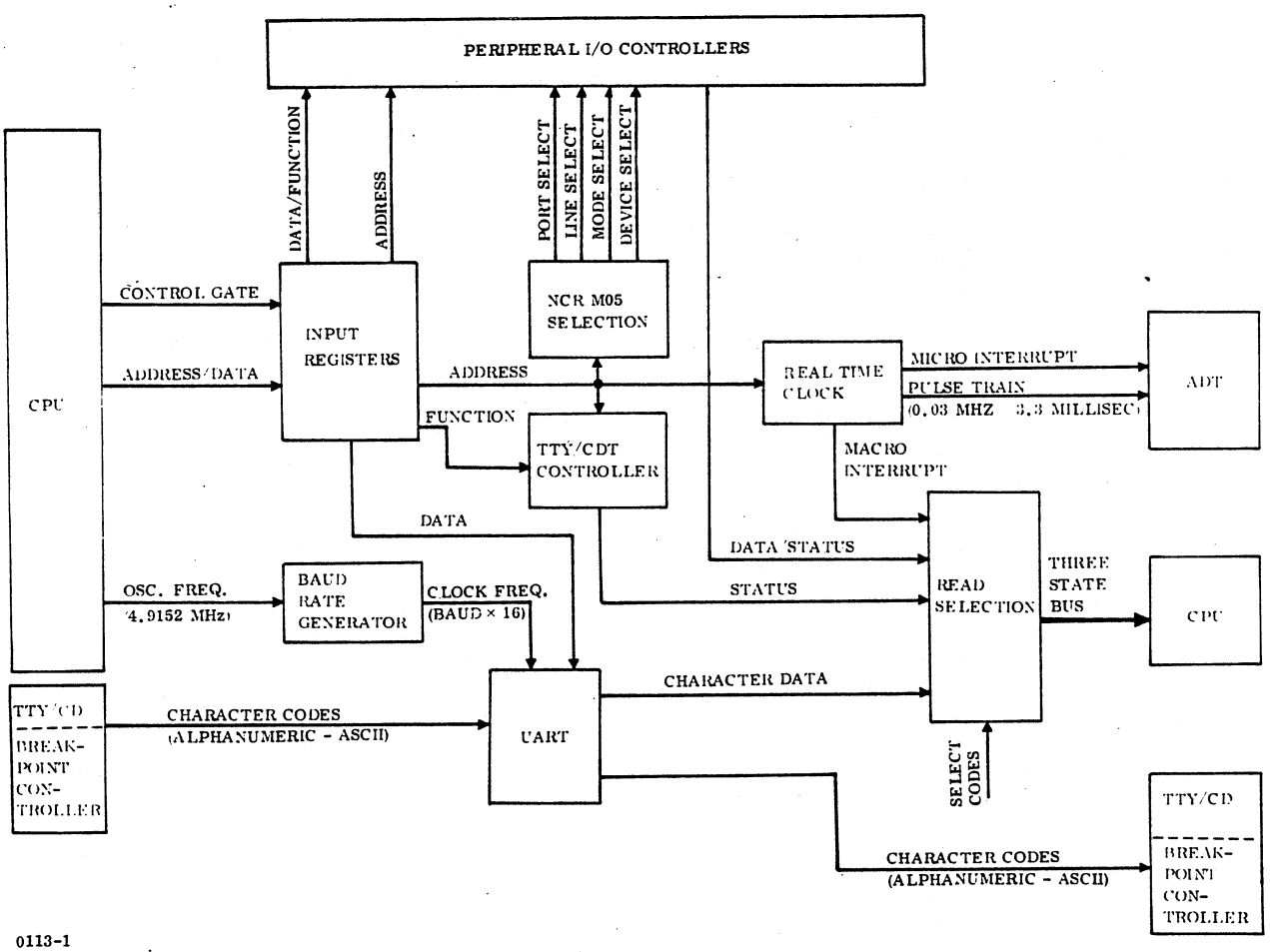


Figure 1-9. I/O-TTY Controller Functional Block Diagram

The contents of the address word are deciphered by the internal I/O-TTY controller logic or are applied directly to the peripheral I/O controllers. If the address selects an internal function, the I/O-TTY controller processes the data present in the D register. If the address selects a peripheral I/O controller, the data present in the D register is processed by the addressed controller. If the address selects an NCR M05 controller, the address word is processed to determine the selected port, line, mode, and device. Depending upon the mode selected, either the data present in the D register is transferred to the appropriate M05 I/O device, or the data/status word is coupled through the read selection logic to the three-state bus to the CPU.

Internal I/O-TTY controller functions include a real-time clock (RTC) and a communication interface. When the real-time clock function is initiated, a macro interrupt advises the CPU that RTC has been selected and an auto-data transfer (ADT) routine is established. The CPU macro code is not to be involved in the real-time clock sequence until the selected elapsed time limit has been reached or the real-time clock has malfunctioned (lost count). The ADT routine receives the real-time clock pulse train (0.03 MHz) and processes the pulses to determine when the elapsed time limit has been reached. This real-time clock sequence is carried out independently of the CPU except for the macro interrupts.

When the communication interface is selected, the character codes to and from the UART enable the conversion of keyboard characters to machine language and of machine language to character display. All CPU character data to and from the UART is transferred in parallel format, and all character codes (ASCII) are transferred between the TTY/CD and/or the breakpoint controller in serial format. Data is clocked in and out of the UART under control of the baud rate generator; the serial data is clocked at the rate of one bit every 16 baud rate clock cycles.

REFERENCE DATA

ELECTRICAL CHARACTERISTICS

Logic Levels

TTL - Low = 0 volts, high = +5 volts

RS232-C - Excursion between $\pm$3 to $\pm$12 volts

Transmission Characteristics

Teletypewriter - 20 mA current loop

Console display - RS232-C

Panel interface - TTL

CPU to I/O - 90 kHz maximum

Baud rates - 110, 300, 1200, 9600

PHYSICAL CHARACTERISTICS

Logic package - Dual in-line package (DIP)

Transistor package - T05 can

Modularity - Printed circuit card, four-layer

Printed circuit board size - 279 by 356 mm (11 by 14 inches)

ENVIRONMENTAL CONDITIONS

Operational:

Termperature - 4.4°C to 48.4°C
(40°F to 120°F)
(Maximum thermal shock
0.1°C (0.2°F) per min.)

Relative humidity - 10 to 90 percent

OPERATION 2

This section contains a description of the processor system operation. For a complete description of the formats and functions of the processor micro instructions, refer to the associated processor reference manual.

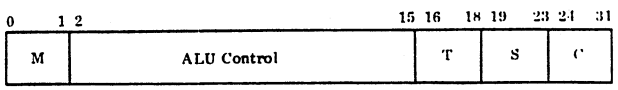
The operating procedures for the processor are described in general terms only, since each use has a different configuration and application for the equipment. The micro memory and macro memory can be loaded via a display console or breakpoint panel and breakpoint controller. Both micro memory and macro memory can also be loaded more conveniently via any peripheral device that has a controller with deadstart capability.

The I/O-TTY controller provides a communication interface between a teletypewriter or a console display and the CPU. Communication rates vary with the device and the application, so the rate select switch on the I/O-TTY controller must be set to the correct rate for the device attached (deadstart for deadstart subsystem installed, and program for display console installed). The basic rate is 9600 baud, but a rate of 110, 300, or 1200 baud may be selected. Refer to the installation manual for complete details.

The procedures to read/write micro memory, read/write macro memory, and enter or display any register are given in section 3.

MICRO INSTRUCTIONS

Each micro-memory address specifies the location of two micro instructions (upper and lower). Each micro instruction (32-bits) is divided into five basic fields (M, ALU, T, S, and C) and is numbered from left to right as bits 0 through 31 as shown below. These five basic fields are defined in table 2-1.



5025

The M field specifies the addressing mode (return, sequential, or jump) used to obtain the next micro-instruction pair from micro memory. This field also specifies the format (one of three) for interpreting the S and C fields (figure 2-1 and table 2-2).

The arithmetic/logical unit control field comprises two subordinate field formats: arithmetic/logical and shift/scale.

The arithmetic/logical subfields (F, A, B, and D) specify function, A source, B source, and destination of the arithmetic or logical operands. The shift/scale subfields (F, R, L, A/AQ, SC) specify the interpretation of the arithmetic/logical operation subfields A and B.

TABLE 2-1. BASIC MICRO-INSTRUCTION FIELDS

Bit Position	Field Definition
0 and 1	The M (mode) field specifies the addressing used in obtaining the next micro-instruction pair from micro memory; it also specifies the format used in interpreting and S and C fields (bits 19 through 31) of the micro instruction.
2 through 15	The arithmetic/logical unit control field specifies the arithmetic and logical unit (ALU) operation code, source operands, and destination of the operation result.
16 through 18	The T (test) field specifies the method of selecting which micro instruction (upper or lower) of the next micro-instruction pair to execute.
19 through 23	The S (special) field specifies subformat selection (bit 19) and special operation codes (bits 20 through 23).
24 through 31	The C (constant or suboperation) field specifies constant (bit 24), jump transform, or micro operation.

Subfield F specifies shift or scale operation. Subfield F (bits 2 through 6) also selects shift or scale operation. An F value of 1E (hex) specifies shift, and value 1F (hex) specifies scale. During shift and scale operations, the N register designates the number of shift and scale bits. Subfields R and L specify right shift or left shift. Subfield A-A/Q specifies whether the A register alone or the A and Q registers together are to be shifted or scaled. Subfield SC specifies the shift control code: enter 0, enter 1, extend sign, or end-around-carry.

The T field tests for the conditional branch of the micro instruction. It specifies which micro instruction, upper or lower, or the next micro-instruction pair to execute. The determination may be based on the result of an arithmetic/logical unit operation of a current micro instruction or on the condition of the K and N registers or interrupts.

The codings in the S and C fields depend upon the contents of the M field. These fields are coded in three formats (figure 2-1).

FORMAT 1

When the M field specifies return or sequential, format 1 is selected. The S field specifies macro-memory read or write operations and alternate codings employed in the A, B, and D fields. Bit 24 (T/T′) specifies that the code in the T field is to be interpreted as the normal T code or as the alternate T′ code. Bit 19, the subformat select bit, determines whether bits 25 through 31 are to be interpreted as C′ and C′′ codes (table 2-2).

FORMAT 2

When the M field specifies jump, format 2 is selected. If a jump to a micro-instruction pair within the same micro-memory page is specified, the subformat select bit (bit 19) is 0. Bits 20 through 23 contain special operation codes, and bits 24 through 31 contain the micro-memory address of the next micro-instruction pair. If a jump to a micro instruction pair on a different micro-memory page is specified, the subformat select bit is a 1. Bits 20 through 31 contain the complete micro-memory address of the next micro-instruction pair.

FORMAT 3

When the M field specifies sequential, format 3 is selected. This format allows one special operation (as specified by the S code) to be performed. This also causes the contents of the C field to be transferred to the N register if bit 19 is 1, or to the K register if bits 19 is 0.

EMULATION

If the basic processor is used to emulate a CDC 1700 Series digital computer, refer to the 1700 Enhanced Processor Reference Manual for details of instruction repertoire, addressing, and interrupts. For emulation of other types of digital computers, refer to the appropriate reference manuals for software instructions.

If the processor is to be used to emulate a computer, the emulator may be contained in read only memory (ROM) or may be deadstart loaded into RAM micro memory.

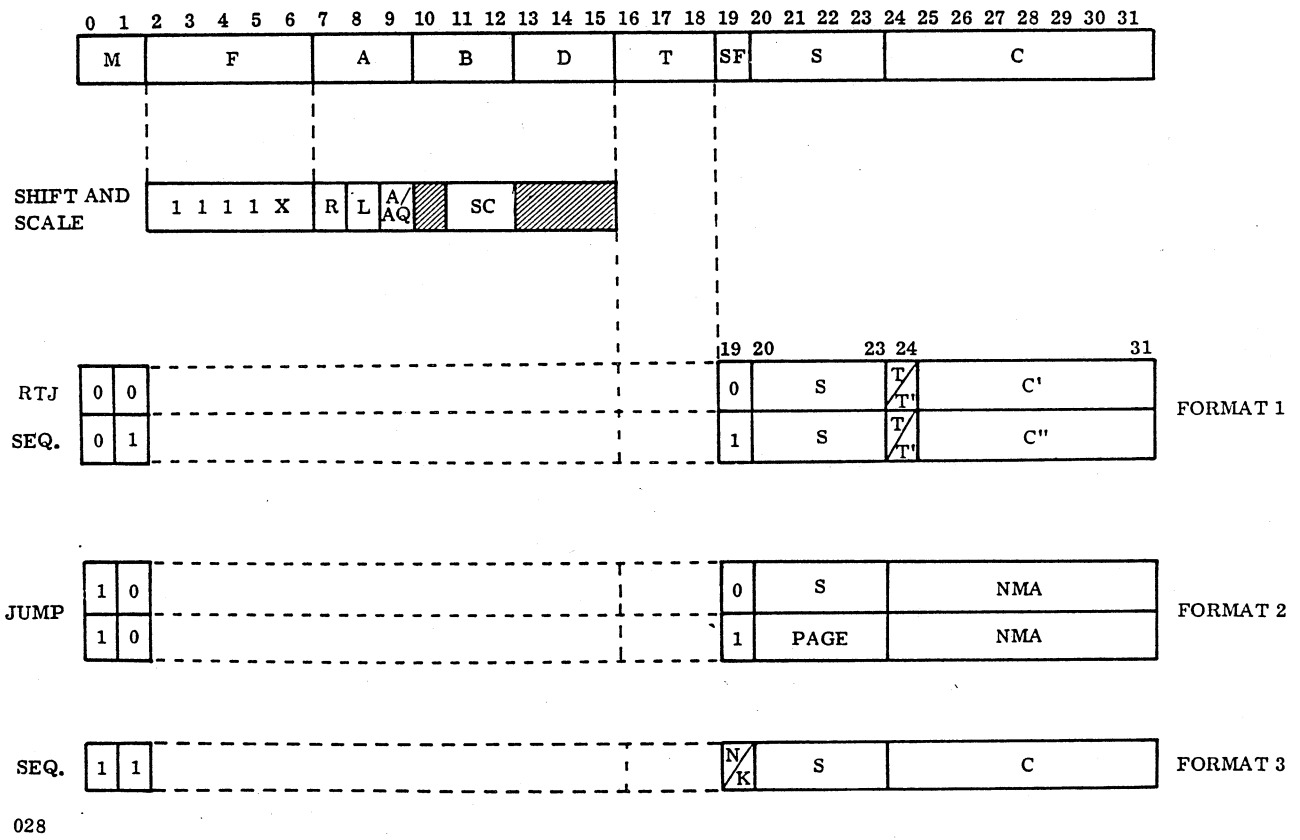


Figure 2-1. Micro-Instruction Formats

TABLE 2-2. MICRO CODE SUMMARY

ADDRESSING MODES

M	Addressing Mode	Format
0 0	Return	1
0 1	Sequential	1
1 0	Jump	2
1 1	Sequential	3

ARITHMETIC OPERATIONS

F Code	Mnemonic
1 0 1 0 0	SUB
1 0 1 0 1	SUBT
1 0 1 1 0	SUB-, SUB-C
1 0 1 1 1	SUB-T, SUB-TC
1 1 0 0 0	ADD
1 1 0 0 1	ADDT
1 1 0 1 0	ADD+
1 1 0 1 1	ADD+T

LOGICAL OPERATIONS

F Code	Mnemonic
0 0 0 0 0	-A
0 0 0 0 1	-A + -B
0 0 0 1 0	-A + B
0 0 0 1 1	ONE
0 0 1 0 0	-A • -B
0 0 1 0 1	-B
0 0 1 1 0	-EOR
0 0 1 1 1	A + -B
0 1 0 0 0	-A • B
0 1 0 0 1	EOR
0 1 0 1 0	B
0 1 0 1 1	A + B
0 1 1 0 0	ZERO
0 1 1 0 1	A • -B
0 1 1 1 0	A • B
0 1 1 1 1	A

SHIFT INSTRUCTIONS
F = 1 1 1 1 0; (N) = Number of Shifts

Bit Code					Mnemonic	Operation
7	8	9	11	12		
1	0	0	0	0	AR0E	Shift A right; enter 0 at most significant bit
1	0	0	0	1	ARSE	Shift A right; with sign extension
1	0	0	1	0	AREA	Shift A right; with end-around carry
0	1	0	0	0	AL0E	Shift A left; enter 0 at least significant bit
0	1	0	0	1	AL1E	Shift A left; enter 1 at least significant bit
0	1	0	1	0	ALEA	Shift A left; with end-around carry
1	0	1	0	0	AQR0E	Shift A/Q right; enter 0 as most significant bit in A
1	0	1	0	1	AQRSE	Shift A/Q right; with sign extension
1	0	1	1	0	AQREA	Shift A/Q right; with end-around carry
0	1	1	0	0	AQL0E	Shift A/Q left; enter 0 at least significant bit of Q
0	1	1	1	0	AQLEA	Shift A/Q left; with end-around carry

TABLE 2-2. MICRO CODE SUMMARY (Contd)

SCALE INSTRUCTIONS
F = 1 1 1 1 1

Bit Code		Mnemonic	Operation
7 8 9	11 12		
0 1 0	0 0	SL0E	Scale A left; enter 0 as least significant bit
0 1 0	0 1	SL1E	Scale A left; enter 1 as least significant bit
0 1 0	1 0	SLEA	Scale A left; with end-around carry
0 1 1	0 0	SDL0E	Scale A/Q left; enter 0 as least significant bit in Q
0 1 1	0 1	SDLEA	Scale A/Q left; with end-around carry

A CODE

A Code	Mnemonic	ALU (A) Source Data	A Code	Mnemonic	ALU (A) Source Data
0 0 0	F2	File 2 register	1 0 0	A	A register
0 0 1	P	P register	1 0 1	F	F register
0 1 0	I	I register	1 1 0	F1	File 1 register
0 1 1	X	X register	1 1 1	MEM	Main (macro) memory read

A′ CODE

A′ Code	Mnemonic	ALU (A) Source Data	A′ Code	Mnemonic	ALU (A) Source Data
0 0 0	SM1	SM1 register			times with end-around carry (A register remains unchanged)
0 0 1	M1	Interrupt mask 1 register	1 0 1	A*	Double precision A register
0 1 0	SM2	SM2 register			
0 1 1	M2	Interrupt mask 2 register	1 1 0	X*	Double precision X register
1 0 0	A*R8	Double precision A register shifted eight	1 1 1	Q*	Double precision Q register
A′ when S = 0 1 1 1 or 1 0 1 0					

TABLE 2-2. MICRO CODE SUMMARY (Contd)

B CODE

B Code	28 29	Mnemonic	ALU (B) Source Data
0 0 0		F2	File 2 register
0 0 1	1 1	ZERO	All zeroes
0 0 1	1 0	N	N register
0 0 1	0 1	K	K register
0 0 1	0 0	N, K	N and K registers
0 1 0		BG	
0 1 1		X	X register
1 0 0		Q	Q register
1 0 1		F	F register
1 1 0		F1	File 1 register
1 1 1		MEM	Main (macro) memory

B′ CODE

B′ Code	Mnemonic	(ALU (B) Source Data) Operation
0 0 0	OPEN	
0 0 1	NRTJ	Complement of RTJ registers to 12 LSBs of selector S2; transfer 1's to four most significant bits of selector 32
0 1 0	INRD	Input data/status from input/output channel
0 1 1	INRS	Input to selector S2 response signals from input/output channel
1 0 0	MMU	Micro-memory upper 16 bits to X register, F field designates B source
1 0 1	MML	Micro-memory lower 16 bits to X register, F field designates B source
1 1 0 or 1 1 1	INTA	Interrupt address encoder
B′ when S = 1 0 0 0		

TABLE 2-2. MICRO CODE SUMMARY (Contd)

D CODE

D Code	Mnemonic	Destination
0 0 0	NOP	None
0 0 1	P	Selector S3 output to registers P, AB (macro-memory address buffer register)
0 1 0	I	Selector S1 output to registers I, AB
0 1 1	Q	Selector S3 output to registers Q, AB
1 0 0	F1	Selector S3 output to registers F, AB; write data in file 1 register at address specified by K register.
1 0 1	A	Selector S3 output to registers A, AB
1 1 0	X	Selector S3 output to registers X, AB
1 1 1	F	Select S3 output to registers F, AB

D′ CODE

D′ Code	Mnemonic	Destination
0 0 0	IOD	Selector S3 output to I/O-TTY data register
0 0 1	IOA	Selector S3 output to I/O-TTY address register via I/O-TTY data register
0 1 0	MMU	Selector S2 output to micro-memory upper
0 1 1	MML	Selector S2 output to micro-memory lower
1 0 0	M1	ALU output to mask 1 register
1 0 1	SM1	ALU output to SM1 register
1 1 0	M2	ALU output to mask 2 register
1 1 1	SM2	ALU output to SM2 register
D′ when S = 1 0 0 1 or 1 0 1 0		

TABLE 2-2. MICRO CODE SUMMARY (Contd)

D" CODE

D" Code	Mnemonic	Destination
0 0 0	NOP	None
0 0 1	A*LHW	Output of selector S1 to double-precision A* register, shifted left one-half word with end-around carry
0 1 0	X*LHW	Output of selector S1 to double-precision X* register, shifted one-half word with end-around carry
0 1 1	Q*LHW	Output of selector S1 to double-precision Q* register, shifted one-half word with end-around carry
1 0 0	NOP	None
1 0 1	A*	Output of selector S1 to double-precision A* register
1 1 0	X*	Output of selector S1 to double-precision X* register
1 1 1	Q*	Output of selector S1 to double-precision Q* register
D" when S = 1 0 1 1		

DD" CODE

DD" Code	Mnemonic	Destination
1 0 1	AA*	Output of selector S3 to A register and output of selector S1 to double-precision A* register
1 1 0	XX*	Output of selector S3 to X register and output of selector S1 to double-precision X* register
1 1 1	FQ*	Output of selector S3 to F register and output of selector S1 to double-precision Q* register
DD" when S = 0 0 0 1		

TABLE 2-2. MICRO CODE SUMMARY (Contd)

T CODE

T′Code	Mnemonic	Test
0 0 0	*L	Execute the lower micro instruction of this micro-instruction pair
0 0 1	U	Execute the upper micro instruction of the next micro-instruction pair
0 1 0	L	Execute the lower micro instruction of the next micro-instructio pair
0 1 1	KZU	If initial K register contents are zero, execute the upper micro instruction of the next micro-instruction pair
1 0 0	NZU	If initial N register contents are zero, execute the upper micro instruction of the next micro-instruction pair
1 0 1	INTU	If the interrupt and interrupt mask are coincidental, execute the upper micro instruction of the next micro-instruction pair
1 1 0	NU	If the arithmetic/logical unit output sign bit is negative on completion of this micro instruction, execute the upper micro instruction of the next micro-instruction pair
1 1 1	ZL	If the arithmetic/logical unit output is zero on completion of this micro instruction, execute the lower micro instruction of the next micro-instruction pair
T when MIR24 = 0		

T" CODE

T′Code	Mnemonic	Test
0 0 0	*L	Execute the lower micro instruction of this micro-instruction pair
0 0 1	LQL	Execute the lower micro instruction of the next micro-instruction pair if the least significant bit of Q register is 1 at start of the micro instruction
0 1 0	K7L	Execute the lower micro instruction of the next micro-instruction pair if the least significant bit of the K register is set
0 1 1	OVFL	Execute the lower micro instruction of the next micro-instruction pair if overflow exists
1 0 0	BTU	Execute the upper micro instruction of the next micro-instruction pair if the least significant bit of the C field is 1
1 0 1	LQ*L	Execute the lower micro instruction of the next micro-instruction pair if the least significant bit of the double-precision Q register is 1 at the start of this micro instruction
1 1 0	COL	Execute the lower micro instruction of the next micro-instruction pair if carry-out results from arithmetic/logical unit arithmetic operation
1 1 1	Z*L	Execute the lower micro instruction of the next micro-instruction pair if the output of the arithmetic/logical unit is 0 at the completion of this instruction
T′ when MIR24 = 1		

TABLE 2-2. MICRO CODE SUMMARY (Contd)

S CODE

S Code	Mnemonic	Command
0 0 0 0	NOP	None
0 0 0 1	DDPP	Alternate D field coding (DD")
0 0 1 0	RPT	Repeat the current micro-instruction pair if the N register is not equal to 0
0 0 1 1	READ	Read the word from macro memory at the location contained in the address buffer
0 1 0 0	WRITE	Write selector S3 data in macro memory at the location contained in the address buffer
0 1 0 1	L8EA	Left shift eight bits end-around of selector S3
0 1 1 0	F2WR	Write F register data into file 2 at the location specified by the N register at the beginning of this micro instruction
0 1 1 1	AP	Alternate A field coding (A')
1 0 0 0	BP	Alternate B field coding (B')
1 0 0 1	DP	Alternate D field coding (D')
1 0 1 0	APDP	Alternate A and D field coding (A' and D')
1 0 1 1	DPP	Alternate D field coding (D")
1 1 0 0	GATEI	Gate selector S1 output to I register
1 1 0 1	HALT	Stop operation of the processor at the completion of this micro instruction if the halt bit of the SM register is 1
1 1 1 0	RTJ	Transfer the address of the next sequential micro-instruction pair to the RTJ register
1 1 1 1	CLRNP	Clear the N and P registers

FORMAT 1
BIT 19 = 0

C' Code	Mnemonic	Action
0 0 X X X X X		
0 1 0 0 0 0 0	WRCH/0	Write character (eight bits) at the output of selector S3 at the main memory location specified by the address buffer
0 1 0 0 0 0 1	WRCH/1	Write character (eight bits) at the output of selector S3 at the main memory location specified by the address buffer
0 1 0 0 0 1 0	WRCH/2†	Write character (eight bits) at the output of selector S3 at the main memory location specified by the address buffer
0 1 0 0 0 1 1	WRCH/3†	Write character (eight bits) at the output of selector S3 at the main memory location specified by the address buffer

† Applicable to 32 bit processor only

TABLE 2-2. MICRO CODE SUMMARY (Contd)

FORMAT 1 (Contd)
BIT 19 = 0

C′ Code	Mnemonic	Action
0 1 0 0 1 0 0	RMW	Read modify write, read data from macro memory, modify and write new data at the same address. Requires software instruction to complete the write operation.
0 1 0 0 1 0 1	WRHW0	Write (bits 0 through 15) the output of selector S3 at the memory location specified by the address buffer
0 1 0 0 1 1 1	WRHW1	Write (bits 16 through 31) the output of selector S3 at the memory location specified by the address buffer
0 1 0 1 0 0 0	WRPB	Write the protect bit
0 1 1 X X X X	GATEIXT	Open for special application at T4 strobe
1 0 0 0 1 0 1	INCK	Increment the contents of the K register by 1
1 0 0 1 1 0 1	INCN	Increment the contents of the N register by 1
1 0 0 0 1 0 0	DECK	Decrement the contents of the K register by 1
1 0 0 1 1 0 0	DECN	Decrement the contents of the N register by 1
1 0 0 0 0 0 0	CLRK	Clear the K register
1 0 0 1 0 0 0	CLRN	Clear the N register
1 0 1 X X X X	SETF/j	Set SM register flag j; XXXX is the value of j (0 to 15)
1 1 0 X X X X	CLRF/j	Clear the SM register flag; XXXX is the value of j (0 to 15)
1 1 1 0 0 0 0 or 1 1 1 0 0 0 1	RQLXN	The Q register and one destination register (P, A, F, or X) provide a double-length register. The combined register is shifted left one bit position. The arithmetic/logical unit sign bit complement is entered at the least significant bit of the Q register.
1 1 1 0 0 1 1	RQR1E	Shift the combined destination and Q registers right one bit; enter 1 (0) in the sign position of destination register.
1 1 1 0 0 1 0	RQR0E	Shift the combined destination and Q registers right one bit; enter 1 (0) in the sign position of destination register.
1 1 1 0 1 0 0	RL0E	Shift the destination register left one bit; enter 1 (0) in the lowest bit position.
1 1 1 0 1 0 1	RL1E	Shift the destination register left one bit; enter 1 (0) in the lowest bit position.
1 1 1 0 1 1 0	RR0E	Shift the destination register right one bit; enter 1 (0) in the lowest bit position.
1 1 0 1 1 1 1	RR1E	Shift the destination register right one bit; enter 1 (0) in the lowest bit position.

TABLE 2-2. MICRO CODE SUMMARY (Contd)

FORMAT 1
BIT 19 = 1

C" Code	Mnemonic	Action
0 0 0 X X X X	TMA/j	Obtain the next micro-instruction pair from the main memory location sepcified by MA transform selector setting j; XXXX is the value of j (0 to 15).
0 0 1 X X X X	TMAK/j	Obtain the next micro-instruction pair from the main memory location specified by MA transform selector setting j; set the K register to the value specified by K transform selector setting j. XXXX is the value of j (0 to 15).
0 1 0 0 X X X	GITMAK/j	Gate the output of macro memory to the IXT register and perform the TMAK/j operation; XXX is the value of j (0 to 7).
0 1 0 1 X X X	GITMAK/xt	Gate the output of macro memory to the IXT register. Perform transform on the upper 16 bits of the micro- instruction register (MIR).
0 1 1 X X X X	TK/j	Set the K register to the value specified by K transform selector setting j: XXXX is the value of j (0 to 15).
1 0 0 X X X X	TN/j	Set the N register to the value specified by N transform selector setting j: XXXX is value of j (0 to 15).
1 0 1 X X X X	SUB Requires program protect	Set the upper bounds; transfer the output of selector S3 to the upper bounds register (XXXX) in macro memory.
1 1 0 X X X X	SLB	Set the lower bounds; transfer the output of selector S3 to the lower bounds register (XXXX) in macro memory.

FORMAT 2

C Field	Action
X X X X X X X X	Address of the next instruction. If bit 19 = 1, then S = page (page jump)

FORMAT 3

C Field	Action
X X X X X X X X	If bit 19 = 0 transfer the C-field value (bits 24 to 31) to the K register. If bit 19 = 1, transfer the C-field vlaue (bits 24 to 31) to the N register

INSTALLATION 3

Refer to the CYBER 18 site planning kit and installation manuals for detailed installation instructions for processor systems and peripheral subsystems.

THEORY OF OPERATION 4

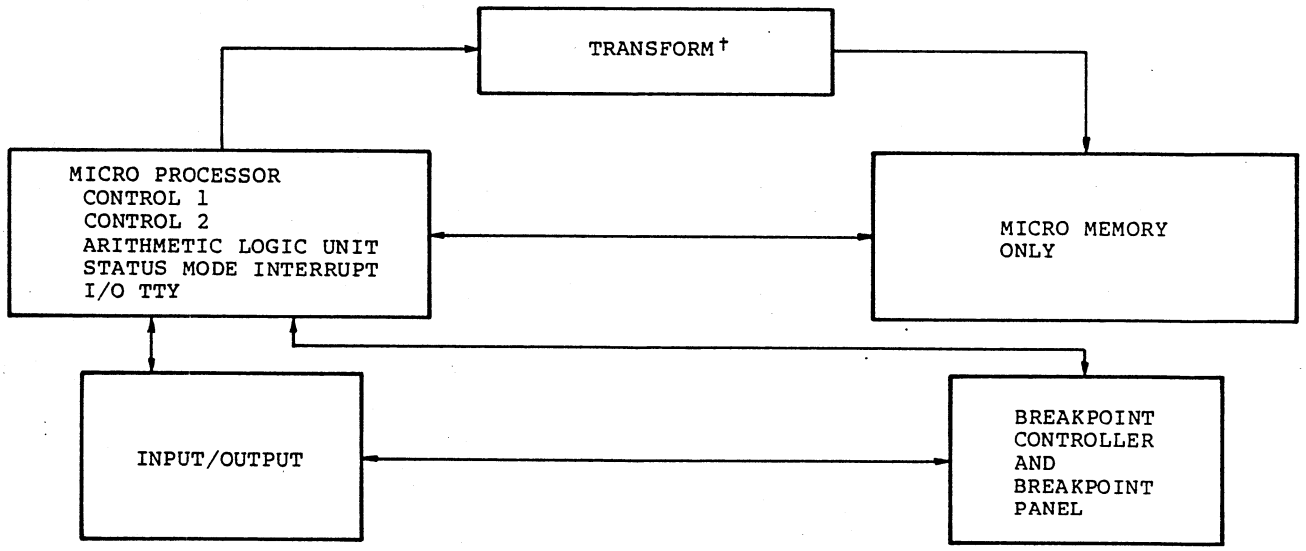
This section presents a functional description of the basic micro-programmable processor. The basic processor consists of the control 1, control 2, arithmetic/logical unit (ALU), status mode interrupt (SMI), transform, and I/O TTY modules that are required for all processor configurations. The theory of operation is presented in terms of the individual module functions. Descriptions of the mnemonics used in this section and in the logic diagrams are included in appendix A.

GENERAL

The basic elements of the processor may be configured in various forms to serve different applications. The hardware modules used in the basic processor are required for all processor configurations. The basic processor can be configured as a single-level processor, as indicated in figure 4-1, or as a multilevel processor, as in figure 4-2.

The single-level processor treats micro memory as main memory. In this configuration there is a direct one-to-one relationship between the instruction read from storage and its decoding and execution.

The multilevel processor configuration includes two memory systems: a macro memory and a micro memory. With this configuration, a macro-instruction program is stored in macro memory and a micro-instruction program is stored in the micro-memory system. The micro memory is smaller and faster in access and cycle times than the macro memory. A set of micro-memory subroutines corresponds to each macro instruction stored in macro memory. Each subroutine contains a sequence of micro instructions for executing a particular macro instruction. The hardware decode of the macro instruction allows micro control to effect a branch to the appropriate subroutine in micro memory. This is referred to as transform action. The transform operation is application-dependent. It is possible to perform the micro instruction decode without the use of the transform module shown in figure 4-2, but decoding is much faster when the transform module is incorporated into the configuration. The transform hardware performs a partial decode to direct program branching, to set parameters, and to initiate arithmetic or logical operations. These operations are performed in parallel.



†TRANSFORM HARDWARE IS REQUIRED TO LOAD THE N/K REGISTER AND TO ENCODE MIR FOR MORE EFFICIENT OPERATION.

Figure 4-1. Single Level Processor Configuration

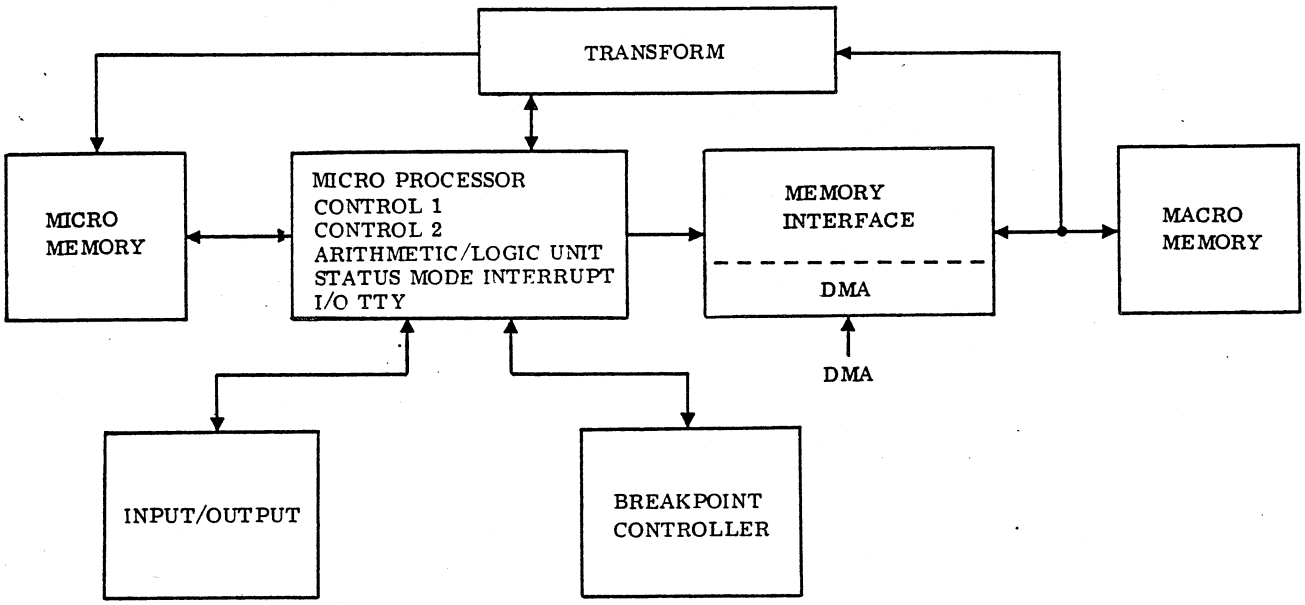


Figure 4-2. Typical Multilevel Processor Configuration

ORGANIZATION

Figure 4-3 is a general overview of the processor system functional block diagram.

Figure 4-4 presents the detailed block diagram of the basic processor. This diagram shows that the registers of the processor are interconnected primarily by selectors. Each selector is a multiplexer that provides switching of data to output from one of several input positions. The numbers in parentheses inside the selector blocks indicate the width of the selector. Figure 4-4 shows primarily the block diagram and the data path of the arithmetic/logical section, the micro memory and micro control section (control 2 module), the status mode interrupt section, a typical transform section, and some important registers of the input/output and macro memory sections. The control and timing hardware of the control 1 and control 2 modules are not shown in this diagram.

CONTROL AND TIMING

Control 1 generates the timing and control signals for the basic processor. The operation of this assembly is described in detail in section 5.

MICRO MEMORY AND MICRO CONTROL SECTION

Figure 4-5 is a block diagram of the control 2 module. The following functional areas are in the micro memory and micro control section.

Micro memory (MM) (not on control 2)

Micro-instruction register (MIR)

Page/memory address register (P/MA)

T field test multiplexer

Memory address counter (PS/MAC)

Return jump register (RTJ)

N and K registers

Selectors S4 and S6

Bit generator

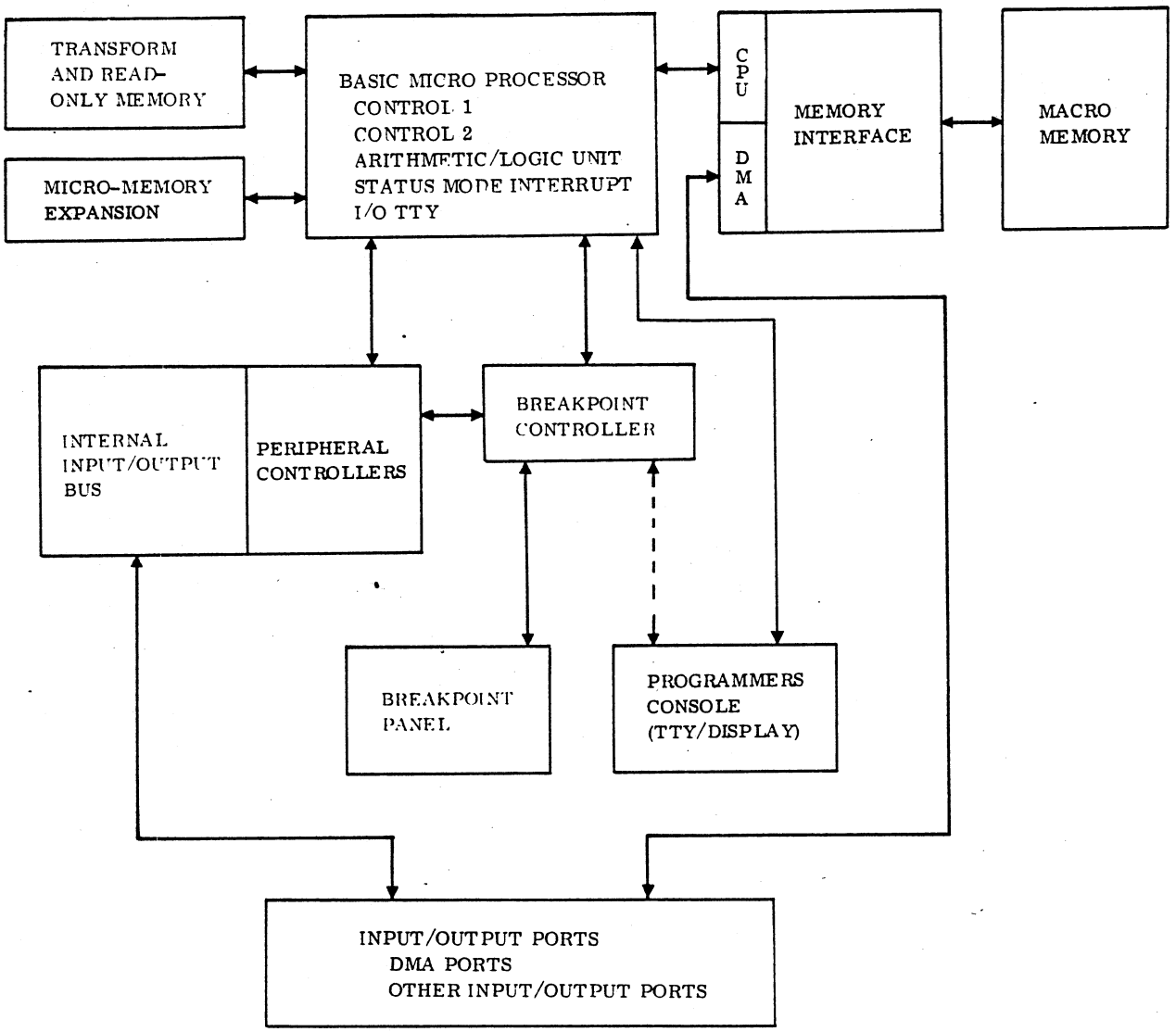


Figure 4-3. Typical Functional Block Diagram

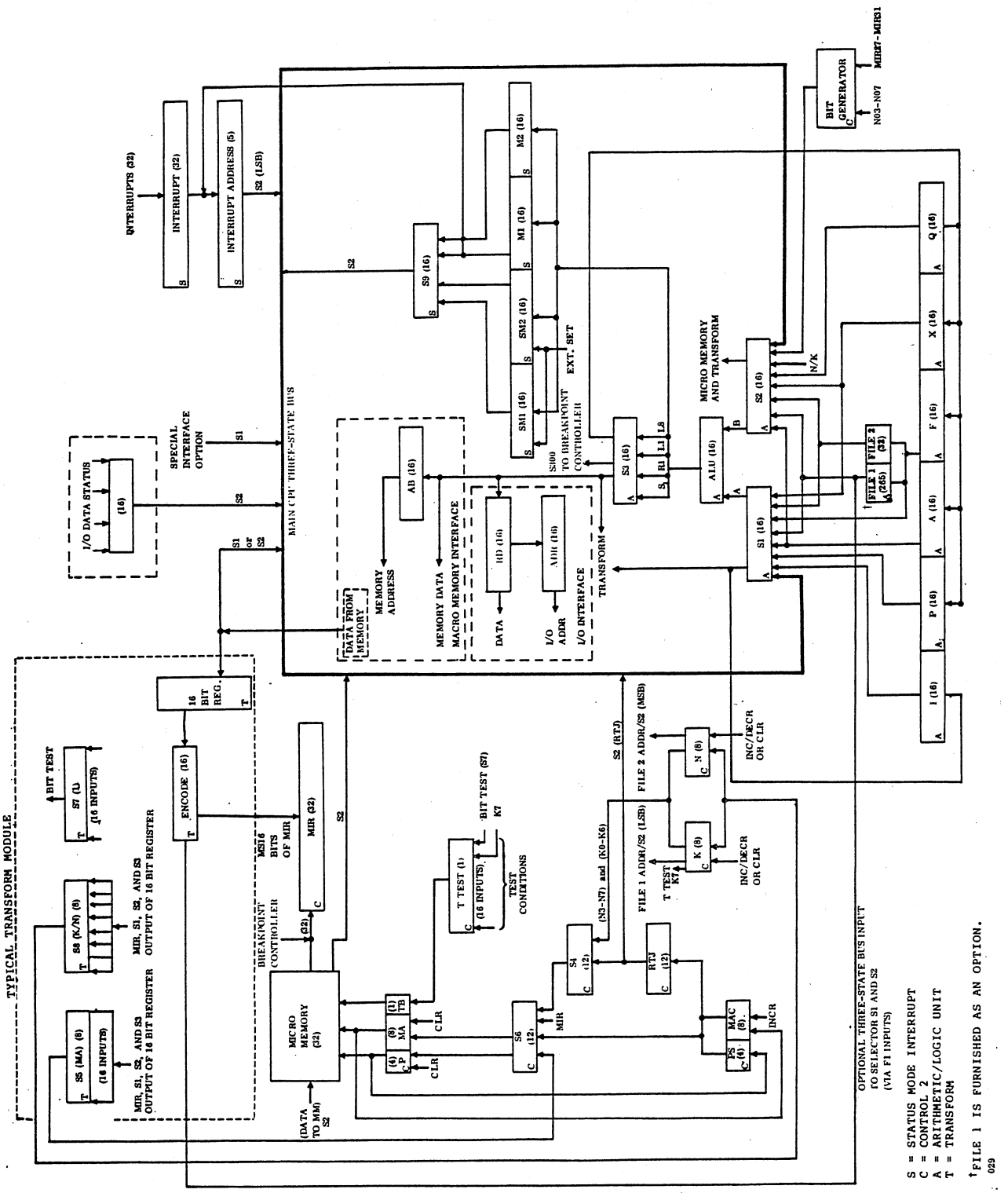
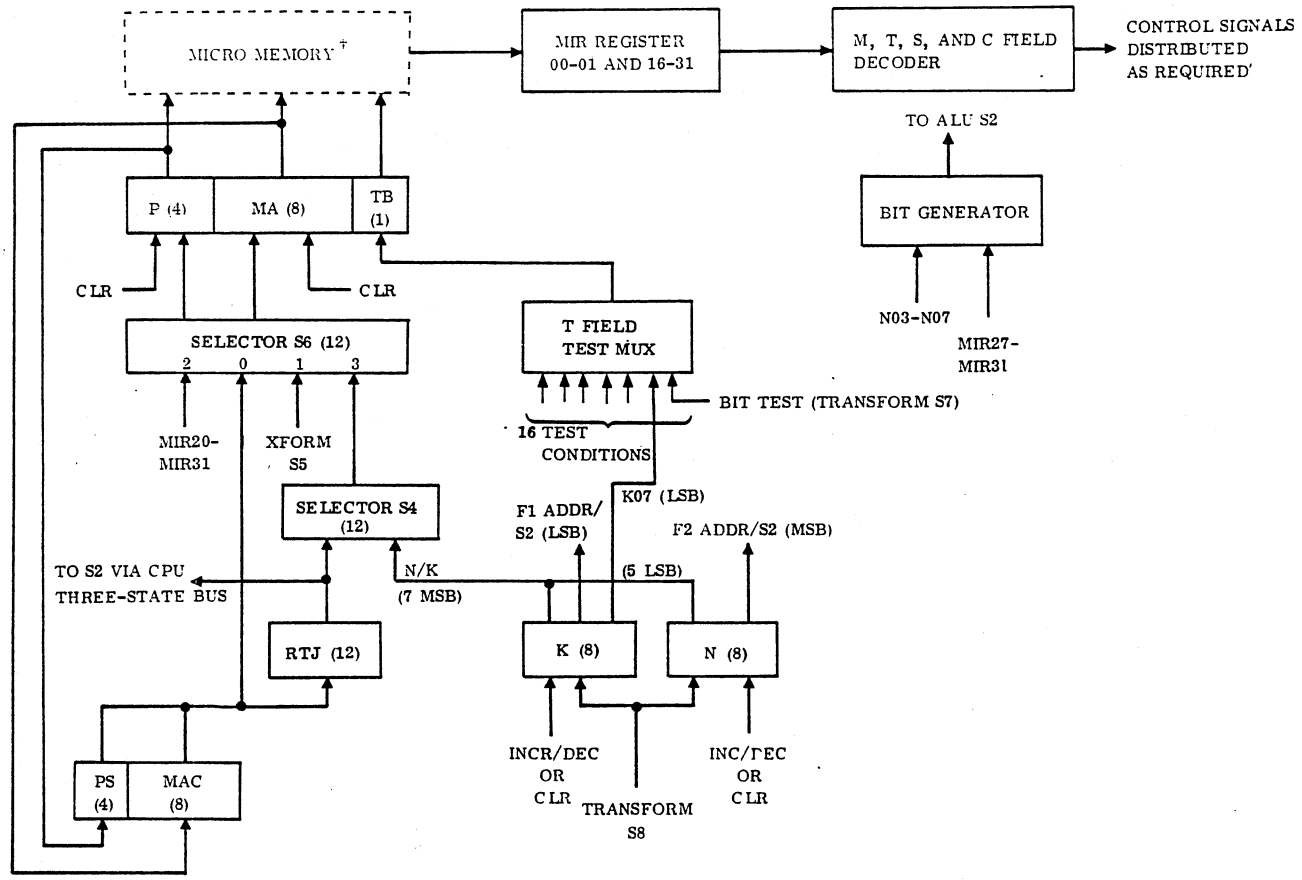


Figure 4-4. Processor Detailed Block Diagram



† THE MICRO MEMORY IS NOT LOCATED ON THE CONTROL 2 MODULE.

NOTES:

1. THE NUMBERS INSIDE THE SELECTOR BLOCKS INDICATE THE SELECTOR POSITIONS.
2. THE NUMBERS IN PARENTHESES INDICATE THE WIDTH OF REGISTERS AND SELECTORS.

036

Figure 4-5. Control 2 Module Block Diagram

MICRO MEMORY

The micro memory is not located on the control 2 module, but the micro-memory description is given here to provide continuity in the micro control description.

The micro memory is normally used to store the firmware (sets of micro instructions) that controls the operation of the microprocessor.

The micro memory is available as a read/write memory or as a read-only memory. Each memory word contains two 32-bit micro instructions, referred to as the upper micro instruction (MM00 through MM15) and the lower micro instruction (MM16 through MM31).

Each 256 words (that is, 256 instruction pairs) is termed a page. The memory can consist of as little as one page (512 micro instructions) or up to 16 pages (8192 micro instructions).

READ-ONLY MEMORY

A read-only memory may be used in processor configurations intended for fixed applications. Read-only memory is irreversibly programmed with application-dependent firmware, such as the 1700 emulator that is required to emulate the CDC 1700 Series computers. This read-only memory normally resides on the transform module and contains 512 micro-memory words (1024 micro instructions).

READ/WRITE MICRO MEMORY

Read/write memory is available in units of 512 or 2048 micro instructions for applications that require a reprogramming capability. This read/write micro memory is either loaded from an external device or data can be written into the micro memory under control of the micro program. The micro memory gets its input data from the output of selector S2. The output of micro memory is connected to the micro-instruction register (MIR) and to the main CPU three-state bus. The micro memory is read out to the MIR in 32-bit format, but micro-memory data is transferred to the X register via the main CPU three-state bus 16 bits at a time only.

MICRO-INSTRUCTION ADDRESSING

The micro memory is addressed via the page/memory address (P/MA) and test bit registers.

The four bits from the P portion of the P/MA register define which one of 16 pages is to be addressed. The eight MA bits define the locaton of one of 256 words on that page. Each of the 256 words holds two 32-bit micro instructions, an upper and a lower. The next micro instruction is executed from the upper or lower location as defined by test bit TB. The T field of the current micro instruction specifies via the T field test multiplexer how test bit TB is to be set. Micro-instruction sequencing does not execute the automatic overflow of addressing from the MA register to the P register. That is, the MA address wraps around within a page and cannot be automatically incremented into the next page when the address reaches 255.

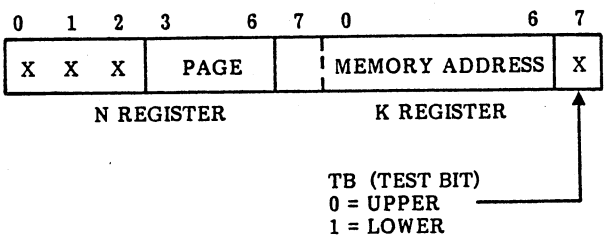
P/MA register loading is explained below. Selector S6 selects one of four sets of 12-bit inputs to set the P/MA register. The four sets are as follows.

Postion 0 - The four bits of the page storage resister PS are input to P, and the eight bits from the memory address counter (MAC) are input to the memory address (MA) register.

Position 1 - Micro-instruction bits MIR20 through MIR23 are input to P, and eight bits from S5 in the transform module are input to MA.

Postion 2 - Micro-instruction bits MIR20 through MIR23 are input to P, and MIR24 through MIR31 are input to MA.

Position 3 - The input to P and MA is via the 12-bit-wide selector S4. In one position S4 selects the 12-bit return jump address from RTJ, and in the other position selects the combined N/K register as follows:



5026

Two micro-memory addressing modes are controlled by SM113. If set, SM113 selects the MA transform via selector S5; if not set, SM113 selects the combined N/K registers.

The eight-bit memory address counter, MAC, is used to determine the next address within the current page. The current location in MA is transferred to MAC at each micro-memory reference and then incremented by one. MAC does not sequence to another page. Location 0 within a page follows location 255. Whether or not the MAC is used to obtain the next micro instructon depends on the current instruction in the MIR.

The four-bit page storage register, PS, is used to determine the page for the next instruction. At each micro-memory reference, the current page number is transferred to PS from the page register, P. PS may or may not be used to set the page for the next micro instruction, depending on the sequencing mode indicated by the current instruction.

The 12-bit return jump (RTJ) register can accept and save the next sequential address. Under control of the micro instruction, it can store the contents of the PS register and the memory address counter. It is not changed until issued a command to save another address. The RTJ register may provide the return jump location to the P/MA register via selector S6, or it may be transferred to the MP processor organization in complemented form via selector S2.

The 32-bit micro instruction register, MIR, holds the current micro instruction. The contents of MIR are decoded to control the operation of the microprocessor. MIR is normally loaded with a 32-bit instruction from micro memory. The upper or lower micro instruction to be loaded is based on test bit TB, determined by the execution of the preceding micro instruction.

Micro instructions may also be transferred to MIR from the optional breakpoint controller. The optional transform module has the capability of transferring a partial micro instruction to the upper 16 bits of MIR.

TRANSFORM OPTIONS

The optional transform feature of the MP processor provides a means of forming micro-memory addresses from any pattern of bits from the selectors, registers, or data paths of the processor. These addresses may be used to sequence the micro program, to set the contents of the N and K registers, and to set the upper 16 bits of the MIR. Transform details vary with the application.

A typical transform is shown in figure 4-4. In this diagram, the output of selector S5 connects to the MA register via S6. The 16 positions of selector S5 provide means for selecting the micro address from one of 16 different 8-bit patterns. The registers, selectors, or data paths forming the sources for these 8-bit patterns are defined for each particular application by the micro-program designer and are implemented by backplane and transform module wiring.

The output of selector S8 provides eight bits for loading the N and K registers. Each of the eight selector input positions can accept an 8-bit pattern from the processor. The sources and construction of these bit patterns are also specified by the designer to fit the particular application.

Figure 4-4 does not show the actual connections to the selection input. In general, these may be taken from selectors S1, S2, S3, the SM registers, the input/output data registers, the MIR, the macro-memory read data bus, or micro memory.

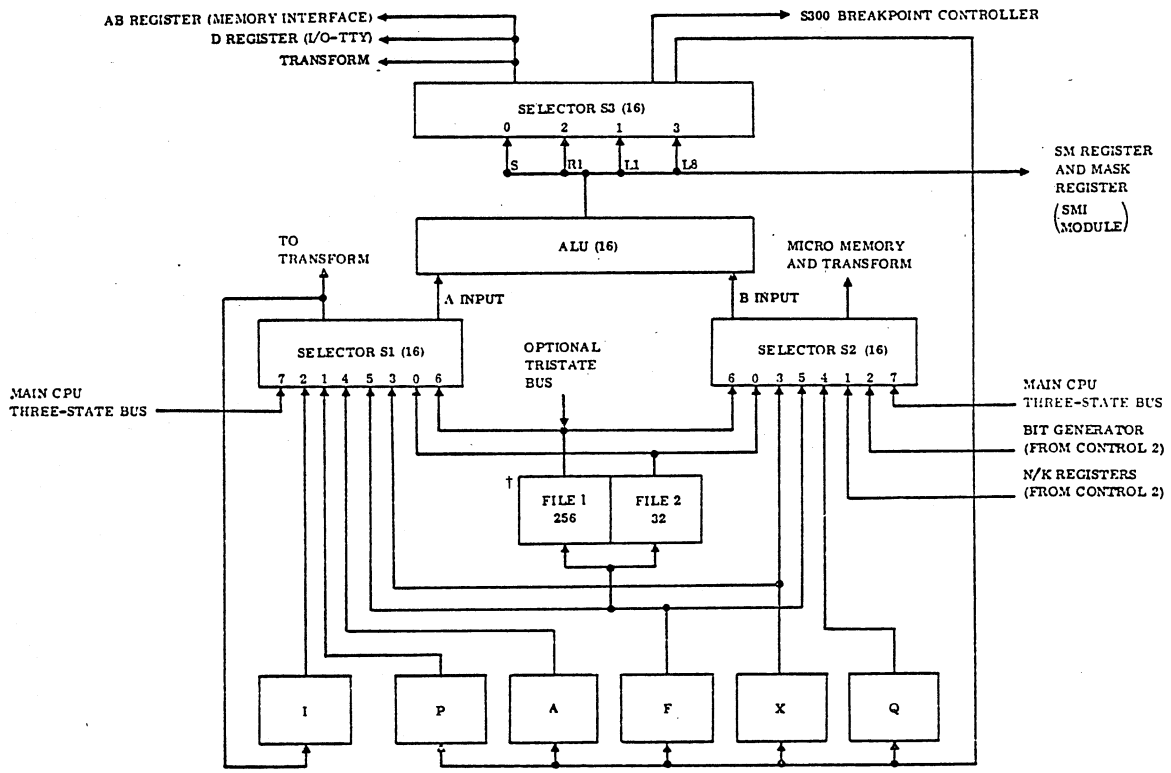
In the typical transform sample shown, the 16-bit holding register can accept macro memory data under control of the micro instruction. The encoder block generates a 16-bit partial micro instruction based on the information held in the register. This may be loaded directly into the upper 16 bits of MIR to control processor arithmetic functions during macro emulation. Additional hardware features may exist in other transform modules designed for special applications. A detailed description of the transform module is given later in this section.

BIT GENERATOR

The bit generator is located on the control 2 module but affects the operations of the arithmetic/logical unit. Refer to the following Arithmetic and Logical Section description for the bit generator application.

ARITHMETIC AND LOGICAL UNIT SECTION

The arithmetic and logical section consists of the following blocks: arithmetic/logical unit, selectors S1, S2, and S3, primary registers I, P, A, F, X, and Q, file 2, and file 1 (optional). Figure 4-6 shows the arithmetic/logical unit block diagram.



†FILE 1 IS OPTIONAL

NOTES:

1. THE NUMBERS INSIDE THE SELECTOR BLOCKS INDICATE THE SELECTOR POSITION.
2. THE NUMBERS IN PARENTHESES INDICATE THE WIDTH OF REGISTERS AND SELECTORS.

038

Figure 4-6. Arithmetic/Logical Unit Module Block Diagram

The arithmetic/logical unit operates on two full-word inputs, A and B, to perform the arithmetic or logical function specified by the micro instruction. The result is input to the status/mode registers (SM1 and SM2), the mask registers (M1 or M2), or selector S3.

The A input word is 16 bits from selector S1; the B input word is 16 bits from selector S2. The arithmetic/logical unit output is 16 bits. This connects to the inputs of selector S3 in four different arrangements so S3 can transfer the word with no shifts, one-bit right shift (R1), one-bit left shift (L1), or end-around eight bits left shift (L8), depending on the input selected.

The result of the arithmetic or logical operation is stored at one or more destination registers. Possible destinations are:

Either of the status/mode registers, SM1 or SM2

Either of the mask registers, M1 or M2

Any of the registers P, A, F, X, or Q

File 1 (optional) and file 2 via register F

Memory interface register AB (macro memory address word)

Memory interface (data word)

Breakpoint controller (one bit only, S300)

I/O-TTY register RD (data output word)

I/O-TTY register ADR (I/O address word)

The micro instruction specifies the destination as well as the A source, the B source, and the operation to be performed.

The arithmetic operation may be either ones complement or twos complement arithmetic. The arithmetic operation mode is set by the micro instruction via SM101 in the status/mode register.

The split adder mode is also available as one of the MP features. This allows separate operations to be carried out simultaneously in the upper eight bits and lower eight bits of the arithmetic/logical unit. This mode is also set by the micro instruction via SM103 in the status/mode register.

The primary data registers I, P, A, F, X, and Q are not limited to any specific uses, but the register names are appropriate to their use in an emulation application.

A description of the primary registers and the files is as follows:

I Register - This 16-bit register is used when the microprocessor serves as an emulator. It is used to hold a software instruction during execution. Input to register I comes directly from the output of selector S1, so data on the three-state bus can be stored directly in register I and simultaneously input to the arithmetic/logical unit for some other operation. This is particularly useful in configurations using macro memory.

P Register - This 16-bit register receives data from selector S3 and outputs to the A input of the arithmetic/logical unit via S1. In computer emulation configurations, it is normally used to contain the macro program instruction counter.

A Register - The 16-bit A register may be used for data shifts, either by itself or in conjunction with the Q register as a double-length shift register. The shift function is independent of the arithmetic/logical unit and S3. This general-purpose register inputs from S3 and outputs to the A input of the arithmetic/logical unit via S1.

F Register - This 16-bit general-purpose register inputs from S3 and outputs to the A input of the arithmetic/logical unit via S1 or the B input via S2. The F register serves as an entry register for file 2 or file 1 when either of these is a destination of an arithmetic/logical unit operation.

S Register - The X register is a 16-bit general-purpose register whose input is from S3. It outputs to the arithmetic/logical unit A input via S1 and to the B input via S2.

Q Register - The 16-bit Q register is a general-purpose register that outputs to the arithmetic/logical unit B input via S2. It may be shifted left or right in conjunction with the A register. This shift is independent of the arithmetic/logical unit and S3.

File 1 - File 1 contains 256 16-bit words addressed by the contents of the K register. The register output shares the three-state bus with an optional source. This may be from a transform or an external device. A status mode bit selects either the file 1 output or the option. The data is sent to the arithmetic/logical unit A input via S1 or the B input via S2.

File 2 - File 2 contains 32 16-bit words addressed by the lower five bits of the N register. It delivers its output to the arithmetic/logical unit A input via S2 and to the arithmetic/logical unit B input via S1. File 2 is intended as a source for constants, but it may be used as a general-purpose file.

Bit Generator (BG) - The bit generator is capable of setting one bit to the B input of the arithmetic/logical unit via S2. The single bit can be set at any of the 16 bit positions in the word. The bit position is established by the least significant five bits in the micro instruction (MIR27 through MIR31) or by the lower five bits of register N. Status mode bit SM102 is set, the bit generator position is determined by bits N03 through N07.

Selector S1 - S1 provides for the selection of one of eight inputs for delivery to the arithmetic/logical unit (A input), to the I register (if I is a destination register), and to the transform module. Input to the selector is indicated below:

Position	Input Source
0	File 2
1	P
2	I
3	X
4	A
5	F
6	File 1 or external input (optional three-state bus)
7	Main CPU three-state bus (data from macro memory, SM and mask registers, algorithm option)

Selector S2 - Similarly, S2 provides for the selection of one of eight inputs for delivery to the arithmetic/logical unit (B input), the micro memory, and the transform module. Input to selector S2 is indicated below:

Position	Input Source
0	File 2
1	N and K registers
2	Bit generator
3	X register
4	Q register
5	F register
6	File 1 or external input (optional three-state bus)
7	Main CPU three-state bus (data from macro memory, interrupt address, macro memory, input/output data/status register, breakpoint panel, and return jump (RTJ) register)

STATUS MODE INTERRUPT SECTION

The status/mode interrupt module contains the status/mode registers, mask registers, the interrupt register, and selector S9. Figure 4-7 is a block diagram of the status mode interrupt module.

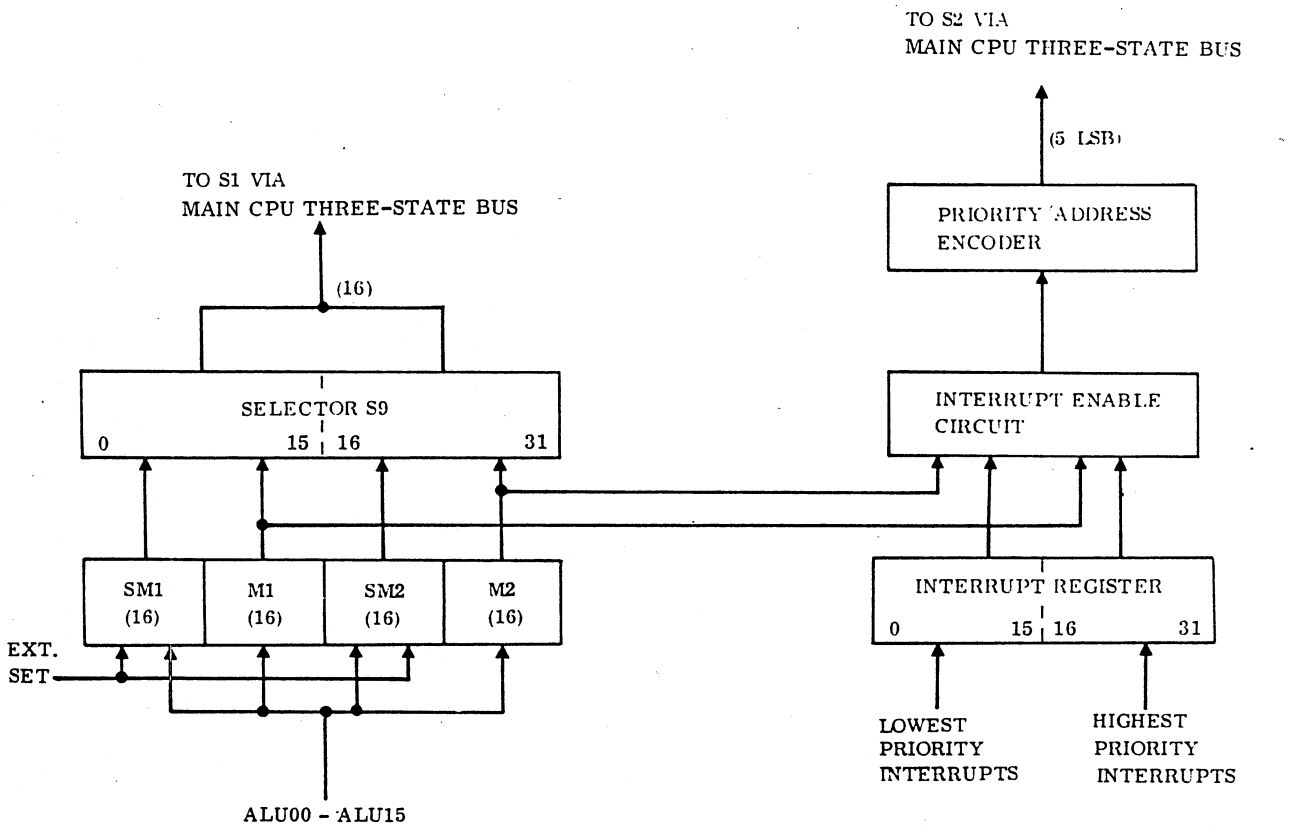


Figure 4-7. Status Mode Interrupt Module Block Diagram

STATUS/MODE REGISTERS

The status/mode registers (SM1 and SM2) allow the micro program to control the mode of operation and to examine the status of certain internal and external conditions. Each status/mode register has 16 bits, for a total number of 32 status/mode bits in the MP processor. Some of these status/mode bits can be set by an external signal, some by micro instructions. However, all status/mode bits can be set or reset by transferring information to the status/mode register from the output of the arithmetic/logical unit. Master clear clears both SM1 and SM2. Depending on the status/mode bits, they may have true or complement output or both.

A table provided in section 5 summarizes the characteristics of the status/mode bits. The contents of the status/mode registers can be used as the A source of the arithmetic/logical unit by transferring the status/mode register output via selector S9 through the main CPU three-state bus and selector S1. See the A′ code under Micro Instructions, section 2.

MASK REGISTERS

The main purpose of the mask (M1 and M2) registers is to enable and disable the desired interrupt lines. To enable an interrupt line, its corresponding mask bit must be set. Mask register M1 controls interrupt lines 00 through 15 and M2 controls interrupt lines 16 through 31. Each mask register has the same number of bits as the basic processor word length (16 bits). The mask bits can only be set/reset by transferring information from the arithmetic/logical unit outputs. Master clear does not clear the mask bits. Similar to the status/mode register, the contents of the M register can be used as the A source of the arithmetic/logical unit. See the A′ code under Micro Instructions, section 2.

INTERRUPT REGISTER AND INTERRUPT ADDRESS ENCODER

The interrupt system of the basic processor is implemented as a sampled data system at the micro-program level. The INTU micro instruction (in the T field) is used to sample the interrupt system to see if the corresponding mask register bit of any interrupt present is set to 1. Because of the interrupt sampling method, the interrupt signals must be in steady state. If a pulse-type interrupt is required, the pulse interrupt signal is used to set a bit in the status/mode register; this status/mode bit is then wired to the interrupt system. On recognizing this interrupt, the micro program can clear the interrupt condition by clearing that status/mode bit. The interrupt register is a 32-bit register that stores the conditions of the interrupt lines. The output register is then compared with the outputs of the mask register to control the interrupt recognition.

Once the interrupts are recognized, the interrupt address encoder selects the highest priority interrupt and generates its corresponding address. The interrupt address generated is the same as its priority level; i.e., the highest priority interrupt (line 00, corresponding to mask bit M115) generates a 00 interrupt address and the lowest priority interrupt (line 31, corresponding to mask bit M200) generates a 31 interrupt address. The interrupt address encoder must be read in by micro instruction INTA in B′ code following the interrupt test to be sure of the correct interrupt line address. The output of the interrupt address encoder is the complement of the interrupt address for input to S2. For example, to transfer the interrupt address to the X register, the micro instruction should be coded with -B code in the F field, INTA in the B′ field, and X in the D field. This causes the output of the interrupt address encoder to go through the main CPU three-state bus, through selector S2, through the arithmetic/logical unit to be complemented, through selector S3, and finally to the X register. The interrupt system also allows the option of activating interrupts in groups of eight interrupt lines. This is a function of backplane wiring. Any unused status/mode bit that has complement output can be used to activate the desired interrupt groups. However, the desired interrupt groups can be permanently enabled by grounding the appropriate enable input of the interrupt priority encoder. The interrupt lines corresponding to the various interrupt groups are shown below:

Group Number	Interrupt Line
Group 0	Interrupt lines 00 through 07
Group 1	Interrupt lines 08 through 15
Group 2	Interrupt lines 16 through 23
Group 3	Interrupt lines 24 through 31

SELECTOR S9

Selector S9 is a 16-bit wide selector that selects one of four sources (SM1, SM2, M1, or M2) to be the A source of the arithmetic/logical unit. The output of selector S9 is connected to selector S1 via the main CPU three-state bus.

STATUS MODE AND INTERRUPT ASSIGNMENTS

The assignment of status and mode conditions to specific bits of the status/mode register is a design function. Refer to the appropriate system status/mode bit assignment table for complete details.

Table 4-1 lists typical functions assigned to the status mode register bits. Bit functions left blank are not used. Table 4-2 shows the typical interrupt assignments for the basic processor; the interrupts are listed in descending priority with interrupt line 00 having the highest priority and interrupt line 31 having the lowest. Interrupts 00 through 15 are micro interrupts and interrupts 16 through 31 are macro interrupts. The interrupt assignments in table 4-2 are prewired for the standard backplane of the basic MP processor. The open pins are available for customer use. Refer to specific system interrupt assignments for any variation from the standard assignment of the basic processor.

TABLE 4-1. TYPICAL STATUS/MODE REGISTER BIT ASSIGNMENT

SM Bit	Name	Function
100	Double Precision	Not included.
101	Ones Complement	1 = The arithmetic/logical unit operates in ones complement arithmetic mode for addition and subtraction. 0 = Operations are in twos complement arithmetic mode for addition and subtraction.
102	BG Input From N	1 = The bit generator is controlled by the lower five bits of the N register (N03 through N07). 0 = The bit generator is controlled by the lower five bits of the micro instruction (MIR27 through MIR31).
103	Adder Split	1 = The arithmetic/logical unit is split into two independent arithmetic/logical units at the adder split point for arithmetic operation. If ones complement is selected, the lower portion of the arithmetic/logical unit operates in ones complement and the upper portion always operates in twos complement.
104	Macro BKP Interrupt	Set if the macro memory address is equal to the breakpoint register and select conditions are met. Select conditions are determined by FCR14 and FCR15, and FCR14 must be set to 1.
105	Protect Fault	Set if a protect violation occurred.
106	Active Interrupt System	1 = Enables the interrupt system to allow the INTU command to examine the interrupt system. 0 = The interrupt system is disabled.
107	Open	
108	Memory Parity Error	Set if the macro-memory interface detects a parity error during a read operation with macro memory.
109	Micro Halt	1 = Any micro instruction with a HALT code in the S field stops the operation of the processor on completion of the micro instruction. 0 = The HALT code in the S field of any micro instruction is ignored.
110	Overflow	Set on detection of an overflow condition. Three selectable overflow conditions may be tested: a. Arithmetic overflow - Micro instruction performing the arithmetic operation is an add or subtract with overflow test and the arithmetic result is inconsistent with the sign of the operands and arithmetic operation. b. Binary overflow - Micro instruction performing the arithmetic operation is an add or subtract with overflow test, and a carry-out of the most significant bit of the arithmetic/logical unit has occurred. This condition is tested if the optional binary overflow status/mode bit is set to 1. c. Decimal overflow - Not applicable to the basic processor.
111	Enable F1	1 = Enables F1 to selectors S1 and S2. 0 = Enables an external source to selectors S1 and S2.
112	Binary Overflow Test	Set to monitor arithmetic/logical unit output for binary overflow for setting SM110.
113	Select XT/MA	1 = The MA transform determines the micro-memory address for read/write micro-memory operand references. 0 = The N/K register determines the micro-memory address for read/write micro-memory operand references.

TABLE 4-1. TYPICAL STATUS/MODE REGISTER BIT ASSIGNMENT (Contd)

SM Bit	Name	Function
114	Pre-enable Interrupt System	Not used in the basic processor. Applicable for enhanced processor only. If set, it activates the interrupt system by allowing SM106 to be set on execution of a second GITMAK micro instruction. Cleared by hardware once SM106 is set.
115	Open	
200	Auto-Data Transfer (ADT)	Set when the location of the ADT table is determined. See Auto-Data Transfer in the CYBER 18 Processor with MOS Memory (Macro Level) Reference Manual.
201	Strobe/Read	Strobe/read signal of NCR M05 or A/Q channel. See Auto-Data Transfer in the CYBER 18 Processor with MOS Memory (Macro Level) Reference Manual.
202	Enable Port/Write	Enable port/write signal of NCR M05 or A/Q channel. See Auto-Data Transfer in the CYBER 18 Processor with MOS Memory (Macro Level) Reference Manual.
203	Terminate	Set to terminate the ADT sequence during the last data transfer or because of an addressing error. See Auto-Data Transfer in the CYBER 18 Processor with MOS Memory (Macro Level) Reference Manual.
204	Deadstart	1 = The deadstart mode is selected on the breakpoint panel. 0 = The deadstart mode is disabled on the breakpoint panel.
205	Open	
206	Open	
207	Select Page (XT/MA)	1 = An MA transform can be executed across a page boundary by setting the desired page in the S field of the micro instruction. Normal S field operations are disabled. 0 = An MA transform is limited to the page in which the micro instruction containing the MA transform command resides. The S field of the micro instruction is decoded for normal operations.
208	Open	
209	Open	
210	Write Data to Panel Device	These bits are required for panel simulation.
211	Read Data from Panel Device	
212	Write Data to Auxiliary Control Function Register	
213	Open	
214	Pre-enable Console Interface to MIR	1 = The output of the console interface is pre-enabled to MIR. A page jump or return jump micro instruction must be executed with this mode bit set to 1 to force the actual selection. If the processor is master cleared, the console interface is enabled to MIR. 0 = The output of the micro memory is pre-enabled to MIR. A page jump or return jump micro instruction must be executed with this mode bit set to 0 to force the actual selection to the page and address selected.
215	Macro Run	Set by console GO (macro mode). Cleared by console STOP or firmware when the selective stop condition is detected. The cleared state initiates a constant micro interrupt (lowest priority).

TABLE 4-2. STANDARD INTERRUPT LINE ASSIGNMENT

Mask Bit	Interrupt Line Number		Function
	Micro	Macro	
M115	0		Magnetic tape controller (NRZI only) interrupt†
M114	1		TTY/console display controller ADT interrupt
M113	2		Paper tape reader/punch and card punch interrupt
M112	3		Unassigned
M111	4		Line printer controller ADT interrupt
M110	5		Unassigned
M109	6		Unassigned
M108	7		Tape cassette controller ADT interrupt
M107	8		Real-time clock data interrupt
M106	9		Magnetic tape controller (NRZI only) interrupt†
M105	10		Communication line adapter ADT interrupt
M104	11		Card reader controller ADT interrupt
M103	12		Panel simulation (keyboard input)
M102	13		Panel simulation (display output)
M101	14		Panel request to CPU (used during emulation)
M100	15		Stop signal; suspends emulation
M215		0	Protect parity and power failure interrupt
M214		1	TTY/console display controller program interrupt
M213		2	Paper tape reader/punch and card punch interrupt
M212		3	Unassigned
M211		4	Line printer controller program interrupt
M210		5	Unassigned
M209		6	Unassigned
M208		7	Tape cassette controller program interrupt
M207		8	Real-time clock program interrupt
M206		9	Magnetic tape controller (NRZI only) program interrupt
M205		10	Communication line adapter program interrupt
M204		11	Card reader controller program interrupt
M203		12	Magnetic tape controller (NRZI and phase encode) interrupt
M202		13	Unassigned
M201		14	Storage module drive (SMD) program interrupt
M200		15	Breakpoint interrupt

†Wire ORed on the processor backpanel.

TRANSFORM

The 1700 Transform with Binary-Coded Decimal arithmetic (BCD) is used to emulate the 1700 computer instruction repertoire when it is combined with the basic microprocessor to form the 1700 enhanced processor. The emulation process utilizes both hardware and firmware for more efficient operation.

The firmware consists of many micro-code subroutines that emulate 1700 macro instructions; therefore, it is also called the 1700 emulator. For each 1700 macro instruction, there exists a corresponding subroutine required to emulate it. To start the emulation, the macro instruction is read out from macro memory by a portion of the micro program. The macro instruction is then decoded by hardware; this hardware decode is called the transform. The transform provides the micro program with the capability to select patterns of bits from the registers and the data-transmission path of the processor to form the micro-memory address. This micro-memory address selects the appropriate micro-code subroutine to emulate the macro instruction. More than one transform operation may be required to completely emulate a macro instruction. The transform also sets the parameters, generates the micro code needed for the arithmetic and logical operation (refer to the MIR Encode) during the emulation process, and sets the contents of the N and K registers.

There are three types of transforms:

MA transform

K or N transform

Combined MA and K transform

The transform commands are coded in the C" (double prime) field of the micro instruction as TMA/j, TMAK/j, TN/j, TK/j, GITMAK/j, and GITMAK/XT. The letter j is decoded from the lower four bits (MIR28 through MIR31) of the micro-instruction register for the MA transform and the lower three bits (MIR29 through MIR31) for the K and N transform. These bits specify the selector position of the MA transform and of the K and N transform. Table 4-3 lists the operations that result when the above transform commands are executed. Figure 4-8 is the block diagram of the 1700 transform module.

TABLE 4-3. TRANSFORM OPERATIONS

Mnemonics	Operation
TMA/j	Obtain next micro-instruction pair from the address specified by MA transform (selector S5), setting j.
TK/j	Set K register to value specified by K transform (selector S8), setting j.
TN/j	Set N register to value specified by N transform (selector S8), setting j.
TMAK/j	An MA and K transform is executed based on the value of j.
GITMAK/j†	1. Output data from macro memory is gated into the instruction transform (IXT) register. 2. An MA and K transform is executed based on the value of j.
GITMAK/XT†	1. Output data from macro memory is gated into the IXT and IXT′ registers. 2. One of eight MA transforms is executed based on the macro instruction loaded into the IXT register (selected from selector S5, positions 8 through 15). The K register is always transformed from S8, position 7. 3. The most significant 16 bits of the micro instruction register (MIR) are loaded with a micro command encoded from the macro instruction residing in the IXT register. This operation is referred to as MIR transform (XT/MIR). The least significant 16 bits of MIR are loaded from micro memory.

†These commands must be executed in the micro instruction following a read command.

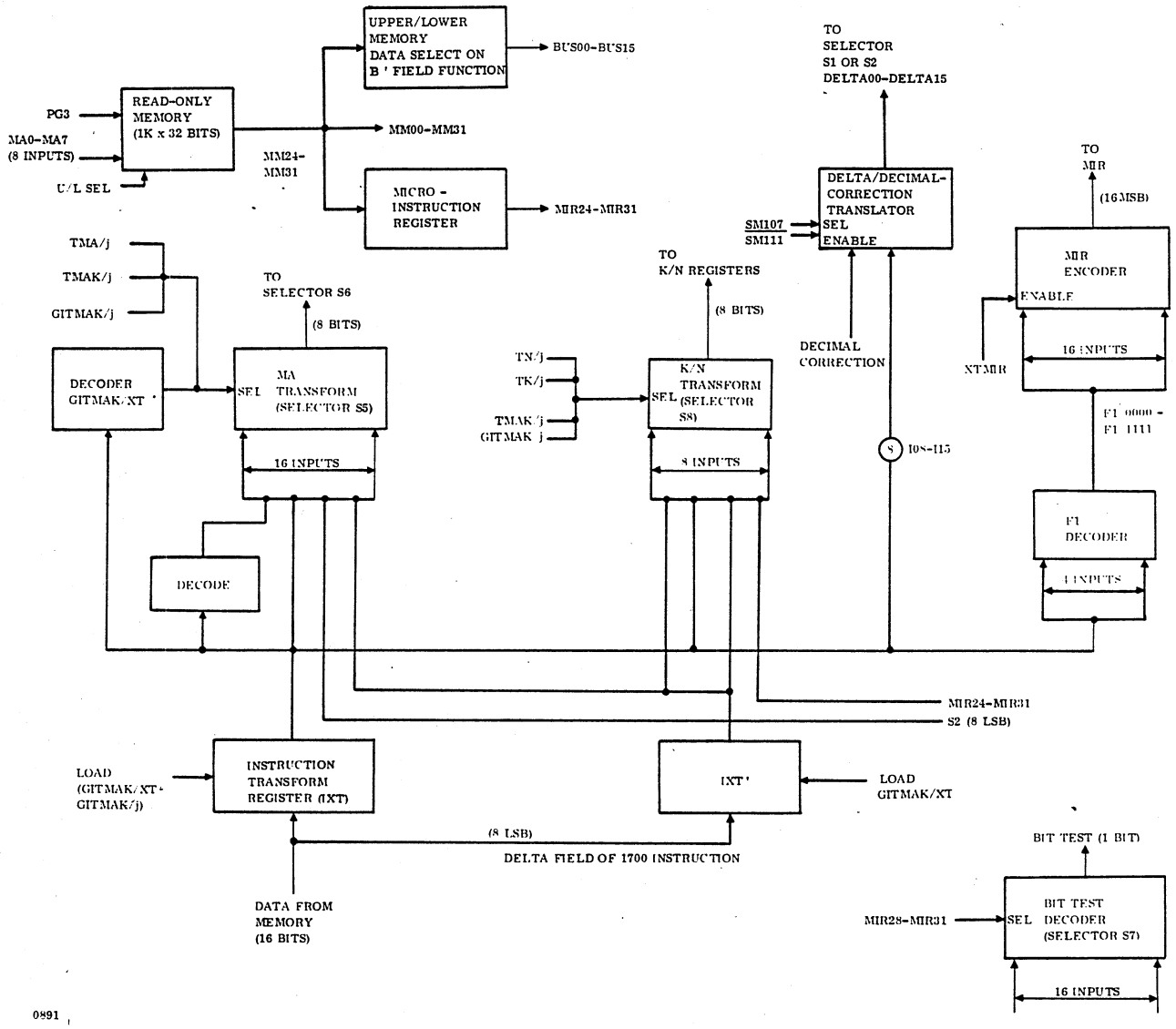


Figure 4-8. 1700 Transform Module Block Diagram

INSTRUCTION TRANSFORM (IXT) REGISTER

This is a 16-bit register that holds the macro instruction currently being emulated. It receives inputs directly from macro (main) memory. The IXT register outputs are sent to the MA transform, K/N transform, delta translator, and via the F1 decoder to the MIR encoder to be transformed. Other IXT register outputs are also sent to the GITMAK/XT decoder to generate the proper select signals for the MA transform during a GITMAK/XT operation. This register is loaded by executing a macro-memory-read micro instruction followed by a micro instruction with a GITMAK/j or GITMAK/XT in the C" field. The IXT register can also be loaded by executing a micro instruction with the C′ (prime) field equal to 011XXXX to generate a general-purpose strobe at time T4 (GATEIST-/). This signal is available at the backpanel at N25 and can be used as a scope sync pulse.

IXT′ REGISTER

The IXT′ (prime) register is an 8-bit register that holds the eight least significant bits (delta field) of the 1700 macro instruction. This register provides emulation of the 1700 enhanced instructions that have double-word format. Inputs to the IXT′ register are obtained directly from macro memory. The macro-memory data is gated into this register by application of GATEXTMIR, which is generated only during the GITMAK/XT command. IXT′ register outputs are sent to the MA and K/N transforms S5 and S8 respectively.

MA TRANSFORM

The MA transform (selector S5) is an 8-bit-wide selector that is used to form micro-memory addresses. The MA transform provides selection of one of 16 different micro-memory address (MA) transforms. These MA transforms specify one of 256 64-bit micro-instruction pairs contained within a micro-memory page. The selection of MA transform is determined by the j value of transform commands (TMA/j, TMAK/j, and GITMAK/j) or depends upon the macro instruction via transform hardware (GITMAK/XT commands).

Figure 4-9 indicates 16 different instruction transforms. Transform selections (j value) 0 through 7 and 9 through C are applicable to emulation of enhanced 1700 instructions. Transform

INSTRUCTION	j VALUE	TRANSFORM OUTPUT (SELECTOR S5) 0 1 2 3 4 5 6 7
XT/INT	0	1 0 1 1 1 1 S2 (11) A†
XT/IR2	1	1 0 1 1 1 IXT' (11 - 12) 0
XT/F3A	2	1 1 0 1 1 IXT' (13 - 15)
XT/DEST	3	0 0 1 1 1 IXT' (13 - 15)
XT/COM	4	1 1 1 IXT' (11 - 15)
OPEN (ZERO)	5	0 0 0 0 0 0 0 0
XT/F2	6	1 0 1 0 IXT' (8 - 9) Δ= 0 0
XT/S2	7	S2, LOWER 8 BITS (08 through 15)

INSTRUCTION	j VALUE	TRANSFORM OUTPUT (SELECTOR S5) 0 1 2 3 4 5 6 7
XT/SK	8	1 1 0 IXT' (08 - 11) 0
XT/SH or XT/DRP	9	1 0 1 1 0 IXT' (08 - 10)
XT/IR or XT/F3	A	1 1 1 0 IXT' (12 - 15)
XT/F or XT/F4	B	1 0 0 IXT' (00 - 03) 0
XT/IM or XT/SKIP 2 or XT/SCI	C	1 1 1 1 IXT' (08 - 11)
XT/F1	D	0 1 F= 0 IXT (04 - 07) Δ= 0
XT/F1*	E	1 0 1 0 0 B† IXT (06 - 07)
XT/FM	F	1 0 1 1 1 0 C† D†

†A = (IXT=0500) $\overline{(SM105)}$ + S209
B = (F1=00xx) + MULTILEVEL INDIRECT MODE
C = Protect violation
D = (Δ =0) + C

0892

Figure 4-9. MA Transforms

selections 8 through F are applicable to emulation of the basic 1700 instruction set. The patterns of bits are derived from hardwired connections to +5V and ground, IXT and IXT' registers, CPU selector S2, and special conditions (such as protect violation or indirect-address mode) that are decoded from the macro instruction. Table 4-4 lists the 16 different MA transforms applied for different types of macro instructions and for different addressing modes. Whenever the GITMAK/XT command is executed, one of the eight MA transforms (8 through F) is selected, based on the macro instruction being emulated. Tables 4-5, 4-6, and 4-7 list the MA transforms for the basic 1700 instruction-set storage reference (F ≠ 0), register reference, and inter-register reference instructions, respectively. The MA transforms for the enhanced instructions are selected by MA transform of instruction XT/F1 (j value D).

During GITMAK/XT operations the MA transform select signals are derived from the macro-instruction decode inputs to the GITMAK/XT decoder. During GITMAK/XT operations, the MA transform select signals are derived from the MIR28 through MIR31 inputs obtained form the micro instruction.

K/N TRANSFORM

The K/N transform (selector S8) is an 8-bit-wide selector that chooses one of eight instruction transforms to be loaded into the CPU K and N

TABLE 4-4. MA TRANSFORM APPLICATIONS

Instruction	MIR28-MIR31 = j	Application
XT/INT	0	Micro/macro interrupt
XT/IR2	1	Inter-register type 2 instruction
XT/F3A	2	Field instruction
XT/DEST	3	Register destination
XT/COM	4	Commerical instruction
	5	Not used (all zeros)
XT/F2	6	F2 (address mode) for enhanced instruction
XT/S2	7	Selector S2 (lower eight bits)
XT/SK	8	Skip instruction
XT/SH or XT/DRP	9	Shift instruction or decrement and repeat instruction
XT/IR or XT/F3	A	Inter-register instruction with M not the origin or miscellaneous instruction
XT/F or XT/F4	B	F (OP CODE) field or OP CODE for storage reference type 2 and field instruction
XT/IM or XT/SKIP 2 or XT/SCI	C	Inter-register with M origin, skip instruction type 2, or scientific instruction
XT/F1	D	F1 (address mode) field
XT/F1*	E	Alternate F1 field
XT/FM	F	Miscellaneous F1 field

TABLE 4-5. 1700 STORAGE REFERENCE TRANSFORMS DURING GITMAK/XT OPERATION

Mode	F1 (Binary)	Hexadecimal	Delta	Instruction	MIR Transform
Absolute Constant	0000	0	≠0 =0	XT/F XT/F1	Δ → X,AB P + 1 → P, AB
Absolute Constant	0001	1	≠0 =0	XT/F XT/F1*	Δ + (00FF) → X, AB P + 1 → P, AB
Absolute Constant	0010	2	≠0 =0	XT/F XT/F1*	Δ + (Q) → X, AB P + 1 → P, AB
Absolute Constant	0011	3	≠0 =0	XT/FM XT/F1*	Δ + (00FF) → X, AB P + 1 → P, AB
Indirect Storage	0100	4	≠0 =0	XT/F1* XT/F1*	Δ → X, AB P + 1 → P, AB
Indirect Storage	0101	5	≠0 =0	XT/F1* XT/F1*	Δ → X, AB P + 1 → P, AB
Indirect Storage	0111	7	≠0 =0	XT/F1* XT/F1*	Δ → X, AB P + 1 → P, AB
Relative 16-bit relative	1000	8	≠0 =0	XT/F XT/F1	P + Δ (SE) → X, AB P + 1 → P, AB
Relative 16-bit relative	1001	9	≠0 =0	XT/F1 XT/F1	P + Δ (SE) → X, AB P + 1 → P, AB
Relative 16-bit relative	1010	A	≠0 =0	XT/F1 XT/F1	P + Δ (SE) → X, AB P + 1 → P, AB
Relative 16-bit relative	1011	B	≠0 =0	XT/F1 XT/F1	P + Δ (SE) → X, AB P + 1 → P, AB
Relative indirect Relative indirect	1100	C	≠0 =0	XT/F1* XT/FM	P + Δ (SE) → X, AB P + 1 → P, AB
Relative indirect Relative indirect	1101	D	≠0 =0	XT/F1* XT/FM	P + Δ (SE) → X, AB P + 1 → P, AB
Relative indirect Relative indirect	1110	E	≠0 =0	XT/F1* XT/FM	P + Δ (SE) → X, AB P + 1 → P, AB
Relative indirect Relative indirect	1111	F	≠0 =0	XT/F1* XT/FM	P + Δ (SE) → X, AB P + 1 → P, AB

TABLE 4-6. 1700 REGISTER REFERENCE TRANSFORMS DURING GITMAK/XT OPERATION

F1 (Binary)	Instruction	MIR Transform	Comment
0000	XT/F1	NOP	Selective stop (Δ=0) Instruction enhanced (Δ≠0)
0001	XT/SK	P + 1 → P, AB	Skip
0010	XT/F1	P + Δ (SE) → F, AB	Input to A
0011	XT/F1	P + Δ (SE) → F, AB	Output from A
0100	XT/F1	NOP	Enable interrupt (Δ=0) Instruction enhanced (Δ≠0)
0101	XT/F1	NOP	Inhibit interrupt (Δ=0) Instruction enhanced (Δ≠0)
0110	XT/F1	Q → X, AB	Set program protect (Δ=0) Instruction enhanced (Δ≠0)
0111	XT/F1	Q → X, AB	Clear program protect (Δ=0) Instruction enhanced (Δ≠0)
1000	†	†	Inter-register
1001	XT/F1	P + 1 → P, AB	Increase A
1010	XT/F1	P + 1 → P, AB	Enter A
1011	XT/F1	NOP	Pass (Δ=0)
1100	XT/F1	P + 1 → P, AB	Enter Q
1101	XT/F1	P + 1 → P, AB	Increase Q
1110	XT/F1	Δ (INT) → X, AB	Exit interrupt
1111	XT/SH	A → F	Shift

† See 1700 Inter-Register Transforms, table 4-7.

TABLE 4-7. 1700 INTER-REGISTER TRANSFORMS DURING GITMAK/XT OPERATION

Instruction				Condition							
XT/IR XT/IM				M not origin register (I12 ≠ 0) M is origin register (I12 = 0)							
MIR Transform (MIR Fields)				Conditions							
						Origin			Destination		
F Bits 2-6	A Bits 7-9	B Bits 10-12	D Bits 13-15	LP I8	XR I9	A I10	Q I11	M I12	A I13	Q I14	M I15
ADDT 11001	-	-	-	0	0	X	X	0	X	X	X
A B 01110	-	-	-	1	0	X	X	0	X	X	X
A + B 01001	-	-	-	0	1	X	X	0	X	X	X
(-A) + (-B) 00001	-	-	-	1	1	X	X	0	X	X	X
ADD+ 11010	P register 001	Zeros 001	P register† 001	X††	X	X	X	1	X	X	X
-	Ones 110	-	-	X	X	0	X	0	X	X	X
-	A register 100	-	-	X	X	1	X	0	X	X	X
-	-	Ones 110	-	X	X	X	0	0	X	X	X
-	-	Q register 100	-	X	X	X	1	0	X	X	X
-	-	-	NOP 000	X	X	X	X	0	0	0	0
-	-	-	A register† 101	X	X	X	X	0	1	X	X
-	-	-	Q register† 011	X	X	X	X	0	0	1	X
-	-	-	F register 111	X	X	X	X	0	0	0	X

†NOP if protect violation detected

††X = Don't care condition

registers. Figure 4-10 indicates the eight K/N transform assignments. Bit patterns for creation of the transform outputs are derived from direct ground connections, eight bits from CPU selector S2 (S208 through S215), IXT and IXT' registers, and eight bits from the CPU micro-instruction register (MIR24 through MIR31). The K/N transform is enabled by receipt of an S8ENABLE/ (low). The S8ENABLE/ (high) disables the K/N transform during the clear-K-register, clear-N-register, and clear-N-and-page-register commands. These clear commands allow all zeros to be loaded into the respective K or N register.

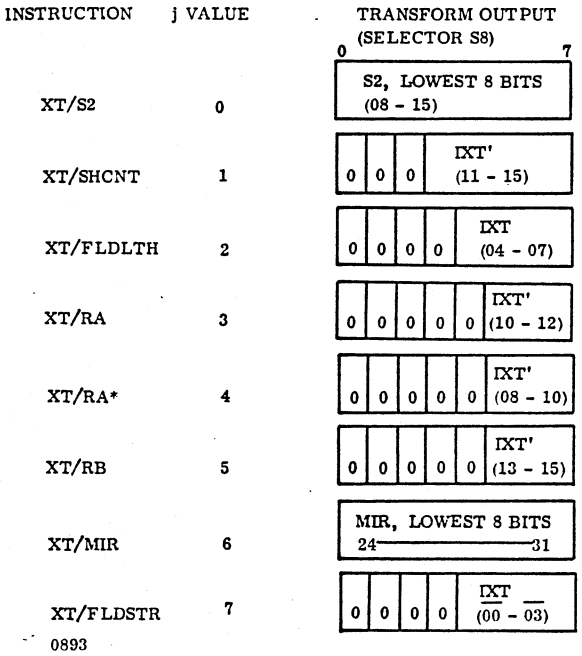


Figure 4-10. K/N Transforms

BIT TEST DECODER

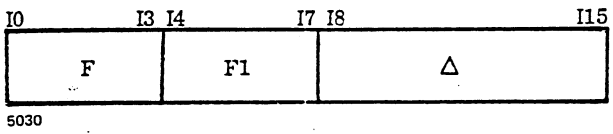
The bit test decoder (selector S7) is a 1-bit-wide selector that provides for up to 16 different conditions (external/internal) to be tested. These tests determine whether the upper or lower micro instruction shall be executed from the next micro-instruction pair. The test bit is selected by the least significant four bits of the micro-instruction register (MIR28 through MIR31). The bit test output is sent to the T field test multiplexer on the control 2 module and is tested if the T field of the micro instruction contains a BTU command. Table 4-8 lists the bit conditions to be tested by the 1700 emulator during the emulation process.

MIR ENCODE

During GITMAK/XT command the macro instruction is encoded from the F1 bits of the 1700 instruction. The F1 bit values select the various configurations of the upper 16 bits (MM00 through MM15) that are loaded into the CPU MIR register. These MIR bits control the arithmetic functions for read-next-instruction cycles and other required operations to provide efficient execution. The various types of MIR transforms, based on the macro instruction being emulated, are listed in tables 4-5, 4-6, and 4-7 for storage, register, and inter-register reference instructions, respec- tively. Figures provided in section 5 show the MIR transform selections for storage reference instructions and the MIR transform selections for register reference instructions. These figures illustrate the selection conditions and not the sequential steps that select the MIR transform.

DELTA/DECIMAL-CORRECTION TRANSLATOR

To emulate certain basic and enhanced 1700 instructions, the delta field of the macro instruction (figure 4-11) must be modified:



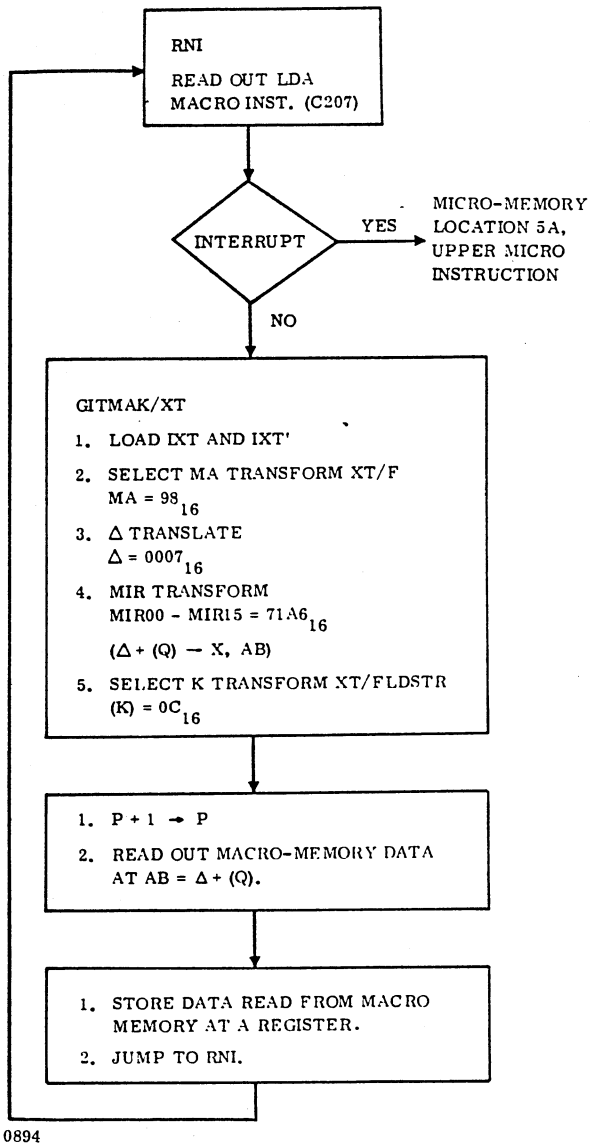


Figure 4-11. Step-by-Step Emulation of Macro-Instruction LDA

TABLE 4-8. EMULATION BIT TEST CONDITIONS

Test Bit	Operation	Selector S7	
		Pin	Position
BTU00	Not assigned	8	0
I02	Execute upper micro instruction if I02 is a 1.	7	1
I07	Execute upper micro instruction if I07 is a 1.	6	2
I06	Execute upper micro instruction if I06 is a 1.	5	3
IND00FF	Execute upper micro instruction if STORE 00FF (index 1) status is true.	4	4
SM105/ (PROTECT FAULT)	Execute lower micro instruction if storage-protect fault is detected.	3	5
SELSTOP	Execute upper micro instruction if selective-stop switch is set.	2	6
SELSKIP/	Execute lower micro instruction if selective-skip switch is set.	1	7
SM108 (PARITY ERROR)	Execute lower micro instruction if storage-parity error is detected.	23	8
BTU09	Not assigned	22	9
DELTA'=0	Execute upper micro instruction if delta equals 0 (IXT8 through IXT15 = 0)	21	10
EA=OPER	Execute lower micro instruction if the effective address equals the operand.	20	11
EVENPAR	Execute upper micro instruction if memory-parity line is true (even parity).	19	12
I00	Execute upper micro instruction if I00 is a 1.	18	13
MULTIND	Execute upper micro instruction if multilevel direct address mode is selected.	17	14
SM105+SM108	Execute upper micro instruction if previous macro memory write cycle was aborted (caused either by parity error or protect fault).	16	15

This modification is necessary before the delta field can be processed for operations indicated by MIR transforms. Table 4-9 lists the conditions and modified delta field.

The delta translator translates the delta field according to the type of macro instructions being emulated as indicated in table 4-6. The delta translator is enabled by conditions of SM107, SM111, F field, and F1 field:

SM107 is not set (decimal arithmetic correction logic is disabled).

SM111 is not set (the CPU file F1 output to selectors S1 and S2 is disabled, and delta-translator output to selectors S1 and S2 is enabled).

F = 0 or F1 = 1000 is low (the inter-register reference instructon is not selected).

When an inter-register reference instruction is emulated, the delta translator is disabled. This causes the delta ($FFFF_{16}$) to be sent to selectors S1 and S2.

The decimal-correction logic of the translator is enabled when SM107 is set and SM111 is not set. During emulation of 1700 decimal arithmetic functions, this correction logic determines when to correct illegal characters (A-F) into decimal numbers by adding or subtracting six (0110_{16}) to the proper character(s).

READ-ONLY MICRO MEMORY

The micro memory of the 1700 transform module is a read-only memory (ROM) that has been preprogrammed with the 1700 instruction emulator. This ROM micro memory consists of 512 64-bit words (two pages). Each word consists of two micro instructions that are referred to as upper (32-bit) and lower (32-bit) instructions. Each word in the micro memory is addressed by the memory address bits (MA0 through MA7, and PG3). The MA0 through MA7 specify one of 256 words (micro-instruction pairs, 32-bit upper and 32-bit lower instruction within a page). PG3 selects the page (page 0 or 1) in which the instruction resides. Selection of the upper or lower micro instruction is determined by the T and T′ field decode (MIR16 through MIR18) of the previous micro instruction. The selected instruction is then loaded into the MIR register.

TABLE 4-9. DELTA TRANSLATIONS

Conditions	Delta (Δ)
a. (F=0 (F1 = 0xxx) b. Enhanced instructions: (F=0) (F1 = 4 + 5) (r = 0)†	Δ = 0 0 0 0 0 0 0 0 I08 I09 I10 I11 I12 I13 I14 I15
a. (F=0) (F1 = 1xxx) b. (F=0) (F1= 2 + 3) c. Enhanced instructions: (F=0) (F1 = 4 + 5) (r = 0)†	Δ (SE) (with sign extend) = C C C C C C C C I08 I09 I10 I11 I12 I13 I14 I15 Constant = I08
a. (F=0) (F1=1) b. (F=0) (F1= 0 + 6)	Δ (SK) (for skip instruction) = 0 0 0 0 0 0 0 0 0 0 0 0 I12 I13 I14 I15
a. (F=0) (F1=E)	Δ (INT) (for interrupt instruction) = 0 0 0 0 0 0 0 1 I08 I09 I10 I11 I12 I13 I14 I15
a. (F=0) (F1=8)	Δ (FFFF) = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

†Refer to the CYBER 18 processor reference manual for type 2 storage reference instructions, enhanced instructions, and field reference instructions. Flag r is the relative address flag represented by IXT08.

1700 EMULATOR

The 1700 emulator is the firmware program that emulates both the basic and enhanced 1700 computer instruction set. The emulator also contains firmware for handling I/O operations, macro and micro interrupts, auto-data-transfer (ADT) operations, scientific/commercial-application decimal arithmetic and panel simulation. The 1700 emulator consists of micro-memory subroutines that are preprogrammed into the read-only memory. These subroutines are completely contained in the two pages (0 and 1, 1024 32-bit micro instructions) of the ROM micro memory. The 1700 emulator consists of many micro-code subroutines. Each subroutine micro code performs a specific task, such as generating and controlling the operations required to emulate a macro instruction. For example, the micro-code subroutine titled the decode-next-instruction subroutine is executed at the beginning of every new macro-instruction emulation. This subroutine resides in micro location 058_{16} and 059_{16} and includes the following micro instructions:

Increment to next instruction (INI).

Read the next instruction (RNI) and check for interrupt. Go to process-micro/macro-interrupt subroutine if an interrupt occurred.

Transform GITMAK/XT on the next instruction (XNI) if no interrupt occurred.

At the end of every macro-instruction emulation, a jump command is used to branch to specific locations in the decode-next-instruction subroutine to start the emulation of the next macro instruction.

MISCELLANEOUS CIRCUITRY

Protect-Violation Detecting Circuit

To emulate the 1700 series computer-protect function, a combination of hardware and firmware (1700 emulator) is required. The 1700 computer incorporates a program-protect function that inhibits access into protected programs by unprotected programs. The function develops around a memory protect bit that is contained in each word of macro (main) memory. The protect-violation conditions detected by the 1700 transform hardware are:

An attempt to execute an unprotected, privileged instruction (EIN, IIN, SPB, CPB, EXT, or inter-register with destination M) with protect-switch set

An attempt to execute a protected instruction following the execution of an unprotected instruction

An attempt to execute an unprotected, miscellaneous instruction with the protect-switch set

All other violations, such as an attempt by an unprotected instruction to write into a protected storage location, are detected by the macro-memory interface hardware and processor hardware.

Interrupt Enable

The 1700 computer systems require that the interrupt function be activated after one instruction has been executed following the enable interrupt instruction (EIN). A combination of hardware and firmware is required to prevent the interrupt function from being activated during the execution of the instruction following EIN if the instruction is either:

A storage reference instruction with multilevel-indirect-address mode that requires more than one memory reference (more than one RNI cycle)

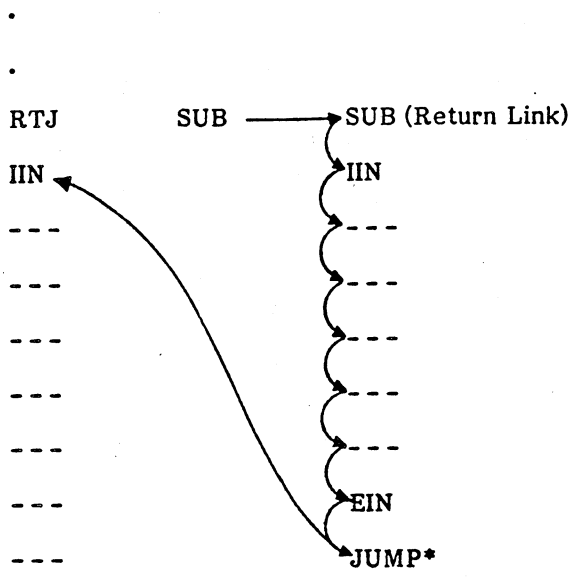
An enhanced instruction with double-word format

The implementation of the interrupt function is described below.

Status mode bit SM114 is first set by the 1700 emulator whenever the EIN is emulated to pre-enable the interrupt function. If the instruction following EIN is neither a storage reference instruction (F = 0) nor an enhanced instruction (except a type 2 skip instruction), the hardware enables the SETSM106 at the next RNI cycle, which in turn sets SM106 to enable the CPU interrupt function. If the instruction following EIN is either a storage reference or enhanced instruction, SM106 is set after the 1700 emulator has executed a SUB- operation (increment P counter). The SUB- operation indicates the end of the macro-instruction emulation. Once status mode bit SM106 is set, SM114 is cleared by the hardware.

1700 series computers also require that the inhibit interrupt instruction (IIN) take precedence over enable interrupt instructions (EIN). If the IIN instruction is executed after one instruction following an EIN (as in the following example), the macro-interrupt function must be disabled.

Main Program



The 1700 emulator implements the above example as follows: Whenever an interrupt is detected, the interrupt transform (XT/INT, MA transform 0) is performed. The XT/INT transform forms the micro-memory address of BC_{16} or BE for the micro interrupt (S211 = 0) or macro interrupt (S211 = 1), respectively. If the interrupt is a macro interrupt, the emulator ignores the macro interrupt and rereads the macro instruction. (This has the same effect as disabling the macro-interrupt function.) If the interrupt is a micro interrupt, the emulator processes the micro interrupt as normal.

If the macro interrupt being emulated is not an IIN instruction and there is no false interrupt, the XT/INT transform forms micro-memory address DB_{16} pr BF_{16} for the micro interrupt or macro interrupt, respectively. The interrupt is then processed accordingly by the micro- or macro-interrupt subroutine residing at the above address.

XTBLKT4 Signal Generation

During any operation except GITMAK/XT, the destination is decoded and strobed at time T4. However, during a GITMAK/XT transform operation, the previously read data is strobed into the destination register by the trailing edge of RESUME. (Refer to the processor manuals for more details.) This allows the correct delta field to be used in emulation of the macro instruction.

BLKM100 Signal Generation

During macro step mode, the macro-halt interrupt must be blocked while either a memory reference instruction or an enhanced instruction is emulated. The macro-halt interrupt is enabled only at the end of instruction execution. The following example with step-by-step operation of the emulator shows the need for the block mask bit 100 (BLKM100/) signal.

Assume that the machine is in step mode and is idling.

1. A macro GO command is executed (enter I: with function control register bit 12 clear).

2. The GO command sets SM215 (clear macro-halt interrupt).

3. The RNI cycle is executed to read the instruction. Assume that the instruction is a memory reference instruction with multilevel-indirect-address mode. Check for interrupt. Since SM215 is set, there is no macro-halt interrupt.

4. Perform the GITMAK/XT transform operation.

5. Hardware clears SM215 of either the breakpoint controller or the I/O-TTY mode whenever the CPU is in step mode and GITMAK/XT is executed. This sets the macro-halt interrupt (enabled by mask bit M100).

6. The emulator executes a multilevel-indirect-address subroutine to look for the effective address. The interrupt is also checked.

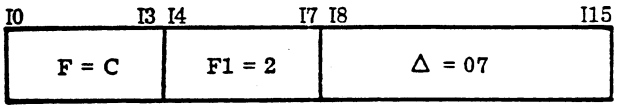
7. Since the macro interrupt is set, the emulator branches to the interrupt subroutine and the instruction cannot be completed.

To avoid the problem in step 7, BLKM100/ must be generated to block the macro-halt interrupt from being set to step 5. BLKM100/ is set low during a GITMAK/XT operation whenever a memory reference instruction or an enhanced instruction is decoded.

At the end of the above macro-instruction emulation, the emulator executes a SUB- operation (increment P counter), which sets the BLKM100/ signal high. BLKM100/ is also set high by the master clear.

TYPICAL 1700 MACRO-INSTRUCTION EMULATION

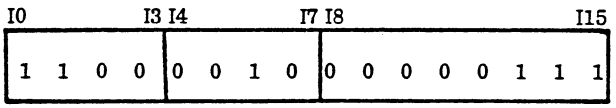
The following 1700 macro instruction, load A register, is selected to demonstrate the transform operation and the emulation of a macro instruction.



5032

The above macro instruction loads the A register with the contents of the storage location specified by the effective address (EA). EA is Δ + (Q). Figure 4-11 shows the step-by-step emulation of this macro instruction in the form of a flow chart. Figure 4-12 lists the macro-code subroutines required to emulate the instructions. First, the above instruction must be read out from macro memory using the read-next-instruction (RNI) cycle. Refer to the decode-next-instruction subroutine at location 058_{16}. The interrupt is also checked. Assuming that there is no interrupt at this time, then the transform-next-instruction (XNI) cycle (the GITMAK/XT command) is executed. Execution of the GITMAK/XT command causes the following sequence of events.

1. Data from memory is loaded into the IXT and IST' registers. The IXT register now contains:



5033

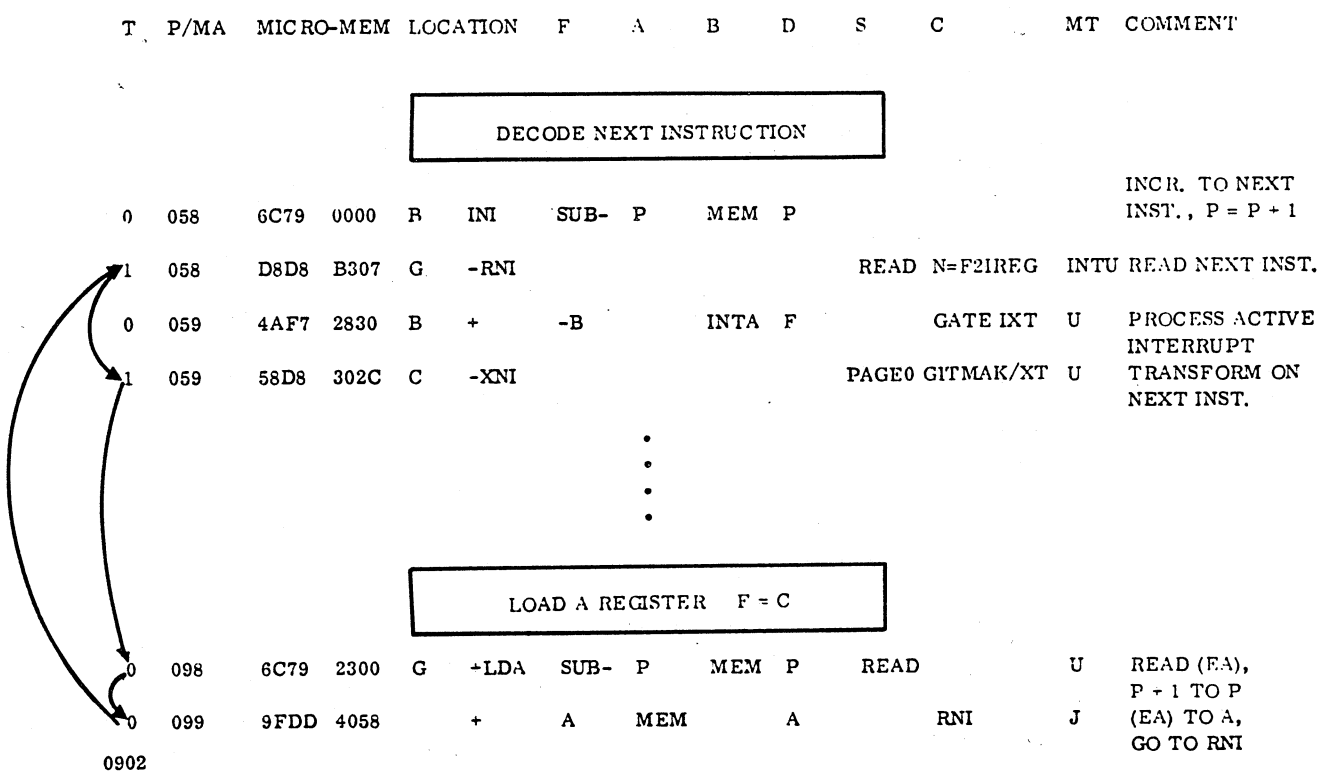
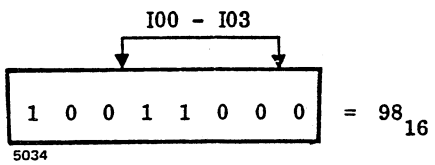


Figure 4-12. Micro-Code Subroutines to Emulate LDA Micro Instructions

2. The output of the IXT register is decoded into j value select signals to select the MA transform. Refer to section 5 for MA transform selection. For the macro instruction used in this example, the XT/F MA transform is selected since j value equals B_{16}. The output of the MA transform becomes:



which is then applied to CPU selector S6 to designate the new micro-memory address (MA). The new micro-memory address, 98_{16}, points to the load-A-register subroutine of the 1700 emulator.

3. If required, the outputs of the IXT and IXT' registers are also sent out to the delta translator circuit to be modified. In this example the output of the delta translator contains the following:

Δ = 0 0 0 0 0 0 0 0 I08 I09 I10 I11 I12 I13 I14 I15
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1

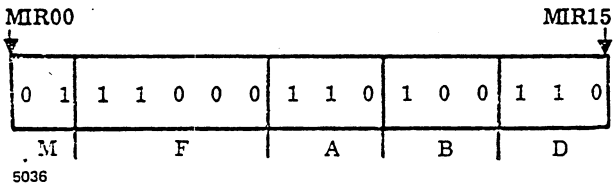
5035

Refer to table 4-9 for more detail.

4. Simultaneously with the operations in step 2 and 3, the output of IXT is encoded to form the upper 16 bits (MM00 through MM15) of micro memory. Refer to section 5.

   During the GITMAK/XT operation, the read-only memory ICs containing the upper 16 bits of micro memory data are disabled to allow the output of the MIR encoder to be loaded into MIR at time T5.

   The upper 16 bits of MIR now contain:



   Where:

   F = 11000 indicates that this is an A+B operation

   A = 110 selects the output of file 1 as the A input to the ALU. Since status mode bit SM111 is clear, the output of the delta translator is used as the A input.

   B = 100 selects the Q register as the B input to the ALU.

   D = 110 transfers the results of the add operation to X and the macro-memory address AB registers.

When executed, the above 16 bits form the effective address Δ + (Q).

5. K transform 7 (XT/FLDSTR) is always selected during the GITMAK/XT opration (that is, the K register is loaded with $0C_{16}$). However, since SM113 is not set by the emulator, the contents of the K register is not used for emulation of this instruction.

After the GITMAK/XT transform operation is completed, the upper micro instruction of the load-A-register subroutine at location 98_{16} is loaded into the MIR and executed. From this micro instruction, the P counter is incremented by 1; and data from macro memory at address + (Q), determined by the content of the macro-memory address register AB, is read out. The upper micro instruction is executed next. From this micro instruction, the data read from macro memory is loaded into the A register and a jump is made to the RNI cycle. This completes the emulation of the load-A-register (LDA) macro instruction.

I/O-TTY CONTROLLER

The I/O-TTY controller module provides the following basic functions:

Real-time clock - The real-time clock, in conjunction with the micro code, appears as a 1700-type peripheral to the macro-level programmer. The real-time clock is used mainly to measure elapsed time.

I/O-Teletypewriter/Console Display Controller - The teletypewriter to be used is the Teletype Model ASR/KSR 33/35. The console display is the CONTROL DATA Model 92423 Conversational Display Terminal.

Internal Peripherl Controller Bus - This function provides all input/output data lines, interrupts, and control signals necessary to generate, in conjunction with the 1700 emulator, both an internal CDC 1700 A/Q bus and the NCR M05 set/sample I/O bus.

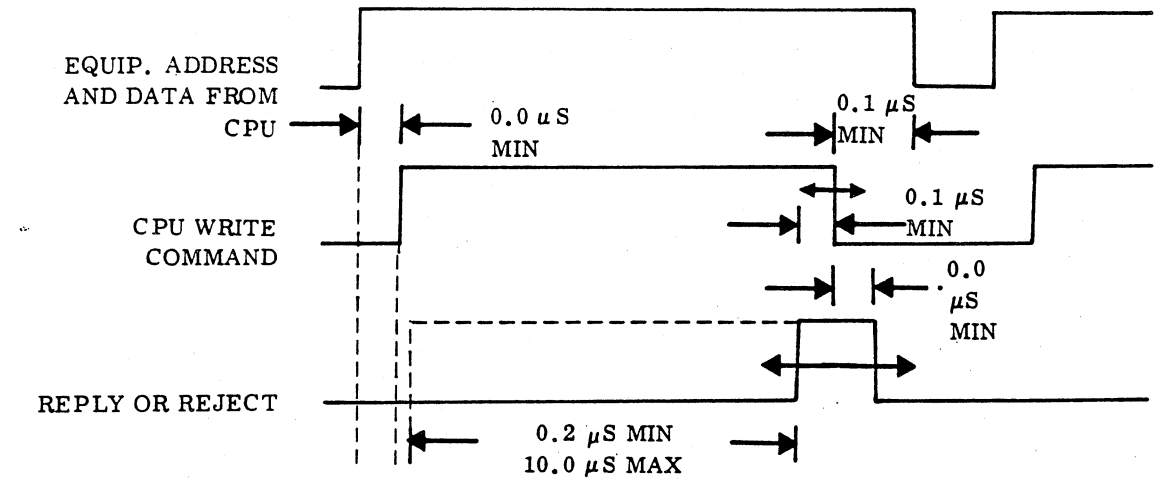
Breakpoint Controller Simulation - This option allows the operator to use the programmers console in panel mode to control and display all MP processor registers and memory without the need for a breakpoint controller. This option is a functon of firmware.

The module contains two registers that are required during 1700 A/Q and NCR M05 I/O operations. The input/output data register (RD) is used as the output data register and corresponds to the A register in a CDC 1700 Series computer. The input/output address register (ADR) is used as the peripheral address register and, if the processor is operating as a 1700 emulator, corresponds to the Q register in a 1700 Series computer.

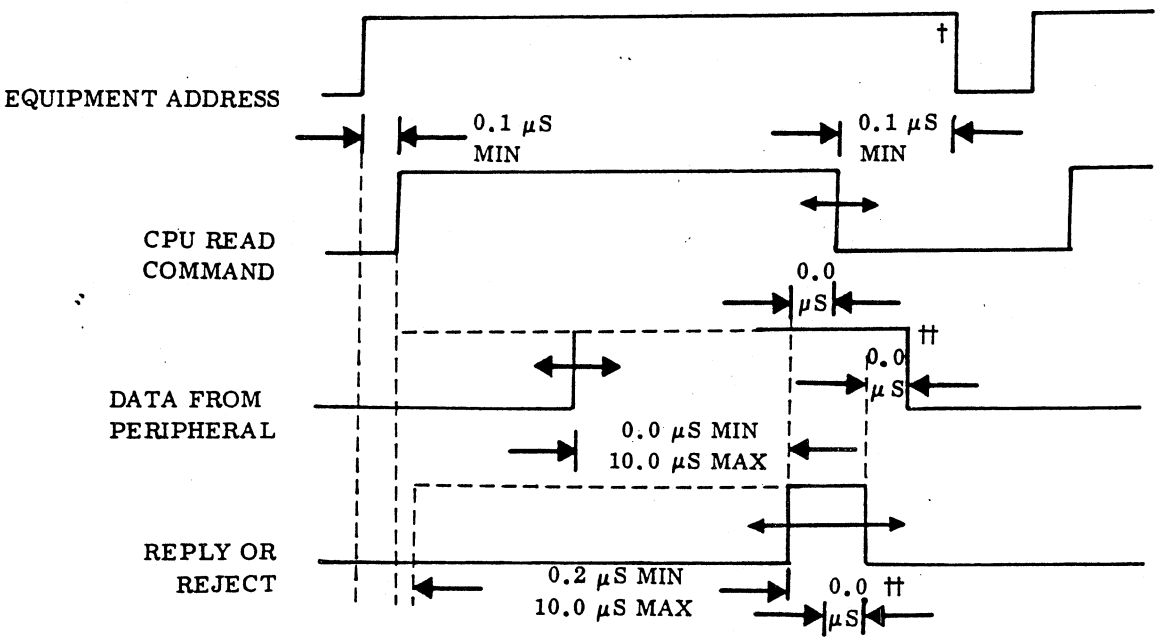
The I/O-TTY controller contains all the logic for the program control of data to and from the computer peripheral I/O controllers (A/Q and M05); of the internal TTY/CD control, data, and status, and generation of the real-time clock elapse time base and interrupts. This controller, in conjunction with the basic microprocessor, 1700 emulator, and peripheral I/O controllers, provides an I/O technique that is fully compatible with the 1700 A/Q and M05 I/O schemes.

The A/Q scheme utilizes standard Control Data data/address word and reply/reject conventions to process and address I/O data within the processor (figure 4-13). Figure 4-14 illustrates the overall TTY controller timing.

WRITE COMMAND



READ COMMAND



† ADDRESS AND DATA LINES NEED NOT DROP AFTER THE CPU READ COMMAND DROPS, AND IN GENERAL, THEY WILL NOT CHANGE UNTIL THE BEGINNING OF THE NEXT I/O OPERATION.

†† THERE ARE NO MAXIMUM TIMES SPECIFIED FOR THESE EVENTS BUT IT IS ESSENTIAL THAT THEY BE KEPT TO AN ABSOLUTE MINIMUM. THEREFORE, IT IS STRONGLY RECOMMENDED THAT COMBINATIONAL LOGIC BE USED TO CONTROL THESE SIGNALS.

0114

Figure 4-13. A/Q I/O Timing

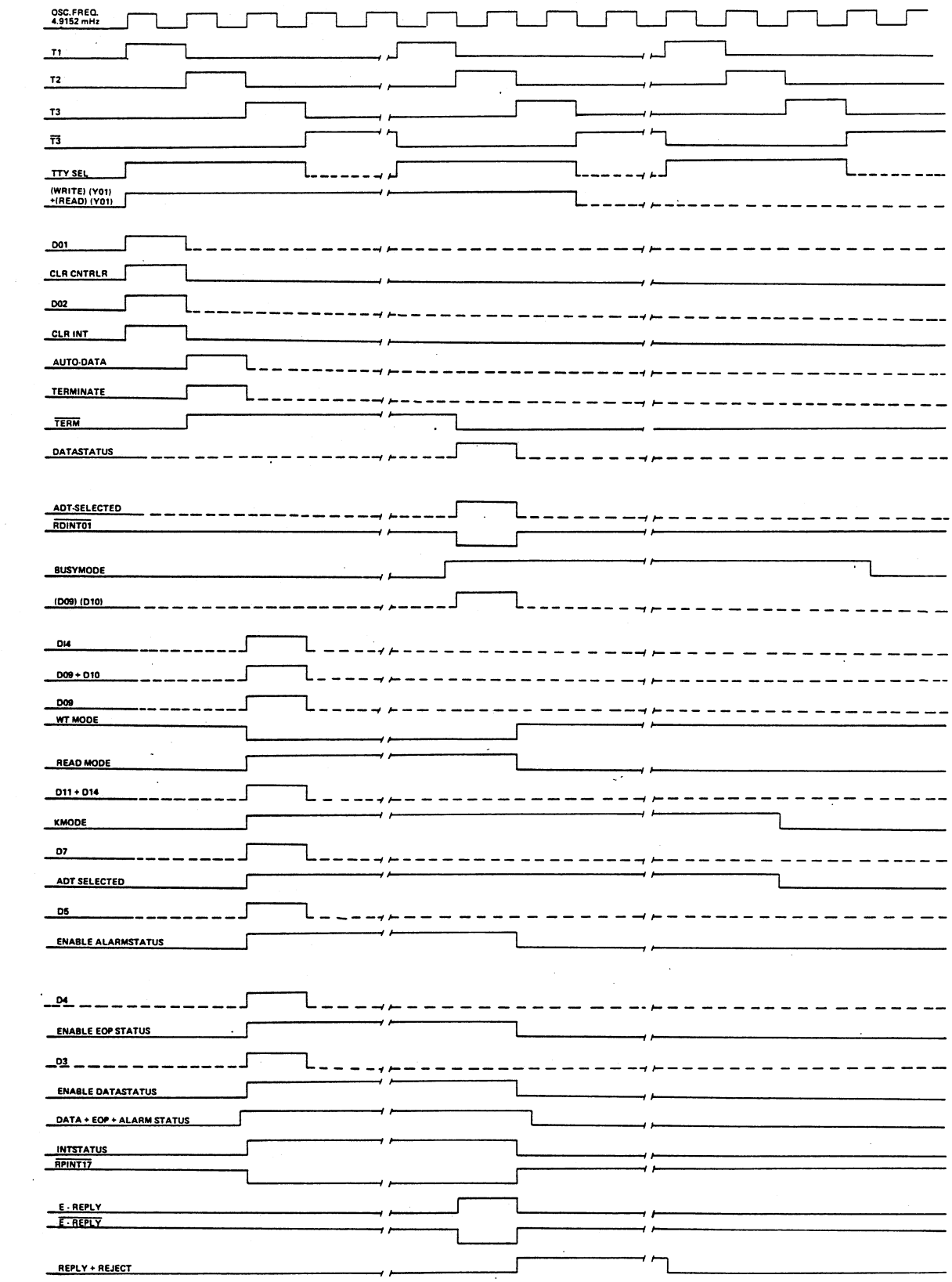


Figure 4-14. Overall TTY Controller Timing

The M05 scheme employs a set/sample (write/read) technique whereby the data is placed on the input/output line and remains until a strobe pulse occurs within the I/O timeframe (figure 4-15).

PERIPHERAL I/O CONTROLLERS

The data signal flow block diagram for the I/O-TTY controller is presented in figure 4-16. Figure 4-17 is the control line signal flow block diagram.

1700 Computer A/Q Scheme

All write data and address words transferred from the CPU to the peripheral I/O controllers are coupled through the I/O-TTY controller D and Y input registers. The D register holds false (complement) data; it couples the data words from the CPU A register to the peripheral I/O controller via the send lines (SD01 through SD16). The Y register holds true data; it couples address words from the CPU Q register to the peripheral I/O controllers via the address lines (ADR01 through ADR16). (Y register bits ADR12 through ADR16 are low only when the transfer (ADT) mode of operation has been selected.)

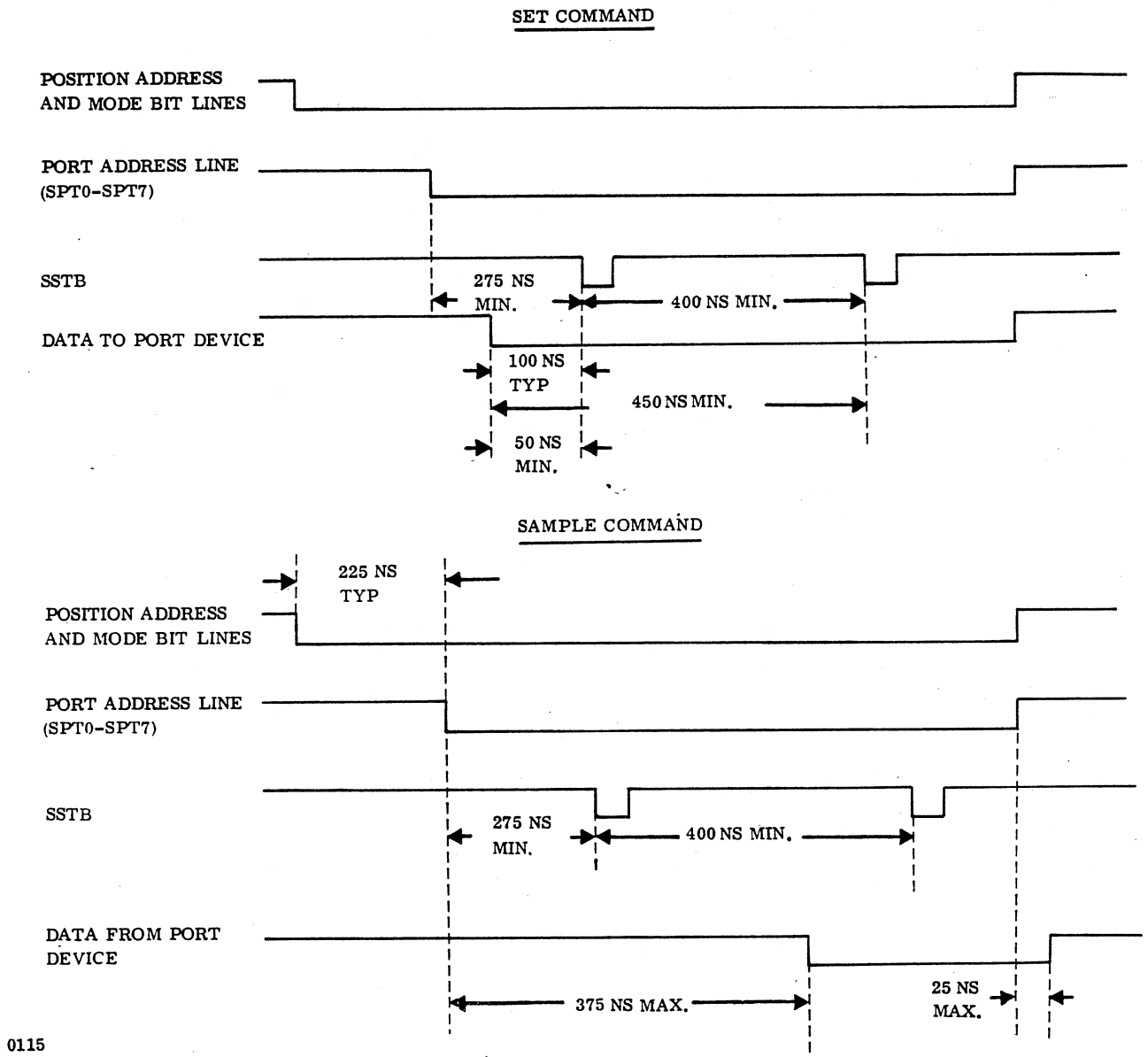


Figure 4-15. M05 I/O Timing

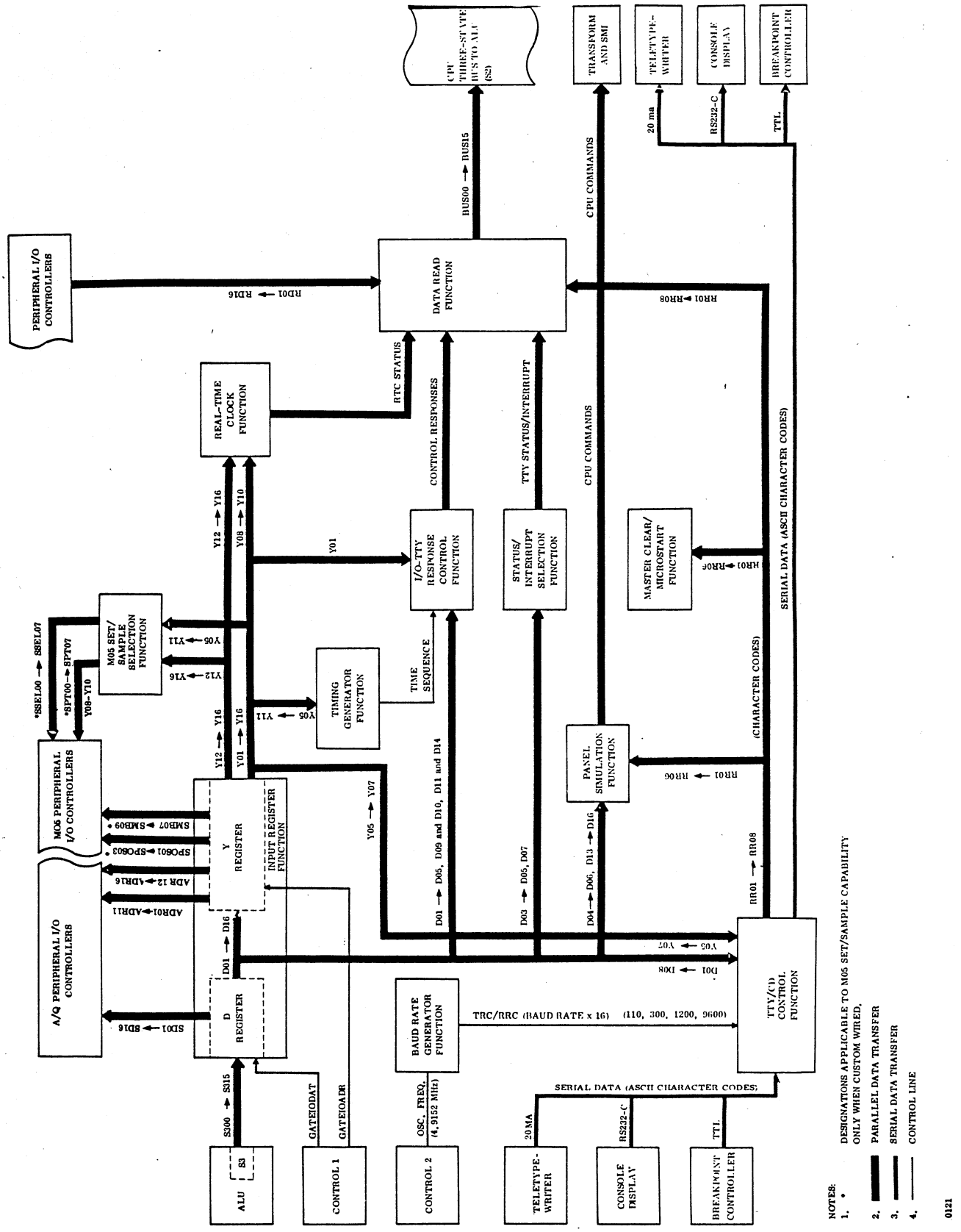


Figure 4-16. I/O-TTY Controller Data Signal Flow Block Diagram

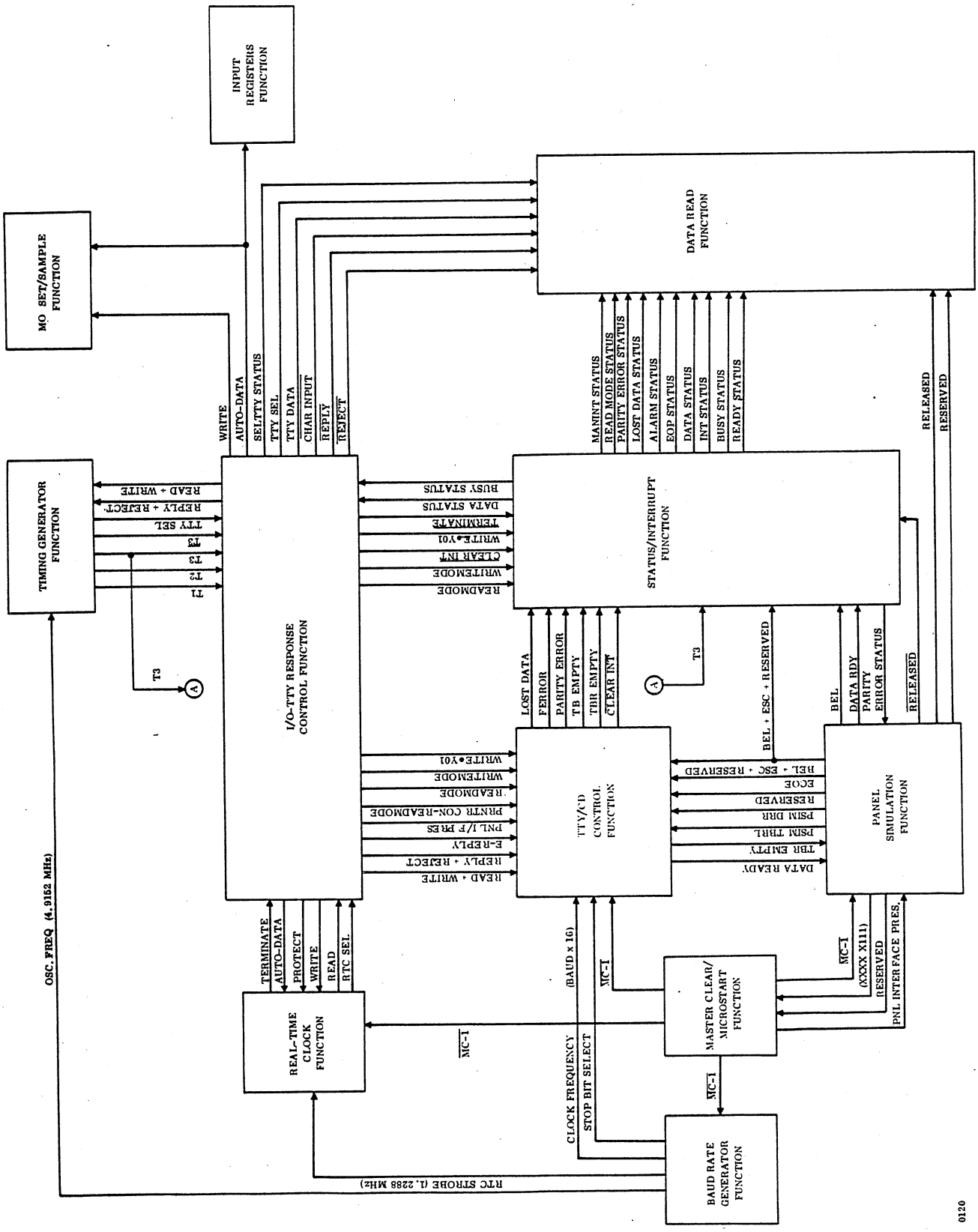


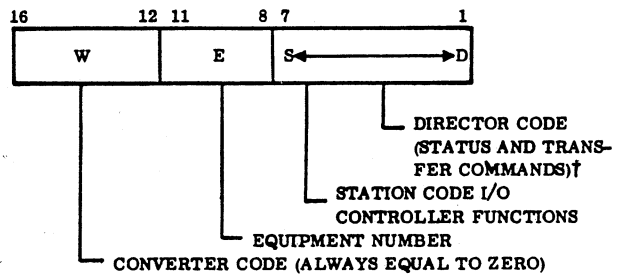
Figure 4-17. I/O-TTY Controller Control Line Flow Block Diagram

The data and address words are gated into the respective registers under control of the D′ micro instructon (000 = gate I/O data, 001 = gate I/O address). The D′ micro instruction is a 56-nanosecond negative-going pulse that strobes data from the CPU arithmetic/logical unit (ALU) into the D register when the leading edge swings negative. If D′ is equal to 001, the data is strobed from the D register to the Y register when the trailing edge of the gate poke swings positive. All read data is transferred via the open collector lines (RD01 through RD16) to the data read multiplexers. The read data is then transferred to the CPU three-state bus via bus lines BUS00 through BUS15 under control of the program-controlled data read multiplexers.

M05 SET/SAMPLE Scheme

The M05 set/sample scheme provides program-controlled selection of the active I/O port, line, device, and mode. The selection is deciphered from the contents of the port select (SPT00 through SPT07), line select (SSEL00 through SSEL07), position select (SPOS01 through SPOS03), and mode select (SMB07 through SMB09) lines.

The address word in the Y register is decoded in accordance with the CDC 1700 Series WES/D convention (figure 4-18).

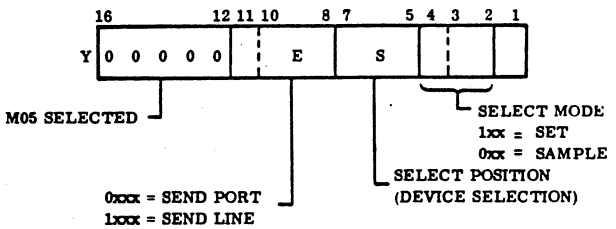


† The S and D fields have no bound, but may not overlap.

5028

Figure 4-18. WES/D Convention

The contents of the Y register indicate the selections as shown in figure 4-19.



5029

Figure 4-19. Y Register SET/SAMPLE Selections

The send port selection is designated by the octal value of Y-register bits 8 through 10 when Y11 is low. The line selection is designated by the octal code value of bits Y08 through Y10 when Y11 is high. The position (device) selection is designated by the octal code value of bits Y05 through Y07. The mode selection is determined by the state of bit Y04 (SMB09). When Y04 is high, the set mode is selected, placing the data word (16 bits) in the D register on the send lines (SD01 through SD16). When Y04 is low, the sample mode is selected; the data word on read data lines RD01 though RD16 is transferred via the open collector bus to the program-controlled data read multiplexers. The data is placed on the CPU three-state bus via bus lines BUS00 through BUS15 under control of the data read multiplexers.

Y register bits 5 through 16 determine whether the M05 set/sample scheme has been selected. This is indicated when the address word W-field bits (Y12 through Y16) are all low (W = 0), the E field (Y08 through Y11) hexadecimal value is greater than 8, and the S field (Y05 through Y07) is selecting a device.

Internal I/O Control

The internal TTY/CD control logic provides the data paths and control signals required for the man/machine communication link with the CPU. Y-register bits are applied to the timing generator and I/O-TTY response control functions to produce the time segment that ensures data transfer, reply, or reject responses are properly timed. The timing generator employs Y-register bits 5 through 7 to ensure that the TTY selected (TTYSEL) condition exists before generating time periods T1, T2, T3, and T3. The I/O-TTY response control function uses Y-register bit 1 to determine the write and read mode responses. (The real-time clock is described later in this section.)

The D register data bits are applied to the I/O-TTY response control, status/interrupt selection, panel simulation, and TTY/CD control functions to designate the director function and provide the character codes to the peripheral device. The I/O-TTY response control function deciphers the D register to determine the function designated, and the response provided agrees or disagrees to produce the proper reply or reject control response. Bits D03 through D05 and D07 indicate to the status/interrupt selection function the appropriate status and interrupt response. Bits D04 through D06 and D13 through D16 indicate to the panel simulation function the appropriate CPU command. D-register bits 1 through 8 are sent to the TTY/CD control function to provide character codes to the teletypewriter, console display, and/or breakpoint controller. The character code data bits are applied to the TTY/CD control function in parallel format and are strobed out to the teletypewriter, CD, and/or breakpoint controller in serial format, under control of the baud rate generator, at the rate of one bit per 16 Hz of clock frequence.

Character codes (ASCII codes) can be injected into the CPU from the teletypewriter, CD, or breakpoint controller. The character codes are clocked into the TTY/CD control function by the baud rate generator, where they are converted to parallel format (RR01 through RR08) and applied to the panel simulation, master clear/microstart, and data read functions. The panel simulation and master clear/microstart functions decipher the character data (RR01 through RR06) to provide the reserve (ESC), release (@), manual interrupt (BEL), and master clear (?) commands selectable at the keyboards. The data applied to the data read function is gated onto the CPU three-state bus.

The real-time clock decodes Y register data to determine whether the real-time clock function has been selected. The actual elapsed time pulses (3.3 ms) are processed by the ADT logic and only the status signals are coupled to the CPU.

The real-time clock status, control response, TTY status/interrupt, read data (RD01 through RD16), and character data (RR01 through RR08) conditions present at the data read function are multiplexed to the CPU three-state bus. Selection conditions applied to the data read function multiplexer indicate the transfer of one of the five data conditions to the bus lines (BUS00 through BUS15). These bus lines place the selected data conditions on the CPU three-state bus.

OPTIONS

In addition to the six standard modules that form the basic micro processor, the following optional modules are available for use in a complete application computer system.

BREAKPOINT CONTROLLER

The breakpoint controller (panel interface) provides the following functions:

Basic man/machine interface via the breakpoint panel (maintenance panel) and/or RS232-compatible programmers console. The interface provides for all common computer control panel functions such as start, stop, master clear, auto-restart, enter/display register, and enter/display macro and micro memory.

A data path for peripheral controllers to transmit ASCII characters to the processor to perform most operations that can be performed from the breakpoint panel or programmers console. This feature is normally used to load macro or micro memory from devices such as a card reader. This operation is referred to as deadstart operation.

Figure 4-20 shows the basic signal flow of the breakpoint controller module. For a complete description, refer to the breakpoint controller and breakpoint panel hardware reference/maintenance manual.

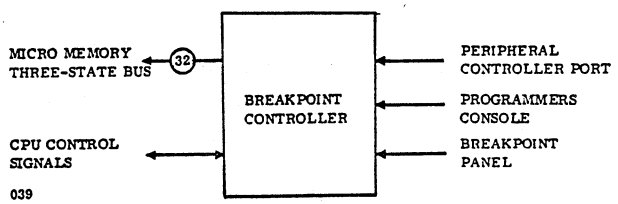


Figure 4-20. Basic Signal Flow

MOS MEMORY INTERFACE

The MOS memory comprises a data interface, address interface, and one to four MOS memory arrays. The address interface provides timing and control for the MOS memory array, processes memory requests, and processes addresses from the CPU and peripheral controllers. The data interface provides processing of the write and read data transfers.

The MOS memory is configured as a one-bank, two-port memory (local operation) that can be expanded to two banks and three ports (external operation). Each bank may have only one memory request active at one time. However, each bank operates independently, thereby permitting concurrent access by different ports (local CPU, local DMA, and external) to each bank.

Figure 4-21 illustrates the MOS memory configuration. Refer to the MOS memory subsystem hardware reference/maintenance manual for a detailed decription.

MNEMONICS

Refer to the glossary (appendix A) for a description of the processor external and internal signals.

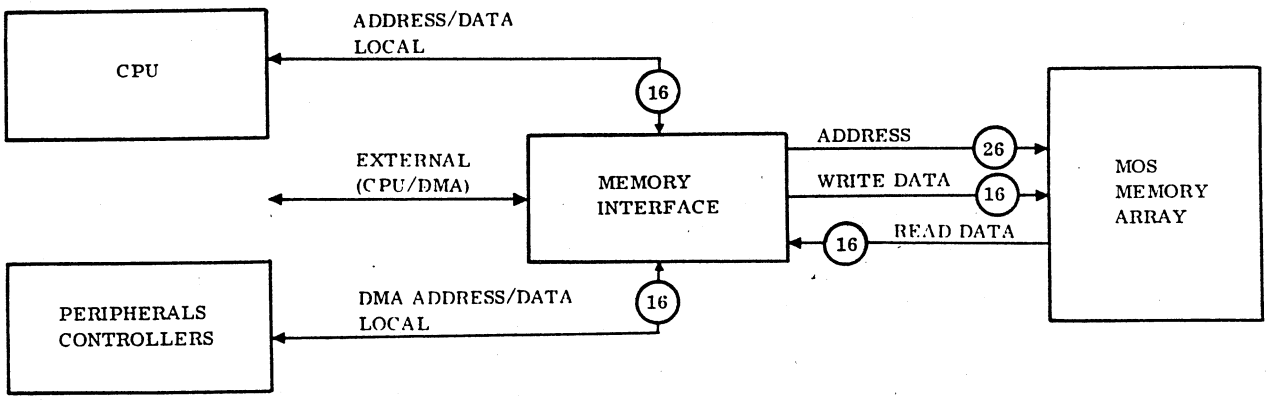


Figure 4-21. MOS Memory and Interface Basic Signal Flow

DIAGRAM DESCRIPTIONS 5

This section describes the modules (control 1, control 2, status mode interrupt (SMI), arithmetic/ logic unit (ALU), transform, and I/O TTY) of the basic MP processor and their circuit functions. The logic diagrams are contained in the field print package manual. Refer the to preface for publication title and number.

Note that within the processor logic, bit 00 is always the most significant bit and bit 15 (or 31) is always the least significant bit.

CONTROL 1

Figure 5-1 is a functional block diagram of the control 1 module. The locations of individual functional blocks within the logic diagram are indicated by the logic sheet numbers in the upper right corner of the boxes.

The control 1 module consists of the following functional areas:

MP processor operating control enables and fanouts: power up, master clear, micro halt, micro run, auto restart

Main timing generator

- Clock generator
- Odd/even time generator
- Extend timing circuits
- Micro memory time generator

Micro-instruction register (MIR02 through MIR15)

Decode logic for the A, B, D, and F fields of the micro instruction including the control signals for the selectors, registers, and arithmetic/logic unit (ALU)

- Data to enter during right and left shift of the A register
- Data to enter during right and left shift of the Q register

Overflow condition detector

Carry-in generator circuit for the ALU

MAIN TIMING GENERATOR

The main timing generator consists of the following circuits:

Clock generator

Odd/even time generator

Extend timing circuits

Micro-memory timing generator

CLOCK GENERATOR

The clock generator produces a chain of symmetrical square-wave clock pulses (CLK" at B2-8 and ODDCLK at C3-12) of 56 nanoseconds time duration. It also activates the shift clock and terminates scale during scale and shift operation. Refer to figure 5-2 for more details.

The clock pulses (ODDCLK) are coupled to the odd/even time generator, which generates the eight basic timing signals, T0 through T7. The timing signals are distributed throughout the processor and comprise the micro instruction basic execution time of 168 nanoseconds with extended timing depending upon the class of micro instruction. The CLK" pulses at B2-8 generate micro-memory timing and extend timing. CLK" is enabled only when T1 is high.

The clock generator consists of an oscillator network made up of a delay line and two NAND gates of B3. The NAND gate outputs are used to generate the master clock for control timing. The clock generator is started when the processor is placed in the micro-run mode (flip-flop A3-9 is high). Stopping the processor makes A3-9 low, which stops the clock generator. The $\overline{\text{MEM}}$ signal also stops the clock generator; it is present during a macro memory reference and causes the processor to stop after the current micro instruction is executed and wait for macro memory to resume (T2 and T3 high).

The clock generator is restarted when the processor receives the $\overline{\text{RESUME}}$ signal from macro memory. During other extend timing functions, the clock (ODDCLK) input to the odd/even time generator is disabled by the EXTEND signal (with T1 and T2 high). This clock input is enabled by the RESTART signal when the time extension expires.

Regardless of what operation is taking place, normal cycle or extended timing, the basic fetch-execute cycle remains the same. The instructions are always fetched from micro memory during T1 through T3 and are executed from T5 to next T5. Even during extended times, where additional execution time is required, the basic cycle uses the same time frame.

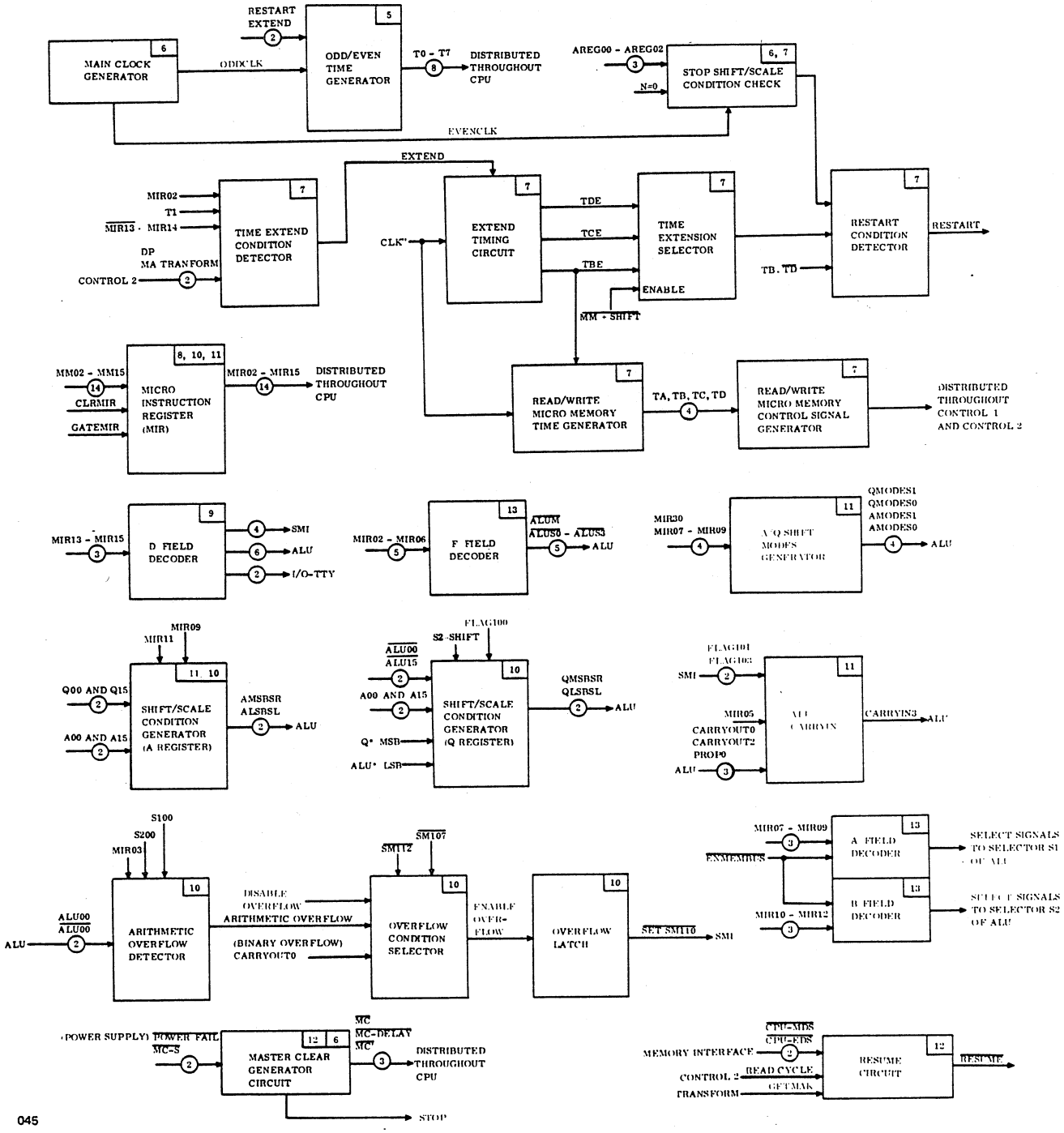


Figure 5-1. Control 1 Functional Block Diagram

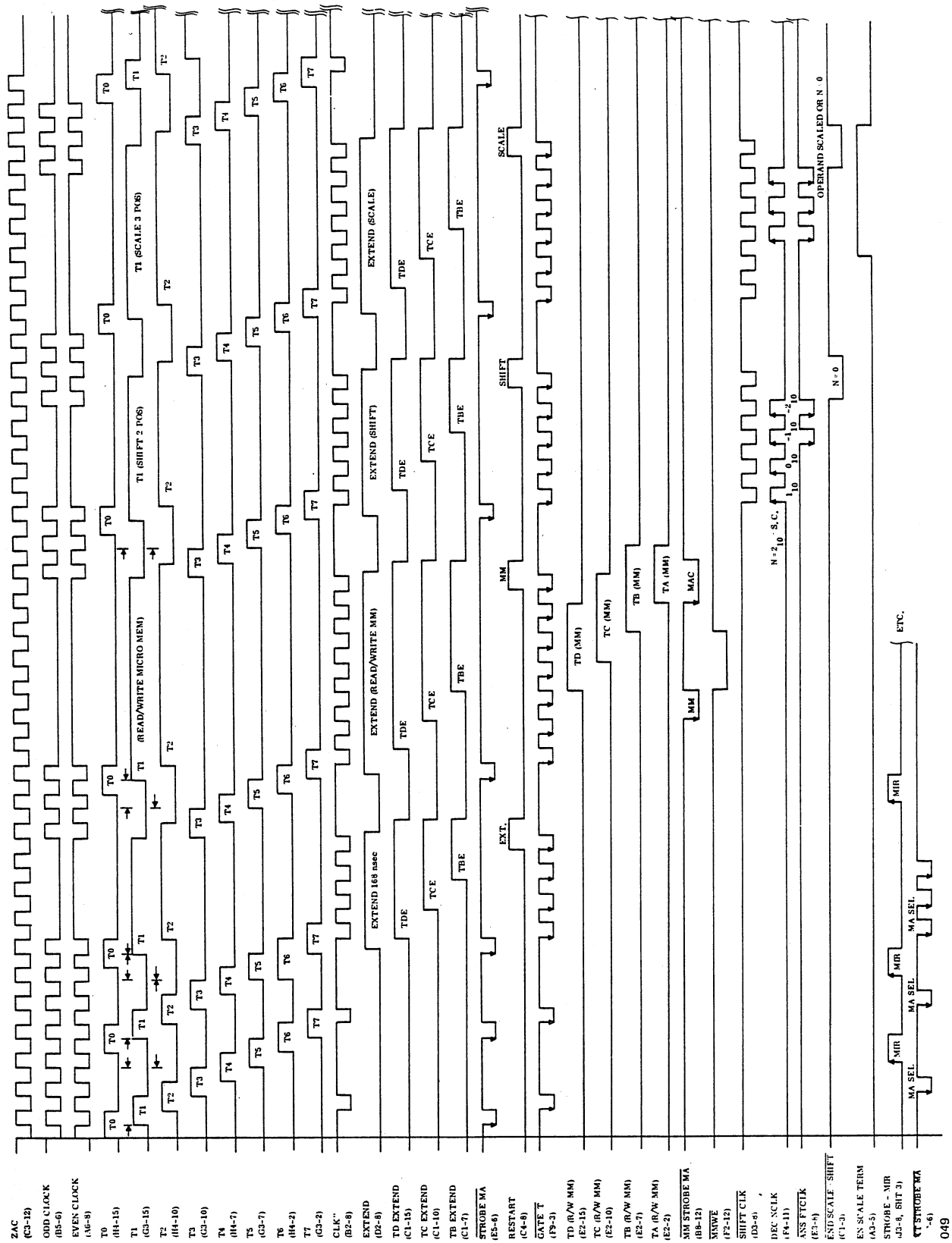


Figure 5-2. CPU Timing Control (Sheet 1 of 2)

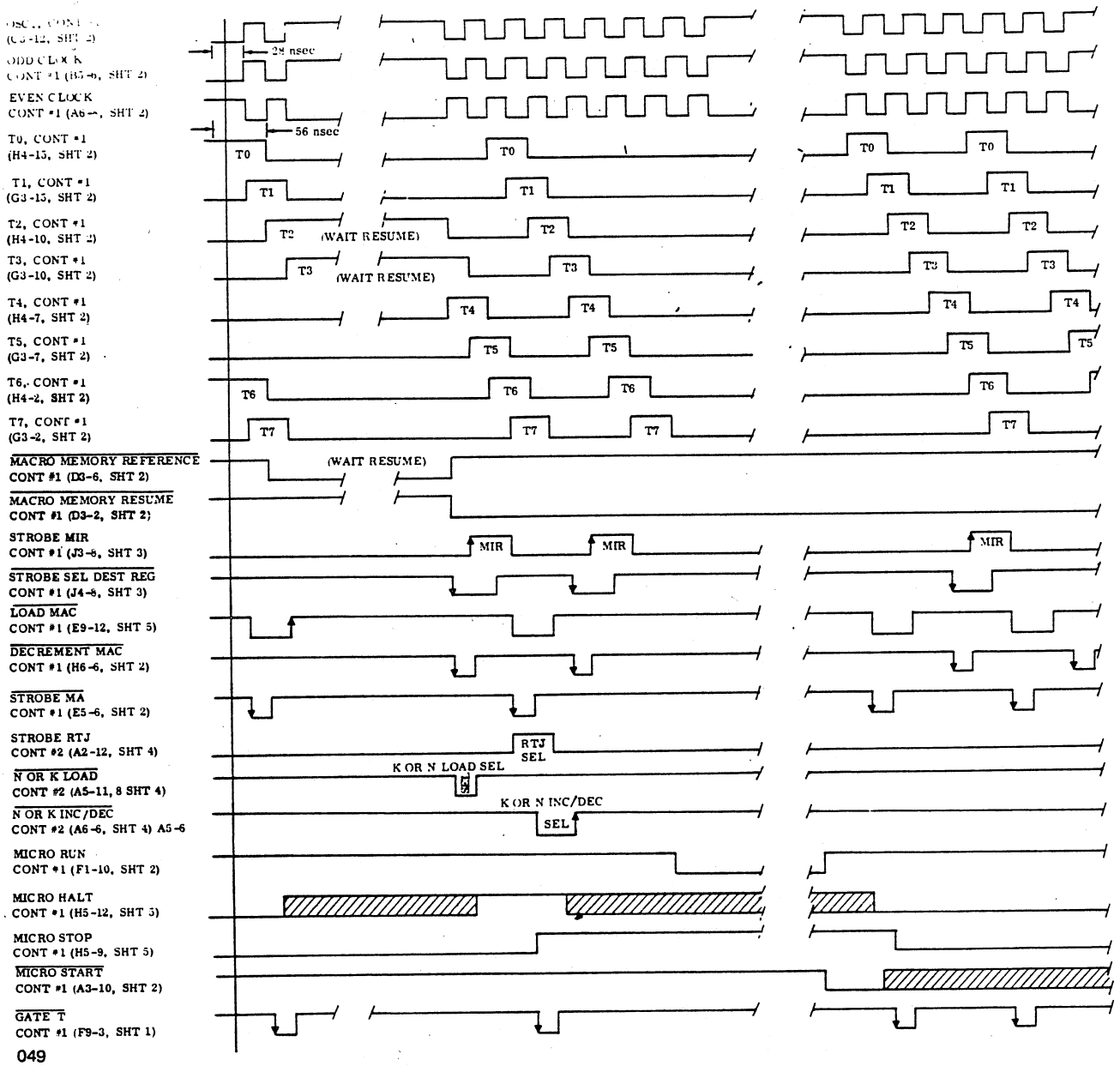


Figure 5-2. CPU Timing Control (Sheet 2 of 2)

Odd/Even Time Generator

The odd/even time generator furnishes the processor with the sequence timing signals used in executing the micro instruction. The odd/even time generator consists of eight flip-flops that are placed in different logic states by the odd and even clock pulses. These odd and even clocks are generated at B5-6 and A6-8, respectively. The outputs of the timing generator flip-flops, T0 through T7, are used throughout the processor to sequence processor functions. During an operation, T0 and T1 are coincident with T6 and T7. The eight flip-flops are prevented from changing states either when the STOP signal is present at G2-5 or when the clock generator output is disabled by the MICRORUN or $\overline{\text{MEM}}$ signal at B3 or by the EXTEND signal at B5-9. When the STOP signal is present, the inputs to the T0 flip-flop (H4-13) and the T1 flip-flop (G3-13) are low, which prevents T0 and T1 from occurring, thus causing the odd/even timing to stop when T6 and T7 are active without T0 and T1. This allows the processor to complete the micro instruction before being stopped. During processor stepping, the STOP signal is removed long enough to allow one complete cycle (T0 through T7) with each step.

Disabling the clock generator output (CLK) with the EXTEND signal causes the odd/even time generator to stop with T1 and T2 active. Disabling the clock generator output (CLK) with the MEM signal causes the odd/even time generator to stop with T2 and T3 active.

Extend Timing Circuits

The extend timing circuits are used to stop and restart the odd/even time generator, providing extensions of the normal processor sequence timing in 56-nanosecond increments. The extension is necessary to allow the completion of certain processor operations that exceed the basic micro-instruction execution time. The following processor operations require time extension of the odd/even time generator:

Memory address (MA) transform

Add and subtract micro instructions

Micro instructions specifying NU, ZL, COL, and Z*L (T field) (refer to section 2)

Micro instructions specifying A′ and B′ fields

Scale of shift operation

Micro-memory data read/write references

The EXTEND signal is generated at D2-8 during time T1 as follows:

$$\text{EXTEND} = \text{MAXFORM} \bullet T1 + \overline{\text{MIR13}} \bullet \text{MIR14}$$
$$\text{DP} \bullet T1 + (\text{LT+AP+BP}) \bullet T1 + \text{MIR02} \bullet T1$$

The extend time generator and decoder provide different time extensions beyond the basic micro-instruction execution time. The extend time generator consists of three flip-flops of C1 that furnish timing signals TDE, TCE, and TBE to the extend time decoder during the above processor operations. The extend time generator also enables the micro-memory time generator during micro-memory operand references (when TCE at C1-15 is high) and enables A and/or Q register shifting during shift or scale operations (C1-3 is high). The extend time decoder D1 selects the time extension from outputs of the extend time generator according to the specific function being performed (micro-instruction class A, B, C, or D). Refer to table 5-1. The decoder D1 is not used during micro-memory references and shift/scale operations, so it is disabled during these functions.

TABLE 5-1. TIME EXTENSION

Selector				Time Extension (Beyond 168 nanoseconds)	Micro-Instruction Class††
S3	S2	S1	S0		
0	0	0	0	00 ns†	A
0	0	0	1	56	B
0	0	1	0	112	C
0	0	1	1	112	C
0	1	0	0	56	B
0	1	0	1	112	C
0	1	1	0	112	C
0	1	1	1	168	D
1	0	0	0	00†	A
1	0	0	1	56	B
1	0	1	0	112	C
1	0	1	1	112	C
1	1	0	0	112	C
1	1	0	1	168	D
1	1	1	0	112	C
1	1	1	1	168	D

†Time extension is 00 ns because the EXTEND signal is false.

††Refer to micro-instruction classification in the CYBER 18 Processor with MOS Memory Reference Manual for more details.

The extend time generator is initiated when the EXTEND signal is true; it advances with each clock pulse CLK" (B2-8). The presence of the EXTEND signal also causes the odd/even time generator to stop at times T1 and T2. The extend time decoder generates a RESTART signal (except for micro-memory references and shift/scale micro instructions) after the required time for the processor function has elapsed. The RESTART signal enables the odd/even time generator to continue at T1 and T2. Figure 5-3 is a simplified extend timing circuit diagram; it shows the time extension beyond the basic 168 nanoseconds selected according to the specific processor operation to be performed.

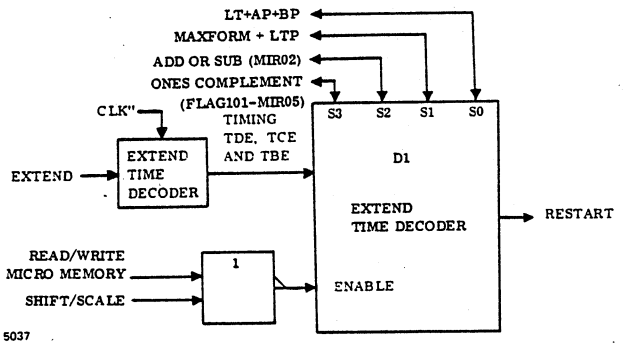


Figure 5-3. Simplified Extend Timing Circuit Diagram

The extend timing for a shift or scale operation is equal to 280 nanoseconds plus 56 nanoseconds for each shift. During a shift or scale operation, the EXTEND signal is true because MIR02 is true. The EXTEND signal initiates the extend time generator and stops the odd/even time generator at times T1 and T2. Output $\overline{\text{TERMINATE SCALE}}$ of the C4 flip-flop has already been set when C1-15 is low and remains set until the stop scale conditions are met (AREG00 ≠ AREG01). $\overline{\text{TERMINATE SCALE}}$ combines with the SC+SHIFT signal (F = 1111x) to enable the shift clock during time T1. Once enabled, the shift clock is produced each time the EVENCLK signal (C2-8) is received from the clock generator. Each shift clock pulse generates $\overline{\text{ANSHFTCLK}}$ and/or $\overline{\text{QSHIFTCLK}}$, which cause the appropriate register, A and/or Q, to scale left with each shift clock pulse. The scale operation continues until N equals 0 or A-register bits AREG00 and AREG01 are unequal (the scale complete condition). When the N register is equal to 0 or the scale is complete, ENDSHIFT/SCALE at C2-11 goes high. This high drives the terminate shift/scale flip-flop output at (C1-3) low, disabling $\overline{\text{ANSHIFTCLK}}$ and $\overline{\text{QSHIFTCLK}}$, and causing the RESTART signal to be generated. The stop scale when N equals 0 is necessary when the number to be scaled is all zeros or all ones (that is, the number cannot be scaled).

The scale flip-flop A3 is set after TDE is high during the scale operation to allow the scale complete conditions to be monitored. The shift operation is almost the same as the scale operation except that the shift clock also generates the NSHIFT clock, which causes the N register to decrement twice before the registers are actually shifted. (Refer to the shift/scale timing portion of figure 5-2, CPU Timing Control.) When TCE at C1-15 goes high, the $\overline{\text{ANSHFTCLK}}$ signal is generated and continues to decrement the N register until N equals 0. The $\overline{\text{QSHIFTCLK}}$ signal is generated as required in the same manner as $\overline{\text{ANSHFTCLK}}$.

During the shift operation AREG00 and AREG01 have no effect on the operation; the shift operation is stopped only when N equals 0. Extend timing for a macro memory request or an I/O operation is initiated by the $\overline{\text{MEM}}$ signal at D3-6 when it goes low at time T2. $\overline{\text{MEM}}$ is generated by the following equation:

$$\overline{\text{MEM}} = \overline{\text{WAITRESUM}} \bullet \overline{\text{RESUME}} \bullet \text{T2} \bullet$$

$$(\text{RESTART} + \overline{\text{EXTEND}})$$

The $\overline{\text{MEM}}$ signal stops the clock generator with the odd/even time generators at T2 and T3. If an extend operation is in progress, the $\overline{\text{MEM}}$ signal llops the clock generator when the RESTART, signal occurs. The clock generator is restarted again when the $\overline{\text{RESUME}}$ signal is received, causing the $\overline{\text{MEM}}$ signal to go high.

Micro-Memory Time Generator

The micro-memory time generator consists of four flip-flops of E2 that are used to generate timing during micro-memory read and write operations. When the micro-memory time generator is enabled (TCE signal at C1-15 goes high), CLK" pulses causes the flip-flops to change to logic states TA, TB, TC, and TD, representing the micro-memory time functions.

During the micro read or write operation, the extend time generator is initiated by the BP (read) or DP (write) signal and the odd/even time generator is stopped at T1 and T2. When output C1-11 of the extend time generator goes low, the micro-memory time generator is started. The extend time generator also generates the $\overline{\text{MMSTBMA}}$ (B8-12) and $\overline{\text{MMSTBPAGE}}$ (C8-8) signals, which in turn generate the GATEMA and GATEPAGE signals, respectively. GATEMA and GATEPAGE cause the

outputs of selector S6 to be gated to the PMA register; that is, either the output of S5 in the transform module or of the N/K register is selected, depending upon status mode bit SM113. The first CLK" pulse disables GATEMA and GATEPAGE and enables the $\overline{\text{MMWE}}$ signal if the write operation is specified. The $\overline{\text{MMWE}}$ signal allows the data to be written into micro memory. The S6CONS1 signal from the micro-memory time generator is transmitted to the control 2 module where it generates S6S1 and S6S0 = 00, enabling the next micro-memory sequential address to be selected from the MAC register. If a micro-memory read operation is specified, the $\overline{\text{MMGATEX}}$ signal causes the data to be gated from micro memory to the X register. At the same time, GATEMA and GATEPAGE are generated a second time and the $\overline{\text{ENTESTMUX}}$ signal is disabled, causing the next upper sequential address to be gated into the MA register. The micro-memory time generator produces a RESTART signal at time $\overline{\text{TD}}$ TB, which enables the odd/even time generator to continue from T1 and T2. The RESTART signal is reset at time T4. At T5, the micro-memory time generator returns to its initial state.

MICRO-INSTRUCTION REGISTER

The control 1 module contains ALU control bits 02 through 15 of the micro instructions. The outputs of the micro memory are latched in the D flip-flops by GATEMIR, which is generated at time $\overline{\text{T5}}$ or by $\overline{\text{CONGATMIR}}$ from the breakpoint controller.

D-FIELD DECODER

The D-field decoder translates the D field of the micro instruction into a signal that gates data to the appropriate destination. MIR13 through MIR15 are used to decode the destination field. The destination field consists of the D code and extended D′, D", and DD" codes as follows:

DP → D′ if S field = 1001 or 1010
DPP → D" if S field = 1011 Not applicable for
DDPP → DD" if S field = 0001 the basic processor

Refer to Micro Instructions, D field, in section 2 for more details.

The D code is decoded by a three- to eight-line decoder, L9, which is enabled at time T4 by the GATED signal and if DP-B or $\overline{\text{DPP-B}}$ code is selected. Table 5-2 lists the outputs of the L9 decoder.

TABLE 5-2. L9 OUTPUTS

MIR13	MIR14	MIR15	Output Pin	Operation†
0	0	0	15	NOP
0	0	1	14	P = (S3) → P, AB
0	1	0	13	I = (S1) → I, AB
0	1	1	12	Q = (S3) → Q, AB
1	0	0	11	F1 = (S3) → F1, AB via frequency
1	0	1	10	A = (S3) → A, AB
1	1	0	9	X = (S3) → X, AB
1	1	1	7	F = (S3) → F, AB

†( ) denotes output
AB denotes macro memory address buffer register

The outputs of L9 are used to generate the necessary gating signals for the P, I, Q, F, A, and X registers as well as the WEF1 signal for writing into F1.

Similarly, the D′ code is decoded by the three- to eight-line decoder, K10. The gating signals necessary to perform the operation are generated from outputs of K10 as shown in table 5-3.

The gating signal $\overline{\text{GATEAB}}$ gates the output of S3 to the AB macro memory address buffer register. This signal is generated at time T4′* whenever the destination field contains the D codes, except when the D code equals 0000, which is the NOP command, or when it is blocked by the $\overline{\text{BLKABFF}}$ signal at K7-5. This signal is low if a D-field command to load the AB is issued in the next micro instruction following the micro instruction with a D code of 001. (Transfer the output of S3 to P, AB.) Refer to D code in the CYBER 18 Processor Reference Manual for more details.

The $\overline{\text{GATEAB}}$ signal is generated as follows:

$$\overline{\text{GATEAB}} = \text{T4′*} \bullet (\text{MIR13} + \text{MIR14} + \text{MIR15}) \; (\text{DP} \bullet \overline{\text{DPP}} \bullet \overline{\text{DDPP}}) \bullet \overline{\text{BLKABFF}}$$

F-FIELD DECODER

The F field (bits MIR02 through MIR06) specifies the logic or arithmetic operation to be performed by the A and Q registers. Table 5-4 lists the translation of F codes into the ALU control signals.

A AND Q REGISTER CONTROL SIGNALS

During the shift or scale operation (F = 1111x) or during the shift operation of the Q register in conjunction with a destination (C′ = 111xxxx), MIR07 through MIR09 and MIR30 are used to generate the control signals for the A and Q registers.

The control signals for shift/scale operations of the A and Q register are generated by the quad 2 input multiplexer A9. QMODES1, QMODES0, AMODES1, and AMODES0 are used to control the Q and A registers, respectively. Refer to the A and Q register descriptions for the ALU module for more details. Control signals AMODES1 and AMODES0 are always decoded from MIR07 and MIR08 when $\overline{\text{SC+SHIFT}}$ is low (A or A/Q shift independently of ALU). Control signals QMODES1 and QMODES0 are decoded from MIR07 and MIR08 when $\overline{\text{SC+SHIFT}}$ is low and from MIR30 when $\overline{\text{SC+SHIFT}}$ is high (Q is shifted in conjunction with a destination register via S3). These control signals are generated as shown in table 5-5.

TABLE 5-3. K10 OUTPUTS

MIR13	MIR14	MIR15	Output Pin	Operation†	Gating Signals
0	0	0	15	(S3) → D register	$\overline{\text{GATE I/O DATA}}$
0	0	1	14	(S3) → Y register	$\overline{\text{GATE I/O ADR}}$
0	1	0	13	Not connected	-
0	1	1	12	Not connected	-
1	0	0	11	(ALU) → M1	$\overline{\text{GATEM1}}$
1	0	1	10	(ALU) → SM1	$\overline{\text{GATESM1}}$
1	1	0	9	(ALU) → M2	$\overline{\text{GATEM2}}$
1	1	1	7	(ALU) → SM2	$\overline{\text{GATESM2}}$

†( ) denotes contents of

TABLE 5-4. CONVERSION OF F CODE INTO ALU CONTROL SIGNALS

F Code MIR02-MIR06										
Bit	2	3	4	5	6	$\overline{ALUM}$	$\overline{ALUS3}$	$\overline{ALUS2}$	$\overline{ALUS1}$	$\overline{ALUS0}$
	0	1	1	0	0	0	0	0	1	1
	0	1	1	1	0	0	0	0	0	1
	0	1	1	0	1	0	0	0	1	0
	0	1	1	1	1	0	0	0	0	0
	0	1	0	0	0	0	0	1	1	1
	0	1	0	1	0	0	0	1	0	0
	0	1	0	0	1	0	0	1	1	0
	0	1	0	1	1	0	0	1	0	0
	0	0	1	0	0	0	1	0	1	1
	0	0	1	1	0	0	1	0	0	1
	0	0	1	0	1	0	1	0	1	0
	0	0	1	1	1	0	1	0	0	0
	0	0	0	0	0	0	1	1	1	1
	0	0	0	1	0	0	1	1	0	1
	0	0	0	0	1	0	1	1	1	0
	0	0	0	1	1	0	1	1	0	0
	1	0	1	0	0	1	1	0	0	1
	1	1	0	0	0	1	0	1	1	0
	1	0	1	0	1	1	1	0	0	1
	1	1	0	0	1	1	0	1	1	0
	1	0	1	1	0	1	1	0	0	1
	1	1	0	1	0	1	0	1	1	0
	1	0	1	1	1	1	1	0	0	1
	1	1	0	1	1	1	0	1	1	0
	1	1	1	1	0	1	0	0	0	0
	1	1	1	1	1	1	0	0	0	0

Note: All ALU control signals are complemented before connection to ALU chips. The ALU control signals are generated according to the following equations:

ALUS3 = MIR03
ALUS2 = MIR04
ALUS1 = MIR05 MIR02 + MIR04 MIR02
ALUS0 = MIR03 MIR02 + MIR06 MIR02
ALUM = MIR02

TABLE 5-5. A AND Q REGISTER CONTROL SIGNALS

SC+SHIFT (Pin 1)	QMODES1 (Pin 9)	QMODES0 (Pin 12)	AMODES1 (Pin 4)	AMODES0 (Pin 7)
Low	MIR07 or High†	MIR08 or High†	MIR08	MIR07
High	MIR30	MIR30	High	High

NOte: MIR30 = 0, Shift left
1, Shift right

MIR07 = 1, Shift or scale right
MIR08 = 1, Shift or scale left

†If MIR09 = 0, then QMODES1 and QMODES0 are both high since the Q register is inhibited to shift or scale.

A AND Q SHIFT/SCALE OPERATIONS

Data Input to A and Q

The appropriate data to be entered into the A register during right or left shift is generated by the dual four-to-one multiplexer C5. Refer to the micro-programming section for shift and scale operation codes and conditions. AMSBSR is data to enter during right shift and ALSBSL is data to enter during left shift of the A register. Output pin 7 (AMSBSR) is OR-wired with output pin 8 of D6 so that when MIR11 and MIR12 = 00, the A side of C5 is disabled and D6 is enabled, causing AMSBSR to be low. If MIR11 and MIR12 ≠ 0, AMSBSR and ALSBSL are generated as follows:

MIR11 (Pin 2)	MIR09 (Pin 14)	AMSBSR (C5, Pin 7)	ALSBSL C5, Pin 9)
0	0	A00 or low†	High
0	1	A00	Q00
1	0	A15	A00
1	1	Q15	Q00

Similarly, QMSBSR is data to enter during right shift and QLSBSL is data to enter during a left shift of the Q register. QMSBSR and QLSBSL are generated by the dual four-to-one multiplexer D5. D6 and the B side of D5 are enabled when S2=SHIFT is high or when $\overline{\text{MIR11}}$ is low.

Output pin 9 of D5 is OR-wired with output pin 11 of D6 so that ALSBSL is low when MIR11 = 0. QMSBSR and QLSBSL are selected by the S2=SHIFT signal.

When S2=SHIFT is high, Q is shifted in conjunction with a destination register, and when S2=SHIFT is low, Q is shifted in conjunction with the A register, independently of the ALU. QMSBSR and QLSBSL are generated as follows.

S2=SHIFT (Pin 2)	FLAG100 (Pin 14)	QMSBSR (D5, Pin 7)	QLSBSL (D5, Pin 9)
0	0	A15	A00 or low††
0†††	1	A15	A00
1†††	0	ALU15	ALU00
1	1	ALU*LSB	Q*MSB

Data Input to S3

Data enters the least significant bit of S3 when a shift-left operation is either the most significant bit of the Q register (Q00) or MIR31. This data, MIR31+Q00, is generated by A4, B4, and D7 according to the following equation:

$$\overline{\text{MIR31} + \text{Q00}} = \overline{(\text{MIR29}' + \text{MIR30}') \bullet \text{MIR31}' + (\overline{\text{MIR29}}' \bullet \overline{\text{MIR30}}')\ \text{Q00}}$$

This equation indicates that during the left shift of the destination register in conjunction with the Q register (MIR29 and MIR30 = 00), $\overline{\text{Q00}}$ will be entered. During the left shift of the destination register only (MIR29 and MIR30 ≠ 00), MIR31 will be entered into the least significant bit position of S3 in the ALU module.

†When MIR11 and MIR 12 = 00, AMSBSR is low regardless of MIR09.

††When S2=SHIFT is low and MIR11 is high, ALSBSR is low regardless of FLAG100.

†††These selections are applicable only when the double-precision mode is used and the double-precision hardware (ALU*) is installed. They are not presently available for the basic MP processor.

OVERFLOW DETECTOR

Status bit SM110 is set when an overflow condition is detected. There are two selectable overflow conditions: arithmetic overflow and binary overflow. CARRYOUT0 is generated in the ALU module. The arithmetic overflow is generated by the eight-input multiplexer C6 according to S100, S200, and MIR03. S100 and S200 represent the sign bits of operands A and B to the ALU, respectively. MIR03 determines the add (MIR03 = 1) or subtract (MIR03 = 0) operation. The arithmetic overflow is true whenever the arithmetic result is inconsistent with the sign of the operands and the arithmetic operation. The arithmetic overflow is generated as follows:

MIR03 (Pin 9)	S200 (Pin 10)	S100 (Pin 11)	Arithmetic Overflow (Pin 5)
0	0	0	Low (no overflow)
0	0	1	$\overline{ALU00}$
0	1	0	ALU00
0	1	1	Low (no overflow)
1	0	0	ALU00
1	0	1	Low (no overflow)
1	1	0	Low (no overflow)
1	1	1	$\overline{ALU00}$

The no overflow condition always exists for the subtract operation with two same-sign operands and for the add operation with two different sign operands.

The two overflow conditions are selected by the eight-to-one multiplexer B6 according to SM107, SM112, and DISABLEOVERFLOW. The DISABLEOVERFLOW signal is generated by the NAND gate J8 as follows:

$$DISABLEOVERFLOW = \overline{SC+SHIFT} \bullet MIR06 \bullet MIR02$$

Where: SC+SHIFT = High indicates this is not a shift or scale operation

MIR06 = High indicates this is an arithmetic operation and not a logical operation

MIR02 = High selects only the arithmetic operations with the overflow test

Refer to Micro Instructions, F code, in section 2 for more details.

Table 5-6 lists the overflow condition selected by B6.

TABLE 5-6. OVERFLOW CONDITION SELECTED BY B6

DISABLEOVERFLOW† (Pin 9)	SM112 (Pin 10)	SM107 (Pin 11)	ENOVERFLOW (Q) (Pin 5)	ENOVERFLOW (Pin 6)
0	0	0	Low	High
0	0	1	CARRYOUT0 (Binary Overflow)	$\overline{CARRYOUT0}$
0	1	0	Not available	
0	1	1	C6-5 (Arithmetic Overflow)	$\overline{C6-5}$
1	0	0	Low	High
1	0	1	Low	High
1	1	0	Low	High
1	1	1	Low	High

† The outputs of multiplexer B6 are connected to the J5 flip-flop to generate the SET SM110 signal, which is used directly to set SM110 (the overflow status mode bit) in the SMI module.

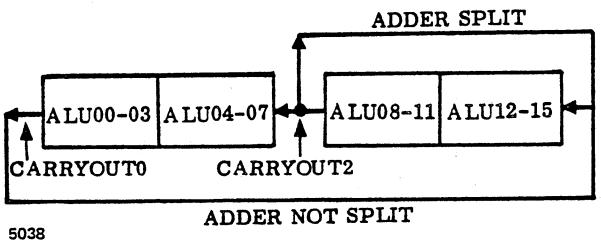
CARRYIN3 TO ALU CHIPS

The carry-in, CARRYIN3, for the lowest-order bit ALU chip is generated by the following equation:

```
CARRYIN3 = (MIR04 + MIR05 FLAG101) +
           (MIR05 • FLAG101) + (FLAG103 •
           FLAG101 • CARRYOUT2 +
           FLAG103 • FLAG101 • CARRYOUT0)
           +FLAG103 • FLAG101 • PROP0 •
           PROP1 + FLAG103 • FLAG101 •
           PROP1 • PROP21 • PROP31)
```

Since the ALU chips perform add and subtract operations in ones complement only, the appropriate CARRYIN3 signal is required, depending on the ones or twos complement mode, adder-split mode (16- or 32-bit machine), or a combination of these modes of operation. Refer to the arithmetic F codes and the arithmetic operation table of the ALU chip in the ALU module for more details.

The first two terms of the equation generate the forced carry CARRYIN3 to acquire the desired operation specified by the F codes, depending upon whether the machine is in ones or twos complement mode. If ones complement mode is selected, the carry-in generated by the third term is either CARRYOUT0, if the adder is not split, or CARRYOUT2, if the adder is 8/8 split as follows:



The fourth term generates a carry-in for the special case when the bit position of the B operand contains the complement of the corresponding bit in the A operand (for example, A = 1001 and B = 0110). As indicated in the logic drawing, $\overline{\text{PROP1}}$ at pin 207 is grounded for the 16-bit machine. $\overline{\text{PROP21}}$ at pin 10 and $\overline{\text{PROP31}}$ at pin 210 are grounded for the 32-bit machine.

STOP LOGIC

The stop logic is used to send a STOP signal to the control timing module any time the S field contains a HALT code and SM109 is set, a micro instruction is being stepped at the panel, or breakpoint sends a breakpoint stop signal. The STOP signal is generated at time T2 whenever the micro-halt conditions are met (K1-8 is high). The STOP signal stops the operation of the processor on completion of the micro instruction; for example, the timing generator is stopped at times T6 and T7 without T0 and T1. After halting, T6 and T7 are cleared.

MASTER CLEAR SIGNAL GENERATOR

The master clear signals are generated during normal master clear, power-up, and power-down as described below.

Master Clear

When the master clear switch on the control panel is pressed or a question mark (?) is typed on the console display (in panel mode), $\overline{\text{MC-S}}$ is generated at L6-9. When $\overline{\text{MC-S}}$ is low, an $\overline{\text{MC}}$ signal produced at L2-6 clears the memory interface so it does not accept any more requests. One-shot K2 is triggered when the trailing edge of MC-S goes from low to high, producing the MC-DELAY pulse at L2-3. This 16-microsecond $\overline{\text{MC-DELAY}}$ pulse is used to clear the SM1, control 1, control 2, transform, micro memory, breakpoint controller, and input/output modules. The trailing edge of $\overline{\text{MC-DELAY}}$ terminates the MC signal at L2-6.

Power-Up Mode

During power up, +5 V from the power supply is generated first, which causes the one-shot K2 to become a free-running oscillator. The oscillating signal at K2-9 causes the L1 flip-flop pin 5 to go low. This produces the $\overline{\text{MC}}$ signal at L2-6 and also $\overline{\text{MCDELAYED}}$. When the $\overline{\text{POWER FAIL}}$ signal goes high ($\pm$12 V and $\pm$15 V are available), it sets L1-5 to high, triggering the one-shot K2 to terminate $\overline{\text{MC-DELAY}}$ and $\overline{\text{MC}}$ after approximately 16 microseconds. The $\overline{\text{POWER FAIL}}$ signal also stops the free-running oscillator K2. If auto-restart is enabled (L1-12 is high), the $\overline{\text{SM215}}$ signal is produced, which sets macro-run status mode bit SM215. The micro-run signal at A3-9 is also set, allowing the basic clock to run.

Power-Down Mode

During the power down or power failure mode, the $\overline{\text{POWER FAIL}}$ signal triggers the one-shot K2 to free running. Approximately 0.6 microsecond later, the oscillating signal at K2-9 causes the L1 flip-flop pin 5 to go low. This, in turn, generates $\overline{\text{MC}}$ and $\overline{\text{MCDELAY}}$. The $\overline{\text{POWER FAIL}}$ signal is also transmitted to the SMI module to produce an interrupt, INT16, indicating to the processor that power failure has occurred. The software then stores all registers and flag bits required for resumption when the power is on again. Approximately 2 milliseconds after the $\overline{\text{POWER FAIL}}$ signal goes active, the $\overline{\text{MC}}$ signal is generated to clear the memory interface and lock out any further requests.

MISCELLANEOUS LOGIC

S2 Control Signals

Position select signals $\overline{\text{S2S2}}$, $\overline{\text{S2S1}}$, and $\overline{\text{S2S0}}$ of selector S2 correspond to MIR10 through NIR12 of the B code, respectively. In the B′ code ($\overline{\text{BP}}$ signal is true), however, these control signals are forced low at time T0 to select the three-state bus as input.

S1 Control Signals

Position select signals $\overline{S1S2}$, $\overline{S1S1}$, and $\overline{S1S0}$ of selector S1 correspond to MIR07 through MIR09 of the A code, respectively. In the A′ code ($\overline{AP}$ signal is true), however these control signals are forced low at time T0 to select the three-state bus as input.

B′ Decoder

The B′ decoder is enabled when the S field contains 1000 ($\overline{BP}$ signal is low) and the macro memory bus is disabled ($\overline{ENMEMBUS}$ is high). When enabled, MIR10 through MIR12 are decoded to generate the control signals for the read micro memory, enable I/O data and address path, enable the output of the register to the three-state bus, and enable the interrupt address to S2 via the three-state bus.

A′ Decoder

A′ codes are decoded to generate the $\overline{ENSM1}$ and $\overline{ENSM2}$ signals, which control the selection of the contents of the M1 or SM1 and M2 or SM2 registers to be used as the A source.

RESUME

The $\overline{RESUME}$ signal is generated from the CPU to allow the main timing generator to continue to run at times T2 and T3 during macro memory reference. $\overline{RESUME}$ is generated by one of the following signals:

CPU memory data resume ($\overline{CPU-MDS}$), when the micro instruction contains a read macro memory command followed by a GETMAK command. At the trailing edge of $\overline{CPU-MDS}$, output pin 9 of the F8 flip-flop is set high and the $\overline{RESUME}$ signal is generated after being delayed 80 nanoseconds by delay line G7. The $\overline{RESUME}$ signal is delayed so that the data from the memory can be properly gated into the IXT register in the transform board for emulation. Refer to the transform description later in this section for more details. $\overline{RESUME}$ is reset at time T4 or by a master clear.

The leading edge of the CPU memory early data resume signal ($\overline{CPU-EDS}$), whenever the micro instruction does not contain the GETMAK command. $\overline{RESUME}$ is reset at time T5 or by master clear.

CONTROL 2

Figure 5-4 shows the functional block diagram of the control 2 module with logic sheet numbers designating the locations of individual functional blocks within the logic diagrams.

Control 2 is divided into the following functional parts:

Bit generator

K and N registers

Micro-memory address registers and control circuitry

- Page/micro-memory register (P/MA)

- Page storage register (PS)

- Micro-memory address counter (MAC)

- Return jump register (RTJ)

- Selector S4

- Selector S6

Micro-instruction register: MIR00, MIR01, and MIR16 through MIR31

M-, S-, C-, and T-field decoders and miscellaneous control circuitry

Macro-memory interface control and miscellaneous circuitry

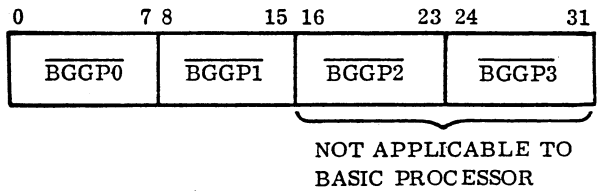
BIT GENERATOR

The main function of the bit generator is to generate a 1-bit at any position in a word as input to the B side of the ALU. Control to drive the bit generator is derived either from a micro instruction (MIR27 through MIR31) or from the lower five bits of the N counter (N03 through N07). The bit generator consists of four dual one-to-four demultiplexers: E9, F9, G8, and E8. FLAG102 (SM102) selects which input drives the bit generator as follows:

1 = Enables G8 and F9 and disables E8 and E9

0 = Enables E8 and E9 and disables G8 and F9

The corresponding outputs of these multiplexers are OR-wired together. N05 through N07 and MIR29 through MIR31 are decoded by F9 and E9 respectively to determine the position of the 1-bit within an 8-bit group ($\overline{BG0}$ through $\overline{BG7}$). N03 and N04 and MIR27 and MIR28 are decoded by G8 and E8 to determine the position of the 8-bit group within a word, as follows:



$\overline{BGGP0}$ and $\overline{BGGP1}$ are used to enable $\overline{BG0}$ through $\overline{BG7}$ at the input of selector S2 of the ALU module.

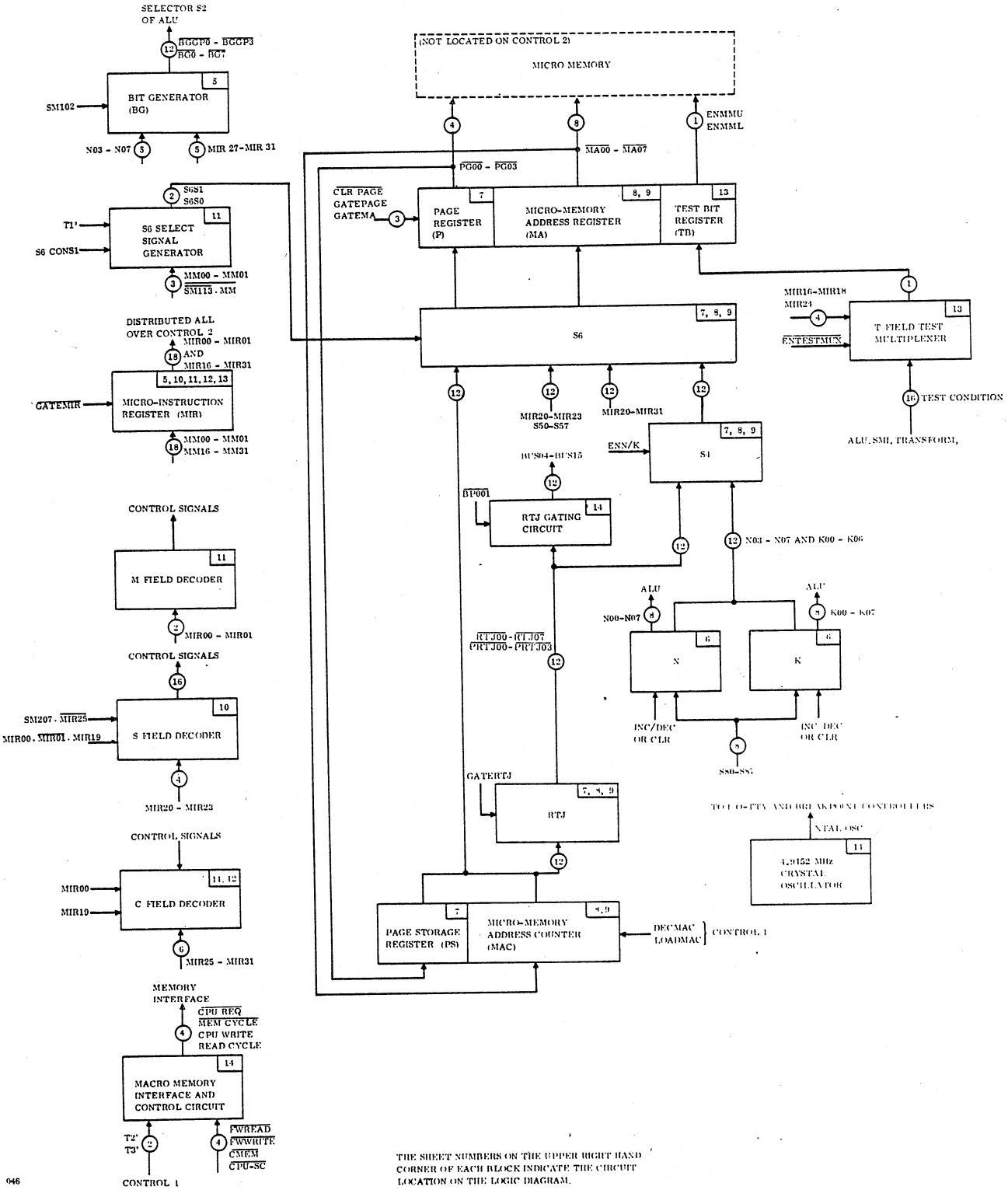


Figure 5-4. Control 2 Functional Block Diagram

K AND N REGISTERS

The K register is an 8-bit counter that can be cleared, decremented, incremented, or loaded from selector S8 of the transform module. It consists of two 4-bit binary up/down counters, K8 and H8. K-register outputs K0 through K7 are used to address file 1 (optional), or they may be selected as input to selector S2 or as part of the micro-memory address (combined with N) in addition to any program usage as a counter. The value of K can be tested against zero by the micro instruction. The K register receives inputs from selector S8 on the transform module. The $\overline{K=0}$ signal is transmitted to the T-field test multiplexer, A9, for monitoring by the T field of the micro instruction.

The N register is an 8-bit counter that can be cleared, decremented, incremented, or loaded from S8. It consists of two four-bit binary up/down counters, K7 and H7. N register outputs N0 through N7 are used to address file 2, to control shift and scale operations, as input to selector S2 of the ALU module, or to combine with the contents of the K register to form the micro-memory address. The value of N can be tested against zero by the micro instruction. The N register receives input from selector S8 on the transform module. $\overline{N=0}$ is sent to the T-field test multiplexer, A9, for monitoring by the T field of the micro instruction. The control signals for the K and N registers are decoded by multiplexer E2 from the C′ code of the micro instruction as shown below. Multiplexer E2 is enabled when $\overline{S2=KN}$ is low (that is, MIR25 through MIR27 = 100).

MIR28 (Pin 3)	MIR29 (Pin 13)	Operation
0	0	CLRK
0	1	IN/DEC K†
1	0	CLRN
1	1	IN/DEC N†

$\overline{NLOAD}$ is generated at time T3 MEM during the N transform (TN/j), clear N register (CLRN), clear page and N register commands, and during micro-instruction format 3 with MIR19 set, as indicated below:

$$\overline{NLOAD} = \overline{NXFORM + CLRN + CLRPGN + (MIR00 \bullet MIR01 \bullet MIR19)} \bullet T3 \bullet MEM$$

During the CLRM and CLRPGN commands, the N register is cleared by loading all zeros to the N register (selector S8 of the transform module is disabled).

Similarly, the KLOAD signal is also generated at time T3 • $\overline{MEM}$ during K transform and clear K register (CLRK) and during micro-instruction format 3 with MIR19 not set as indicated below:

$$\overline{KLOAD} = \overline{KXFORM + CLRK + (MIR00 \quad MIR01 \bullet \overline{MIR19}) \bullet T3 \bullet \overline{MEM}}$$

Refer to Micro Instructions, section 2, for more details of these micro command codes.

$\overline{KCLK}$ is generated at time T2′ during the increase/decrease K register command (C′ = 100010x). NCLK is generated at T2′ during the repeat command (S = 0010), during the increase/decrease N register command (C′ = 100110x), or whenever NSHIFT clock is present during the shift or scale operation.

MICRO-MEMORY ADDRESS REGISTERS AND CONTROL CIRCUITRY

P/MA Register

The P/MA register provides addressing for the micro memory. The P portion is the 4-bit page register G5, which selects one of 16 pages. The P portion of the register can be set from the micro-instruction S field (MIR20 through MIR23) when a page jump is specified, from the PS register, or from the RTJ register. The MA portion is an 8-bit register that selects one of the 256 locations within a page. The MA portion of the register can be set from the micro-instruction C field (MIR24 through MIR31) for jump addresses, from the MAC register, from selector S5 on the transform module. Outputs of the P/MA register are sent to the micro memory for addressing and to the PS/MAC register for updating.

PS/MAC Register

The PS register is a 4-bit register that is made up of D flip-flop G4. The MAC register is an 8-bit counter that consists of two 4-bit counters, H2 and K4. Depending on the sequencing operation specified by the micro instruction, the PS/MAC register may be used to provide the next micro-memory address through the P/MA register.

In operation, the MAC register increments the contents of the MA register as follows: The complement of the MA register is transferred to the MAC register, the contents of MAC is then decremented by one (input pin 5 = high → count down), and then the complement of MAC is transferred to the MA register. When the contents of the MAC register is transferred to the MA register, the data is complemented by selector S6,

† When MIR31 is set, control signal KDOWN or NDOWN is low (count up); if not set, MIR31 = 0 is set, control signal KDOWN or NDOWN is high (count down).

providing the MA register with an incremented address. The LOADMAC signal is generated at time T1′ during read or write micro memory and DECMAC is generated at time $\overline{T3}$ • T2. Refer to the control 1 logic description for more details.

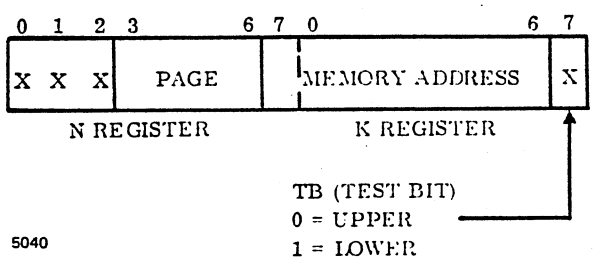
Return Jump Register

The RTJ register is provided to capture the location of the next micro-instruction pair at any time specified by a micro instruction (RTJ command is S field) regardless of the contents of the M field. When the capture is specified, the MAC register is incremented and stored in the RTJ register. The contents of RTJ is unchanged until the next command is given to save a new address. Control signal GATERTJ is generated when the RTJ command (S = 1110) is decoded. The output of RTJ is gated to the P/MA register through S6 to perform the return operation, or it may be read into the organization through selector S2.

Selector S4

Selector S4 consists of three quad 2 input multiplexers: G2, H3, and J4. It selects either outputs of the RTJ register or outputs of the K and N registers as inputs to selector S6. During read or write micro memory, T1 is high, allowing the N/K register combination to be selected. When T1′ is low, outputs of the RTJ register are selected since $\overline{ENN/K}$ is normally pulled up to +5 V dc.

Outputs of the K and N registers are input to selector S6 to provide micro-memory operand addresses outside the current operating page as follows:



The least significant bit of the K register, K07, determines the upper or lower of the next micro-instruction pair to be executed via the T-field test multiplexer, A9.

Selector S6

Selector S6 is a 12-bit selector consisting of six dual four-input multiplexers: H5, G1, H1, J1, K2, and J5. It is used to select inputs for the P/MA register. Selector S6 derives its inputs from the PS/MAC register, selector S5 of the transform module, the MIR register (MIR20 through MIR31), and selector S4. Table 5-7 lists the S6 input selected with each binary value of S6S0 and S6S1.

TABLE 5-7. S6 INPUT SELECTION

Select Input		Source Input to S6
S6S1	S6S0	
0	0	PS/MAC register
0	1	Selector S5 (transform module) + MIR20 through MIR23
1	0	MIR20 through MIR31
1	1	Selector S4

Select signals S6S1 and S6S0 are generated by multiplexer D2 as follows:

S6CONS1	T1′	S6S1	S6S0
0	0	$\overline{MM01}$	$\overline{MM00+MM01}$
0	1	$\overline{SM113\ MM}$	1
1	0	0	0
1	1	0	0

During read or write micro memory (MM = 1), the time generator is set at time T1′ (T1′ is high). Initially, S6CONS1 is low, allowing selector S6 to select input from selector S5 of the transform module if status mode bit SM113 is high, or to select the combination of bits from the N/K register as the micro-memory address if SM113 is low. When S6CONS1 goes high, it allows selector S6 to select the previous sequential address from the PS/MAC register. Refer to Read/Write Micro Memory, section 4, and the timing diagrams of the control 1 module in figure 5-2 for more details.

During the time that T1′ and S6CONS1 are low, select signals S6S1 and S6S0 are decoded according to mode bits MM00 and MM01 as follows:

	MM00	MM01	S6S1	S6S0
RETURN	0	0	1	1
SEQ.	0	1	0	0
JUMP	1	0	1	0
SEQ.	1	1	0	0

MICRO-INSTRUCTION REGISTER

The MIR portion of the control 2 module holds bits MIR00, MIR01, and MIR16 through MIR31 of the micro

instruction. These bits are used to decode the M, S, C, and T fields of the micro instruction. The outputs of micro memory are gated into the MIR register at time T5 by the GATEMIR signal.

M-Field Decoder

MIR00 and MIR01 and subformat bit MIR19 are used to decode the M field as shown in table 5-8.

The coding in the S and C fields depends upon the format/mode of the micro instruction. Refer to Micro Instructions, section 2, for more details.

S-Field Decoder

MIR20 through MIR23 of the S field are decoded by two three-to-eight line decoders, B2 and C2. When MIR20 is low, C2 is enabled and B2 is disabled; when MIR20 is high, B2 is enabled and C2 is disabled. Both decoders B2 and C2 are disabled under one of the following conditions:

Status mode bit SM207 is high and MIR25 is low under C" code (format 1) and MIR19 = 1. This includes the C" codes TMA/j, TMAK/j, GITMAK/j, GITMAK/xt, and TK/j. These conditions allow the MA transform to be executed across the page boundary.

MIR00 and MIR01 = 10 (format 2) and MIR19 = 1, signifying a jump to a different page. Refer to Micro Instructions, section 2, for restrictions on the S field. The S field is decoded as shown in table 5-9.

TABLE 5-8. M-FIELD DECODING

MIR00	MIR01	MIR19	Format/Mode
0	0	0	Format 1/return mode
0	0	1	Not applicable
0	1	0	Not applicable
0	1	1	Format 1/sequential mode
1	0	0	Format 2/jump mode (same page)
1	0	1	Format 2/jump mode (different page)
1	1	0	Format 3/sequential mode
1	1	1	Format 3/sequential mode

TABLE 5-9. S-FIELD DECODING

MIR20 (Pin 6, B2) (Pin 4, C2)	MIR21 (Pin 3)	MIR22 (Pin 2)	MIR23 (Pin 1)	Operation
0	0	0	0	N/C = NOP
0	0	0	1	$\overline{\text{DDPP}}$ = Alternate D field coding DD"
0	0	1	0	$\overline{\text{RPT}}$ = Repeat if N ≠ 0
0	0	1	1	$\overline{\text{FWREAD}}$ = Read macro memory
0	1	0	0	$\overline{\text{FWWRITE}}$ = Write macro memory
0	1	0	1	$\overline{\text{L8EA}}$ = ALU output is shifted left, end-around
0	1	1	0	$\overline{\text{F2WR}}$ = (F) → F2
0	1	1	1	$\overline{\text{AP}}$ = Alternate A field coding A′
1	0	0	0	$\overline{\text{BP}}$ = Alternate B field coding B′
1	0	0	1	$\overline{\text{DP}}$ = Alternate D field coding D′
1	0	1	0	$\overline{\text{APDP}}$ = Alternate A and D field, A′ and D′
1	0	1	1	$\overline{\text{DPP}}$ = Alternate D field coding D"
1	1	0	0	$\overline{\text{GATEI}}$ = Output S1 to I register
1	1	0	1	$\overline{\text{HALT}}$ = Stop processor operation when SM109 is set
1	1	1	0	$\overline{\text{RTJ}}$ = Next sequential address to TRJ register
1	1	1	1	$\overline{\text{CLRPGN}}$ = Clear N and page register

During the repeat command, the ITR • $\overline{N = 0}$ signal is generated to block the selection of the next micro-instruction pair (GATEMA is inhibited). Various control signals are generated from outputs of the S-field decoders, depending upon the specific operation being performed.

C-Field Decoder

The C field consists of MIR25 through MIR 31. The first three bits, MIR25 through MIR27, are decoded by two three-to-eight line decoders, D7 and E7. Both of these decoders are partially enabled when MIR00 is low (format 1 only). When MIR19 is low (C′ coding), decoder E7 is completely enabled and D7 is disabled; when MIR19 is high (C" coding), D7 is enabled and E7 is disabled. Table 5-10 lists the operations decoded from MIR25 through MIR27 of the C field during format 1.

The outputs of decoders D7 and E7 are used to generate various control signals such as $\overline{S3*S1}$ and $\overline{S3*S0}$, which control the shift operation of S3; FLAG1C, and $\overline{FLAG2C}$, which set and clear the respective FLAG bits, and MAXFORM, KXFORM, and $\overline{NXFORM}$, which initiate the transform operation.

The next lower four bits of the C field (MIR28 through MIR31) continue to be decoded by the dual one-to-four demultiplexer E6. E6 is enabled when

TABLE 5-10. C-FIELD DECODING

MIR19 (Pin4, E7 (Pin 6, D7)	MIR25 (Pin 3)	MIR26 (Pin 2)	MIR27 (Pin 1)	Operation
0	0	0	0	Not used
0	0	0	1	Not used
0	0	1	0	Requred for continued decoding of C′ field
0	0	1	1	GATE IXT; general purpose strobe at time T4
0	1	0	0	Required for continued decoding of C′ field
0	1	0	1	SETF/j
0	1	1	0	CLRF/j
0	1	1	1	Shift desitnation and/or Q register
1	0	0	0	TMA/j
1	0	0	1	TMAK/j
1	0	1	0	GATEMAK/j or GETMAK/xt
1	0	1	1	TK/j
1	1	0	0	TN/j
1	1	0	1	SUB (set upper bounds)
1	1	1	0	SLB (set lower bounds)
1	1	1	1	Not used

CMEM is low (MIR25 through MIR27 = 010). Table 5-11 lists the C′ codes decoded by E6.

T-Field Decoder

The T field is the conditional branch of the micro instruction. It specifies which instruction, upper or lower, of the next micro-instruction pair to execute. The T field consists of MIR16 through MIR18 and MIR24. When MIR24 = 0, it indicates the normal T code; MIR24 = 1 indicates the T′ code. Note that the T′ code is available only for the return and sequential addressing mode (MIR00 = 0). Refer to the Micro Instructions, section 2, for more details on the T field. The T field is decoded by the 16-to-1 multiplexer A9 with MIR16 through MIR18 and MIR24 as the select signals. The output of the T-field test multiplexer is used to set or reset the TB flip-flop C7, which causes the selection of either the upper or lower micro instruction of the next micro-instruction pair. The ENMML and ENMMU signals go directly to micro memory. The ENTESTMUX signal is disabled by the micro memory time generator (control 1 module) any time a micro-memory read/write operation is performed. This causes the next upper micro instruction to be read from the next micro-instruction pair.

The $\overline{\text{SAMPINTS}}$ signal that gates the interrupts into the interrupt register is generated at time T4 T5 except when an interrupt is active and the micro-instruction T field is sampling for an interrupt. This prevents the interrupt address from changing while the micro program is utilizing the interrupt address.

MACRO-MEMORY INTERFACE

Control signals required for the macro memory interface are generated from the outputs of the S- and C-field decoders. $\overline{\text{MEMCYCLE}}$ is produced whenever a read, write, or write character (C′ code) of the macro memory is initiated. The $\overline{\text{CPU-REQ}}$ signal is then generated at time T2′. During read and normal write cycles the CPU-WRITE signal is true when the memory is directed to perform a write cycle and false when a read cycle is desired. During split cycle operation, CPU-WRITE is false for the read portion and is forced true for the write portion.

During the read and the read portion of the split cycle, the macro memory bus is disabled ($\overline{\text{ENMEMBUS}}$ signal is set high) at time T4′ if there is an active interrupt and the micro instruction is sampling for an interrupt ($\overline{\text{BLKENMEN}}$ is low).

CRYSTAL OSCILLATOR

The crystal oscillator consists of the crystal Y1 and two inverter gates of B7. This circuit generates 4.9152-MHz clock pulses for the I/O-TTY and breakpoint controller modules.

TABLE 5-11. C′ DECODING

C′Code (MIR25-MIR31)	E6 Output Pin	Operation
0 1 0 0 0 0 0	7	WRCH/0 (write character 0)
0 1 0 0 0 0 1	6	WRCH/1
0 1 0 0 0 1 0	5	WRCH/2†
0 1 0 0 0 1 1	4	WRCH/3
0 1 0 0 1 0 0	9	$\overline{\text{CPU-SC}}$ during read modified write
0 1 0 0 1 0 1	10	WRHWO (write word 0)
0 1 0 0 1 1 0	11	Not used
0 1 0 0 1 1 1	12	WRHW1†

†Not applicable for basic processor

ARITHMETIC/LOGICAL UNIT

Figure 5-5 shows the functional block diagram of the ALU module with logic sheet numbers designating the locations of individual functional blocks within the logic diagrams.

The ALU module contains the following functional areas:

A, Q, P, I, X, and F register

File 1 and File 2

Selectors S1 and S2

Arithmetic and logical unit with look-ahead carry

Selector S3

A REGISTER

The A register is a general-purpose, 16-bit register consisting of four 4-bit bidirectional universal shift registers: M4, J4, C4, and A4. The A register is capable of shift right, shift left, or parallel loading, depending upon the states of control signals AMODES0′ and AMODES1′ as follows:

AMODES′1	AMODES0′	Operation
0	0	Not applicable
0	1	Right shift (down)
1	0	Left shift (up)
1	1	Parallel loading

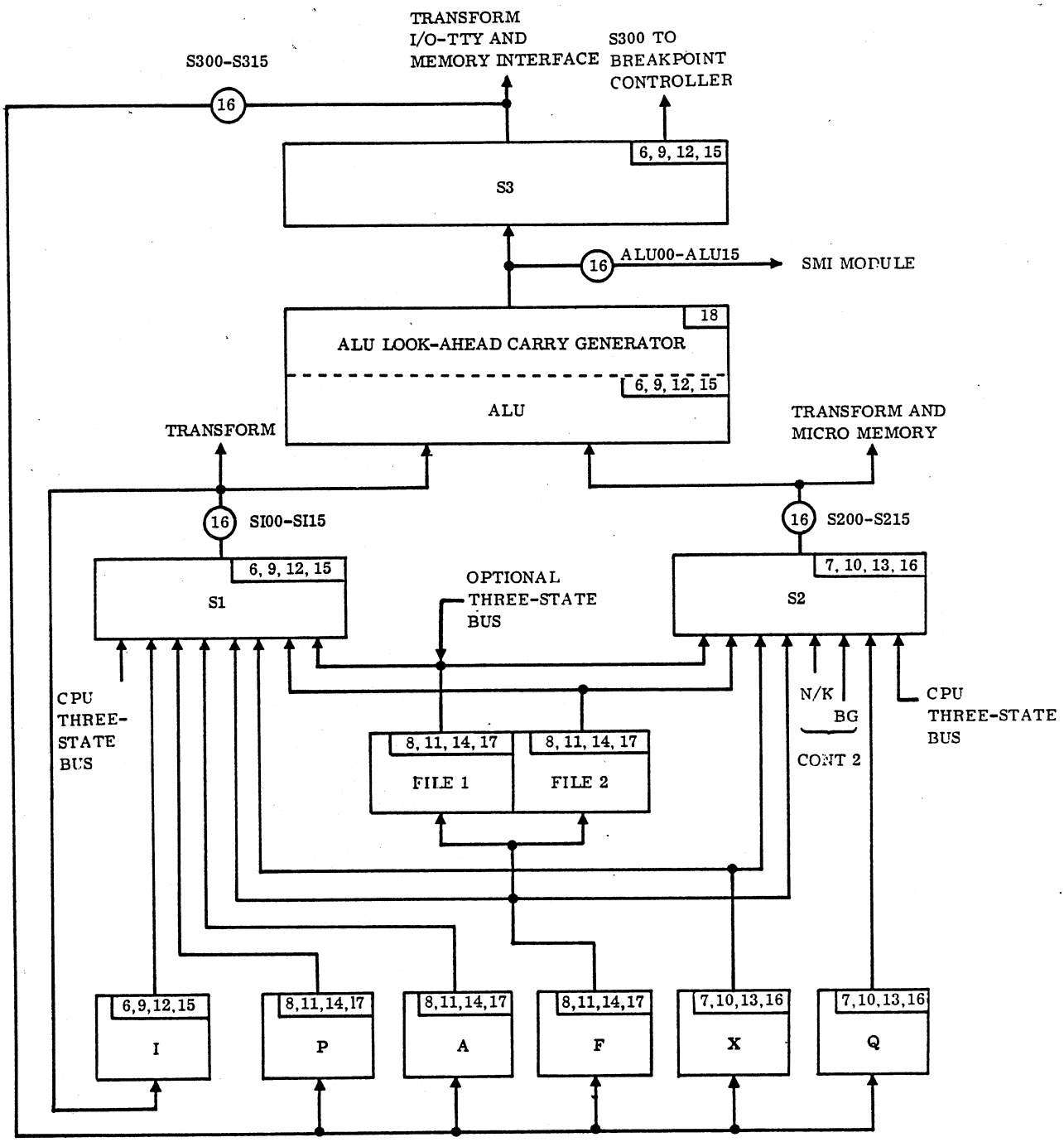
These control signals are decoded from MIR7 and MIR8 of the micro instruction during shift or scale operations. The appropriate data to enter into the vacated positions during right or left shift is AMSBSR or ALSBSL, respectively. AMSBSR and ALSBSL are decoded from MIR11 and MIR12. Refer to the Control 1 for more details. See table 5-12 for a summary of A register characteristics.

Q REGISTER

This 16-bit register consists of four 4-bit shift registers: JK8, JK7, CD8, CD7. It has the same shifting capability as the A register when it is shifted in conjunction with the A register. This register can also be shifted in conjunction with

TABLE 5-12. A, Q, P, I, X, AND F REGISTER CHARACTERISTICS

Register	Input From	Output To	Capabilities	Characteristics and Related Functions
A	S3	S1	Shift right or left with or without Q register, independently of ALU	Hold MSB during double-word length shift with Q register
Q	S3	S2	Shift right or left with A register independently of ALU	Hold LSB during double length shift with A register; shift right or left in conjunction with destination register (A, P, X, F) through ALU
P	S3	S1	Holding register	May hold the software instruction for emulation
I	S1	S1	Holding register	May hold the software instruction for emulation
X	S3	S1, S2	Holding register	May be used for transferring information to I/O section
F	S3	S1, S2 F1, F2	Holding register	Hold data being stored in F1 or F2



NOTE: THE SHEET NUMBERS IN THE UPPER RIGHT HAND CORNER OF EACH BLOCK INDICATE THE CIRCUIT LOCATION ON THE LOGIC DRAWING.

037

Figure 5-5. ALU Functional Block Diagram

any destination via selector S3, so the control signals for the Q register are decoded from MIR07 through MIR09 and from MIR30 (C field).

Similarly, the data to enter during right shift (QMSBSR) and left shift (QLSBSL) are generated in the control 1 mode from MIR11 and MIR12. The clock of Q register GATE Q is generated when Q is the destination register, and when W is shifted or scaled. See table 5-12 for a summary of Q register characteristics.

P, X, I, AND F REGISTERS

These 16-bit registers are made up of positive-edge triggered D-type flip-flops. Inputs of the P, X, and F registers are connected to outputs of selector S3; the I register inputs are connected to outputs of S1. All the gating signals, GATE I, GATE X, GATE P, and GATE F, are decoded from the D field (MIR13 through MIR15) of the micro instruction in the control 1 mode.

Table 5-12 summarizes the capabilities and characteristics of the P, X, I, and F registers.

FILE 1 AND FILE 2

File 1 is a 256-word by 16-bit read/write memory (RAM). File 2 is a 32-word by 16-bit read/write memory. File 1 consists of 16 three-state RAM chips; each chip contains 256 one-bit words; each chip furnishes a specific bit position in each of the 256 words. Note that in the logic diagrams these RAM chips are drawn with dotted lines to indicate that file 1 is an optional feature.

File 1 is addressed by the contents of the K register, K0 through K7. The $\overline{WEF1}$ signal, pin 12 is generated at time T6 when F1 is the destination register (D field = 100). File 1 is enabled for read/write only when status mode bit SM111 is set to 1 ($\overline{ENF1}$ = 0). When SM111 is set, it also enables the output of selector S1 or S2.

File 2 consists of eight open collector RAM chips; each chip contains 16 by 4-bit words. File 2 is addressed by the five low-order bits of the N register. N04 through N07 are used as common addresses and N03 (or $\overline{N3-1}$ = inverse of N3) is used to select the upper (or lower) 16-word group. The corresponding outputs of the upper and lower groups of RAMs are OR-wired together. $\overline{WEF2}$, pin 3, is generated at time T1 to allow writing into file 2 when the S field contains 0110. Refer to Micro Instructions, section 2, for more details of S codes.

SELECTOR S1 AND S2

Selector S1 enables one of eight inputs to the A side of the ALU. Selector S2 enables one of eight inputs to the B side of the ALU. The select signals of S1 and S2 are determined by the A field (bits 07 through 09) and the B field (bits 10 through 12), respectively, of the micro instruction. These A and B field bits are decoded by the control 1, A and B field decoders to produce selector signals (S1S0, S1S1, S1S2, and S2S0, S2S1, S2S2) that are applied to pins 09, 10, 11 of selectors S1 and S2. Refer to the control 1 logic description for the decoding of the select signals S1 and S2. Table 5-13 lists the source input selected at S1 and S2 for each select address.

TABLE 5-13. SELECT ADDRESS SOURCE INPUTS

A- or B-Field Code† (S1, S2 Chip Input Pins			Source input for	
09	10	11	Selector S1	Selector S2
0	0	0	F2	F2
0	0	1	P	N/K
0	1	0	I	BG
0	1	1	X	X
1	0	0	A	Q
1	0	1	F	F
1	1	0	F1 or external input††	F1 or external input††
1	1	1	Bus	Bus

†A field code selects S1; B field code selects S2.

††If SM111 is set, select F1 as input to S1 or S2. If it is not set, select the external input from the MIR encoder of the transform board as input to S1 or S2.

ARITHMETIC/LOGICAL UNIT

The ALU unit consists of four ALU chips (L9, G9, DE9, and A9) and one look-ahead carry generator (B12). THe ALU receives A and B inputs from S1 and S2, respectively. It processes these two inputs in one of two modes: logical mode if ALUM (input pin 8) equals 1, or arithmetic mode if ALUM equals 0. The ALUS3-2, ALUS2-2, ALUS1-2, and ALUS0-2 control signals (input pins 3, 4, 5, and 6, respectively) select the functions of ALU according to table 5-14.

TABLE 5-14. ALU FUNCTION SELECTION

Function Select Signals				ALU Operation		
					Arithmetic ALUM = 0	
ALUS3-2 (Pin 3)	ALUS2-2 (Pin 4)	ALUS1-2 (Pin 5)	ALUS0-2 (Pin 6)	Logical ALUM = 1 (Pin 8)	Cn = Low (No Carry)	Cn = High (No Carry)
0	0	0	0	$F = \overline{A}$	Not used	
0	0	0	1	$F = \overline{AB}$		
0	0	1	0	$F = \overline{A} + B$		
0	0	1	1	F = 1		
0	1	0	0	$F = \overline{A + B}$		
0	1	0	1	$F = \overline{B}$		
0	1	1	0	$F = \overline{A + B}$		
0	1	1	1	$F = A + \overline{B}$		
1	0	0	0	$F = \overline{AB}$		
1	0	0	1	F = A + B		
1	0	1	0	F = B		
1	0	1	1	F = A + B		
1	1	0	0	F = 0		
1	1	0	1	$F = \overline{AB}$	Not Used	
1	1	1	0	F = AB		
1	1	1	1	F = A		
0	1	1	0		A - B - 1	A - B
1	0	0	1		A + B	A + B + 1
1	1	1	1		A	A + 1

These control signals are decoded from the F field, MIR03 through MIR06, of the micro instruction. MIR02 selects the ALU mode (ALUM = $\overline{MIR02}$). The look-ahead carry generator B12 provides look-ahead carries for four ALU chips for high-speed operation. An ALU chip transmits carry generate (pin 17) and carry propagate (pin 15) to the look-ahead carry generator, which in turn generates a carry to the next higher order bit ALU chip. Figure 5-6 shows the arrangement of ALU chips and the loop-ahead carry generator. The split adder feature is implemented by blocking the carry-in between two portions of the adder. For 8/8 adder split, CARRYIN1 is blocked from going into the ALU04-ALU07 chip. In addition, the upper eight bits of the adder always operate in twos complement, while the lower eight bits may operate using ones or twos complement arithmetic depending on the ones complement flag bit 101. The output of ALU is connected to S3 and is also available for connection to SM1, SM2, M1 and M2 of the SMI module. The ALU also generates the A = B signal, indicating that ALU inputs A and B are of equal value.

Since the ALU chips perform the subtract operation in ones complement, the appropriate CARRYIN3 signal must be generated in some cases to obtain the desired results. Refer to the control 1 logic description for more details on the CARRYIN3 generator.

SELECTOR S3

Selector S3 consists of eight dual four-input multiplexers (L11, M11, G11, HJ11, DE11, F11, A11, and B11) that are capable of shifting data left or right one place, left eight places, and end-around or straight transfer prior to making data available to the A, P, X, or F registers. Select signals S3S0-1 and S3S1-1 are decoded from the S code (left shift eight places and end-around) and from the C' code (MIR25 through MIR31) with MIR30 indicating right or left shift. Refer to the control 2 logic descriptions and Micro Instructions section for more details. Select signals S3S0-1 and S3S1-1 control the S3 operation as shown in table 5-15.

TABLE 5-15. S3 OPERATION SELECTION

Select Input		S3 Operation
S3S1-1	S3S0-1	
0	0	Straight transfer
0	1	Left shift one place
1	0	Right shift one place
1	1	Left shift eight places, end-around

During right shift one place of both the single- and double-word length registers, MIR31 is used to set or clear the most significant bit of the destination register. During left shift one place of the combined destination and Q register, Q00 is entered as data to the least significant bit position of S3. However, during the left shift of the destination register alone, MIR31 is used to enter the vacated position. Refer to the control 1 module description for the generation of the $\overline{MIR31+Q00}$ signal.

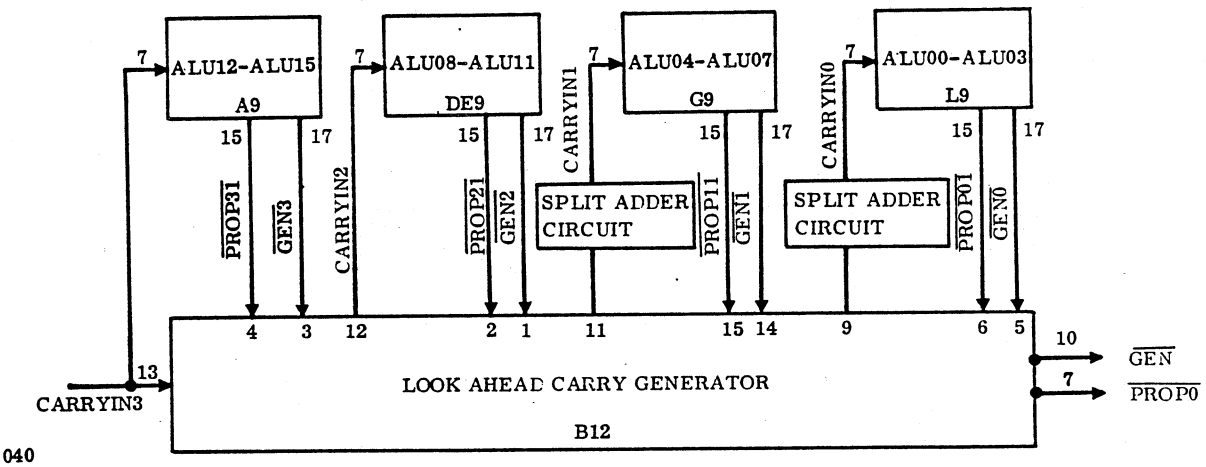
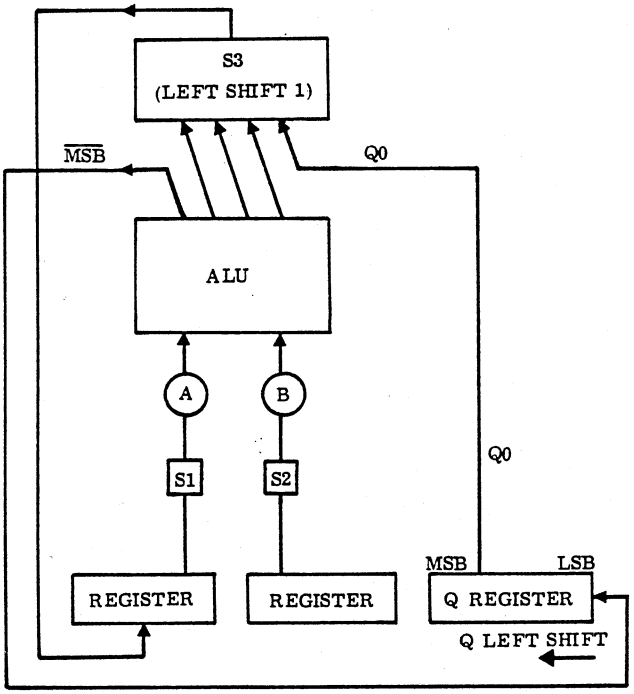


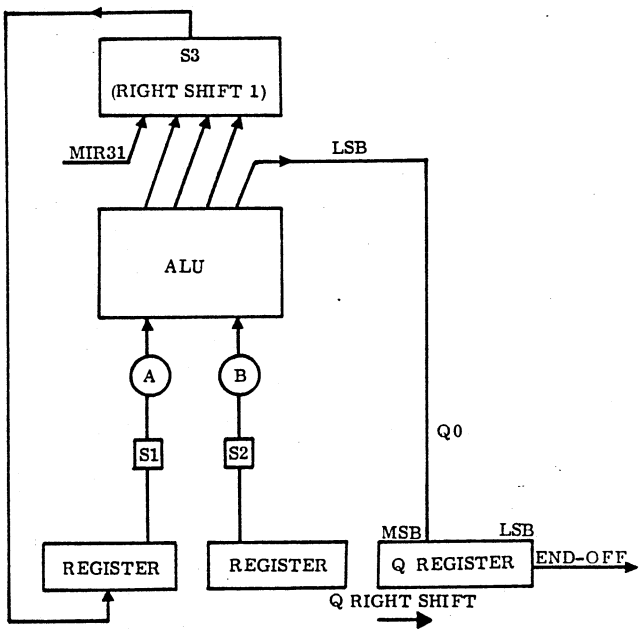
Figure 5-6. Simplified ALU with Look-Ahead Carry Generator

During a double-word length left or right shift one place using S3, the ALU and Q register connections are as shown in figures 5-7 and 5-8. These connections enable the Q register to shift one place in conjunction with the A, P, F, or X register.



050

Figure 5-7. Example of Double-Word Length Left Shift One Place



051

Figure 5-8. Example of Double-Word Length Right Shift One Place

The output of S3 is fed back to the A, P, F, X, or Q register and is available for connection to the transform module, the I/O-TTY module, and the memory interface module. Bit S200 of selector S3 is connected to the breakpoint controller module.

STATUS MODE INTERRUPT MODULE

Figure 5-9 shows the functional block diagram of the status mode interrupt module with logic sheet numbers designating the locations of the individual functional blocks within the logic diagrams.

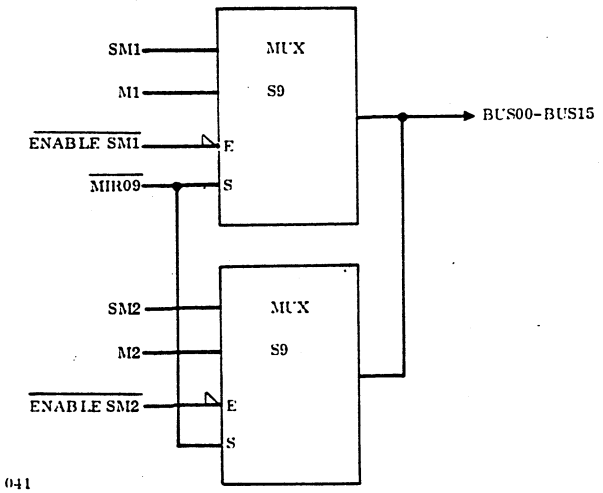


Figure 5-9. Simplified Diagram of Selector S9

The status mode/interrupt (SMI) module consists of the following functional areas:

Status mode registers, SM1 and SM2

Mask registers, M1 and M2

Selector S9

Interrupt system

Data from memory interface to bus

STATUS MODE REGISTERS

The SM register allows the micro program to control the processor's mode of operation and examine the status of certain internal and external conditions. The processor can access one or two SM registers, referred to as SM1 and SM2. SM1 and SM2 are word-length registers that are made up of D-type flip-flops with direct set and clear inputs. FLAG100 through FLAG103 and FLAG200 through FLAG203 (also referred to as SM100 through SM103 and SM200 through SM203) can be directly set or cleared depending on control signals FLAG1C and $\overline{\text{FLAG2C}}$ (decoded from the SETF/j and CLRF/j command of the C field) at multiplexers C9 and J9. The flag position to be set or cleared is decoded from MIR30 and MIR31. In addition, these flag bits, like other SM bits, can be set or reset by the micro program by transferring information to the SM register from the output of ALU. Processor master

clear also clears the SM bit. Table 5-16 summarizes the various characteristics of the SM bits.

The GATESM1 and GATESM2 signals, which gate data into the SM register, are decoded at time T4 when D' = 101 and 111, respectively. Refer to the Control 1 section for D-field decoding.

MASK REGISTERS

The two mask registers, M1 and M2, are word-length registers. M1 associates with interrupts RDINT00 through RDINT15, and M2 associates with interrupts PRINT16 through RPINT31. These mask bits can only be set or reset by transferring information from the ALU output under control of the micro program. The mask register allows the interrupt system to recognize any present interrupt whose corresponding mask bit is set. The M1 register consists of D-type flip-flops C4, B2, B9, and B3. M2 consists of J4, K9, K2, and K3. The inputs of M1 and M2 are connected directly to the ALU output and therefore cannot be shifted. Gating signals GATEM1 and GATEM2 are decoded when D' = 100 and 110, respectively. Refer to Control 1 section for more details. The contents of mask registers M1 and M2 can be used as the A source to the ALU via selector S9 and the three-state bus.

SELECTOR S9

The SM1 and M1 registers are multiplexed by the quad 2 input multiplexers, three-state output C5, B10, B5, and B4. Similarly, the SM2 and M2 registers are multiplexed by J5, K10, K5, and K4. The corresponding outputs of these two groups of multiplexers are OR-wired together and connected to the main CPU three-state bus. Refer to figure 5-10 for a simplified diagram of selector S9. Note that only one group of multiplexers is enabled at any one time by the corresponding enable signals ($\overline{\text{ENABLESM1}}$) or ($\overline{\text{ENABLESM2}}$) (decoded from MIR07 and MIR08) at chip input pin 15. Select signal $\overline{\text{AFLD02}}$ = $\overline{\text{MIR09}}$ is used to select SM1 or M1 and SM2 or M2 as follows:

$\overline{\text{MIR09}}$ = $\overline{\text{AFLD02}}$ = high Select SM1 and SM2

$\overline{\text{MIR09}}$ = $\overline{\text{AFLD02}}$ = low Select M1 and M2

INTERRUPT SYSTEM

The interrupt system is implemented as a sampled data system at the micro-program level instead of as a true interrupt as in a conventional computer. The interrupt logic consists of an interrupt register, a mask register, and a priority/address encoder.

The interrupt system is divided into four groups of eight interrupt lines, referred to as INTGROUP0, INTGROUP1, INTGROUP2, and INTGROUP3. INTGROUP0 has the highest priority, and INTGROUP3 has the lowest. Refer to table 5-17 for a typical interrupt address and priority assignments for the basic processor.

TABLE 5-16. SM BIT CHARACTERISTICS

SM Bits	Functional Characteristics†		
	Set By	Cleared By	Output Available Form
FLAG100 - FLAG103	SETF/j command	CLRF/j command or master clear	True
FLAG200 - FLAG203	SETF/j command	CLRF/j command or master clear	Complement
SM104 - SM107 SM204 - SM207	External input	Master clear	True/complement
SM108 - SM111	External input	Master clear	True
SM112 - SM113	External input	Master clear	Complement
SM114 - SM115	External input	External input or master clear	Complement
SM208 - SM213	External input	Master clear	True
SM214 - SM215	External input	External input or master clear	True

†All SM bits can be set or reset by transferring information from the ALU to the SM registers.

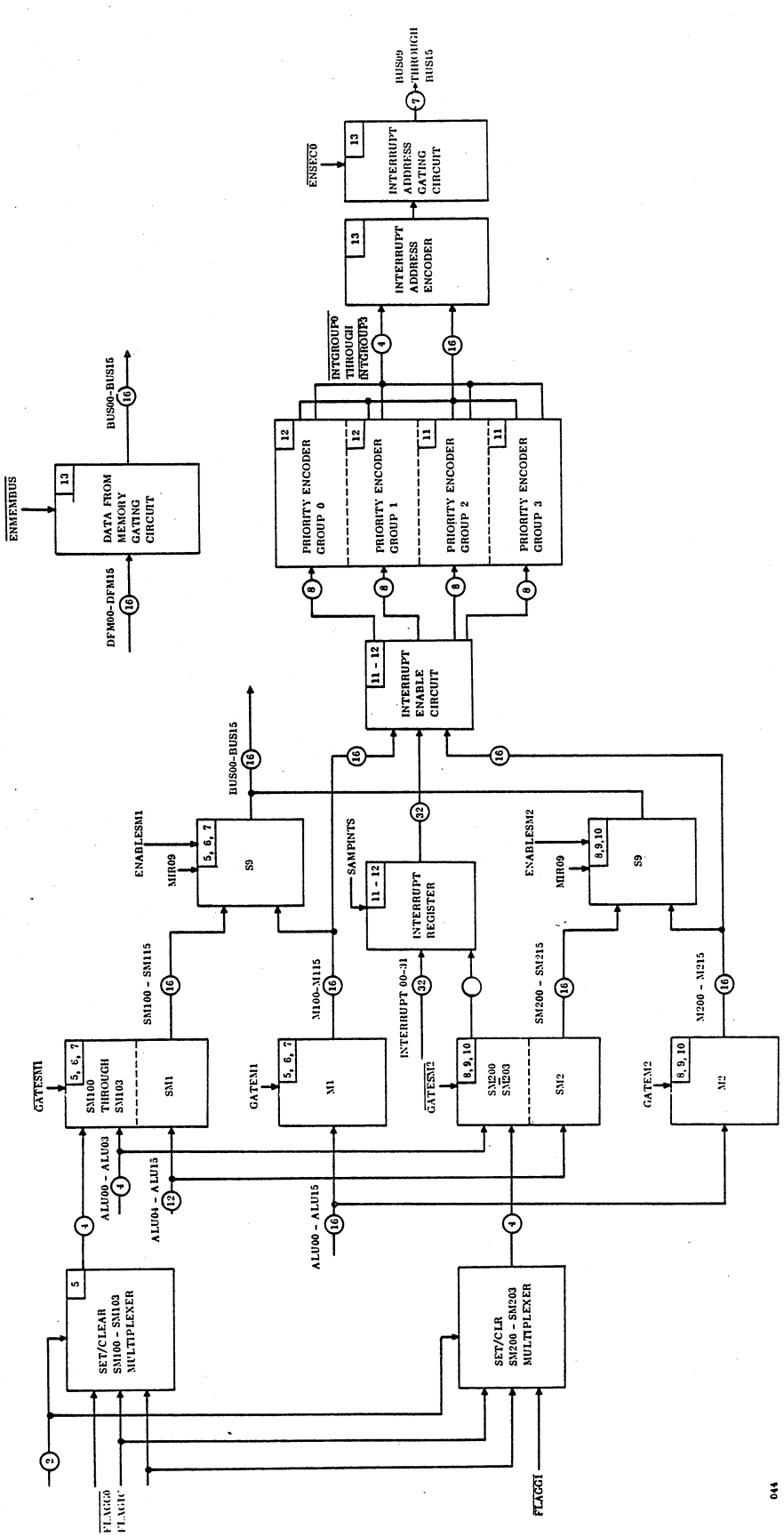


Figure 5-10. SMI Functional Block Diagram

TABLE 5-17. BASIC PROCESSOR INTERRUPT ADDRESSES

Mask Bit	Actual Interrupt Address$_{16}$	Interrupt Line Number		Output of Interrupt Address Encoder BUS11 - BUS15
M200	1F	Group 3	31 Lowest Priority	0 0 0 0 0
M201	1E		30	0 0 0 0 1
M202	1D		29	0 0 0 1 0
M203	1C		28	0 0 0 1 1
M204	1B		27	0 0 1 0 0
M205	1A		26	0 0 1 0 1
M206	19		25	0 0 1 1 0
M207	18		24	0 0 1 1 1
M208	17	Group 2	23	0 1 0 0 0
M209	16		22	0 1 0 0 1
M210	15		21	0 1 0 1 0
M211	14		20	0 1 0 1 1
M212	13		19	0 1 1 0 0
M213	12		18	0 1 1 0 1
M214	11		17	0 1 1 1 0
M215	10		16	0 1 1 1 1
M100	0F	Group 1	15	1 0 0 0 0
M101	0E		14	1 0 0 0 1
M102	0D		13	1 0 0 1 0
M103	0C		12	1 0 0 1 1
M104	0B		11	1 0 1 0 0
M105	0A		10	1 0 1 0 1
M106	09		09	1 0 1 1 0
M107	08		08	1 0 1 1 1
M108	07	Group 0	07	1 1 0 0 0
M109	06		06	1 1 0 0 1
M110	05		05	1 1 0 1 0
M111	04		04	1 1 0 1 1
M112	03		03	1 1 1 0 0
M113	02		02	1 1 1 0 1
M114	01		01	1 1 1 1 0
M115	00		00 Highest Priority	1 1 1 1 1

Note: The interrupt address is the same as its priority level; that is, the highest priority interrupt generates the 00 interrupt address and the lowest priority interrupt generates the 31 interrupt address.

The interrupt register receives and holds the active interrupt signals to be serviced by the processor. The SAMPINTS signal always clocks the interrupt signals to the interrupt registers at time $\overline{T4} \bullet T5$ except when an interrupt is being serviced by the processor. Each set bit in the mask register enables an associated interrupt to be recognized.

First, the active interrupt signals are priority encoded each within its own group by the enabled priority encoder(s) C2, C3, J3, and J2. These priority encoders are enabled by grounding input pin(s) 5 of the desired interrupt group(s). Whenever the INTGROUPx signal at output pin(s) 14 goes active, it indicates that there is an active interrupt within its group.

The priority among the interrupt groups is encoded by priority encoder E1 with input pins 10, 11, 12, and 13 pulled up to +5 V dc (for a 16-bit machine).

The interrupts are encoded in such a way that interrupt 00 has the highest priority and interrupt 31 has the lowest. The interrupt priorities correspond to the interrupt addresses.

The address of the highest priority active interrupt within a group is generated from outputs of interrupt priority encoders by multiplexers H1 and G1. Outputs of group priority encoder E1 generate the group interrupt address. Note that interrupt address BUS11 through BUS15 is the complement of the actual interrupt address. The interrupt address is gated out to the three-state bus by $\overline{ENSEC0}$. The $\overline{INTERRUPT}$/signal (F1-9) is used to inform the processor that an interrupt is active. Interrupt 16, once active, indicates to the processor that a power failure, a protect fault, or a parity error condition has been detected. When the basic processor is used with the 1700 emulator, the 16 lower priority interrupt program interrupts are utilized to generate the standard sixteen 1700 macro interrupts. The 16 higher priority interrupts (data interrupts) are used as micro-level interrupts and are transparent at the macro level.

DATA FROM MEMORY

Data from memory (DFM) is gated out to the CPU three-state bus when the $\overline{ENMEMBUS}$ signal goes low at time T4. Refer to the Control 2 section for more details.

1700 TRANSFORM MODULE

Figure 5-11 shows the functional block diagram of the transform module.

Each block contains one or more numbers in the upper right corner. These numbers correspond to the logic diagram sheet where the function is located. Input and output signals, including the control signals to each block, are also included. This block diagram supplements the logic diagrams.

INSTRUCTION TRANSFORM (IXT) REGISTER

This register is a 16-bit register that holds the macro instruction currently being emulated. The register consists of D-type flip-flops B5, C5, A3, and J2. The macro instruction residing in the register is received from main memory via lines CPUDFM01 through CPUDFM16 (CPUDFM01 is the least significant bit). The outputs from the IXT register, I00 through I15 (I15 is the least significant bit), are coupled to the MA transform (selector S5); K/N transform (selector S8); delta translator; and F1, Δ=0, and GITMAK/XT decoders. Data-from-memory (DFM) bits are clocked into the IXT register on the low-to-high transition of $\overline{GATEIXT}$.

IXT' REGISTER

The IXT' (prime) register is an 8-bit register that holds the eight least significant bits of the macro instruction (CPUDFM01 through 08) from main memory. The register consists of D-type flip-flops B4 and C4. The data present at the D inputs is placed on the output lines (IXT'08 through IXT'15) on the low-to-high transition of the $\overline{GATEXTMIR}$ signal. This signal is only generated during the GITMAK/XT command. The outputs are coupled to the MA and K/N transforms (S5 and S8, respectively).

MA TRANSFORM

The MA transform (selector S5) is an 8-bit-wide selector that generates micro-memory addresses. This selector consists of eight 16-to-1 multiplexers: L2, L3, L4, L5, L7, L9, L10, and L12. These multiplexers permit 16 different micro-memory address (MA) transforms to be specified. Refer to section 4. The 8-bit output (S5-0/ through S5-7/) specifies one of 256 64-bit micro- memory instruction pairs residing within each page of read-only memory. Selection of the MA transforms is determined by the command (TMA/j, TMAK/j, or GITMAK/j) j value or is derived from the macro instruction during the GITMAK/XT command.

During TMA/j, TMAK/j, and GITMAK/j transform operations, select signals S5-S0 through S5-S3 directly correspond to MIR31 through MIR28, which are derived from the micro-memory bits (MM31 through MM28). During the GITMAK/XT transform operations, select signals S5-S0 through S5-S3 are generated based on the macro instruction being emulated. Multiplexer K12 selects one of the above cases depending upon the high or low state of the GETMAK/XT signal applied to K12-1. During the GITMAK/XT operation the output of the IXT register is first coded to select the MA transforms for

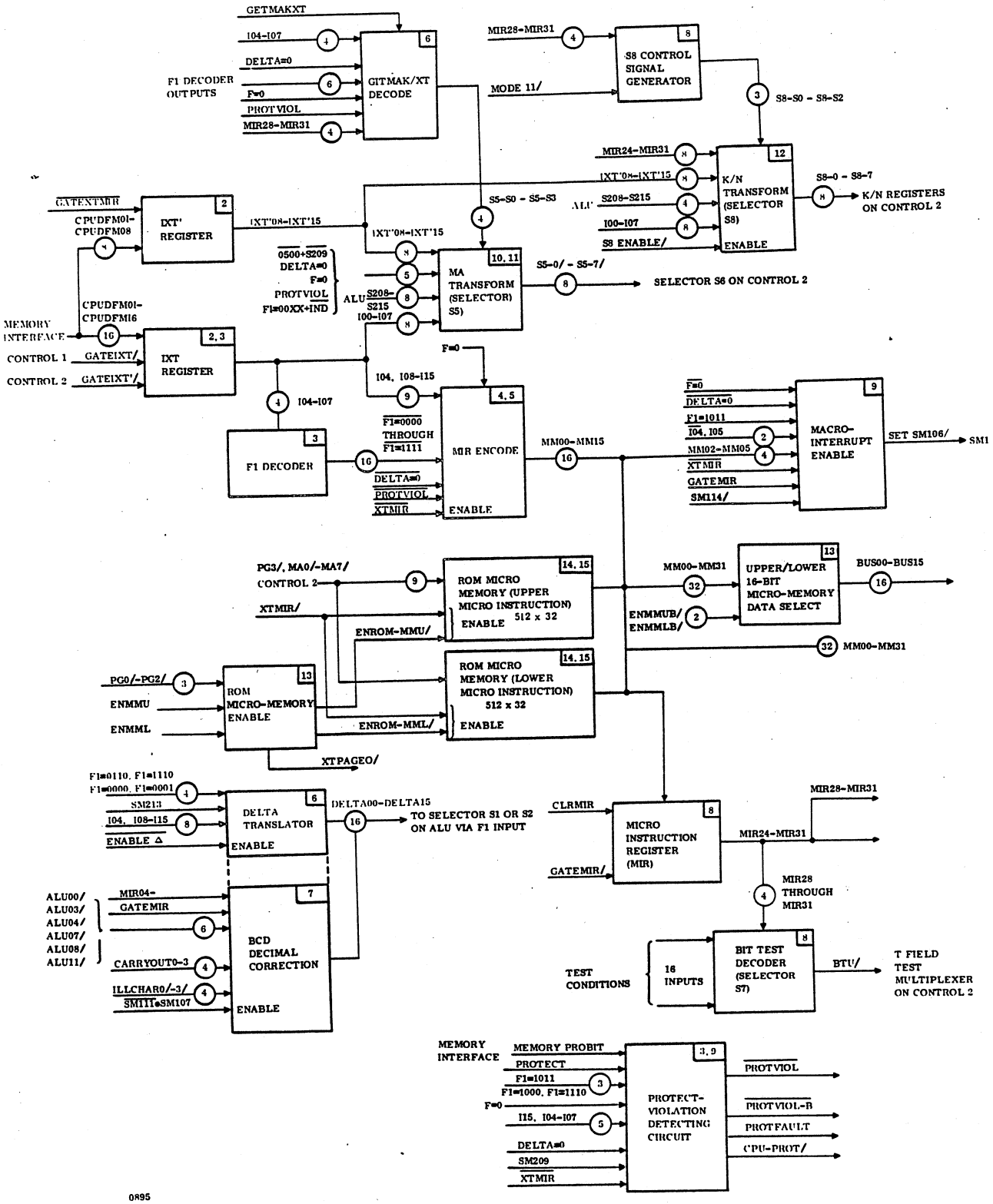


Figure 5-11. 1700 Transform With Binary-Coded Decimal Arithmetic (BCD) Functional Block Diagram

storage reference instructions (figure 5-12) when F is not equal to 0, and then for register and inter-register reference instructions (figure 5-13) when F equals 0. The F equals 0 signal applied to pin 1 of multiplexer D10 selects the MA transforms as follows:

F is not equal to 0	S5-S0 through S5-S3 are generated from the storage reference instructions.
F equals 0	S5-S0 through S5-S3 are generated from the register reference and interregister reference instructions.

The MA transform selections are shown in figures 5-12 and 5-13. These flow charts indicate the selection conditions rather than the sequential steps in selecting the MA transforms.

K/N TRANSFORM

The K/N transform (selector S8) consists of eight 8-to-1 multiplexers (K1, K2, K3, K4, K5, K6, K7, and K9) that are enabled by a low state of the S8ENABLE/ applied to pin 7 of each multiplexer. Selector S8 is used to choose one of eight sources for loading the K/N transform assignments. Refer to section 4. The high state of S8ENABLE/ disables S8 during the clear-N-register (CLRN), clear-K-register (CLRK), and clear-N-and-page-register (CLRNP) commands. These clear commands allow all zeros to be loaded into the N or K register. The input-selection signals S8-S0 through S8-S2 are derived from MIR29 through MIR31 by multiplexer H9 when MODE11/ and $\overline{\text{MIR28}}$ inputs to NAND gate J10 are both high.

Table 5-18 indicates that if MODE11/ is low (sequential address mode, MIR00 and MIR01 = 11), the S8 select signals (S8-S0 through S8-S2) select the MIR transform (XT/MIR, S8 position 6) to load MIR24 through MIR31 bits directly into the K or N register. If MODE11/ is high and $\overline{\text{MIR28}}$ (at H10 pin 3) is low, the S8 select signals select the transform (XT/FLDSTR, S8 position 7) to load into the K or N register. When both MODE11/ and $\overline{\text{MIR28}}$ are high, the selection bits (S8-S0 through S8-S2) are dependent upon the state of MIR bits (MIR29 through MIR31).

TABLE 5-18. K/N TRANSFORM POSITION SELECTION

MODE11/	$\overline{\text{MIR28}}$	S8-S2	S2-S1	S8-S0
1	1	MIR29	MIR30	MIR31
1	0	1	1	1
0	X†	1	1	0

† X = Don't care condition

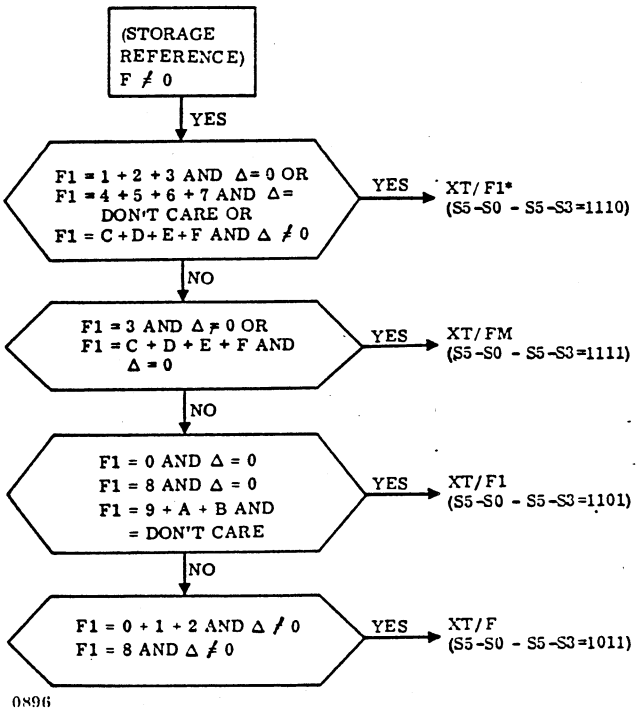


Figure 5-12. MA Transform Selection for 1700 Storage Reference Instructions

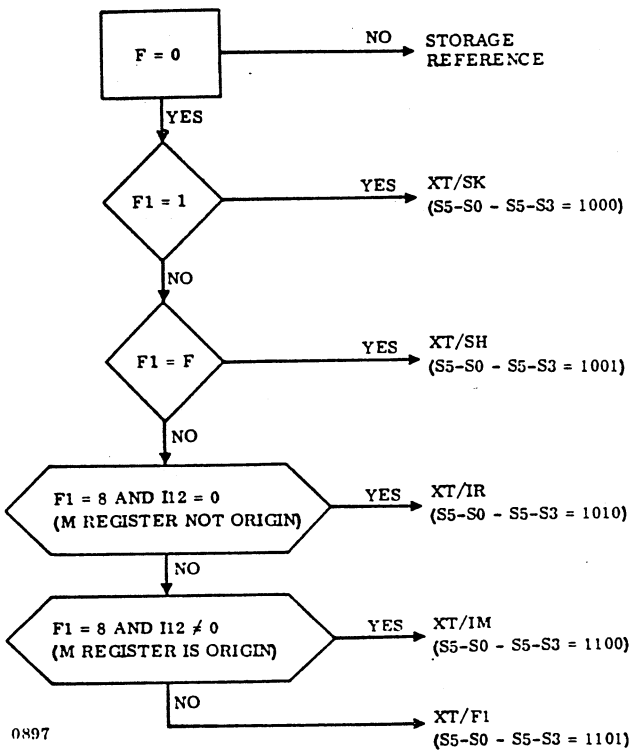


Figure 5-13. MA Transform Selection for 1700 Register Reference and Inter-Register Reference Instructions

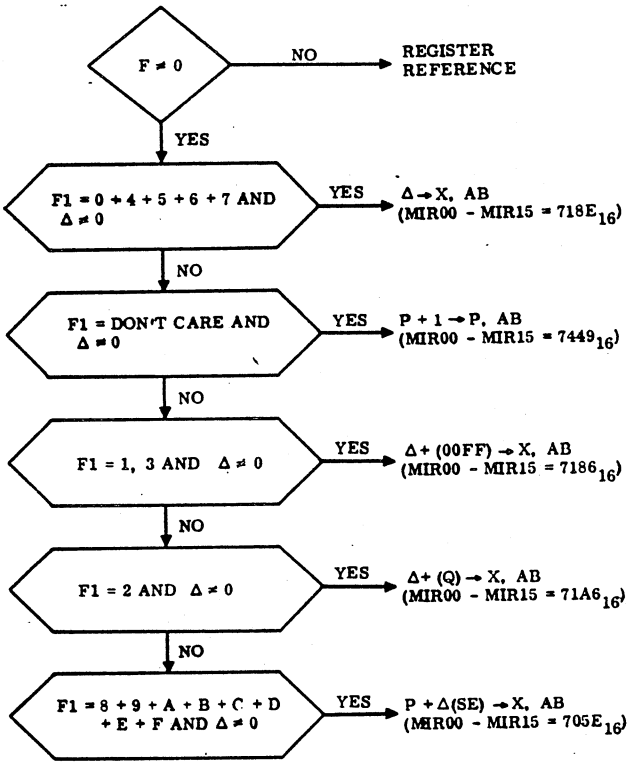
BIT TEST DECODER

The bit test decoder (selector S7) is a 16-to-1 multiplexer (H12) that tests up to 16 external and/or internal conditions. These test conditions are selected by the state of MIR bits (MIR29 through MIR31).

MIR ENCODE

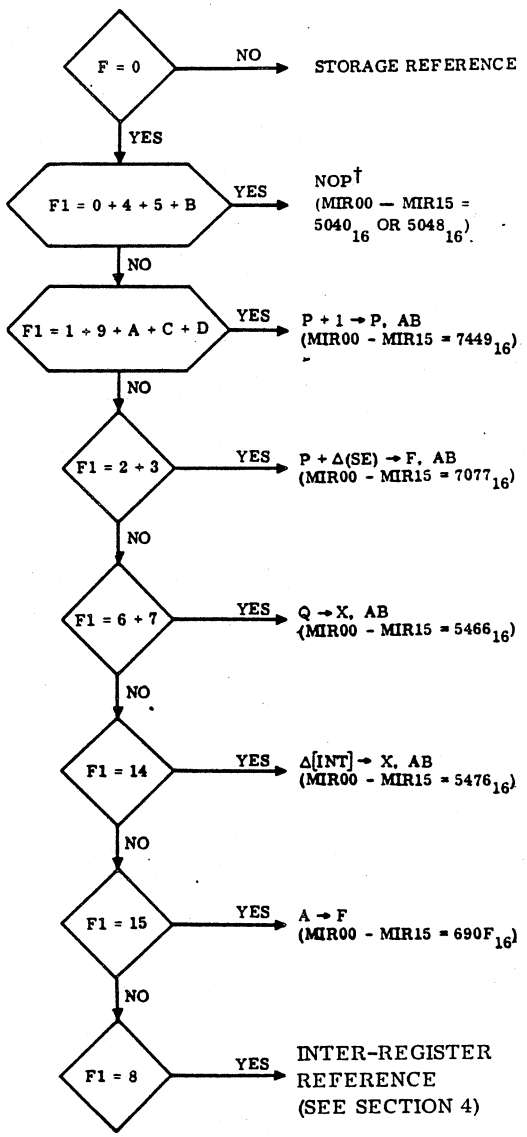
During the GITMAK/XT command the macro instruction is encoded to form the upper 16 bits (MM00 through MM15) of micro memory. These 16 bits are then loaded into the CPU micro-instruction register (MIR), which then determines the mode (MIR00 through MIR01) and the ALU control bits (MIR02 through MIR15) during the emulator's transform-on-next-instruction (XNI) cycle. Three types of MIR transform instructions are created (storage reference, register reference, and inter-register reference) based on the macro instruction being emulated.

The four 2-to-1 multiplexers E5, E6, G5, and G6 select one of two MIR transforms, either storage reference instruction (F≠0, figure 5-14) or register and inter-register reference instructions (F=0, figure 5-15). These multiplexers are enabled by the MIR transform signal ($\overline{\text{XTMIR}}$), which is only generated during GITMAK/XT operations. If a protect violation occurs, the D field of MIR encode is set 000 (no operation, NOP).



0898

Figure 5-14. MIR Transforms of 1700 Storage Reference Instructions



† NOP = THE D FIELD OF THE MICRO INSTRUCTION MUST BE 0 0 0.

0899

Figure 5-15. MIR Transforms of 1700 Register Reference Instructions

DELTA TRANSLATOR/BCD CORRECTION

The delta translator consists of 2-to-1 multiplexers A9, A10, C9, and C10 and associated AND and NAND gates A6, A7, B7, C6, and D6. These elements translate the delta field of the 1700 instruction in accordance with the instruction F1 field selections. The multiplexers are enabled when the SM111 and SM107 are not set and F = 0 or F1 = 1000 are low. This enable allows the bit conditions present on either the multiplexer 0 or 1 input lines to be placed on the delta output lines. The select signal input at pin 1 of the multiplexer is generated as follows:

$$\overline{SM213} \bullet F = 0 \bullet (F1=0+1+6+E)$$

If this select signal is high (the above equation satisfied), delta (Δ) is selected from the 0 inputs of the multiplexer; when it is low, the 1 inputs determine the delta. Status mode bit SM213 is set by the emulator only during emulation of enhanced instructions (that is, type 2 storage reference and field reference instructions).

The BCD correction logic is contained in its entirety on sheet 7 of the logic diagrams. The decimal-correction logic shares the three-state delta bus with the delta translator. Enabling of the decimal-correction logic is performed by the emulator during emulation of decimal arithmetic functions.

The four three-state multiplexers (A11, A13, C11, and C13) are enabled (pin 15 low) when SM107 is set and SM111 is not set. Each multiplexer has its own select-control logic which determines whether a zero (0000) or a six (0110) is placed onto the respective delta lines. The select-control logic decodes the four illegal character signals (ILLCHAR0/ through ILLCHAR3/); the four ALU carry signals (CARRYOUT0 through CARRYOUT3); ALU lines ALU00/, ALU03/, ALU04/, ALU07/, ALU08/, and ALU11/, and MIR04-. The result of this decode selects multiplexer inputs group 0 or group 1, where group 0 = 0110 (six) and group 1 = 0000 (zero).

READ-ONLY MICRO MEMORY

The read-only micro memory (figure 5-16) consists of 512 micro-memory words (64 bit) formatted in two pages (0 and 1).

Each micro-memory word consists of two 32-bit instructions (upper and lower). The upper instructions are contained in four 512- by 8-bit read-only memory (ROM) ICs E9, G9, E12, and G12. The lower instructions are contained in ROM ICs E7, G7, E10, and G10. The outputs of the corresponding upper and lower ROM ICs are wire ORED to form a 32-bit micro-memory three-state bus (MM00 through MM31). The upper instruction ROMs are enabled by ENROM-MMU/, and the lower instruction ROMs are enabled by ENROM-MML/. These enables provide individual selection of the upper or lower micro instruction. ROM ICs E9, G9, E7, and G7 (for MM00 through MM15) are disabled ($\overline{XTMIR}$ at input pin 19 low) during GITMAK/XT operation to permit 16-bit outputs of the MIR encoder (instead of a ROM instruction) to be loaded into the CPU micro-instruction register (control 2).

Each word in the ROM is addressed by memory address bits MA0/ through MA7/ and the least significant page bit PG3/. The MA0/ through MA7/ bits specify one of 256 micro-memory word pairs (upper and lower) within a page. The PG3/ bit selects the page (0 or 1) from which the instruction is read (high = page 0, low = page 1).

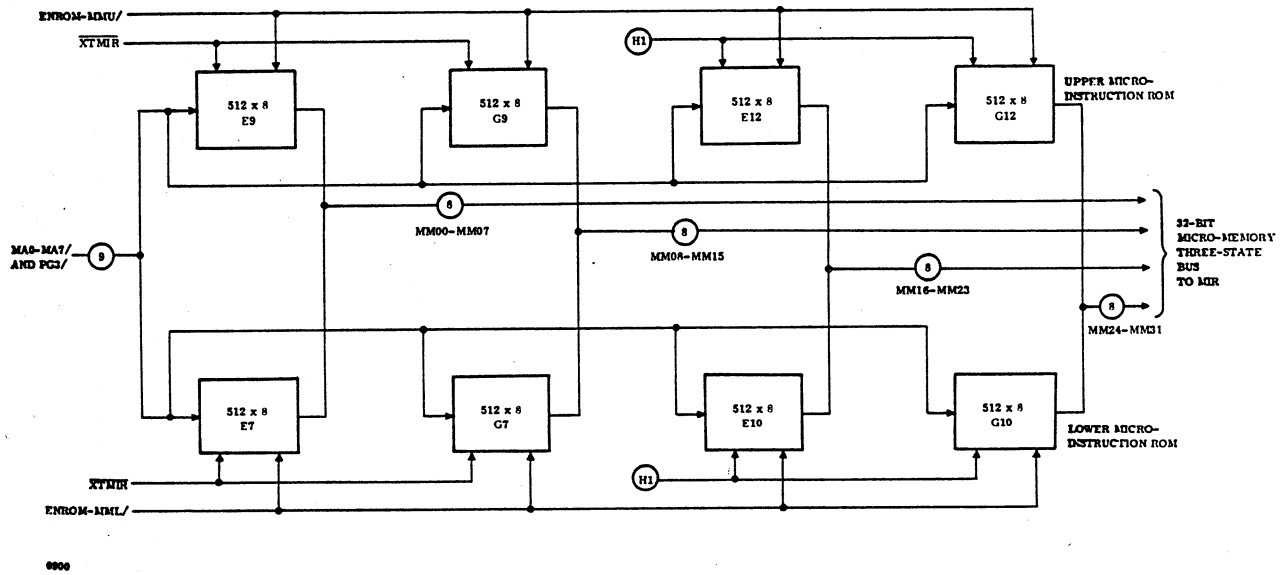


Figure 5-16. Simplified Read-Only Micro Memory Block Diagram

UPPER/LOWER MICRO-MEMORY DATA SELECT

During a read-micro-memory operation, the upper or lower 16 bits of data from micro memory are transferred to the X register via the CPU three-state bus (BUS00 through BUS15). Whenever the micro-instruction B' code contains either a read-micro-memory-upper or read-micro-memory-lower command, an associated ENMMUB/ or ENMMLB/ is generated. These ENMMUB/ and ENMMLB/ signals enable the micro-memory-select multiplexers DE13, E13, FG13, and G13 and select the micro-memory bits (MM00 through MM15 or MM16 through MM31) that will be placed on the three-state bus lines. ENMMUB/ selects upper micro-memory bits (MM00 through MM15) when low and selects the lower micro-memory bits (MM16 through MM31) when high. The selected micro-memory bits are then placed onto the CPU three-state bus when either ENMMUB/ or ENMMLB/ is low.

MISCELLANEOUS CIRCUITRY

Protect-Violation Detecting Circuit

The protect-violation conditions detected by the 1700 transform hardware set the protect violation ($\overline{\text{PROTVIOL}}$) when an attempt is made to execute the following instructions:

When the instruction is unprotected:

- Any inter-register instruction with bit 15 set (F = 0, F1 = 8, and I15 = 1)
- Enable interrupt (EIN), inhibit interrupt (IIN), set protect bit (SPB), and clear protect bit (CPB) instructions (F = 0, F1 = 01xx, and Δ = 0)
- Enable interrupt transform (EXI) instruction (F = 0 and F1 = E)

These violations are detected only when the instruction is unprotected and the protect switch is set. (The protect switch is set via function control register bit 08 in the breakpoint controller or panel simulation firmware when the breakpoint controller is not present.)

When a protected instruction follows the execution of an unprotected instruction: If the previous instruction is an unprotected instruction, then the protected-instruction flip-flop H5 is set during the read-next-instruction (RNI) cycle ($\overline{\text{XTMIR}}$ at J7-1 is low) by the GATEMIR clock. When both the enable-fault detection (SM209) and CPU protect are set, the output at H7-12 will be low if the next instruction is a protected instruction. This causes the PROTFAULT at H7-8 to set. At GATEMIR time, flip-flop H5 is reset, and flip-flop J4 is set simultaneously to keep PROTFAULT signals set for another memory cycle.

One exception is that if an interrupt caused the sequence of executing a protected instruction following the execution of an unprotected instruction, this is not a protect violation. The emulator handles this exception by clearing the SM209 bit whenever an interrupt is detected. This disables the fault-detect system and prevents PROTFAULT at H7-8 from setting. SM209 is set by hardware at time GATEMIR whenever a GITMAK/XT operation is executed.

When execution of an unprotected, miscellaneous instruction with the protect-switch set is attempted. All miscellaneous instructions are privileged instructions. $\overline{\text{PROTVIOL-B}}$ is generated by:

$$\text{G2-6(low)=F1=B} \quad \overline{\Delta = 0} \bullet (\text{PROTECT} \bullet \overline{\text{MEMPROTBT}})$$

All other protect violations are detected by the memory-interface hardware.

Interrupt Enable

Implementation of the interrupt function requires a combination of hardware and firmware. Status mode bit SM114 is first set by the 1700 emulator whenever the enable interrupt (EIN) instruction is emulated to pre-enable the interrupt function. If the instruction following EIN is neither a storage reference instruction (F≠0) nor an enhanced instruction, the hardware causes SETSM106 at J5-8 to be low at the next RNI cycle (XTMIR is high). The logic equation at J5-13 for this condition is:

$$\text{J5-13(low)}=[F=0 \bullet (\overline{F1=B}+\Delta=0) \bullet (I04+\overline{I05}+\Delta=0) \bullet (\text{GATEMIR} \bullet \text{XTMIR})]$$

If the instruction following EIN is a storage reference or enhanced instruction, SM106 is set after the 1700 emulator has executed a SUB-operation (increment P counter). The SUB-operation indicates the end of the macro-instruction emulation. The SUB-operation is decoded at J5-12 if the F field contains 1011 the following:

$$\text{J5-12(low)}=\text{MM02} \bullet \overline{\text{MM03}} \bullet (\text{MM04} \bullet \text{MM05} \bullet \text{GATEMIR})$$

Once status bit SM106 is set, SM114 is cleared by the hardware. If the macro instruction being emulated happens to be an inhibit interrupt (IIN) instruction (instruction = 0500_{16}) or a false interrupt (S209 = 1), then the $\overline{0500 + S209}$ signal at J3-8 is low.

$$\text{J3-8(low)} = \overline{\text{UNP-PROT}} + S209 + [\Delta=0 \bullet (F1=5 \bullet SM105) \bullet F=0)]$$

XTBLKT4 Signal Generator

The block transform at T4 signal (XTBLKT4/) is generated by flip-flop J9 whenever GATEXTMIR is high. Normally J9 is clocked by time T3 to set

J9-5 output high to allow destination to be strobed at time T4. When GATEXTMIR is low, XTBLKT4/ goes low to block the destination register strobe at time T4.

BLKM100 Signal Generator

To avoid emulator branches to the interrupt subroutine during macro-halt interrupt, a BLKM100 signal must be generated. This blocks the macro-halt interrupt from being set whenever the CPU is in step mode and GITMAK/XT is executed. The BLKM100/ is set low during GITMAK/XT operation whenever a memory reference or enhanced instruction is decoded at set input H5-10.

$$H5\text{-}10(low) = (XTMIR \bullet GATEMIR \bullet [\overline{F=0} + (\overline{F1=B} \bullet \Delta=0) + (\overline{I04} \bullet I05 \bullet \overline{\Delta=0})]$$

At the end of the instruction emulation, the emulator executes a SUB- operation (increment P counter) that sets the BLKM100/ signal high. BLKM100/ is also set high by a master-clear-delayed signal (MCDELAYED/).

I/O-TTY CONTROLLER MODULE

Detailed descriptions of the internal I/O-TTY controller functions are illustrated by block diagrams. These diagrams contain blocks that represent groups of logic that perform functions: numbers in the upper and lower right corners of these blocks indicate the logic diagram sheet that contains the functional logic group. The number in the upper right corner pertains to I/O-TTY controller PWA 88909701, and the number in the lower right corner pertains to I/O-TTY controller PWAs 96744241 and 96752182. Also provided is a detailed chip-by-chip description of the logic conditions that produce the required effects to provide the I/O-TTY controller operations. The alphanumeric reference contained in parenthesis ( ) and brackets [ ] in the text designate logic element location on the PWA. The numbers in parenthesis apply to I/O-TTY controller PWA 88909701 and the numbers in brackets apply to I/O-TTY controller PWAs 96744241 and 96752182.

The I/O-TTY logic diagrams contain diagonal (/) and bar (‾) symbols to indicate an active low (not) condition of the signal. The diagonal is used to designate the low condition of external read and write signals and the bar is used to designate the low condition of internal signals (for example, SD01/ through SD16/, ADR01/ through ADR16/; $\overline{D01}$ through $\overline{D16}$, $\overline{Y01}$ through $\overline{Y16}$).

INPUT REGISTERS FUNCTION

The input registers function is shown in figure 5-17.

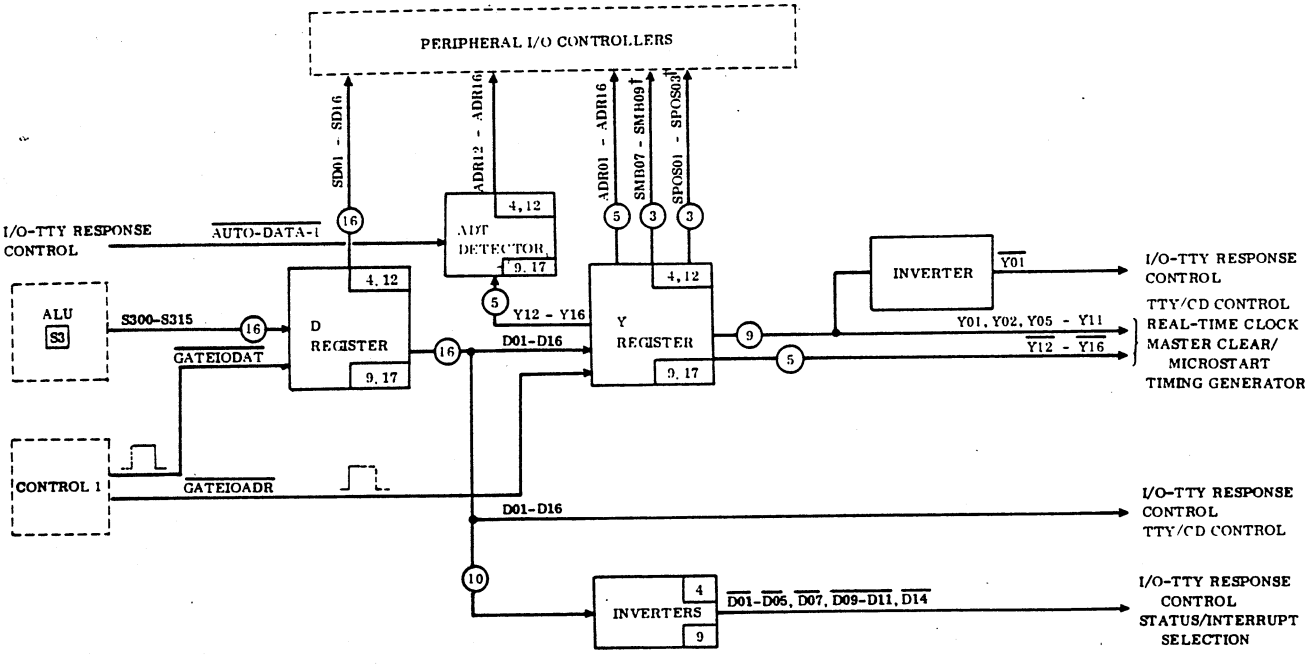


Figure 5-17. Input Registers Function

The I/O-TTY controller input registers receive data and address words from the CPU via selector 3 of the ALU for use in the I/O-TTY and peripheral I/O controllers. The registers are designated D (data) and Y (address) and are direct extensions of the CPU A and Q registers, respectively. The D register is loaded by execution of the CPU micro instruction D', which initiates a control signal within the control 1 module that produces the gate I/O data and gate I/O address signals. These signals gate data and address words into the I/O-TTY controller.

When D' equals 000, the GATEIODAT signal goes low and the data word is transferred from the ALU, selector 3, into the D register via lines S300 through S315. If it is a data word, D01 through D16 and SD01/ through SD16/ (flip-flops A11, C11, E11, and G11) [flip-flops A10, C10, F10, and F9] are set according to the input word bit settings to apply the D-register director functions or character data to the I/O controllers. Figure 5-18 illustrates the I/O-TTY controller director functions.

When D' equals 001, the GATEIOADR signal goes high and the word residing in the D register is transferred to the Y register (flip-flops B11, D11, F11, and H11) [flip-flops A9, C9, G10, and F8]. The data bits of the address word set output lines ADR01 through ADR16 and internal lines Y01 through Y16 low and high to impose the address word contents upon the peripheral I/O controller and I/O-TTY controller functions. Outputs Y12 through Y16 are ANDed at gates (L9 and H11) [L10 and G8] with the automatic data transfer ($\overline{\text{AUTO-DATA-1}}$) signal to force the W field equal to zero during an ADT operation and produce the ADR12/ through ADR16/ signals.

Bits D01 through D05, D07, D09 through D11, and D14 are inverted at (A8, A4, E8, and G4) [A8, C8, M4, and D4] to provide the complement signal to the internal logic and to avoid applying the internal load to the associated output lines (SD01 through SD16). Bit Y01 is inverted to provide both the true and complement signals to the internal functions.

When M05 set/sample capabilities are provided, output lines ADR02 through ADR04 and ADR05 through ADR07 are designated SMB07 through SMB09 and SPOS01 through SPOS03, respectively. The SMB and SPOS lines control the M05 operating mode and the port selection of M05 peripheral devices. When SMB09 is set, the 16 bits of information contained in the data register are placed on output lines SD01 through SD16. When SMB09 is 0, the 16 bits of information on input lines RD01 through RD16 are transferred via the I/O-TTY controller to the three-state bus. Bits SMB07 and SMB08 are not defined and may be employed at the user's discretion. The octal code designation of the selected port, SPOS01 through SPOS03, is transmitted on the port line SPT.

I/O-TTY CONTROL RESPONSE FUNCTION

The I/O-TTY control response function (figure 5-19) provides the I/O-TTY controller control responses (reply, reject, character input, and enable reply) to the CPU three-state bus via the data read function. It also provides the I/O-TTY controller auto-data transfer interrupt signals (RDINT01) to the SMI for incrementing the TTY ADT table sequences, and controller internal signals (SELTTYSTATUS, AUTO-DATA, REPLY+REJECT, and clear interrupt) for processing within the associated function logic.

ADT Detection

The TTY ADT interrupt (TTY-ADT-INT) sends an interrupt to the SMI to increment the auto-data transfer micro-level sequences used by the I/O-TTY controller. After the operating function has selected ADT, the clear ADT detector (latch D4, OR gate A4, and AND gate E1) [latch A5, OR gate A6,

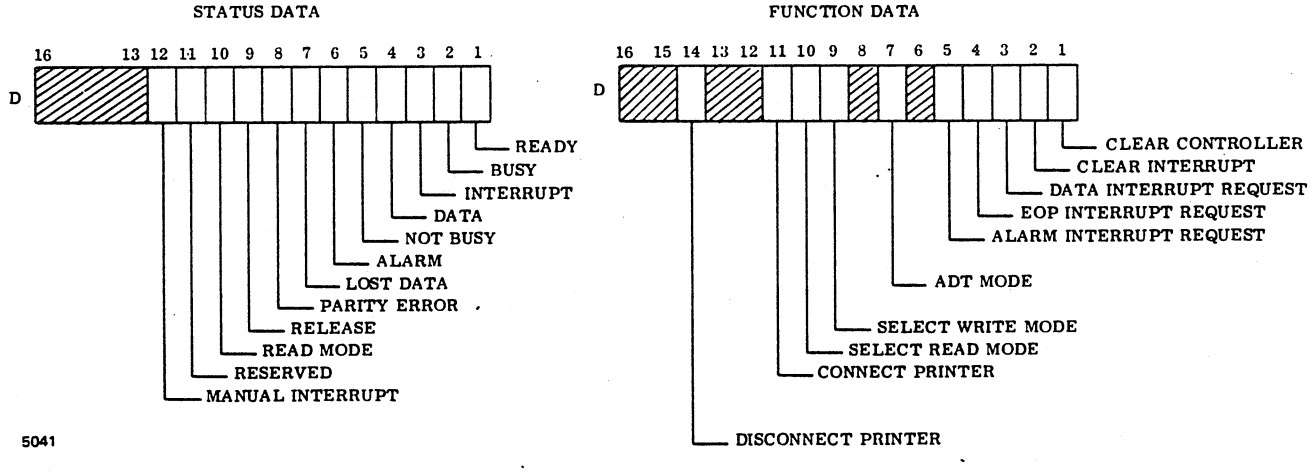


Figure 5-18. D Register Bit Definitions, Director Data

and AND gate D3] monitors the ADT line and permits the ADT sequence each time the data status remains active and the terminate signal is inactive. If the ADT sequence has been completed, a TERMINATE active signal is present at time T2 to inhibit the RDINT01/ to the SMI and to initiate the end-of-operation status that is sent to the status/interrupt selection function.

Normally when the ADT sequence is in progress, the RDINT01/ signal is alternately high and low as the data status signal changes.

The ADT-SELECTED signal is developed at flip-flop (D4) [A5] when director function word bit $\overline{D7}$ produces a high at pin 13. This high sends Q output pin 15 high when the clock input (determined by a WRITE Y1 present at time T3) goes high to produce ADT-SELECTED. This signal remains high until a clear interrupt is applied to flip-flop D4.

RDINT01/ goes low when the clock input (derived by AUTO-DATA and T2 being high at K4) and TERMINATE are high at flip-flop (K1) [C3]. This terminate condition is applied to AND gate (E1) [D3] and AND/OR gate (G6) [E1, D1] to inhibit RDINT01 and initiate the end-of-operation when ADT-SELECTED goes high.

RTC

When the real-time clock function has been activated, it senses that AUTO-DATA, CPU PROTECT, and RTC SEL are all active. These conditions enable the reply/reject selector to send the reply signal response to a CPU read or write if the protect conditions are met. If the protect conditions are not met, a reject signal is sent.

When the real-time clock has been selected (W=0, E=1, S=7), the low output at AND gate (F1) [J3] pin 8 sets flip-flop (E2) [H3]. The next time T3 is high at pin 11, the Q output is high. This high is applied to AND gate (F2) [G7]. If AUTO-DATA/ or CPU-PROT/ at pins 12 or 13 of OR gate (L9) [M5] is low, a protect high is applied to pin (2) [4] of AND/OR gate (G1) [H2]. Since the real-time clock has been selected, pin (3) [5] of (G1) [H2] is also high to drive pin 6 low. This low inhibits the reject signal output at pin (12) [1] and AND gate (F2) [G7] and may be inverted by (G4) [G3] to

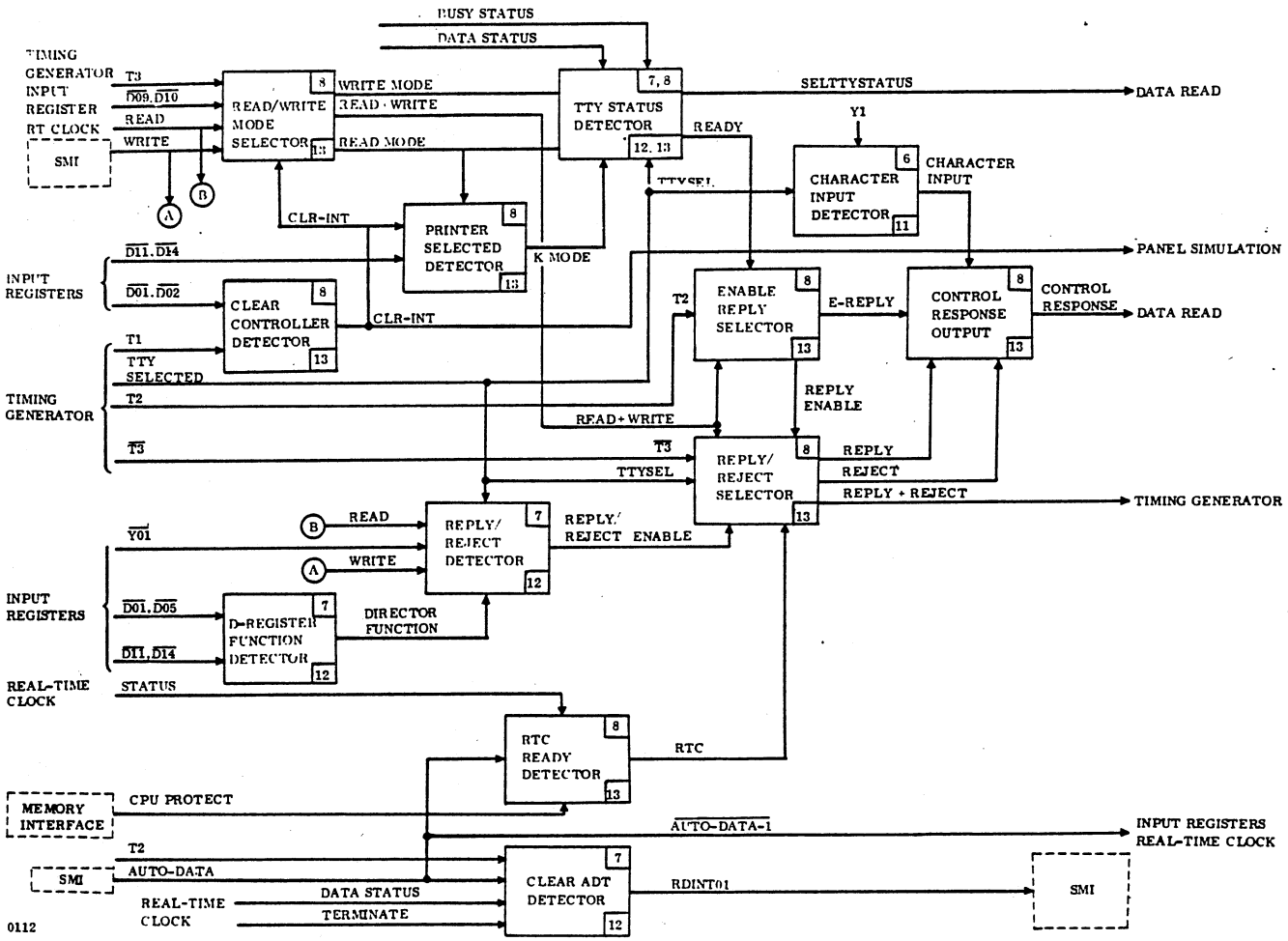


Figure 5-I9. I/O-TTY Response Control Function

enable the reply signal at pin (8) [11]. If the protect condition does not exist, the AND/OR gate output at pin 6 is high. This high enables the reject signal and inhibits the reply signal output of the AND gate (F2) [G7].

Read/Write Selection

When a write or read request is present and the associated bit (D09 = select write mode, D10 = select read mode) is active, read/write mode selector (J2) [K3] produces the read or write mode condition at time T3. This condition combines with the ready conditions (printer connected, data, busy, and TTY selected) to send a READY signal to the enable reply selector. When the ready condition is present at time T2, the enable reply selector produces the $\overline{\text{E-REPLY}}$ control response for the data read function and enables the reply/reject selector. If these conditions do not exist, the reject condition is produced.

When either $\overline{\text{D9}}$ (write mode) or $\overline{\text{D10}}$ (read mode) is high at OR gate (G2) [J2] pins 9 or 10, a high condition exists at pin 13 of AND gate (H2) [K2]. If $\overline{\text{D9}}$ is high, this high also enables read/write flip-flop (J2) [K3]. If the (WRITE)(Y1) condition is high at time T3, the write output is selected; if (WRITE)(Y1) is low, the read output is selected. The selected condition is ANDed with READ, WRITE, and Y1 at AND gate (D1) [G1]. The correct output is coupled through OR gate (C1) [F1] and ANDed with the data status condition. If DATA STATUS is high, the data status direction right $\overline{\text{(DS)(DIR-RT)}}$ low output is applied to OR gate (G2) [J3]. This activates (G2)[J3], which enables flip-flop (E2) [H3]. At time T2, the (E2) [H3] flip-flop is set to provide the enable TTY reply signal ($\overline{\text{E-REPLY}}$) to OR gate (E1) [G1] and to initiate the read or write sequence at the UART.

Character Input

The character input (CHAR INPUT) response is produced when TTYSEL and $\overline{\text{Y1}}$ are high at pins (12 and 13) [1 and 2] of AND gate (D5) [C5]. (TTYSEL is high when AND gate (D7) [F7] detects that the Y register WES fields equal 0 1 1.) This drives the output (pin 11) [pin 3] of AND gate (D5) [C5] low, which is inverted by inverter (D6) [D4] and inversion AND gate (F2) [G7] to provide the $\overline{\text{CHARINPUT}}$ condition at pin (3) [6] of (F2) [G2]. This $\overline{\text{CHARINPUT}}$ is coupled to the CPU via the data read logic (flip-flop J8 and multiplexer J7) [flip-flop G6 and multiplexer F6] to select the character data loading into the lower eight bits of the CPU A register.

Director Functions

When the CPU requests a write operation, all functions that the TTY can perform will reply. The director function bits assigned to printer application (D11 = connect printer, D14 = disconnect printer) and the director function bits assigned for interrupt application (D01 = clear controller, D02 = clear interrupt, D03 = data interrupt request, D04 = end-of-operation interrupt request, and D05 = alarm interrupt request) are applied to the D register function detector. If one or more of these bits is active, a director function active condition is produced and combined with the write request to produce a reply enable. If none of these bits is active, the director function is inactive. This causes a reject enable.

Reply/Reject

Since reply/reject flip-flop (E2) [H3] is latched during time T3, and enable is applied to reply/reject AND gates (F2) [J7]. D-register bits D01 through D05, D11, and D14 are applied to OR gates (C6, D2, and C5) [C6, B5, and B6]. If any of these bits is high, a low condition is applied to OR gate C6 pins (1 or 4) [5 or 4]. If pin 1, 2, 4, or 5 of OR gate C6 goes low, a high is applied to pin (2) [4] of AND gate (C1) [C5]. If this high is present at (C1) [C5] when (WRITE)(Y1) is high, (C1) [C5] applies a low to pin (5) [4] of OR gate (E1) [G1]. (E1) [G1] provides a high level (E-TTYREPLY) to AND/OR gate (G1) [H2] pin (5) [2]. If TTYSEL is also high, a low is applied to AND gate (F2) [G7] pin (12) [1], which inhibits the reject. This (G1) [H2] low condition is also applied to inverter (G4) [G3] to provide a high to AND gate (F2) [G7] pin (9) [12]. This activates (F2) [G7] to provide a reply low ($\overline{\text{REPLY}}$) condition to the CPU via the data read function. If the director function bits are active but (WRITE)(Y1) is low (or vice versa), AND gate (C1) [C5] produces a high output that does not trigger OR gate (E1) [G1] but inhibits AND/OR gate (F2) [G7]. (F2) [G7] sends $\overline{\text{REJECT}}$ to the CPU via the data read function. If READ and Y1 at pins (12 and 13) [1 and 2] of AND gate (C1) [C5] are both high, a low output from (C1) [C5] initiates a reply, providing TTYSEL is also high. If any of these signals is low, the reject is activated.

Printer Selected (K Mode)

When data bits D11 or D14 are active at pins 12 or 13 of OR gate (C5) [B5], a high appears at AND gate (D2 pin 9) [K2 pins 1 and 2]. This high is ANDed with (WRITE)(Y1) and time T3. If all are high, a high is applied to the clock input of flip-flop (J2) [K2]. If data bit D11 is high (connect printer), data bit D14 is low. D14 low is applied to the enable input of (J2) [K3]. When the clock pulse is applied to (J2) [K2], the Q output goes high and is ANDed with BUSYSTATUS at AND/OR gate (G1) [H2]. If both are high, the resulting output (low) is ANDed with (WRITE)(Y1) to produce a high output at AND gate (G2) [J2], which initiates a reject output and inhibits the $\overline{\text{E-REPLY}}$ signal to the UART enable logic. If D11 is high and BUSYSTATUS is not active, the resulting output is AND/OR gate (G1) [H2] is high. This high is ANDed with the (WRITE)(Y1) high at AND gate (G2) [J2]. If both are input high at AND gate (G2) [J2], the reply signals are initiated.

Clear Interrupt

The clear interrupt provides the clear controller signal to printer-connected (K MODE) and UART logic. This function is initiated when the WRITE and Y1 signals at pins 4 and 5 of AND gate (C2) [B5] are high. (WRITE)(Y1) high output is applied to AND gate (C2) [K4] at time T1, pin (11) [8] of

(C2) [K4] goes high. This high is applied to pins (9 and 5) [1 and 5] of AND/OR gate (H1) [K5] to generate the clear controller (CLR-CONT) and clear interrupt (CLR-INT) signals. If D1 is high at gate (H1 pin 10) [K5 pin 13], the clear controller output at pin 8 is applied to flip-flops (J2) [K3] to clear the printer-connected (K MODE) and UART logic. IF D2 is high at (H1) [K5] pin 4, CLR-INT is applied to the ADT-selected, terminate, and manual interrupt logic.

PANEL SIMULATION FUNCTION

The panel simulation function (figure 5-20) is available when associated firmware is supplied. It provides a modified panel mode capability when the breakpoint controller is not supplied. When the microprocessor is operating as a CDC 1700 enhanced computer, the panel simulation function lets the operator control and display all registers and memory that may be used at the macro level. The control conditions provided by the simulated breakpoint controller function are external stop, selective stop, protect, selective skip, enable micro memory write, and multilevel indirect addressing. These conditions are under control of the CDC 1700 enhanced program panel simulation firmware and are available to the CPU via data bits D04 through D06 and D13 through D16. (The bit assignments are listed in table 5-19.) There are two panel simulation interrupts: RDINT12 advises the CPU that data is present and RDINT13 advises the CPU to send data to the UART.

TABLE 5-19. FUNCTION CONTROL BIT DEFINITIONS

Bit	Definition
D04	Enable console echo
D05	Enable micro memory write
D06	Multilevel indirect addressing
D13	External stop
D14	Selective stop
D15	Selective skip
D16	Protect

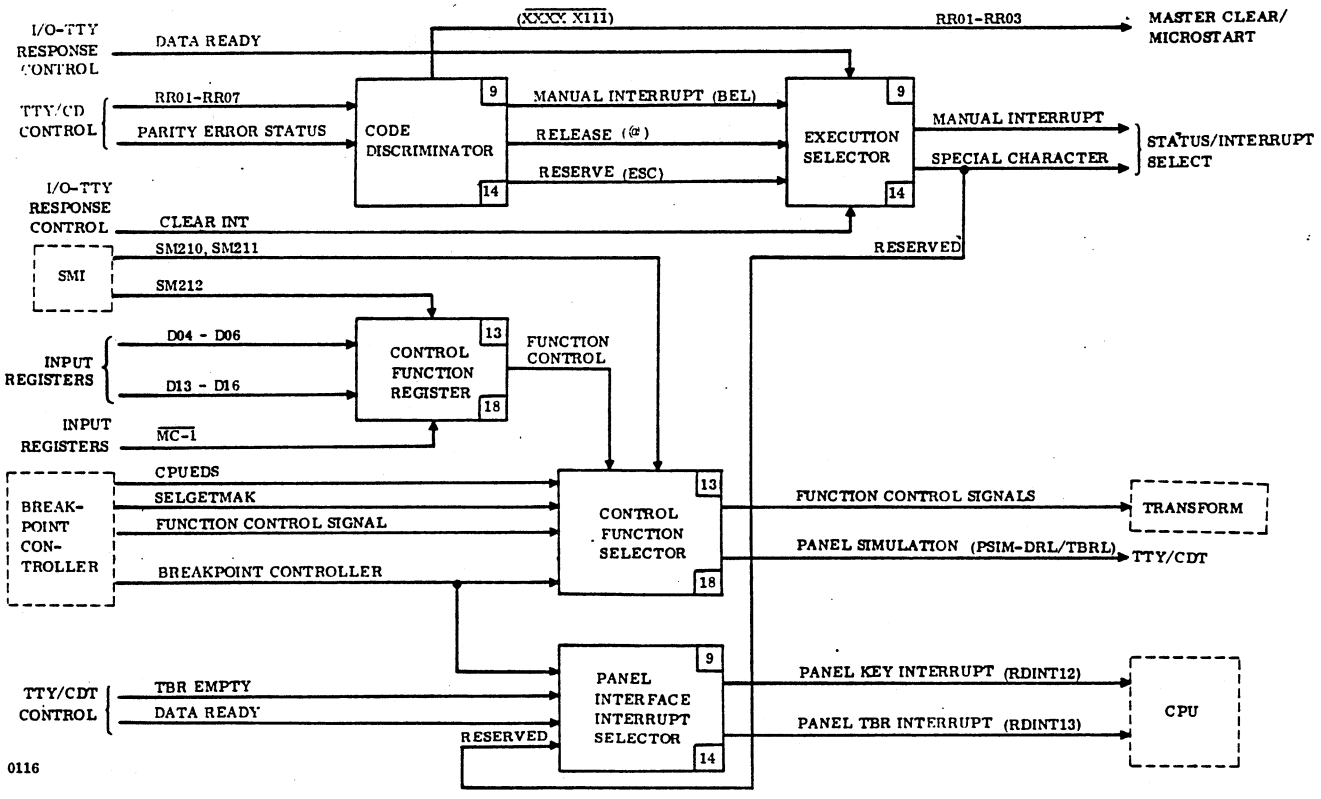


Figure 5-20. Panel Simulation Function

Panel Simulation

When a breakpoint controller is supplied, a low (PNLPRES/) applied to inverter (M4) [G3] and AND gate (K8 and L4) [G2] inhibits the RDINT12/ and RDINT13/ AND gates. The output of inverter (M4) [G3] applies a high to (M3 pin 9) [F2 pin 12] to enable the application of a RESERVED signal (panel simulation active) to the I/O-TTY data ready logic and to the set inputs of 10-to-4-line multiplexers B10 and (H10) [H9], which enables the panel interface inputs (I1) and inhibits the simulation inputs (I0). This enabled condition inhibits the panel simulation ($\overline{\text{PSIM-DRR}}$ and $\overline{\text{PSIM-TBRL}}$) signals by holding the B10, I1c, and I1D inputs at ground. A gound is also applied to I1A of (H10) [H9] to disenable AND gate K10 to inhibit selection of the EXSTOP/ control function.

When the breakpoint controller is supplied, resistor R10 maintains a high condition at the inputs of AND gates (K8 and L4) [G2] and inverter (M4) [G3]. These highs enable the RDINT12/ and RDINT13/ control functions controlled by AND gates (K8 and L4) [G2] and cause a low at the output of inverter (M4) [G3]. With the enable input and set inputs of multiplexers B10 and (H10) [H9] low, the outputs of panel simulation function control register flip-flops (C10 and G10) [B9 and G9] are enabled. Therefore, when data bits D04 through D06 and D13 through D16 contain a control function selection at the next high condition of SM212, the designated function is sent to the CPU. This is true for all except the EXSTOP function; before EXSTOP can be enabled by a high at pin (5) [12] of AND gate K10, the CPU-EDS/ and SELGETMAK/ signals applied to the inputs of AND gate J9 must be low to product a high to pin (4) [13] of K10. If a master clear low ($\overline{\text{MC-I}}$) is applied to the clear input of flip-flops (C10 and G10) [B9 and G9], all outputs are set low no matter what state exists on the CP and D inputs.

Special Character Code

The special character codes ESC, @, and BEL select the reserved status (I/O-TTY controller character data inactive, breakpoint controller character data active), release reserved status (I/O-TTY controller character data active, breakpoint controller character data inactive), and manual interrupt, respectively. Function and status operations within the I/O-TTY controller are still permitted. These character codes, ESC ($1B_{16}$), @ (40_{16}), and BEL (07_{16}) are derived from data bits RR01 through RR07 applied to code discriminator multiplexers (K3 and L2) [E4 and F4]. (K3) [F4] detects the status of RR01 through RR03 and (L2) [E4] detects the status of RR04 through RR07 to initiate the selected action. The Q3 output of multiplexer (K3) [F4] pin 4 is low when RR01 through RR03 are all high ($\overline{\text{XXXX}}$ $\overline{\text{X111}}$) and the Q0 output of (L2) [E4] pin 9 is low when RR04 through RR07 are all low (0000 0xxx). Adding these data bits produces binary code 0000 0111, which equals the BEL code (07_{16}). Lows applied to pins (5 and 6) [8 and 9] of AND gate (L3) [F3] produce the high output (manual interrupt) applied to the manual interrupt detector logic of the status/interrupt function and inhibit the data ready response to the UART. The inhibit is produced when inverter (M4) [G3] provides a low to OR gate (L4) [G2], which produces a high at AND gate (M3) [F2] to inhibit the TTY data register ready ($\overline{\text{TTY-DRR}}$) signal.

All special character codes are produced in the manner described above. Therefore, when both inputs at pins (2 and 3) [8 and 9] of AND gate (L3) [E3] are low (application of ESC code to (K3) [F4] and (L2) [E4], a high is applied to OR gate (M2 pin 12) [F3 pin 5] and AND gate (M3) [F2] pin 2. This high produces a low at (M2 pin 13) [F3 pin 4], and since the @ symbol code has not been selected, a low exists at point 2 of AND gate (M2) [F3]. These two low inputs at (M2) [F3] produce a high at pin 1 of (M1) [F3], which applies a high to pin 12 (D) of flip-flop (M1) [E2]. Since the flip-flop $\overline{\text{Q}}$ output is high, the high and low inputs at pins (5 and 6 of M2) [2 and 3 of E3] apply a low to pin (D) of flip-flop (M1) [E2] (select the not released signal $\overline{\text{RELEASED}}$) [deselect the release signal].

In addition, since data ready (DRDY) is high (indicating the presence of data, the escape code), the inverted output of M4 places a high at pin 1 of AND gate M3. The two high levels at pins 1 and 2 of M3 provide a low to set input pin 10 of flip-flop M1. WIth the low and high levels of pin 10 applied to pins 12 and 11 of M1, the Q output pin 9 becomes high to select the reserved status. [If data ready (DRDY) and escape (ESC) are both high at AND gate F2, a high-to-low transition is applied to one-shot MV L4. If the deadstart signal SM204 is high, this low applied to L4 will activate the one-shot MV. The $\overline{\text{SET-RSRV}}$ output goes high and sets reserve flip-flop E2 to provide an active condition for the breakpoint controller. When the output of AND gate F2 is high; if the SM204 input goes low, the one-shot MV is activated. This also causes the reserve flip-flop to be set.] RESERVED is coupled through the data read function to the CPU three-state bus. The high output at AND gate (L3 pin 1) [E3 pin 10] is also inverted by (M4) [G3] to apply a low at pin (3) [2] of OR gate (L4) [G2]. (L4) [G2] applies a high to pin (12) [9] of AND gate (M3) [F2] and a low to AND gate (K4 pin 9) [B3 pin 10]. Since the data ready signal at (M3 pin 12) [F2 pin 9] is low, the special character code is inhibited from entering the data receive register of the UART.

When the @ symbol code is selected by the TTY/CD, input pins (8 and 9) [11 and 12] of AND gate (L3) [E3] are low, driving pin (10) [13] high. Since the reserved condition is holding pin (11 of M2) [6 of F3] high, the low at pin (13) [4] being ANDed with the high from (L3 pin 10) [E3 pin 13] at AND gate (M2) [F3] produces a low-level output at (M2) [F3] pin 1. This low is applied to pin 12 (D input) of flip-flop (M1) [E2] to send the Q (pin 9)

input low to remove the reserved status; at the same time, the low at AND gate (M2 pin 5) [E3 pin 3] is being ANDed with the low $\overline{Q}$ (pin 8) output of (M1) [E2]. These two low levels at (M2 [E3] produce a high that is applied to the D input of (M1) [E2], setting the Q (pin 5) output high to select the released status. In this case, the special character code is not inhibited from the UART and the @ symbol appears on the output display.

TELETYPEWRITER/CD CONTROL FUNCTION

The teletypewriter/CD control function (figure 5-21) provides the I/O-TTY controller communication link for the transfer of character code data (teletypewriter, RS232-C, or TTL level) to and from the CPU. The communication link provides three transmission interface methods:

Teletypewriter (20 mA current loop)

CD or breakpoint controller (RS232-C level)

M05 or breakpoint controller (TTL level)

All external character codes are transmitted in asynchronous serial format, 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 controller is parallel format of seven- or eight-character code bits.

This function also provides character echo response and sharing of the teletypewriter or CD between the I/O-TTY controller during program mode, and the panel interface controller during panel mode. This sharing is accomplished by processing particular special character codes (ESC, @, and BEL) as control characters. These codes are reserved as control codes and cannot be accepted as message data.

Reserve status - The ESC ($1B_{16}$) code transfers control of the TTY/CD from the I/O-TTY controller to the breakpoint controller.

Release reserve - The @ (40_{16}) code transfers control of the TTY/CD from the breakpoint controller to the I/O-TTY controller.

Manual interrupt - The BEL (07_{16}) code can be used to impose a manual interrupt from the input device rather than from a console manual interrupt switch.

The Y-register status field bits Y05, Y06, Y07, and (W=0)(EQ=1) are deciphered to detect the TTY selected (TTYSEL) condition. TTYSEL is combined with data bit D01 and (WRITE)(Y1) to produce the master reset that activates the UART master reset. The UART is cleared to permit loading of character code words into the UART receive register. Character code words are loaded at the rate of one bit for every 16 clock pulses. TTYSEL high is ANDed with function word bit D01 (clear controller) and (WRITE)(Y1). If all are high at AND gate (A2) [D3], a master reset condition appears at pin (8) [6]. Since a master clear is not in effect, $\overline{MC-1}$ is high and the master reset condition produced by (A2) [D3] is present at OR gate (B1) [C5], the high is applied to the MR input of the UART. The UART logic resets to facilitate loading of data into the UART. At the same time, TTY DATA (a combination of TTYSEL and $\overline{Y1}$) is inverted at (D6) [D4] and compared with the $\overline{REPLY+REJECT}$, READ+WRITE, and $\overline{E-REPLY}$ signal at AND gate (B2) [H5]. If all are high, the result is compared with write mode (WT-MODE) at AND gate (F1) [B2] and with READMODE at AND gate (G2) [B2]. If WT-MODE is high, the transmitter loading is enabled; if READMODE is high, the receiver loading is enabled.

Reciever Enabled

The receiver may be loaded from the teletypewriter, CD, any TTL serial data generator, or any device that produces a compatible RS232-C signal. The teletypewriter transmission line is a 20 mA current loop that is detected by 20 mA-to-TTL converter (Q2) [Q1] (see 20 mA-to-Digital Converter below). The RS232 inputs are ± 3 to ± 12 V amplitude signals that are converted to 0 to +5 V amplitude signals by converters (H5) [H6]. If any one of the four inputs is active at OR gate (H2) [H5], the loading of the UART receiver register is enabled and the data codes are applied to selector multiplexer (E7) [E9]. The input data loaded into the UART (EF6) [DF6] receiver register is in asynchronous serial format; it has one start bit, seven data bits, one parity bit, and one or two stop bits. Data bits loaded into the receiver register are clocked in by the baud rate frequency generator at the rate of one bit every 16 Hz. When the receiver register is loaded and checked for parity, framing, and data status, a data ready signal is generated and the data is shifted to the receiver buffer register and held until a CPU read is generated.

The output data is transferred to the data read function, master clear/microstart function, and panel simulation function (figure 5-20). The parity error (P.E.), framing error (FERROR), and LOST DATA STATUS signals are sent to the status/interrupt function. The data ready signal (DRDY) is sent to the panel simulation function and to AND gate (M3) [F2], where it is ANDed with the restricted codes (BEL, ESC, and @). If the restricted code signal is high, the AND gate low output is coupled through OR gate (H4) [D2] and (F8) [D2] to cause an immediate low at the UART DRR input. This low resets the UART data ready circuit to inhibit the data ready signal to the CPU.

The TTY/panel selector is a 7-to-1 multiplexer (E7) [E9], which selects the CD UART output, character data input, or the breakpoint controller with or without echo response in accordance with the operation code selection signals (breakpoint controller present, reserved, and printer

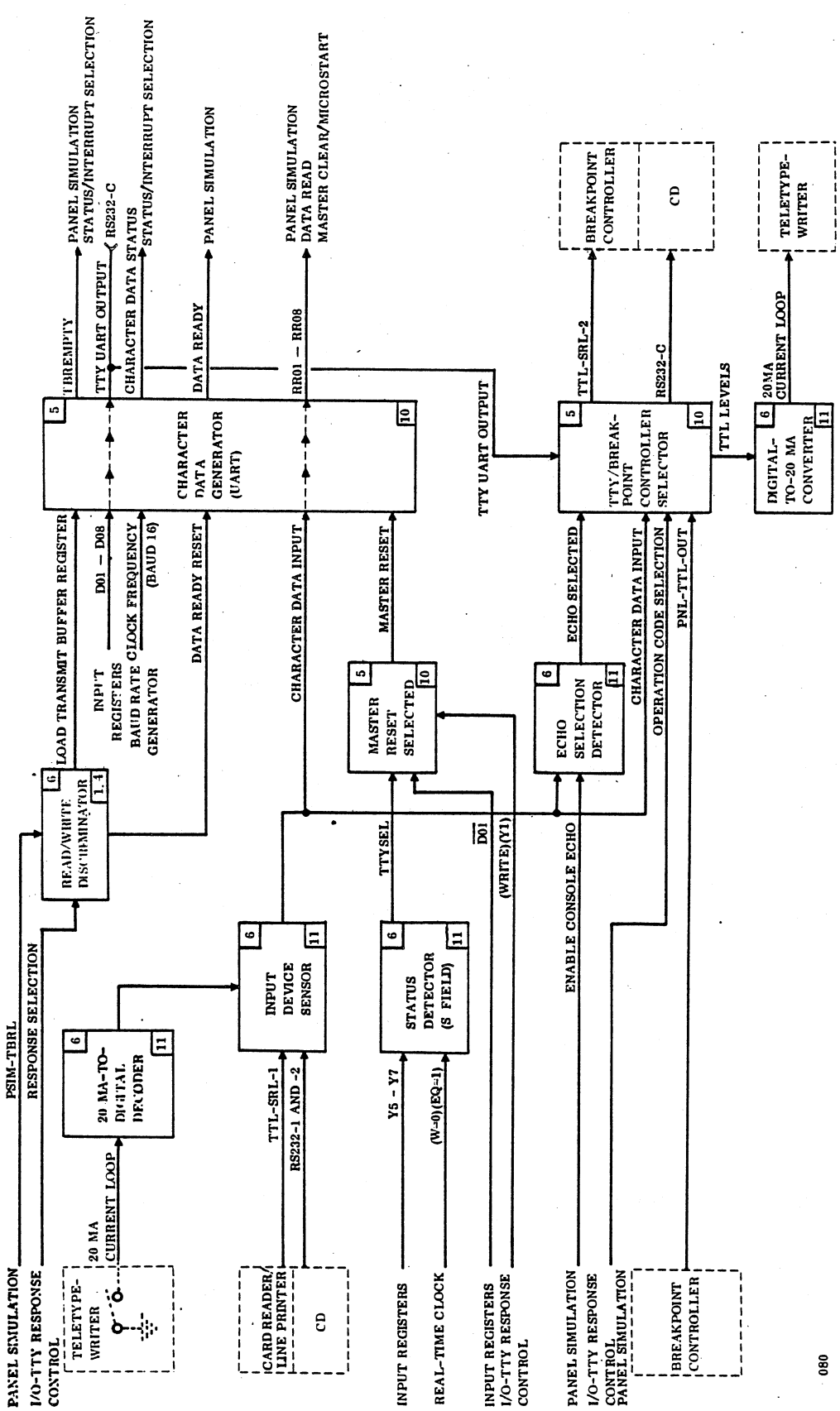


Figure 5-21. Teletypewriter/Console Display Control Function

connected). Refer to table 5-20. These signals are applied to selector inputs S0, S1, and S2 to select the output termination to either TTY, RS232-C, or TTL transmission line.

Transmission line compatibility is provided for the TTY, RS232-C, and TTL communication capabilities. The teletypewriter input and output signals are accomplished via a 20 mA current loop that must be converted to TTL levels for processing within the I/O-TTY controller. Input signals are converted from 20 mA current fluctuations by associated transistor circuits. The RS232-C input signals are converted from the RS232 signal levels (+3 to +12 volts to -3 to -12 volts) to TTL levels (0 to +5 volts) by the RS232 input converter. The RS232 output signals are converted from TTL level to the RS232-C level by a TTL-to-RS232 converter. Since the I/O-TTY controller is compatible with TTL inputs and outputs, the TTL serial and breakpoint controller lines do not require additional circuit elements.

Character data words to be transmitted during a CPU write command are loaded in parallel format into the transmit buffer register from the D register via lines D1 through D8. This load transmit buffer register (TBRL) command is derived from response selection (write mode, E-REPLY, write, reply, and TTY DATA signals) and panel simulation (PSIM-TBRL) signals by the read/write discriminator. When either the response selection signal or panel simulation signal is true, the D-register data is loaded into the transmit buffer register. When the transmit buffer register is filled, the data is immediately transferred to the transmit register. As the word is transmitted, start, parity, and stop bits are added to the data word under control of the UART's word length control logic. This data word is transmitted in serial format to the CPU and respective control device at the rate of one bit for every 16 clock pulses. When the transmit buffer register is empty, a TB register empty signal (TBREMPTY) is sent to the CPU via the status/interrupt selection function, indicating that a new work may be injected. When the transmit and transmit buffer registers are full, a busy signal is sent to the CPU via the status/interrupt selection function.

20 mA-to-Digital Converter

The 20 mA-to-digital converter (Q2) [Q1] is normally conducting due to the positive potential applied to the base. The positive potential is developed by the current flow through (R13, CR1, R14, and R15) [R3, CR1, R7, and R9] when the teletypewriter switch is open. When the teletypewriter switch is closed, ground is applied and current flows through (R13) [R3] to ground, causing the base to become less positive to backward-bias (Q2) [Q1] to cutoff. This cutoff condition of (Q2) [Q1] places approximately +5 volts at the collector to produce the high digital level. When the base voltage rises to forward-bias (Q2) [Q1] on the controller, voltage falls toward ground (approximately 0.6 volt) to produce the low digital level.

Digital-to-20 mA Converter

The digital-to-20-mA converter (Q1) [Q2] is normally backward-biased by the +5 volts applied to the base that holds (Q1) [Q2] cutoff. This inhibits the current path to the teletypewriter. When the digital output at $\overline{Q}$ of multiplexer (H6) [H4] goes low, current flow through (R20 and R21) [R8 and R10] forward-biases (Q1) [Q2] into conduction. This causes current flow from the teletypewriter through Q1 to +5 volts.

UART Description

The general-purpose, programmable, universal asynchronous receiver/transmitter (UART) has three-state outputs: 0, 1, and hi-Z. It is TTL-compatible. It is capable of simultaneously converting asynchronous serial binary characters to a parallel format (receiver) and parallel characters to serial, asynchronous output (transmitter) with start, parity, and stop bits added or verified. The programmable features are word length, information rate, parity, parity inhibit, and stop bit generation (refer to figure 5-22 and table 5-21).

TABLE 5-20. I/O-TTY, BREAKPOINT CONTROLLER, ECHO SELECTION

Selection	PNLPRES (S2)	RESERVED (S1)	KMODE•RMODE (S0)	Selected Output
I/O-TTY	0	0	0	TTY UART-OUT echo†
	0	0	1	
Panel simulation	0	1	0	UART-IN PSIM-echo
	0	1	1	(simulated console echo)
TTY-breakpoint controller shared display	1	0	0	TTY UART-OUT echo
	1	0	1	
	1	1	0	Panel Interface Out
	1	1	0	(PNL-TTY-OUT)

†Echo - The ability to have the selected input character immediately displayed on the teletypewriter or CD.

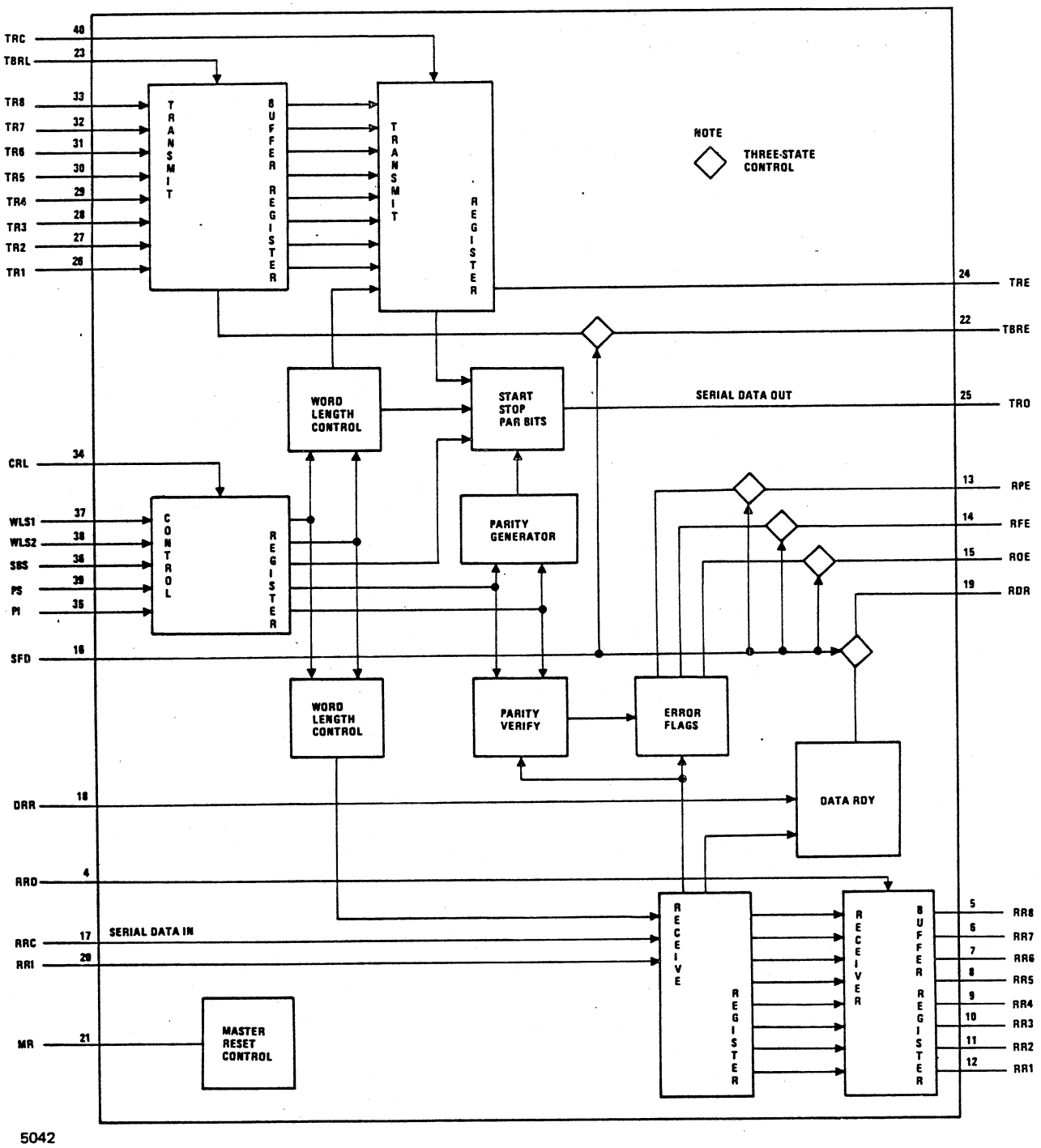


Figure 5-22. UART Functional Block Diagram

TABLE 5-21. UART PIN DESIGNATIONS AND DESCRIPTIONS

Designation	Pin Number	Description	Function
RRB (RBRD)	4	Receiver register	A high-level input voltage, V_{IH}, applied to this line disconnects the receiver holding register outputs from the RR8-RR1 data outputs (pins 5 through 12).
RR8-RR1	5 through 12	Receiver buffer (holding) register data	The parallel contents of the receiver register appear on these lines if a low-level input voltage, V_{IL}, is applied to RRD. For character formats of fewer than eight bits, received characters are right-justified (RR1 = LSB) and the truncated bits are forced to a low output voltage V_{OL}.
RPE	13	Parity error	A high-level output voltage, V_{OH}, on this line indi- cates that the received parity does not compare to that programmed by the even parity enable (PS) control line (pin 39). This output is updated each time a character is transferred to the receiver buffer register. RPE lines from a number of arrays can be bussed together since an output disconnect capability is provided by the status flag disconnect (SFD) line (pin 16).
RFE	14	Framing error	A high-level output voltage, V_{OH}, on this line indicates that the received character has no valid stop bit; that is, the bit following the parity bit (if programmed) is not a high-level voltage. This output is updated each time a character is transferred to the receiver holding register. RFE lines from a number of arrays can be bussed together since an output disconnect capability is provided by the status flag disconnect (SFD) line (pin 16).
ROE	15	Overrun error	A high-level output voltage, V_{OH}, on this line indicates that the data received flag RDR (pin 19) was not reset before the next character was transferred to the receiver holding register. ROE lines from a number of arrays can be bussed together since an output disconnect capability is provided by the status flag disconnect (SFD) line (pin 16).
SFD	16	Status flag disconnect	A high-level input voltage, V_{IH}, applied to this pin disconnects RPE, RFE, ROE, RDR, and TBRE, allowing them to be bus-connected.
RRC	17	Receiver register	The receiver clock frequency is 16 times the desired receiver shift rate.
DRR	18	Data received reset	A low-level input voltage, V_{IL}, applied to this line resets the DRR line.
RDR	19	Data ready	A high-level output voltage, V_{OH}, indicates that an entire character has been received and transferred to the receiver holding register.
RRI	20	Receiver input	Serial input data received on this line enters the receiver register at a point determined by the character length, parity, and the number of stop bits. A high-level input voltage, V_{IH}, must be present when data is not being received.
MR	21	Master reset	This line is strobed to a high-level input voltage, V_{IH}, to clear the logic. It resets the transmitter and receiver registers, the receiver holding register, RFE, ROE, RPE, and DRR, and sets the serial output line to a high-level output voltage, V_{OH}.
TBRE	22	Transmitter buffer register empty	A high-level output voltage, V_{OH}, on this line indicates that the transmitter holding register has transferred its contents to the transmitter register and may be loaded with a new character.

TABLE 5-21. UART PIN DESIGNATIONS AND DESCRIPTIONS (Contd)

Designation	Pin Number	Description	Function
TBRL	23	Transmitter buffer register load	A low-level input voltage, V_{IL}, applied to this line enters a character into the transmitter holding register. A transition from a low-level input voltage, V_{IL}, to a high-level input voltage, V_{IH}, transfers the character into the transmitter register if the register is not in the process of transmitting a character. If a character is being transmitted, the transfer is delayed until the transmission is completed. Upon completion, the new character is automatically transferred simultaneously with the initiation of the serial transmission of the new character.
TRE	24	Transmitter register empty	A high-level output voltage, V_{OH}, on this line indicates that the transmitter register has completed serial transmission of a full character including stop (bit(s). It remains at this level until the start of transmission of the next character.
TRO	25	Transmitter register output	The contents of the transmitter register (start bit, data bits, parity bit, and stop bit) are serially shifted out on this line. When no data is being transmitted, this line remains at a high-level output voltage, V_{OH}. Start of transmission is defined as the transition of the start bit from a high-level output voltage, V_{OH}, to a low-level output voltage, V_{OL}.
TR1-TR8	26 through 33	Transmitter register, data inputs	The character to be transmitted is loaded into the transmitter holding register on these lines with the TBRL strobe. If a character of less than 8 bits has been selected (by WLS1 and WLS2), the character is right-justified to the least significant bit, RR1, and the excess bits are disregarded. A high-level input voltage, V_{IH}, causes a transmission of a high-level output voltage, V_{OH}.
CRL	34	Control register load	A high-level input voltage, V_{IH}, on this line loads the control register with the control bits (WLS1, WLS2, RPE, PI, and SBS). This line may be strobed or hard-wired to a high-level input voltage, V_{IH}.
PI	35	Parity inhibit	A high-level input voltage, V_{IH}, on this line inhibits the parity generation and verification circuits and clamps the RPE output (pin 13) to V_{OL}. If parity is inhibited, the stop (bit(s) immediately follows the last data bit on transmission.
SBS	36	Stop bit(s) select	This line selects the number of stop bits to be transmitted after the parity bit. A high-level input voltage, V_{IH}, on this line selects two stop bits; a low-level input voltage, V_{IL} selects a single stop bit. Selection of two stop bits when programming a 5-bit word generates 1.5 bits from the UART.
WLS2-WLS1	37, 38	Word length select	These lines select the character length (exclusive of parity) as follows: WLS2 / WLS1 / Word Length 0 / 0 / 5 bits 0 / 1 / 6 bits 1 / 0 / 7 bits 1 / 1 / 8 bits
PS	39	Parity select	This line determines whether even or odd parity is to be generated by the transmitter and checked by the receiver. A high-level input voltage, V_{IH}, selects even parity; a low-level input voltage, V_{IL}, selects odd parity.
TRC	40	Transmitter register clock	The transmitter clock frequency is 16 times the desired transmitter shift rate.

STATUS/INTERRUPT SELECTION FUNCTION

The status/interrupt selection function (see figure 5-23) provides I/O-TTY controller status data to the CPU via the data read function and macro interrupts to the SMI. Character data status signals (parity error, lost data, and framing error) are received from the UART error detector logic. They are combined with a ready status signal from the CPU in the character data failure detector logic to produce the character data failure status that is coupled to the TTY status discriminator. If a failure status exists, the particular status is coupled via the data read function to the CPU. P.E., FERROR, and LOST DATA STATUS are inverted by (E5) [D4] and applied to OR gate (E4) [C6]. If any one of the conditions exists at the UART, (E4) [C6] produces a high indicating to the CPU that an alarm status exists. The lost data and ready status highs are applied directly to the data output multiplexer and the parity error status high is inverted by (F5) [D5] before it is coupled to data output multiplexer.

The character data status from the UART is also coupled to the data status detector and combined with the special character codes (BEL, ESC, @), read mode, and the busy status output of the operation detector logic. This busy status is produced from write mode and UART register conditions. If the character data status contains a failure, the data status detector produces an interrupt (INT14) to the SMI and inhibits the data status output. If no failures exist during read

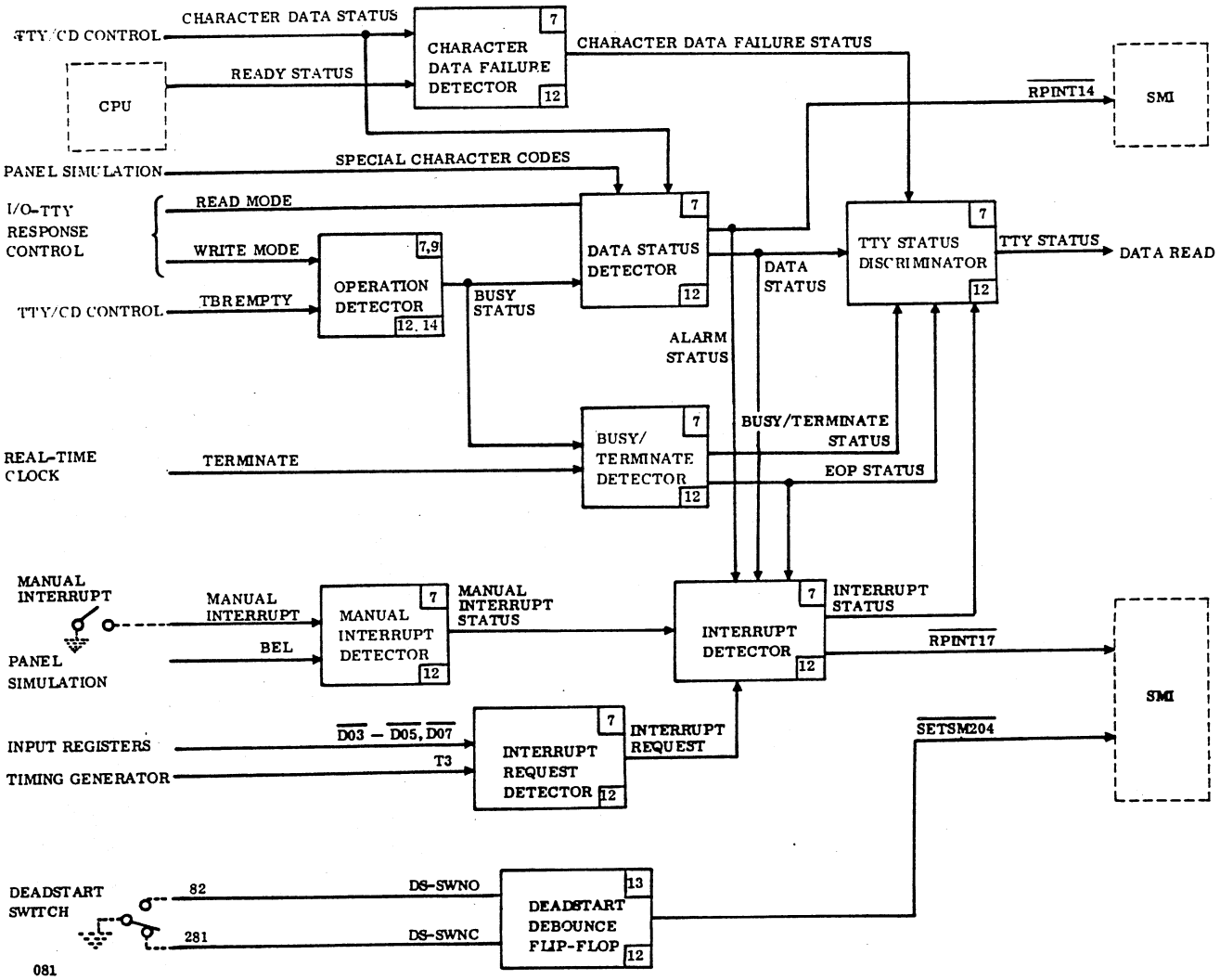


Figure 5-23. Status/Interrupt Selection Function

mode when a special character code is present, the special code data status is coupled to the CPU.

During write mode, if the transmit registers signal indicates that the registers are full, the operation detector produces a busy status. When registers are empty, BUSY STATUS is inhibited or, if a terminate signal is applied to the busy/terminate detector logic, an end-of-operation (EOP) status is sent to the CPU.

Data ready (DRDY) and the special character codes are ANDed by (K4) [B3]; if both are high, the high is ANDed with LOST DATA STATUS and READ MODE at AND gate (E4) [D1]. If all are high (indicating that data is ready, a special code has been selected, a lost data status does not exist, and read operation is selected), AND gate (E4) [D1] produces a low that is inverted by OR gate F1 to produce a DATA STATUS high to the CPU. If, during write mode (high), the transmit buffer register is empty (TBREMPTY high), AND gate (F1) [E1] provides a low to OR gate (F1) [E1], which also provides a write data status to the CPU. When either TBREMPTY or TREMPTY is low, OR gate D1 provides a BUSY STATUS and EOP STATUS to the CPU. When either transmit register is empty, the output of OR gate D1 is high and is inverted by (G4) [A3] to provide EOP STATUS.

When the manual interrupt switch at the console is activated or special character code BEL is injected, the manual interrupt detector produces an associated signal. This manual interrupt status generates an interrupt ($\overline{\text{RPINT17}}$) for the SMI and a manual interrupt status for the CPU. If any of the D03 through D05 and D07 data signals are present during time T3, the interrupt request detector sends an interrupt to the interrupt detector logic. If an alarm, data, or EOP signal is present during these conditions, interrupt $\overline{\text{RPINT17}}$ is produced and an interrupt status is sent to the CPU.

Interrupt status and interrupt RPINT17 are developed from the data, EOP, alarm status, released, or manual interrupt conditions. When data bit D03, D04, or D05 is low, one or more of the (D4) [A5] flip-flops is set. When the clock pulse is present, the associated flip-flop is latched. If the data, EOP, or alarm status signal is high, an interrupt status and $\overline{\text{RPINT17}}$ is produced.

Manual interrupt is accomplished by activating the manual interrupt switch at the console or selecting the BEL code at the input device keyboard. The switch activation sets and resets flip-flop (L1) [C2], which sends a strobe pulse through OR gate (L1) [C2] to flip-flop (K1) [C3]. Since the S and D inputs of (K1) [C3] are tied high, the strobe causes a manual interrupt to be produced by INT17. If the BEL key initiates the manual interrupt cycle, AND gate (M3) [F2] initiates the manual interrupt in the same manner as switch flip-flop (L1) [C2].

A deadstart interrupt (SETSM204) can also be produced by the I/O-TTY controller. If the deadstart switch is activated, the debounce flip-flop is activated for one cycle. This produces a single positive pulse output to apply a clear SET SM204 high level to the SMI without sensitivity to the transient pulses generated by the switch.

When the deadstart switch is activated, flip-flop (L11) [M5] is set and reset. The flip-flop output is inverted by (K11) [E7] to produce a SET SM204/ low.

DATA READ FUNCTION

The data read function (see figure 5-24) provides for the transfer of I/O controller data (character data, TTY/CD status, RTC status, I/O control responses, and I/O data RD01 through RD16) to the CPU three-state bus. The data to be gated onto the three-state bus is determined by a combination of the contents in the micro instruction B′ field and the Y register at the time the micro instruction is executed. The Y register selection is decoded in accordance with the CDC 1700 WES/D convention when the micro instruction B1 field equals 010 (2_8) to designate INRD (input data/status from I/O channel). Which data is selected for transfer to the CPU three-state bus is indicated in table 5-22. The I/O control response selection is determined when micro instruction B′ equals 011 (3_8) to designate INRS (input to S2 I/O response signal) (refer to table 5-23).

TABLE 5-22. Y REGISTER SELECTION OF READ DATA

W	E	S	D	Data Gated to CPU Three-State Bus
0	1	1	0	TTY/CD data
0	1	1	1	TTY/CD status
0	1	7	-	Real-time clock status
-	-	-	-	Data from peripheral controllers (RD01 through RD 16)

TABLE 5-23. MICRO INSTRUCTION B′ FIELD SELECTION

MO5 Peripheral	CDC 1700 Peripheral	Three-State Bus Bit Position
Undefined	Undefined	00 and 01
Position 00	Receiver terminate (R-TERM)	02
Position 01	Reply	03
Position 02	Reject	04
Direction	Character input	05
Undefined	Undefined	06 through 15

This selection is performed by two 2-to-1 multiplexers, (E7 and H7) [E9 and E8], and four 4-to-1 multiplexers (A7, B7, J7, and K7) [F6, G5, C7, and A7]. The 2-to-1 multiplexers effect selection by the signals applied to the S and E inputs, and the 4-to-1 multiplexers effect the selection by decoding the S0 and S1 input conditions when the enable signal is low (refer to table 5-24). All data is received from the peripheral controllers via open collector lines RD01 through RD16 applied to the associated (C7, E8, and G7) [A8, B8, C8, D8, and D9] inverters.

Selector multiplexer (H7) [E8] provides the selection of RTC status and peripheral data lines RD13 through RD16. When RTCSEL is high (refer to table 5-22) and the $\overline{\text{ENABLEIO}}$ signal is low, the RTC limit and LOST COUNT STATUS are transferred to three-state bus lines BUS00 and BUS01. When both RTCSEL and $\overline{\text{ENABLEIO}}$ are low, RD13 through RD16 are transferred to bus lines BUS00 through BUS03.

TABLE 5-24. BUS08 THROUGH BUS15 SELECTOR SELECTION

S1	S0	Data Transferred to CPU Three-State Bus
0	0	I/O data (RD01 through RD08)
0	1	TTY/CD status
1	0	Character data (RR01 through RR08)
1	1	I/O-TTY control responses

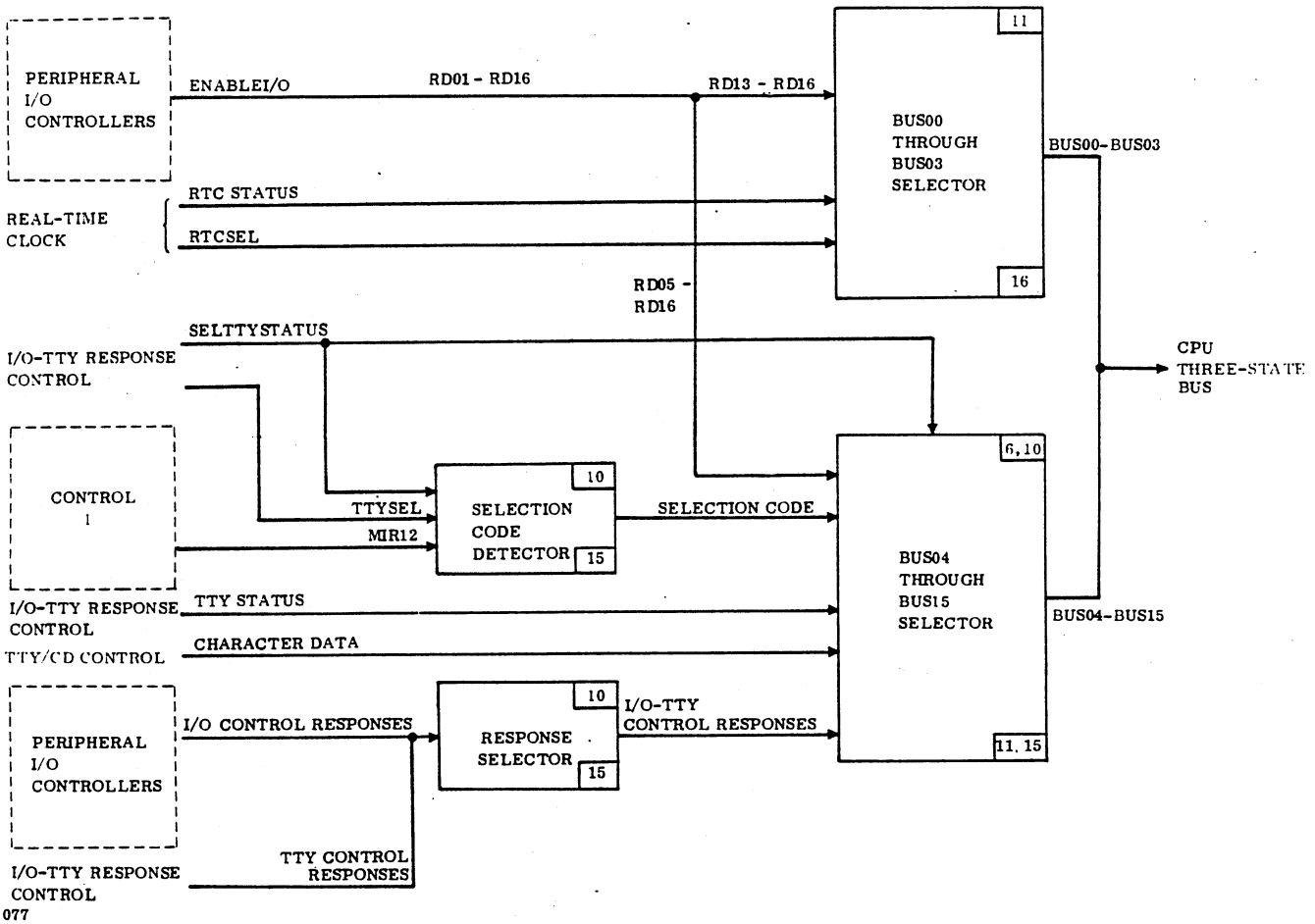


Figure 5-24. Data Read Function

Selector multiplexer (E7) [E9] provides selection of TTY/CD status (MANINTSTATUS, RESERVED, RELEASED, and READMODE) and peripheral data lines RD09 through RD12. When SELTTYSTATUS is high and ENABLEIO is low, the TTY/CD status is transferred to three-state bus lines BUS04 through BUS07. When SELTTYSTATUS and ENABLEIO are low, RD09 through RD12 are transferred to bus lines BUS04 through BUS07.

Selector multiplexers (A7, B7, J7, and K7) [A7, C7, F6, and G5] provide selection of peripheral data lines RD01 through RD08, TTY/CD status and data, and I/O controller control responses. This selection is determined by the signals applied to the multiplexer S0 and S1 inputs (refer to table 5-24). When SELTTYSTATUS is low, (TTYSEL)($\overline{Y1}$) is high, and MIR12 is low, inverter (F7) [L6] provides lows to OR gates (F7)[L6] that set S0 and S1 low. When ENABLEIO becomes low, the data on lines RD01 through RD08 is transferred to the bus lines. When SELTTY STATUS is high, (TTYSEL)($\overline{Y1}$) is high, and MIR12 is low, (F7 pin 11) [L6 pin 6] is low. This low is applied to OR gate (F7 pin 1) [L6 pin 9], which applies a high to S0 input. When ENABLEIO is low, the TTY controller status data at multiplexer inputs DA1 and EB1 is applied to the bus lines. When SELTTYSTATUS, (TTYSEL)($\overline{Y1}$), and MIR12 are all low, pin (6) [11] of OR gate (F7) [L6] is high, setting S1 of (B7) [C7] high. When ENABLEIO goes low, the TTY/CD character code data RR1 through RR8 is transferred to the bus lines. When MIR12 is high, SELTTYSTATUS is low, and (TTYSEL) ($\overline{Y1}$) is high, the output at pin (8) [3] of inverter (F7) [L6] is low. This applies low to both (F7) [L6] OR gates, the outputs of which go high to apply highs to both the S0 and S1 inputs. When ENABLEIO is low, the I/O-TTY controller or other I/O controller control response is transferred to the bus lines.

The I/O-TTY controller and other I/O controller control responses (CHARINPUT, RTERM, REPLY, and REJECT) are all applied to the D-latch flip-flop, (J8) [G6]. If any one of these signals is high at input D, the flip-flop is latched. When the RESYNC-I/O signal applied to CP rises, the latching signal is transferred through the flip-flop to the output multiplexer. If S0 and S1 of the output multiplexer are high, the control response is transferred to the bus lines.

REAL-TIME CLOCK FUNCTION

The real-time clock pulses (figure 5-25) are processed in the CPU auto-data transfer routine to provide the CPU with a time-lapse capability that can be directly related to process program time segments or time of day (see table 5-25). This elasped-time base is produced by a pulse generator that emits a pulse every 3.3 milliseconds. These pulses are coupled from the real-time clock via the data interrupt line (RDINT08) to increment the clock counter field every 3.3 milliseconds. When the clock counter field equals the limit interrupt (LIMITINT-STATUS), the micro instruction send terminate signal (STERM/) is generated by the SMI to enable program interrupt RPINT24/, which terminates the real-time clock sequence. If a lost count interrupt is detected, an $\overline{STERM}$ signal is

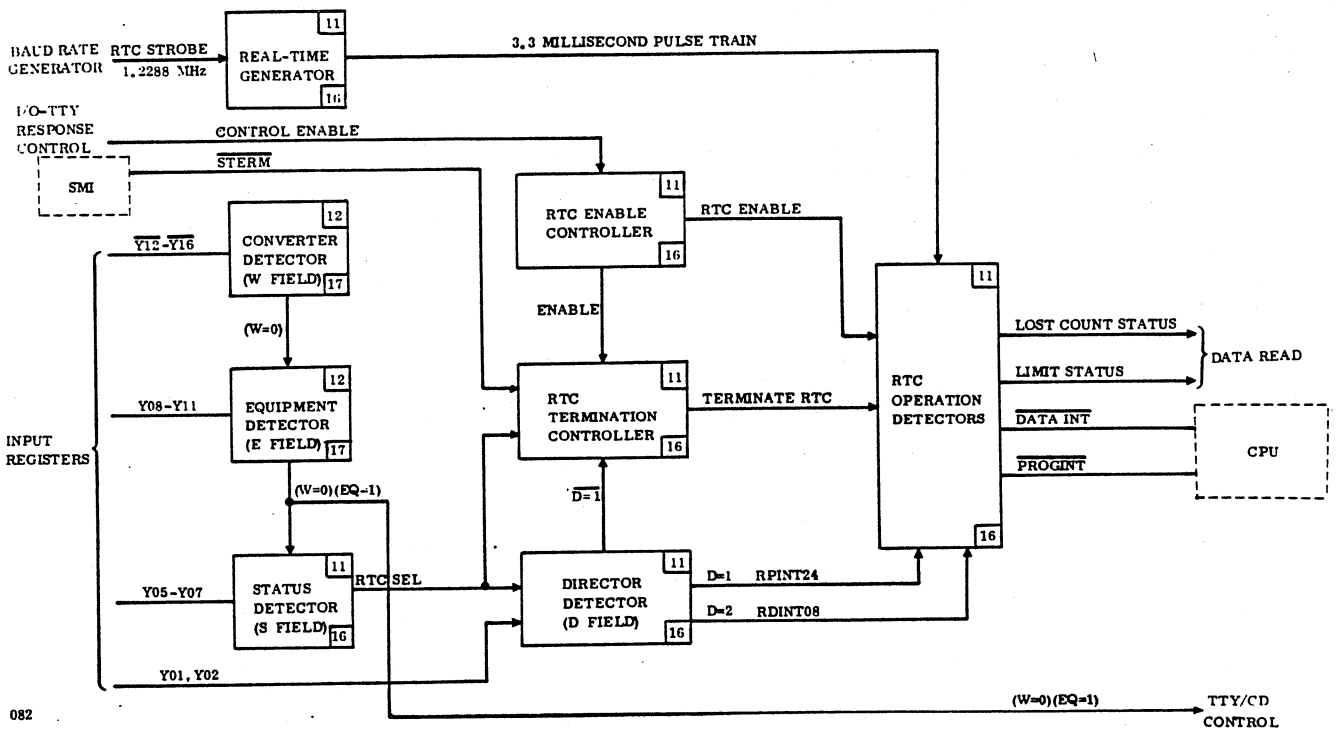


Figure 5-25. Real-Time Clock Function

also generated to enable an RPINT24 signal to terminate the real-time clock sequence. When the sequence has been terminated by either a limit or a lost count interrupt, an associated status signal (LIMITINSTATUS or LOSTCOUNTSTATUS) is produced to advise the CPU which condition terminated the real-time clock sequence.

TABLE 5-25. ADT TABLE FOR RTC SEQUENCE

Bit	ADT	W				E				S			D			
Word	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
2	Clock counter															
3	Clock limit															
4	Not used															

Word 1 contains an address designated as follows:

Bits 4, 5, 6, 7, and 15 must be 1s, and all other bits must be 0s. Where bit 15 designates ADT enabled and W=0, E=1, and S=7, D=0 designates that RTC is selected.

Word 2 contains the clock counter. When enabled, it is initially set to zero and incremented every 3.3 milliseconds.

Word 3 contains the clock limit. When the clock counter contents equals the clock limit, and interrupt occurs.

The converter, equipment, and status detectors decipher the Y register address bits to determine whether or not the real-time clock function code (WES=0 1 7) exists. The converter (W) field condition bits, Y13 through Y16, are detected by AND gate (J10) [F2]. If all bits are ones, a high output is applied to AND/OR gate (G6) [H10]. This high is ANDed with Y12; if Y12 is high or AUTO-DATA at gate (G6) [H10] is high, the $\overline{\text{WEO}}$ signal is generated. $\overline{\text{WEO}}$ is inverted by (G7) [H8] and is coupled through pins (10) [8] and (11) [9] to all peripheral I/O controllers and AND gate (K4) [J8]. The equipment (E) field, bits Y08 through Y11, are then deciphered. The WEO condition is ANDed with the least significant E-field bit, Y08, at AND gate K4. If Y8 and WEO are high and Y9, Y10, and Y11 are high at AND gate (M5) [L8], the resultant (W=0)(EQ=1) condition is produced. The status (S) field, bits Y05 through Y07, are then deciphered. If Y5, Y6, and Y7 are high and (W=0)(EQ=1) is high at AND gate (D7) [C4], the RTCSEL, established by W=0, E=1, and S=7, is generated.

(RTCSEL) [SELRTCSTATUS] combines with Y1 and Y2 at director detector flip-flops (K5 and K6) [J9 and K9]. If Y2 is high (D=2) RDINT08/ is enabled by clearing flip-flop (L6) [K8]. If Y1 is high (D=1) RPINT24/ is enabled by clearing (K5 and L6) [J9 and K8]. (RTCSEL) [SELRTCSTATUS] is also applied to the real-time clock AND gates (L5 and M5) [H7 and L8], reply/reject AND gate (F1) [J3], and AND/OR gate (G1) [H2], and set input multiplexer (H7) [E8]. Gate (M5) [L8] ANDs the (RTCSEL) [SELRTCSTATUS], WRITE, and PROT signals; if all conditions are high (M5) [L8] clears the lost count flip-flop via OR gate (J4) [J10] and clocks the Y2 condition at flip-flop (K6) [K9]. If Y2 is high, the Q high condition of (K6) [K9] enables data interrupt (RDINT08/) flip-flop (L6) [K8].

When the data interrupt flip-flop is enabled, the clock generator (L7 and L8) [M6 and M7] pulses activate the data interrupt flip-flop. This enables lost count flip-flop (L6) [K8] and inverter (M6) [M8]. (M6) [M8] inverts the data interrupt pulse (RDINT08). The data interrupt flip-flop provides the clock generator pulse every 3.3 milliseconds until the real-time clock limit is reached. Lost count flip-flop (L6) [K8] receives the clock pulses and the interrupt flip-flop output pulses to ensure that a data interrupt output is produced for each clock pulse. If (L6) [K8] is not cleared at each pulse, it activates to generate a low at the $\overline{Q}$ output, which produces a program interrupt (RPINT24/) low signal to terminate the real-time clock.

As long as the limit count or lost count conditions have not been set, a clock generator pulse is emitted every 3.3 milliseconds. The SMI and ADT sequences the real-time clock operation until the real-time clock limit is met. The SMI provides a send strobe (READ-SSTB/) that is inverted by (M9) [H8] to produce the read condition. With the READ, AUTO-DATA, and RTCSEL conditions all high, AND gate (L5) [H7] provides a low output or OR gate K4 and inverter (M6) [L5]. The output of K4 goes low to clear the data interrupt flip-flop and to enable continuation of the real-time clock sequence. When the limit counter and the clock count of the ADT agree, a terminate signal (STERM/) is initiated by the SMI. This signal is inverted by (M9) [D8] to apply a TERMINATE high to the input of AND gate (L5) [L9]. When the SMI initiates the next send strobe signal ($\overline{\text{SSTB}}$), which is inverted by (M9) [H8] to produce a high (READ), AND gate (L5) [H2] is activated.

The high inputs (TERMINATE, Y1, and AUTO-DATA READ) produce a low output at AND gate (L5) [L9]. This low sets the limit status flip-flop, which initiates a RPINT24/ low signal. The CPU sends a WRITE signal to terminate the real-time clock sequence, and a limit status signal to the CPU. If, at time during the real-time clock sequecnce, a master clear signal ($\overline{\text{MC-I}}$) is applied to latch flip-flops (K5 and K6) [J9 and K9], the output produces a RPINT24/ condition that also terminates the real-time clock sequence.

The clock generator receives real-time clock strobe pulses from the baud rate generator. These pulses are the result of the crystal frequency 4.9152 MHz being divided by four in the baud rate generator to produce a pulse frequency of 1.2288 MHz. The frequency is divided by 64 by multiplier (L8) [M6] and the resultant 0.0192 MHz frequency is again divided by 64 by multiplier (L7) [M7]. The output of (L7) [M7] is 300 Hz (3.3-millisecond pulse train appearing at the CP inputs of the lost count and interrupt flip-flops). This pulse train is always present and is only permitted out when the real-time clock sequence has been selected. (Refer to figure 5-26 for the real-time clock timing sequence.)

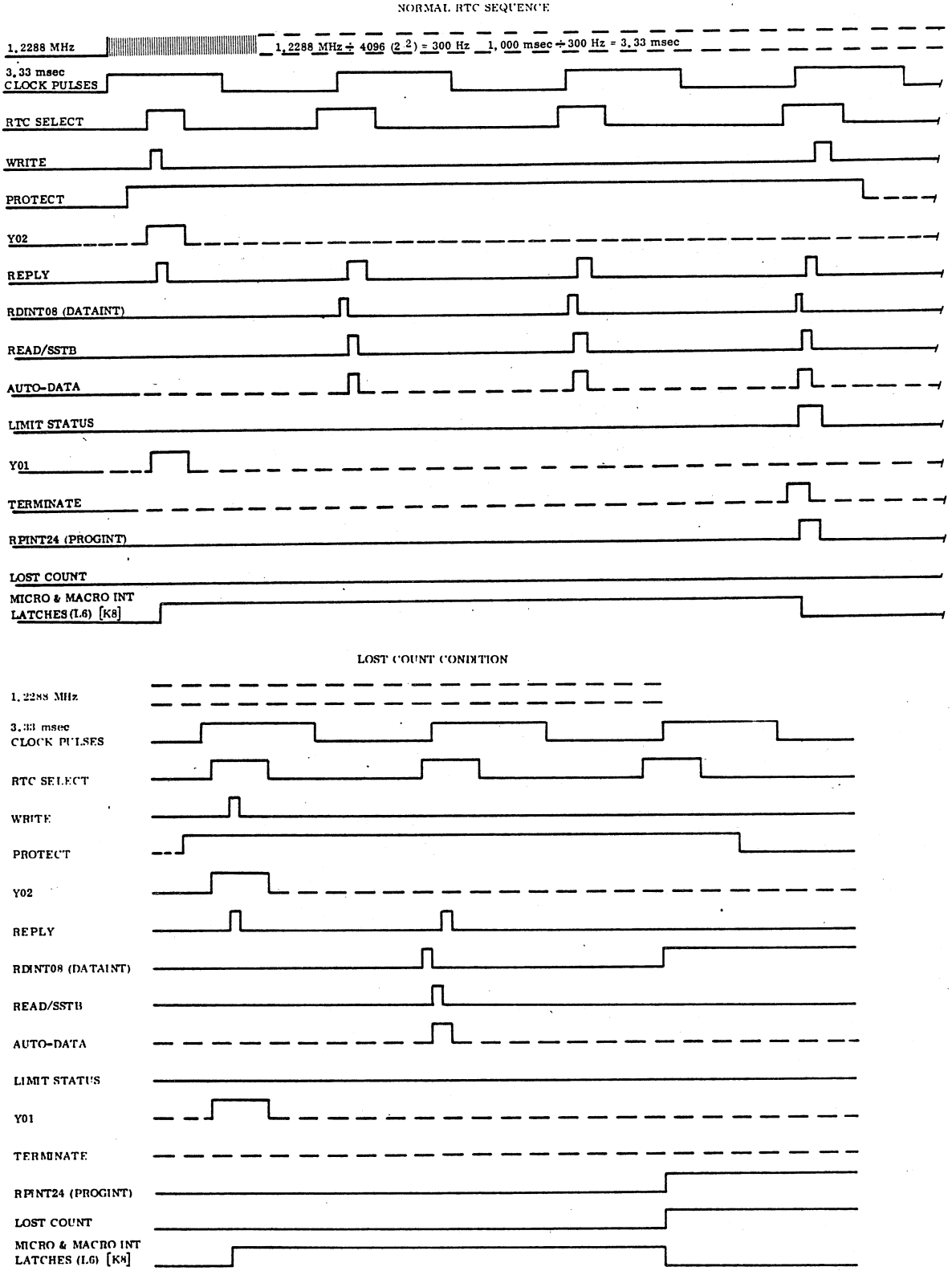


Figure 5-26. Real-Time Clock Timing Sequence

MASTER CLEAR/MICROSTART FUNCTION

The master clear/microstart function (figure 5-27) clears the I/O controllers and initiates the micro programs enable. The master clear operation can be initiated from several sources: the master clear switch, a TTY/CD question mark code (?) or the application of an MC signal from control 1. The microstart is a function of the panel interface not being supplied as part of the computer and the MCDELAYED/ signal not being active. When the breakpoint controller is supplied, the breakpoint controller logic provides the microstart signal to the control 1 module. A master clear signal (MC-S/) can be generated via the I/O-TTY controller only when the breakpoint controller is not supplied. In this case, an external momentary switch must be connected across the master clear input lines to the debounce flip-flop.

A micro program is normally activated by the breakpoint controller; when this controller is not supplied, micro programs are enabled when the I/O-TTY controller supplies a microstart low signal (MICROSTRT/) to the control 1 module. The MICROSTRT/ signal is produced by AND gate K10 when the breakpoint controller present line ($\overline{\text{PNLPRES}}$) and the master clear delayed line (MCDELAYED/) are high. This causes a constant MICROSTRT signal (except during a master clear) when the breakpoint controller is not present.

A master clear start low signal ($\overline{\text{MC-S}}$) initiates a master clear cycle. $\overline{\text{MC-S}}$ is coupled to the control 1 module that, under control of the CPU, produces a master clear low (MCDELAYED/). This signal is inverted by (K2) [L5] to produce the MC signal used to clear the I/O control function. The MC signal is also inverted by additional (K2) [L5] inverters to produce the $\overline{\text{MC-1}}$ and $\overline{\text{MR}}$ signals. The $\overline{\text{MC-1}}$ signal clears the timing generator, panel simulation, and real-time clock functions. $\overline{\text{MR}}$ is coupled across backplane bus terminal 277 to clear all the I/O controllers.

The master clear switch circuit contains a debounce flip-flop to eliminate the switch transients. When the MC switch is activated, backplane terminal 282 goes low and 83 goes high. This causes a high output at the bebounce flip-flop AND gate (L11) [L10] pin 8, which applies a high to OR gate (L11 pin 5) [L10 pin 4]. Highs at pins 4 and 5 of OR gate (L11) [L10] send the output at (L11) [L10] pin 6 low. This high-to-low transition activates one-shot (K9) [L4], which produces a high output at pin 6 for 5 microseconds and inhibits AND gate (L11) [L10] to render the AND insensitive to transient pulses. The one-shot output high and the breakpoint panel present high at AND gate K10 produces an $\overline{\text{MC-S}}$ signal at backplane terminal 74, which is coupled to the control 1 terminal 93.

If $\overline{\text{MC-S}}$ is initiated by a question mark (?) code (UART outputs RR01 through RR08 = $3F_{16}$), gates (J9 and H9) [E3 and C4] detect this character code. AND gate (J9) [E3] ANDs the RR0 • RR1 • RR2 ($\overline{\text{XXXX}}$ X111) low signal (obtained from the panel simulation function) with RR7 low to produce a high

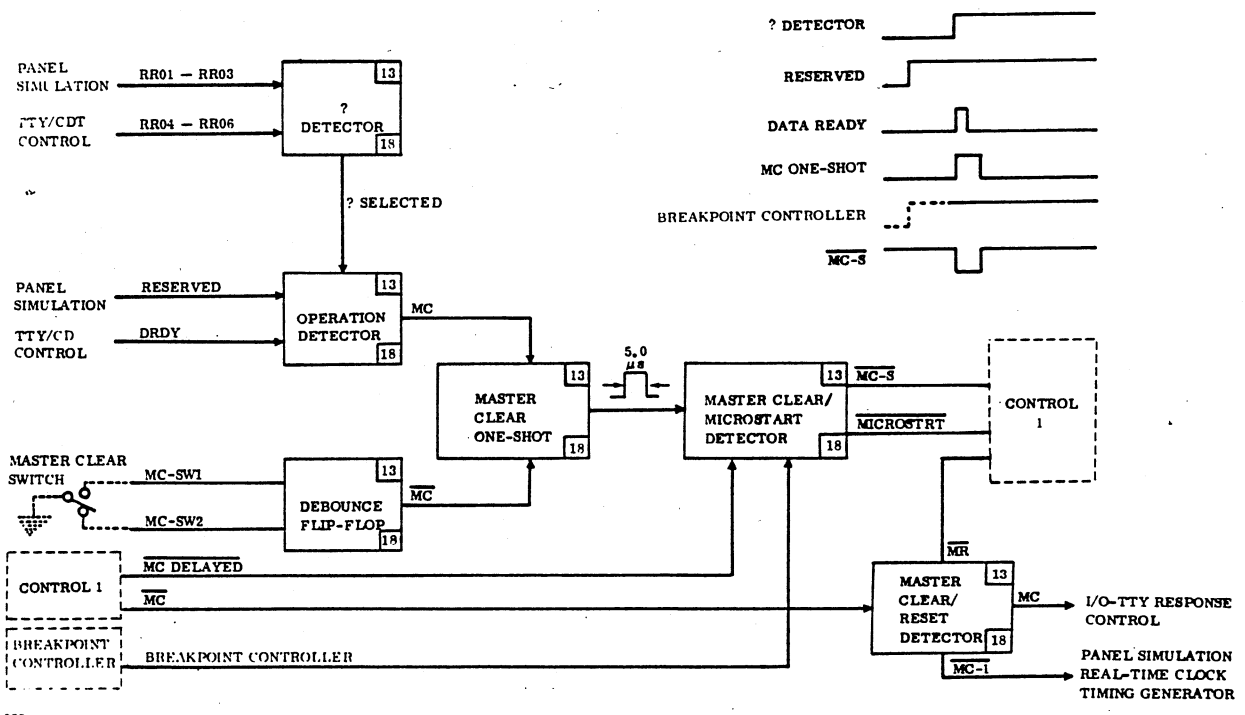


Figure 5-27. Master Clear/Microstart Function

at pin (9) [13] of AND gate (H9) [C4]. This high from (J9) [E3] is ANDed with the high state of RR4, RR5, from RR6 to determine the $3F_{16}$ condition (? code). The ? selected (RR4-RR7) high state is produced by AND gate (H4) [D7]. This question mark selected • RESERVED high condition is then ANDed with data ready (DRDY/) at AND gate (F8) [D2]. If both inputs (pins (12) [2] and 13) are true (high), the output of (F8) [D2] pin (11) [12] rises, triggering one-shot (K9) [L4], which applies a high to pin 2 of AND gate K10. This high at pin 2 and the high at pin 1 (determined by the absence of a breakpoint controller) effect an MC-S/ low condition at pin 3 of AND gate K10. MC-S/ low is coupled to the control 1 module to initiate a master clear cycle.

BAUD RATE GENERATOR FUNCTION

The baud rate generator (figure 5-28) generates the UART clock frequencies required to operate at the four baud rates of 110, 300, 1200, and 9600. These clock frequencies are 16 times the baud rate (UART transmit and receive register shift rate). The various outputs provided by the baud rate generator are a high-level voltage to the UART stop bit select (SBS) input, the clock frequency for the transmit and receive registers (TRC and RRC), the complement of the clock frequency (DSUARTCLK) to the card reader/line printer controller UART, strobe frequency of 1.2288 MHz for the real-time clock, and selectable deadstart baud rates (PNLBAUD0 and PNLBAUD1) for the breakpoint controller. Switch (S1) [SW1] provides for selection of the clock frequencies associated with the controller baud rate requirements as indicated in table 5-26.

TABLE 5-26. BAUD RATE CLOCK FREQUENCY SELECTION

Clock Frequency (Hertz)	Baud Rate	Deadstart Select Segment Number		Program Select Segment Number	
		ON	OFF	ON	OFF
1,760	110	1 and 2	-	3 and 4	-
4,800	300	1	2	3	4
19,200	1200	2	1	4	3
153,600	9600	-	1 and 2	-	3 and 4

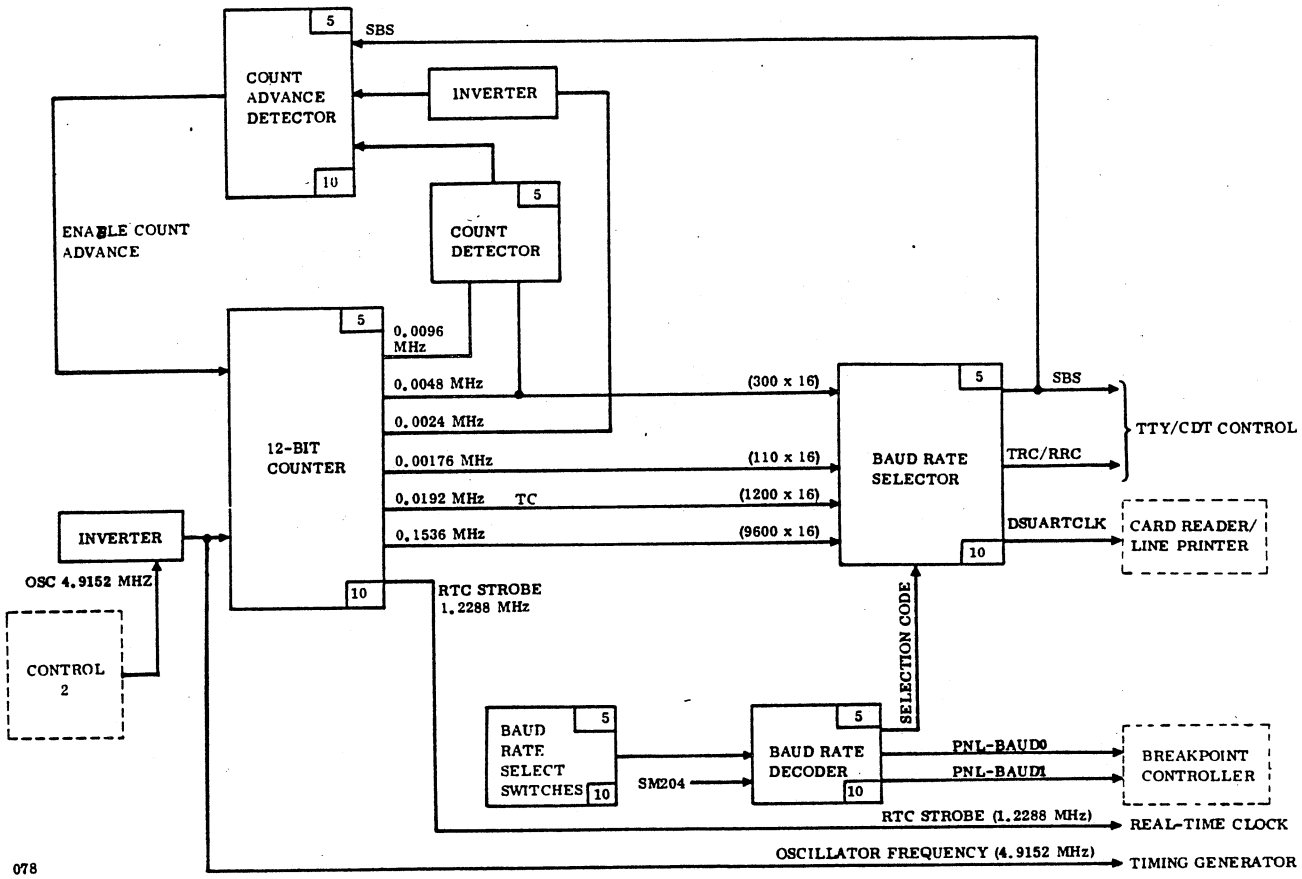


Figure 5-28. Baud Rate Generator Function

The baud rate generator receives a clock frequency of 4.9152 MHz from the crystal-controller oscillator located in the control 2 module of the CPU. This clock frequency is inverted by buffer (A4) [E1] and simultaneously applied to the clock pulse terminals of a 12-bit counter (A5, A6, and B5) [J7, K6, and K7] to produce the various baud rate clock frequencies required by the UART as well as the strobe frequency required by the real-time clock. Counter (A5) [K7] divides the 4.9152 MHz signal by 4 and 16. The divide-by-four output provides the real-time clock strobe frequency of 1.2288 MHz. The divide-by-16 output provides a 0.3072 MHz input to (A6) [K6] where the frequency is again divided by 32 and 256 to provide clock frequencies of 0.1536 MHz and 0.0192 MHz, respectively. The 0.1536 MHz signal is the clock frequency for 9600 baud rate (9600 x 16) and the 0.0192 MHz signal is the clock frequency for 1200 baud rate (1200 x 16). The 0.0192 MHz (TC) signal is coupled to counter (B5) [J7] to generate the clock frequencies for the 110 and 300 baud rates. Clock frequency 0.0048 MHz (300 x 16) is the direct result of the 2^{10} output, but additional manipulation of signals is required to produce the 110 x 16 clock frequency.

High outputs Q0 and Q1 of counter (B5) [J7] are ANDed by (C5) [J8], and the (B5) [J7] low output is inverted by (D6) [H8]. When these outputs are high and coincident with highs from TC of (A6) [K6] and QA of selector (B6) [J6] and AND gate (C6) [H7], a low output is produced by (C6) [H7]. It presets all the counters (A5, A6, and B5) [J7, K6, and K7], thereby advancing the output frequency at Q3 of (B5) [J7] by approximately 570 Hz to produce the clock frequency of 0.00176 MHz (110 x 16). These clock frequencies (9600 x 16), (1200 x 16), (300 x 16), and (110 x 16) are applied to the B inputs of selector (B6) [J6].

The applicable baud rate clock frequency is selected through proper settings of the baud rate select switch (S1) [SW1] segments 1, 2, 3, and 4 (table 5-27). These switches apply lows to the inputs of decoder multiplexer (M11) [M3]. When (M11) [M3] is set by SM204 being high, the deadstart baud rate selection signals PNLBAUD0 and PNLBAUD1 are sent to the breakpoint controller. The enable state select status is applied to S0 and S1 of selector (B6) [J6] to choose the clock frequency that appears at the QB and $\overline{QB}$ terminals of selector (B6) [J6]. When the 110 x 16 baud rate frequency is selected, the QA output is high, and a high applied to the UART SBS input enables the selection of two stop bits per character code. All other baud rate frequencies produce a low at QA to select one stop bit per character code.

TABLE 5-27. BAUD RATE CLOCK FREQUENCY AND STOP BIT SELECTION

Frequency (Hertz)	S1	S0	Baud Rate	Stop Bits
1,760	0	0	110	2
4,800	0	1	300	1
19,200	1	0	1200	1
153,600	1	1	9600	1

The QB output of (B6) [J6] is buffered through AND gate (C2) [J8] to the UART to provide the transmit clock frequency (TRC) and the receive clock frequency (RRC). The complement of the QB output appears at the $\overline{QB}$ output, which provides the deadstart UART clock signal DSUARTCLK to the panel interface.

TIMING GENERATOR FUNCTION

The timing generator (see figure 5-29) provides the basic timing required by the I/O-TTY controller. It provides the T1, T2, T3, and $\overline{T3}$ timing pulses (figure 5-30) for the reply, reject, read mode, printer connected, character input, clear interrupt, and TTY-ADT-INT signals. All timing signals are generated by the 4.9152 MHz crystal-controlled oscillator signal received from the control 2 module via the baud rate generator. When the CPU read or write signals request an output or input operation, a reply occurs within a minimum of 200 nanoseconds and a maximum of 10 microseconds if the peripheral is ready to respond.

If the peripheral is not ready to respond, a reject signal is returned within a minimum of 200 nanoseconds and a maximum of 10 microseconds. The timing generator initiates the sequences required to time these responses.

The converter, equipment, and status detectors decipher the Y-register address bits to determine that the TTY select code exists. The converter detector deciphers bits Y12 through Y16 to determine that the W=0 condition is present and couples the logic condition to the equipment detector. The W=0 condition is combined with bits Y08 through Y11 to develop the E field condition of E=1. This (W=0)(EQ=1) condition is applied to the status detector and combined with bits Y05 through Y07 to determine the (W=0)(E=1)(S=1) conditions.

When this is true, the TTYSEL high, $\overline{\text{REPLY+REJECT}}$ high, and $\overline{T3}$ high are ANDed with the READ+WRITE high output of inverter (E5)[E1] at response detector AND gate (C4) [H1] and provide a high to (C4) [H1] pin 5. If all the $\overline{Q}$ outputs (except $\overline{T3}$) of the four-bit shifter (B4) [J1] are high at (C4) [H1] pins 1, 2, and 4, the timing chain enabled signal is generated and applied to the D (pin 5) input of the shifter (B4) [J1]. This input creates a 202-nanosecond pulse at the Q (pin 2) output coincident with the rise of the oscillator (clock) frequency applied to CP of (B4) [J1]. This pulse triggers a chain reaction to generate the T1, T2, and T3 outputs (T1 triggers T2, which triggers T3). When $\overline{T3}$ is high, the reply/reject flip-flop is latched to produce the REPLY+REJECT signal. The $\overline{\text{REPLY+REJECT}}$ high signal disables the time generator and remains high until the READ+WRITE signal drops coincident with the negative swing of the clock pulse. Also note that the timing chain can be interrupted (aborted) anytime that a $\overline{MC-1}$ low signal is applied to the clear input of the four-bit shifter.

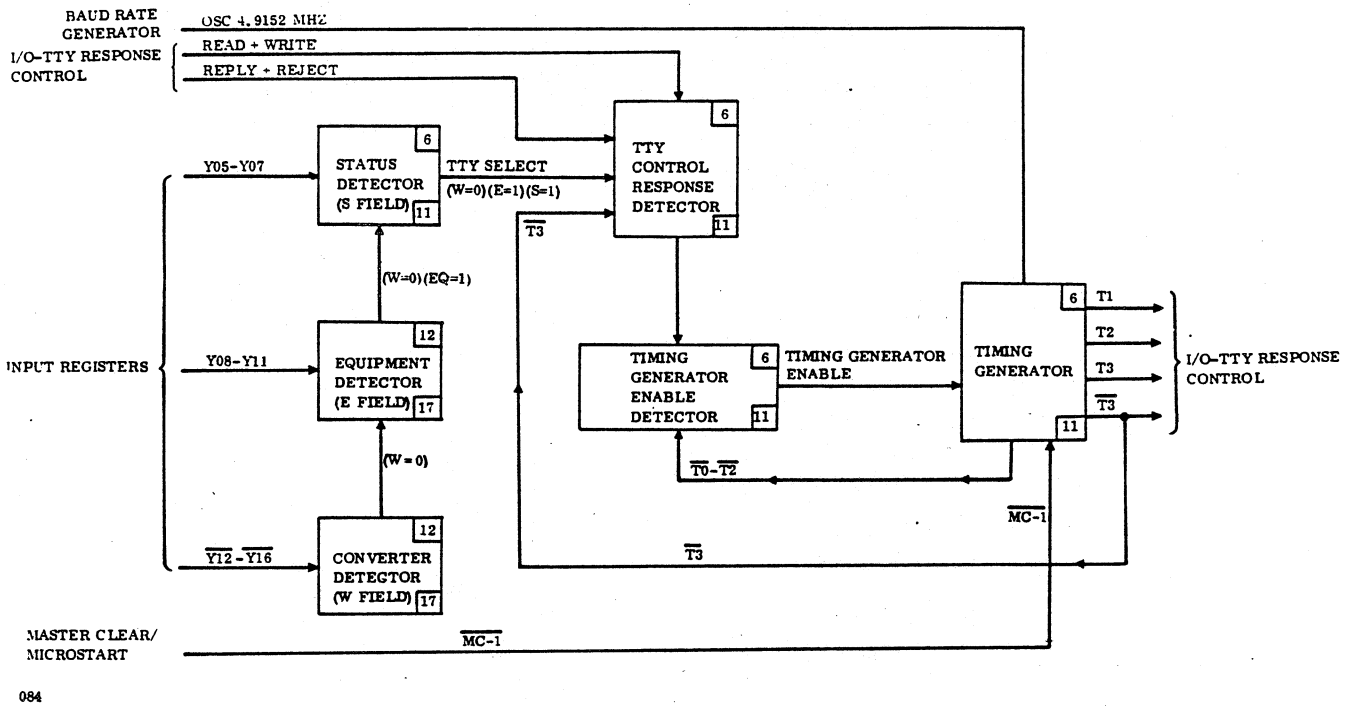


Figure 5-29. Timing Generator Function

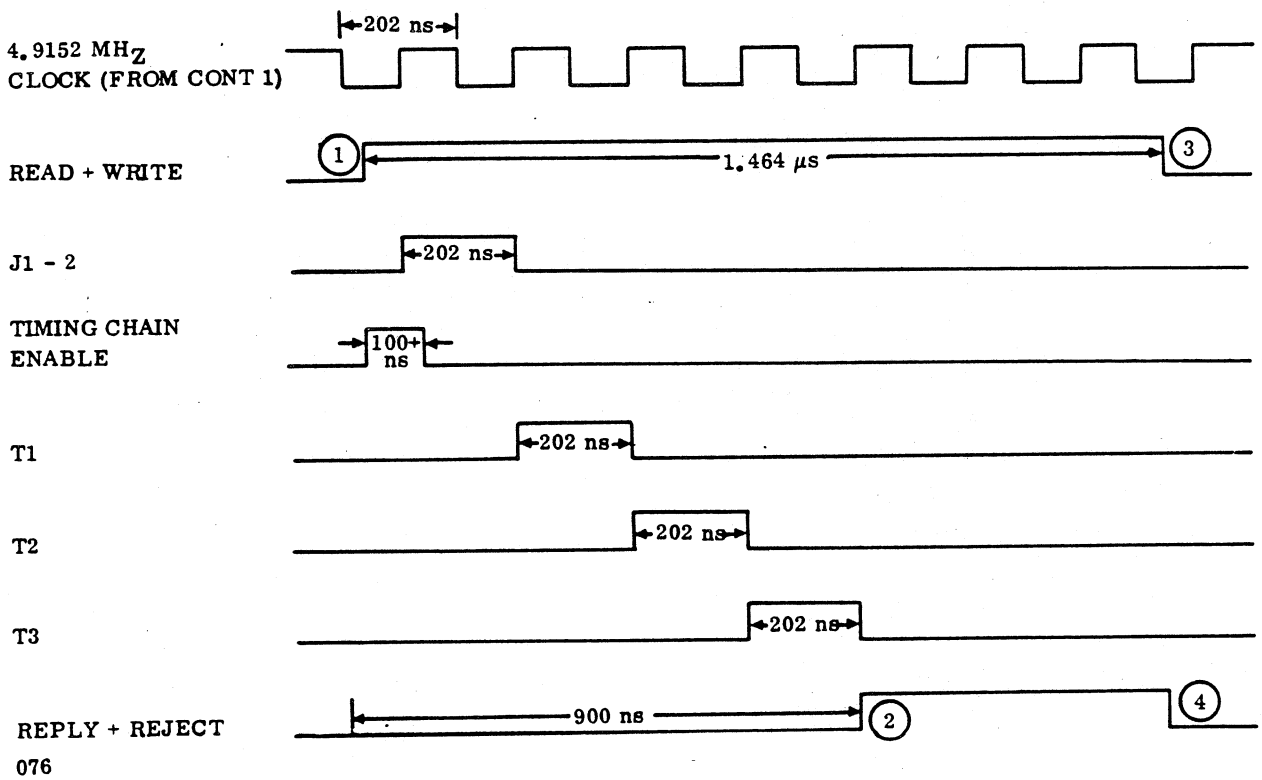


Figure 5-30. I/O-TTY Controller Timing Chart

M05 SET/SAMPLE FUNCTION

This function (see figure 5-31) provides eight port addressing (SPT0 through SPT7) lines and an eight-way parity selection scheme (SSEL0 through SSEL7). Each port is numbered 0 through 7 but designated 1 through 8 (that is, 0=1, 1=2, 2=3, and so forth). Port 8 has the lowest priority. Each port can communicate with a peripheral directly or it may be multiplexed to communicate with up to eight peripherals. This provides for control of up to 64 peripherals (table 5-28). The multiplexed priority (scanning) scheme must be provided from an external means.

TABLE 5-28. 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

The selector multiplexer designates which lines are active by the status of Y-register bits 8 through 11. Whenever the M05 capability is selected, Y11 is set high and when AUTO-DATA is high, these inputs are ANDed and inverted to place a low at the E inputs of multiplexer (M8) [M10]. If Y10 (the D input) is low, one of lines SSEL0 through SSEL7 is selected by the status of the four-position code applied to the S0 and S1 inputs by Y8 and Y9, respectively. Under these conditions, when both D and E are low, SSEL4 through SSEL7 are selected as S0 and S1 change to state 00, 10, 01, and 11, respectively. When D is high and E is low, SSEL0 through SSEL3 are selected as S0 and S1 change to state 00, 10, 01, and 11, respectively.

Y register W-field bits Y13 through Y16 ($\overline{Q}$ high outputs from flip-flops H11) are ANDed at (J10) [F7]. The resultant high is then ANDed with Y12 high at (G6) [H10] to produce W=0. This provides a low that is inverted (G7) [H8] for ANDing with Y11 and WRITE/ at AND gate (L4) [L9]. The Y11 and AUTO-DATA/ signals, if high at (L9) [L10], initiate the sent line selection. The Y11, WRITE/, and W=0 conditions all high at (L4) [L9] initiate the sent port line selection. With the selection condition initiated, the states of Y8, Y9, and Y10 activate the output selection when Y10 is high and the S0 and S1 code selects the particular output line. (Refer to table 5-29.)

TABLE 5-29. SPT AND SSEL LINE SELECTION

Select Lines	S0/Y8	S1/Y9	Y10
SPT00, SSEL00	0	0	0
SPT01, SSEL01	1	0	0
SPT02, SSEL02	0	1	0
SPT03, SSEL03	1	1	0
SPT04, SSEL04	0	0	1
SPT05, SSEL05	1	0	1
SPT06, SSEL06	0	1	1
SPT07, SSEL07	1	1	1

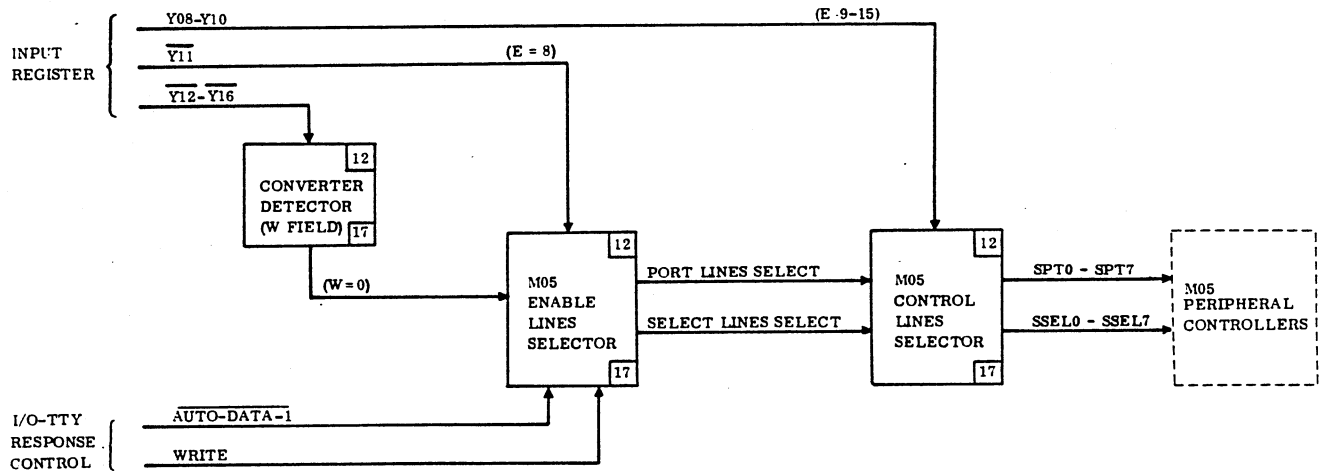


Figure 5-31. M05 SET/SAMPLE Selection Function

All data is transferred from the processor to the peripheral device by the I/O set command (SMB09=1) and from the peripheral device to the processor by the I/O sample command (SMB09=0). Therefore, when the set condition exists, 16 bits of data (SD01 through SD16) are transferred from the D register onto the output lines. When the sample condition exists, 16 bits of data (RD01 through RD16) on the input lines are transferred to the main three-state bus (BUS01 through BUS16). When M05-type I/O controllers are employed, special signal wiring is required.

TYPICAL MEMORY OPERATIONS

This subsection describes two typical operations of the basic micro-programmable processor: read micro memory and write macro memory.

The descriptions are given in terms of timing diagrams, data flow, the step-by-step execution sequence, selector positions, and contents of registers as well as the generated control signals. Refer to previous sections of this manual and the logic diagrams in the field print package manual as necessary to follow thee descriptions.

READ MICRO MEMORY

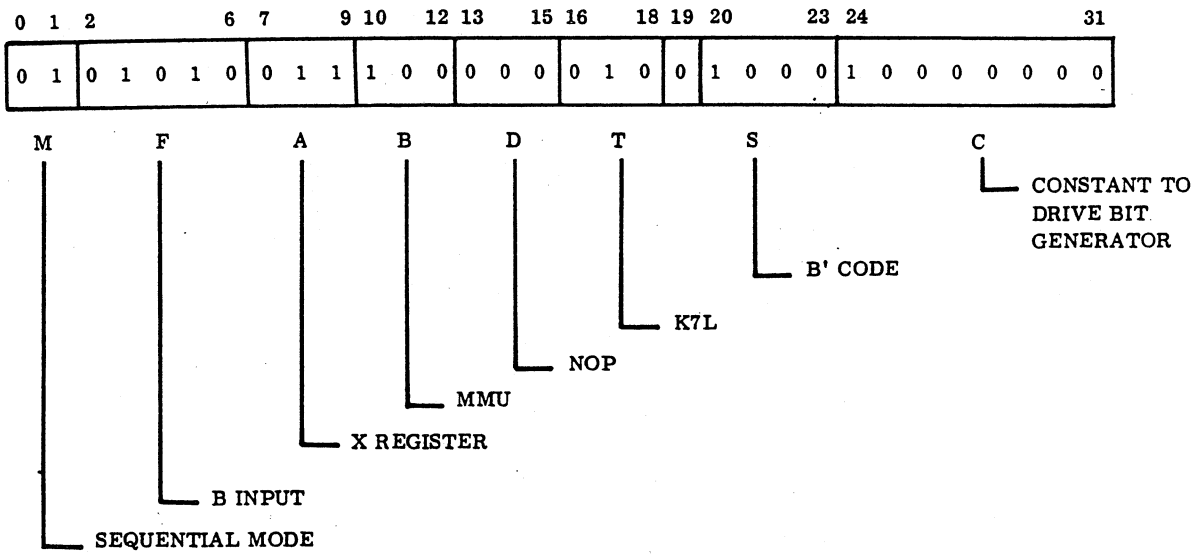
The typical 32-bit micro instruction $54E04880_{16}$ can be written in micro-instruction format as shown in figure 5-32.

Two operations are performed in parallel by this micro instruction when it is executed:

The upper 16 bits of data are transferred from micro memory to the X register.

The next micro instruction is selected for execution.

These operations are accomplished in the following manner.



Where: M = 01 indicates that this is a sequential mode operation.

F = 01010 references the B input.

A = 011 selects the X register as the A input to the ALU. In fact, however, the A field can be anything since the ALU is not using A input in this micro instruction (because of the F code).

B = 100 combined with S=1000 becomes B' code, which indicates read micro memory upper 16 bits.

D indicates an NOP operation.

T = 010 combined with bit 24=1 becomes T' code, which indicates that either the lower or upper micro instruction of the next pair is executed if the least significant bit of the K register (K07) is set or cleared.

S = 1000 indicates the B field is the B' coding.

C indicates a constant = 0 is used to drive the bit generator.

5043

Figure 5-32. Typical Read Micro Memory Instruction

The micro-memory address is first obtained from either the N/K register (control 2) or selector S5 of the transform module depending on status mode bit SM113. The 32-bit micro instruction is then read out from micro memory at this address. However, only the upper 16 bits of the micro instruction are gated to selector S2 via the main CPU three-state bus. The output of S2 goes through the ALU without any operation performed on it, through selector S3, and is finally gated into the X register. Meanwhile, the next sequential address from PS/MAC is loaded into the P/MA register. Figure 5-33 shows the address and data path of the micro instruction. Figure 5-34 shows in flow chart form the steps required to execute the read micro memory instruction. Figure 5-35 shows the detailed timing diagram.

A detailed description of read/micro memory is given below. To execute the micro instruction, it must first be gated into MIR by the leading edge of the GATEMIR signal (at time T5). After the micro instruction has been gated into MIR, all the fields of the micro instruction except the D field can be decoded simultaneously. In this case, since D = 0000 (NOP), it does not matter when the D field is decoded. The S code equal to 1000 indicates that this is an alternate to B′ coding. The decode of B′ code equal to 100 indicates this is a read upper micro memory. (The MM and ENMMUB signals in control 1 are generated.) At time T1 the EXTEND signal in control 1 becomes true, signifying that an extended cycle is required due to the B′ code operation. This enables the extend time generator and stops the odd/even time generator (in the control 1 module) at times T1 and T2. Initially, since the S6CONS1 signal is low, S6 selects either the combined N/K register (if SM113 equals 0) or the MA transform via selector S5 (if SM113 equals 1) to be used as the micro-memory address. As soon as TCE at C1-10 of the extend time generator in the control 1 module goes high, the micro-memory time generator is allowed to run. This micro-memory address (MA transform or N/K) is also gated into the P/MA register in control 2 by GATEMA and GATEPAGE (if SM113 equals 0). The test bit is determined by the least significant bit of K (K07) via the T′ code K7L (T′ = 010) to determine which 32-bit half-word is referenced. With the specified address, the 32-bit micro instruction is retrieved from micro memory, but only the upper 16 bits of data are gated to the main three-state bus.

The data goes through selector S2 of the ALU unit without having any operation performed on it (F = 01010 selects B input) and finally through selector S3. Meanwhile, when the TC signal at E2-10 of the control 1 module goes high, S6 selects the outputs of the PS/MAC register as its input source. At time TA, the $\overline{\text{MMGATEX}}$ signal is generated to gate the desired data to the X register. In addition, test multiplexer A9 in the control 2 module is inhibited since the upper micro instruction of the next sequential location is always selected after the referencing micro instruction. A RESTART signal is generated to allow the odd/even time generator to run again. Figure 5-36 shows the selector positions and contents of important registers in relation to the CPU control timing. Note that the T0 through T7 timing TCE, and TBE are outputs of the extend time generator. TA, TB, TC, and TD are outputs of the micro memory time generator. Table 5-30 lists the locations of these signals in the control 1 module.

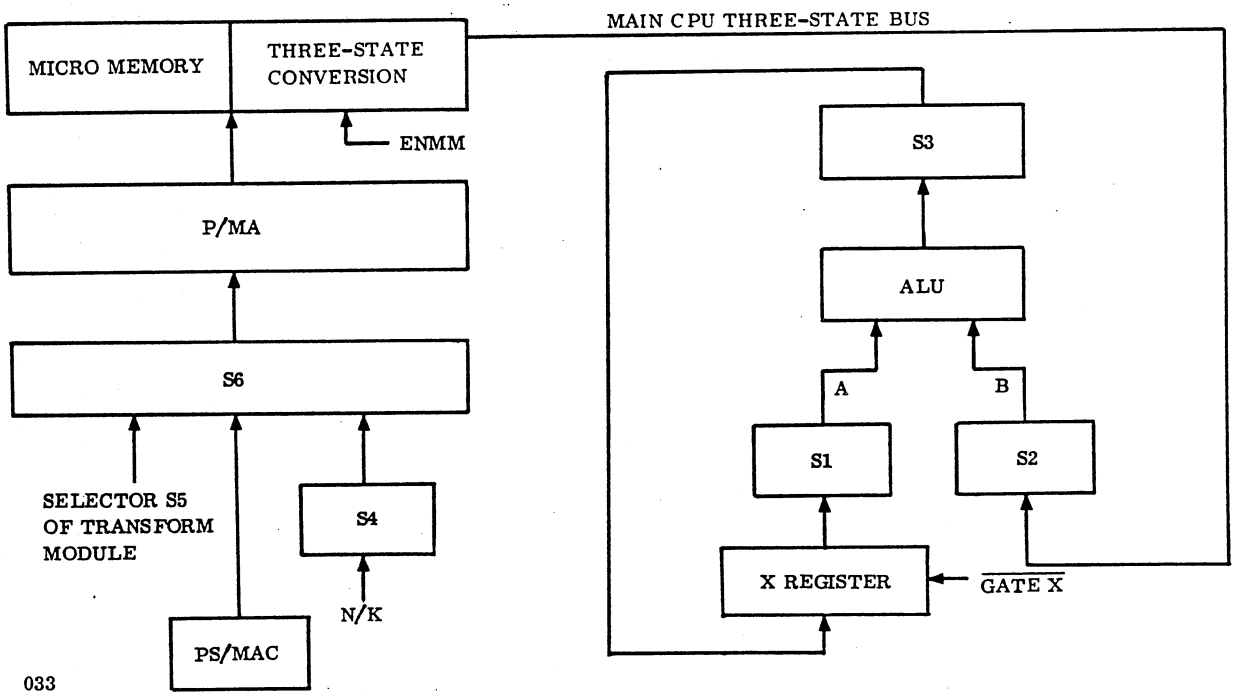


Figure 5-33. Read Micro Memory - Address and Data Path

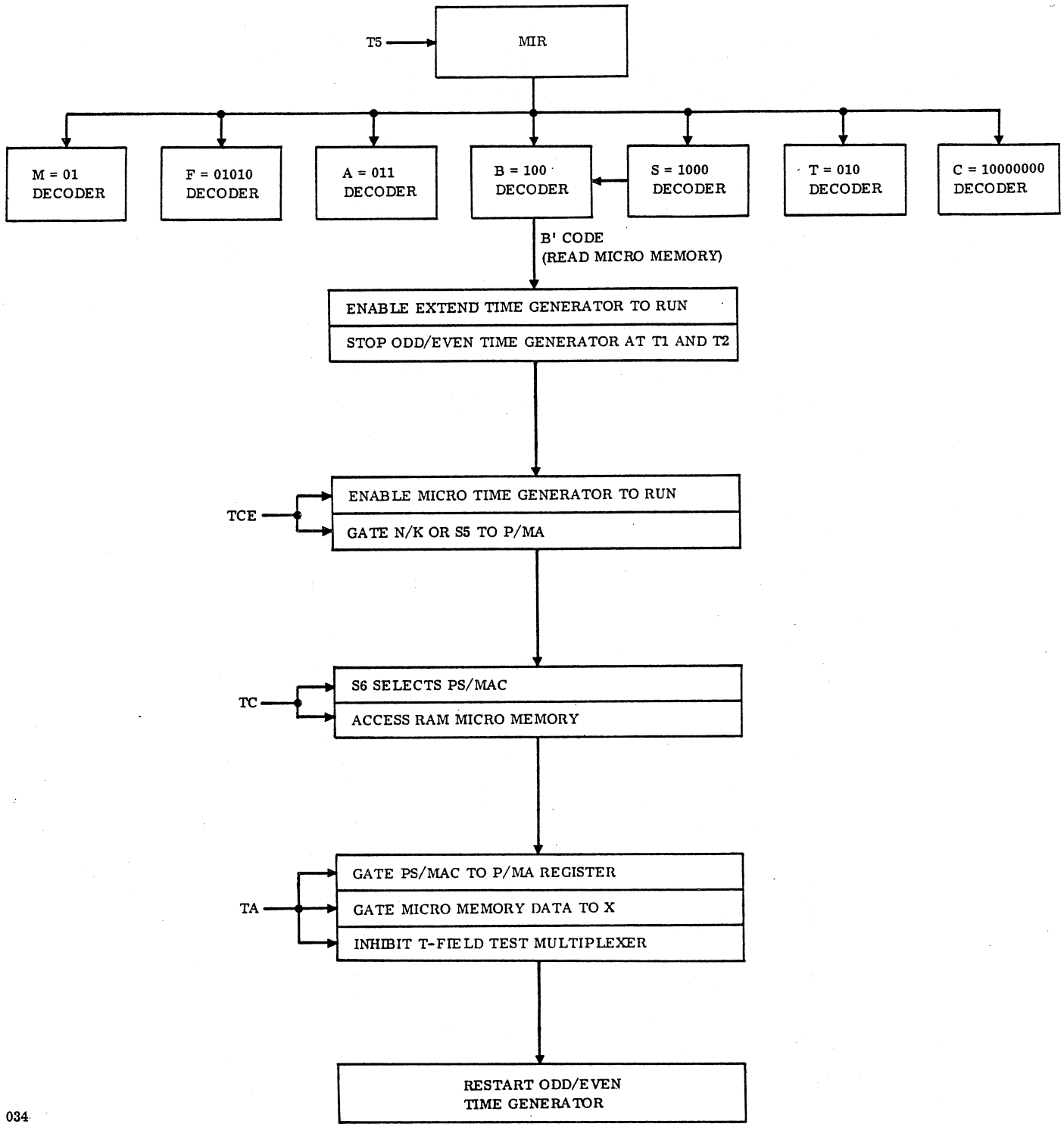
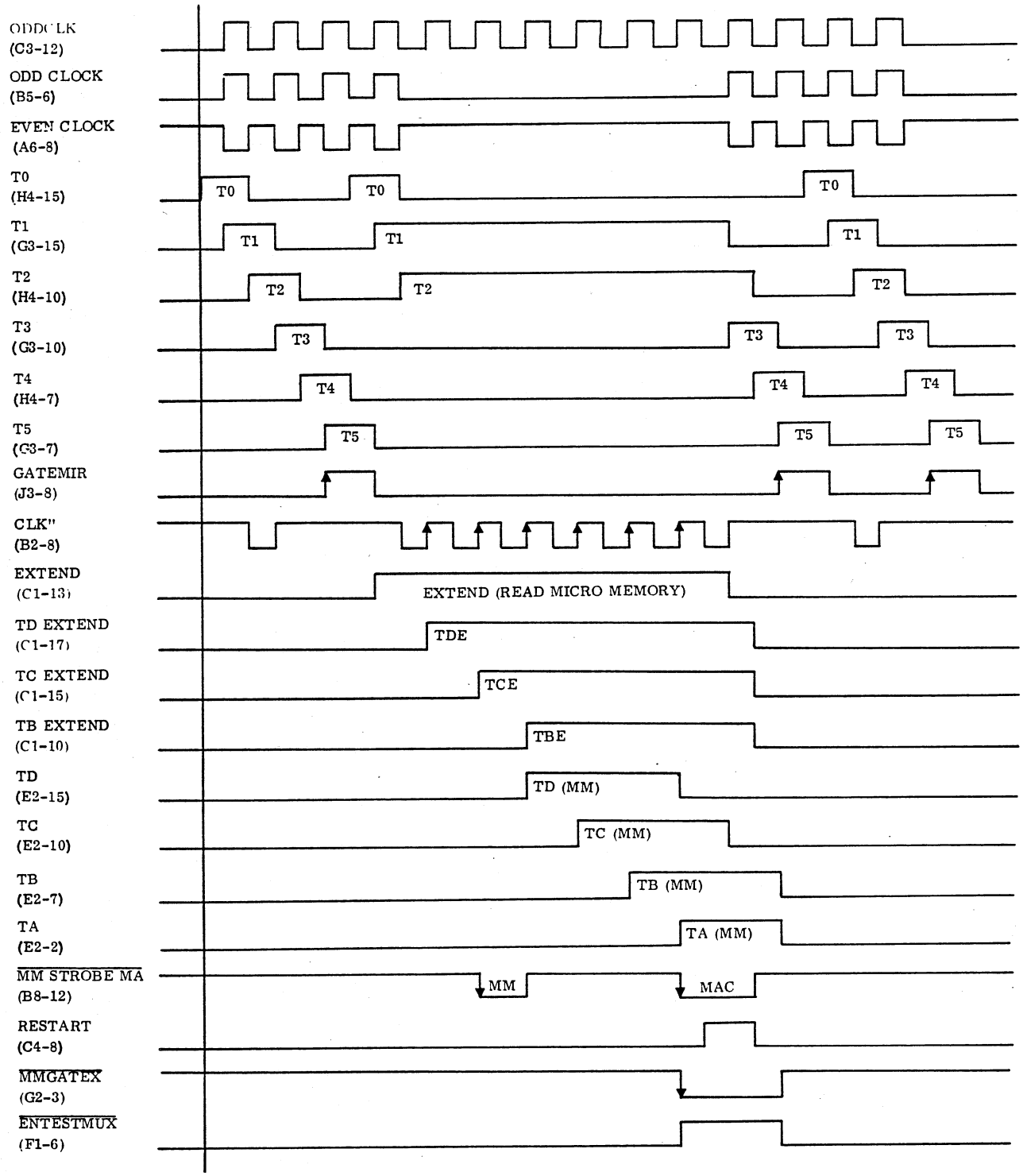


Figure 5-34. Read Micro Memory - Step-by-Step Execution



NOTE: ALL SIGNALS OF THIS TIMING DIAGRAM ARE IN THE CONTROL 1 MODULE

056

Figure 5-35. Read Micro Memory - Detailed Timing Diagram

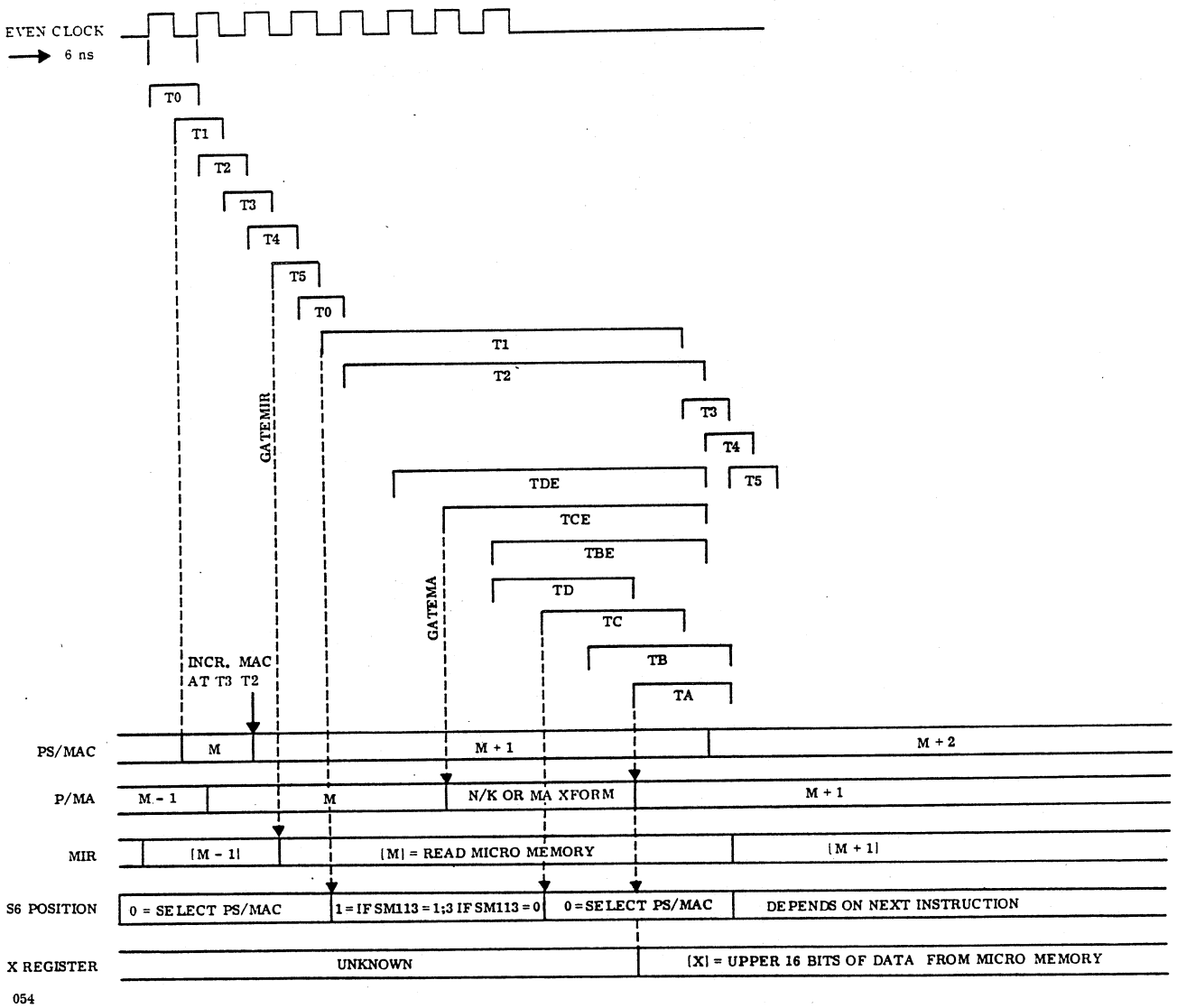


Figure 5-36. Read Micro Memory - Register Contents and S6 Position

TABLE 5-30. READ MICRO MEMORY - LOCATION OF RELATED SIGNALS DURING EXECUTION

Signal Names	Chip and Pin Numbers	Logic Diagram Sheet Number	Signal Names	Chip and Pin Numbers	Logic Diagram Sheet Number
T0	H4-15	5	TCE	C1-15	7
T1	G3-15	5	TBE	C1-10	7
T2	H4-10	5	TA	E2-2	7
T3	G3-10	5	TB	E2-7	7
T4	H4-7	5	TC	E2-10	7
T5	G3-7	5	TD	E2-15	7
T6	H4-2	5	EXTEND	D2-8	7
T7	G3-2	5	RESTART	C4-8	7
TDE	C1-7	7			

ARITHMETIC AND WRITE MACRO MEMORY OPERATIONS

Micro instruction $711F8484_{16}$ is written in micro-instruction format as shown in figure 5-37.

The following suboperations are performed by this micro instruction:

The contents of the A register (selected by S1) is added to the contents of the X register (selected by S2) and put in the F register via selector S3.

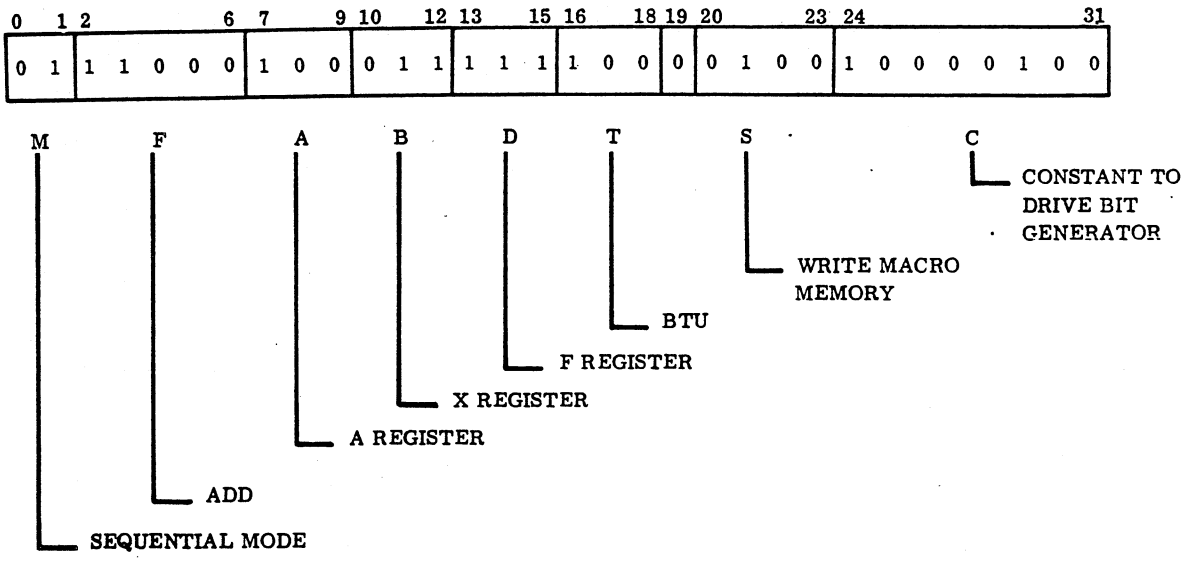
The result of this add operation (from the output of selector S3) is written to macro memory at the address specified by the contents of the AB register. (Assume that the AB register has already been loaded with the desired address.)

Select the upper or lower micro instruction of the next micro-instruction pair via the T field.

Figure 5-38 shows the step-by-step execution of these micro instructions in flow chart form. Figure 5-39 is the detailed timing diagram of the micro instruction.

To start execution of the micro instruction, it is gated to the MIR register at time T5. All fields, except the D field, are decoded simultaneously.

The M field = 01 indicates that this is sequential address mode; therefore, selector S6 at position 0 selects the outputs of PS/MAC as the input source. The T field is decoded at the BTU command; that is, the lower four bits of the C field are used to determine the bit position of the bit test selector (in the transform module). If the bit at position 4 is 1, the upper micro instruction is selected; otherwise, the lower micro instruction of the next micro-instruction pair is selected. The next sequential address from PS/MAC is gated into the P/MA register by the leading edge of the $\overline{\text{GATEMA}}$ signal at time T1 • $\overline{T2}$. This address is also loaded back into the PS/MAC register during time T1 and incremented at time T3 • $\overline{T2}$. The F field is decoded as an arithmetic (add) operation. The S field indicates that this is a write macro memory operation. Both of these operations require a time extension beyond the 168-nanosecond basic cycle. However, since the extend operation (add operation) is in progress, the time extension for the write macro memory is not started until the time extension for the add operation is expired (that is, a RESTART signal is received). Selector S1 selects the A register as the A input to the ALU.



Where: **M = 01 indicates that this is a sequential mode operation.**

F = 11000 indicates an ADD operation.

A = 100 selects the A register as the A input to the ALU.

B = 011 selects the X register as the B input to the ALU.

D = 111 selects the F register as the destination register.

T = 100 combined with bit 24=1 becomes the T' code which indicates that the lower or upper micro instruction of the next pair will be executed when the bit test is 0 or 1, respectively.

S = 0100 indicates the B code = write macro memory operation.

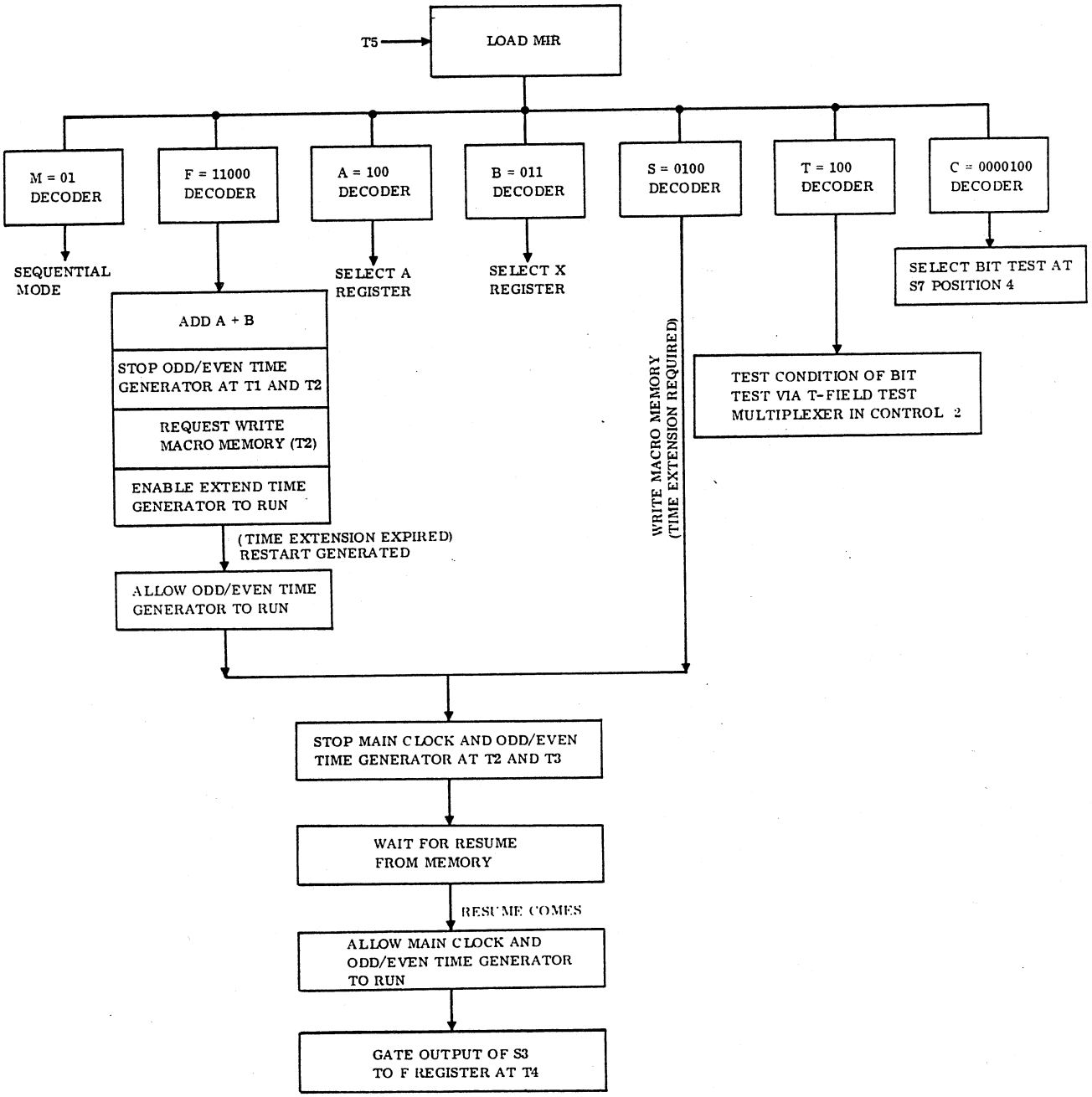
C = 1000100 the lower-order from bits are used to drive the bit generator.

5044

Figure 5-37. Typical Add and Write Macro Memory Instruction

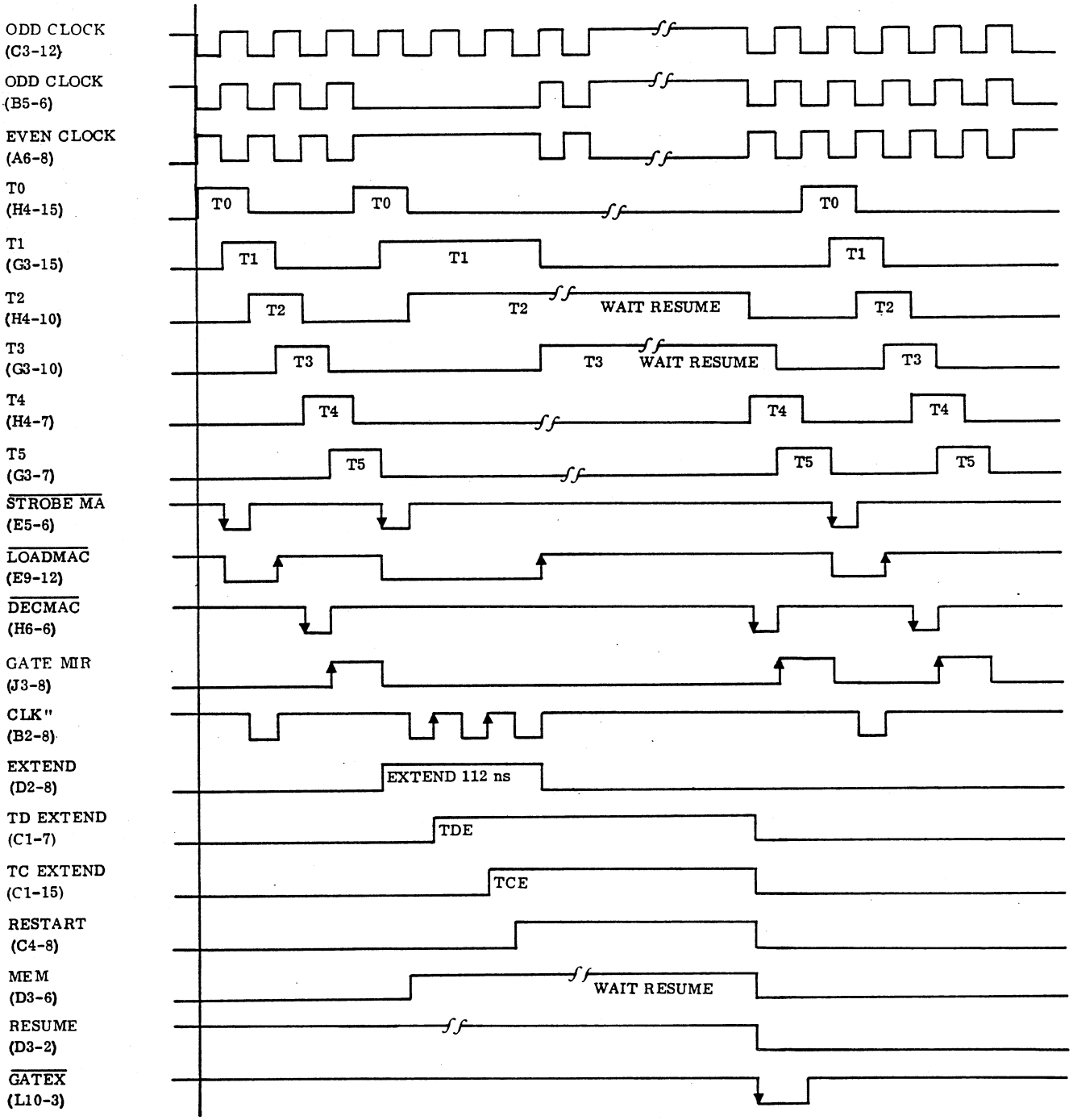
S2 selects the X register as the B input to the ALU. The EXTEND signal is raised at time T1 to stop the odd/even time generator at times T1 and T2. Assuming that the add operation is performed in ones complement, an extra 112 nanoseconds is required for the add operation in the ALU. After the TCE signal in the extend time generator goes active, RESTART is generated indicating that the 112-nanosecond time extension has expired. This resets the $\overline{\text{EXTEND}}$ signal and allows the odd/even time generator to run again. In addition, $\overline{\text{MEM}}$ at D3-6 goes active once the RESTART signal is received, causing the main clock generator to stop oscillating. This in turn stops the odd/even time generator at times T2 and T3 to wait for the RESUME from memory. Actually, part of the write macro memory operation is performed in parallel to the add operation.

The CPU-WRITE signal is generated as soon as the write command is decoded in the S field. The CPU memory request (CPU REQ) is generated at time T2. The data to the memory from the output of selector S3 is sent out to the memory interface module within 100 nanoseconds of the leading edge of the CPU REQ signal; the memory address should have been loaded into the AB register in the previous micro instruction.



032

Figure 5-38. Add and Write Macro Memory - Step-by-Step Execution



NOTE: ALL SIGNALS IN THIS TIMING DIAGRAM ARE ON THE CONTROL 1 MODULE

0545

Figure 5-39. Add and Write Macro Memory - Detailed Timing Diagram

When the CPU early data resume (CPU EARLY DS) signal is received from the memory, the main clock generator and the odd/even time generator are allowed to run again. At time T4, the outputs of selector S3 are gated into the F register. This completes the execution of the micro instruction.

MODULE INTERFACE SIGNAL DIAGRAMS

The module interface signal diagrams show the input and output signals of the control 1 (figure 5-40), control 2 (figure 5-41), ALU (figure 5-42), SMI (figure 5-43), transform (figure 5-44), and I/O TTY (figure 5-45).

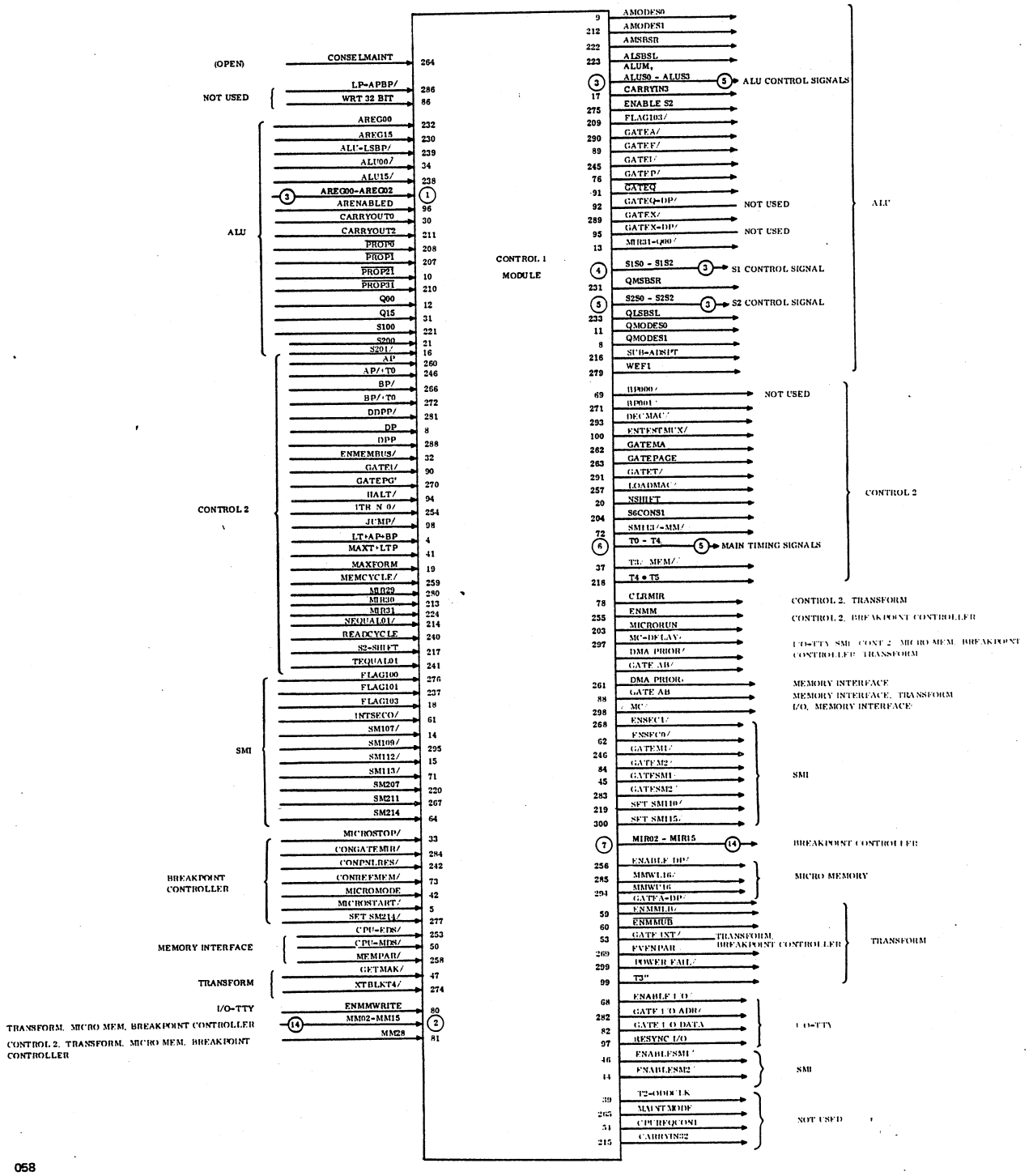


Figure 5-40. Control 1 Interface Signals (Sheet 1 of 2)

(1) AREG00 - AREG02

AREG00	206
AREG01	205
AREG02	7

(2) MM02 - MM15

MM02	27
03	26
04	25
05	24
06	23
07	22
08	55
09	56
10	57
11	58
12	63
13	65
14	66
MM15	70

(3) ALU CONTROL SIGNALS

ALUM	43
ALUS0	49
ALUS1	250
ALUS2	243
ALUS3	244

(4) S1 CONTROL SIGNALS

S1S0/	48
S1S1/	249
S1S2/	247

(5) S2 CONTROL SIGNALS

S2S0/	67
S2S1/	292
S2S2/	273

(6) T0 - T4

T0'/	29
T1'	6
T2'	3
T3'	74
T4'	75

(7) MIR02 - MIR15

MIR02	28
03	225
04	229
05	227
06	228
07	235
08	226
09	38
10	36
11	234
12	35
13	77
14	278
MIR15	79

058A

Figure 5-40. Control 1 Interface Signals (Sheet 2 of 2)

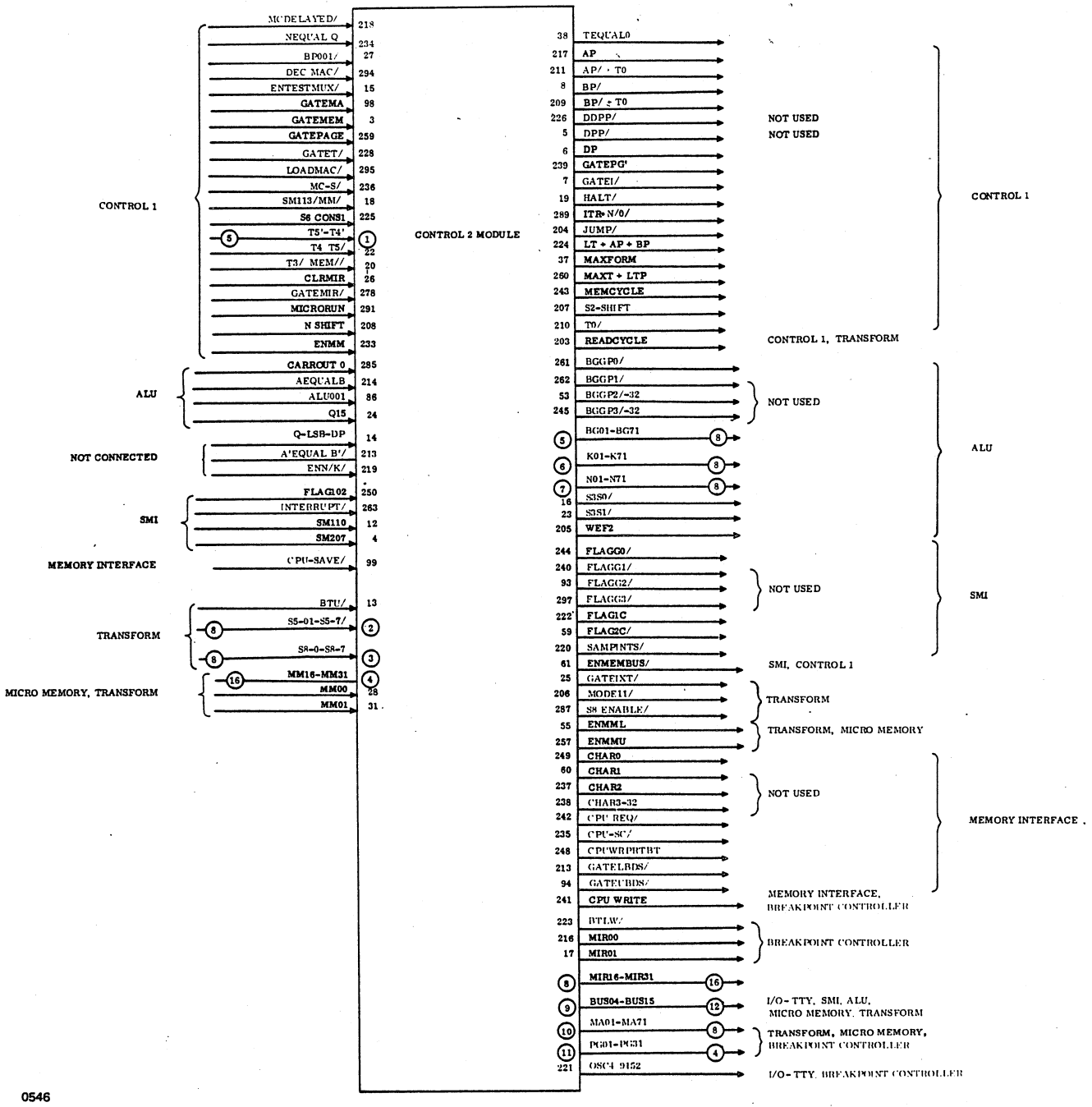


Figure 5-41. Control 2 Interface Signals (Sheet 1 of 2)

(1) T1' - T4'

T1'	212
T2'	296
T3'	62
T4'	21

(2) S5-0/-S5-7/

S5-0/	66
S5-1/	274
S5-2/	290
S5-3/	275
S5-4/	299
S5-5/	298
S5-6/	284
S5-7/	286

(3) S8-0-S8-7

S8-0	70
S8-1	71
S8-2	272
S8-3	269
S8-4	87
S8-5	88
S8-6	92
S8-7	90

(4) MM16-MM31

MM16	40
17	41
18	49
19	48
20	47
21	46
22	45
23	42
24	77
25	78
26	79
27	80
28	81
29	82
30	83
MM31	84

0546A

(5) BG0/-BG7/

BG0/	58
BG1/	256
BG2/	255
BG3/	54
BG4/	57
BG5/	56
BG6/	254
BG7/	253

(6) K0/-K7/

K0/	268
K1/	72
K2/	271
K3/	74
K4/	270
K5/	73
K6/	91
K7/	293

(7) N0/-N7/

N0/	292
N1/	89
N2/	85
N3/	266
N4/	265
N5/	65
N6/	63
N7/	264

(8) MIR16-MIR31

MIR16	231
17	232
18	227
19	33
20	39
21	246
22	36
23	247
24	229
25	29
26	230
27	32
28	280
29	277
30	279
MIR31	281

(9) BUS04-BUS15

BUS04	35
05	43
06	44
07	34
08	11
09	50
10	64
11	67
12	75
13	76
14	10
BUS15	9

(10) MA0/-MA7/

MA0/	273
MA1/	68
MA2/	276
MA3/	69
MA4/	282
MA5/	97
MA6/	96
MA7/	95

(11) PG0/-PG3/

PG0/	283
PG1/	258
PG2/	30
PG3/	267

Figure 5-41. Control 2 Interface Signals (Sheet 2 of 2)

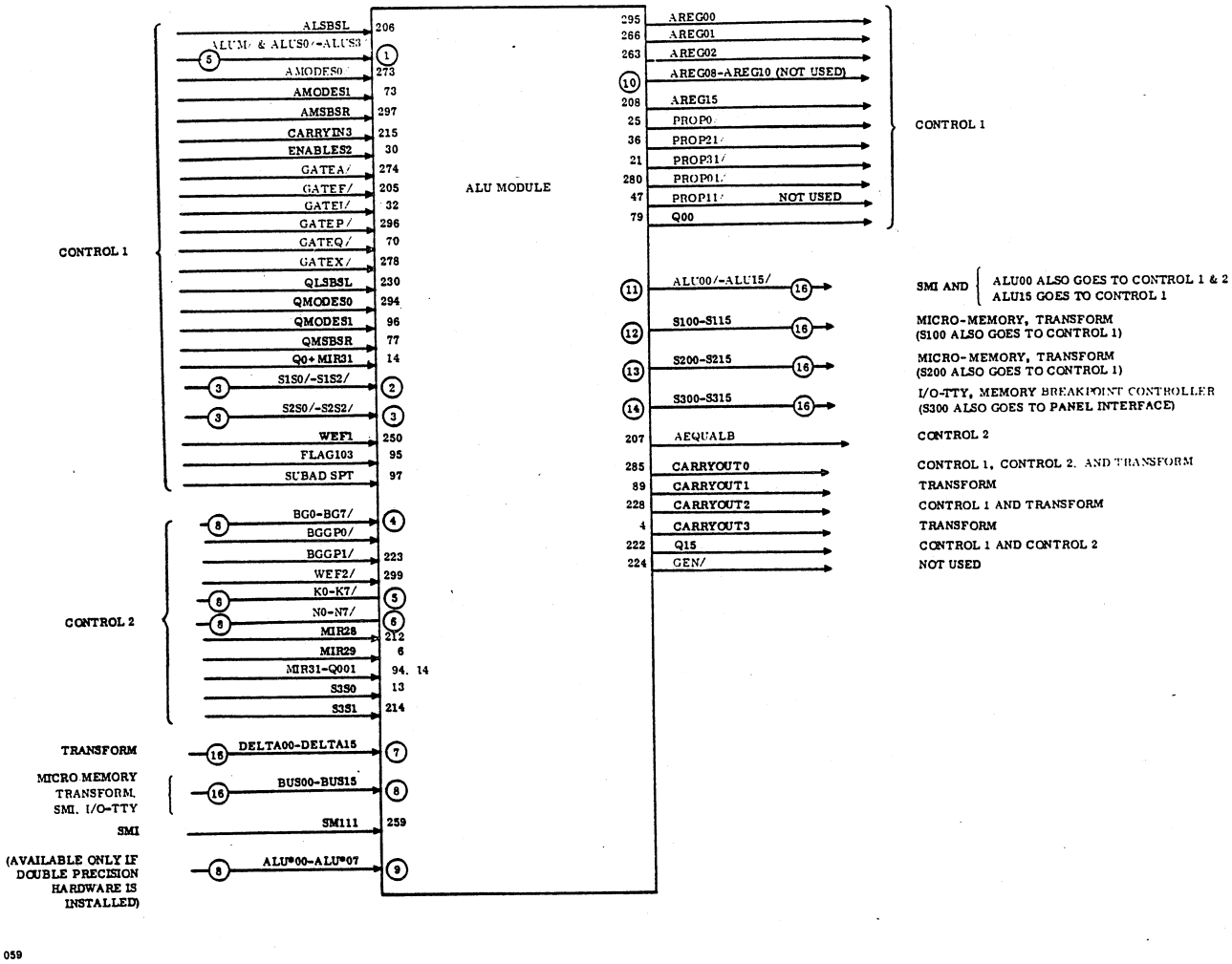


Figure 5-42. ALU Interface Signals (Sheet 1 of 2)

(1) ALUM/ AND ALUS0/-ALUS3/

ALUM/	284
ALUS0/	260
ALUS1/	239
ALUS2/	257
ALUS3/	37

(2) S1S0/-S1S2/

S1S0/	68
S1S1/	267
S1S2/	66

(3) S2S0/-S2S2/

S2S0/	270
S2S1/	69
S2S2/	269

(4) BG0/-BG7/

BG0/	291
BG1/	288
BG2/	293
BG3/	289
BG4/	221
BG5/	211
BG6/	210
BG7/	209

(5) K0/-K1/

K0/	229
K1/	49
K2/	276
K3/	253
K4/	242
K5/	48
K6/	277
K7/	17

(6) N0/-N7/

N0/	256
N1/	255
N2/	240
N3/	298
N4/	100
N5/	98
N6/	99
N7/	300

(7) DELTA00-DELTA15

DELTA00	281
01	286
02	287
03	290
04	254
05	262
06	268
07	272
08	236
09	243
10	245
11	249
12	203
13	216
14	217
DELTA15	220

(8) BUS00-BUS15

BUS00	59
01	60
02	61
03	62
04	35
05	43
06	44
07	34
08	11
09	50
10	64
11	67
12	75
13	76
14	10
BUS15	9

(9) *ALU00-*ALU07

*ALU00	247
01	248
02	237
03	238
04	20
05	219
06	7
*ALU07	213

(10) AREG08-AREG10

AREG08	26
AREG09	246
AREG10	244

(11) ALU00/-ALU15/

ALU00/	86
01/	82
02/	83
03/	85
04/	84
05/	63
06/	58
07/	57
08/	92
09/	93
10/	80
11/	81
12/	42
13/	8
14/	45
ALU15/	46

(12) S100-S115

S100	279
01	78
02	88
03	87
04	265
05	264
06	275
07	74
08	233
09	33
10	235
11	234
12	204
13	27
14	226
S115	227

(13) S200-S215

S200	282
01	283
02	90
03	91
04	71
05	261
06	72
07	271
08	31
09	241
10	232
11	231
12	18
13	16
14	225
S215	19

(14) S300/-S315/

S300/	54
01/	56
02/	55
03/	53
04/	39
05/	41
06/	40
07/	38
08/	23
09/	28
10/	24
11/	22
12/	29
13/	12
14/	15
S315/	5

059A

Figure 5-42. ALU Interface Signals (Sheet 2 of 2)

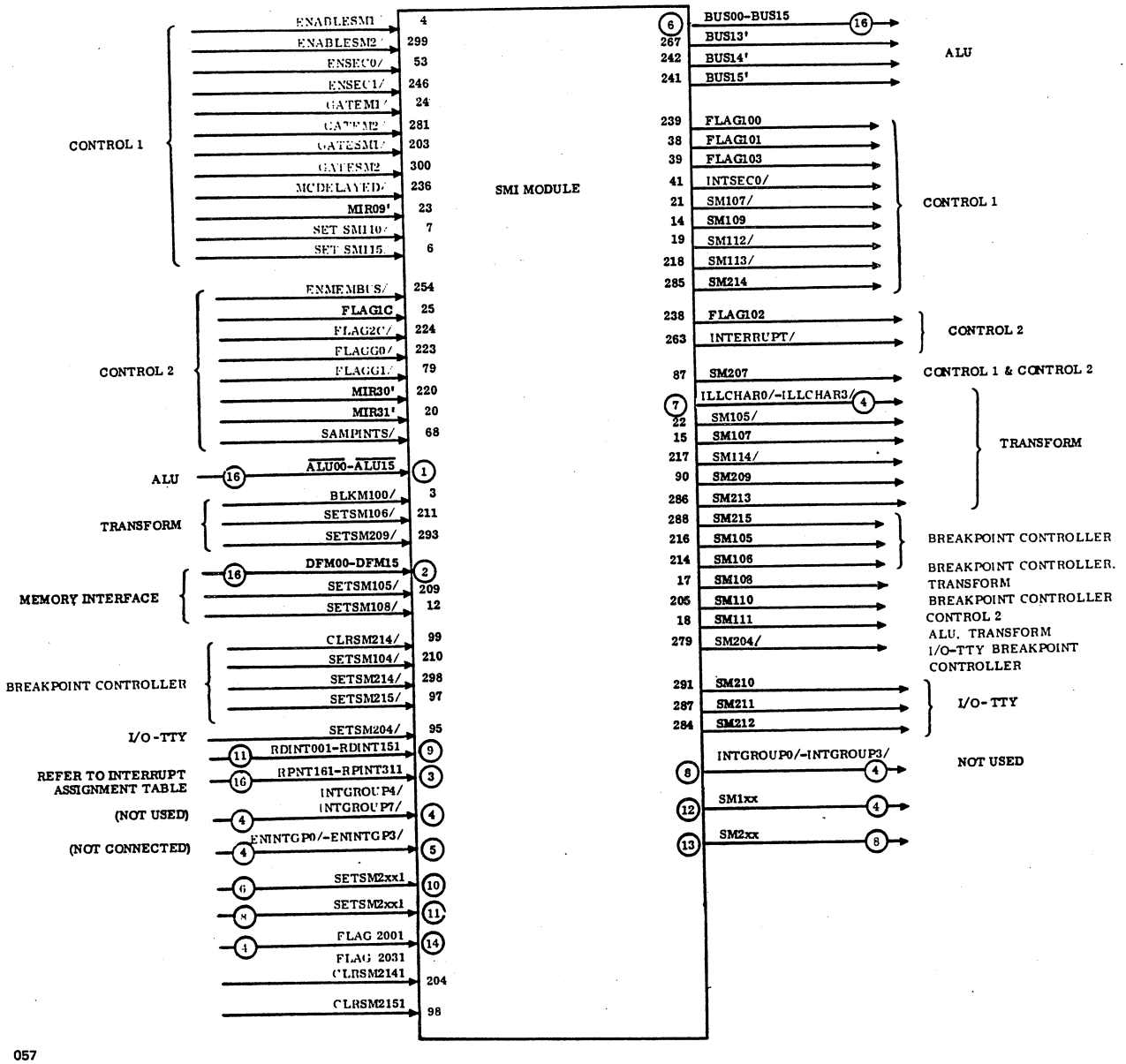


Figure 5-43. SMI Interface Signals (Sheet 1 of 2)

(1) ALU00/-ALU15/

ALU00/	86
01/	82
02/	83
03/	85
04/	84
05/	63
06/	58
07/	57
08/	92
09/	93
10/	80
11/	81
12/	42
13/	8
14/	45
ALU15/	46

(2) DFM00-DFM15

DFM00	259
01	54
02	56
03	256
04	258
05	255
06	55
07	257
08	249
09	250
10	247
11	48
12	248
13	253
14	49
DFM15	47

(3) RDINT00/-RDINT15/

RDINT00/	227
01/	27
02/	32
03/	232
04/	28
05/	31
06/	231
07/	238
08/	30
09/	230
10/	229
11/	29
12/	33
13/	226
14/	233
RDINT15/	234

(4) INTGROUP4/-INTGROUP7/

INTGROUP4/	265
5/	264
6/	36
INTGROUP7/	235

(5) ENINTGP0/-ENINTGP3/

ENINTGP0/	225
1/	26
2/	283
ENINTGP3/	282

(6) BUS00-BUS15

BUS00	59
01	60
02	61
03	62
04	35
05	43
06	44
07	34
08	11
09	50
10	64
11	67
12	75
13	76
14	10
BUS15	9

(7) ILLCHAR0/-ILLCHAR3/

ILLCHAR0/	37
1/	237
2/	261
ILLCHAR3/	260

(8) INTGROUP0-INTGROUP3

INTGROUP0/	262
1/	245
2/	244
INTGROUP3/	243

(9) RPINT16/-RPINT31

RPINT16/	69
17/	269
18/	270
19/	70
20/	72
21/	272
22/	271
23/	71
24/	73
25/	274
26/	74
27/	273
28/	277
29/	276
30/	77
RPINT31	275

(10) SETSM1xx/

SETSM107/	208
109/	207
111/	212
112/	13
113/	206
SETSM114/	213

(11) SETSM2xx/

SETSM205/	295
206/	294
207/	94
208/	296
210/	96
211/	299
212/	91
SETSM213/	292

(12) SM1xx

SM104	16
SM104/	222
SM106/	221
SM115/	219

(13) SM2xx

SM204	89
205	88
205/	78
206	290
206/	278
207	87
207/	280
SM208	289

(14) FLAG200/-FLAG203/

FLAG200/	66
201/	266
202/	240
FLAG203/	40

057A

Figure 5-43. SMI Interface Signals (Sheet 2 of 2)

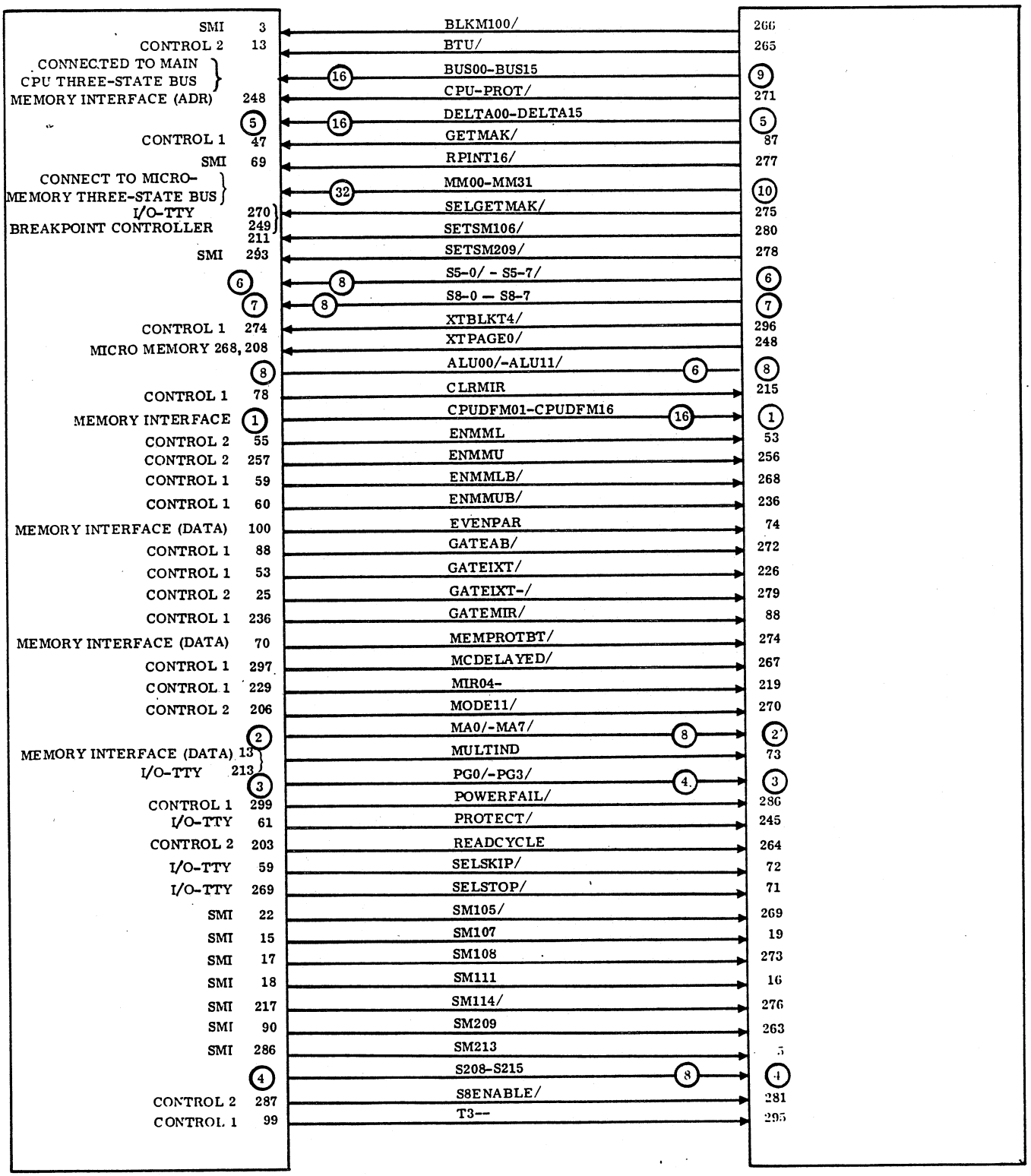


Figure 5-44. 1700 Transform Interface Signals (Sheet 1 of 2)

(1) CPUDFM01-CPUDFM16

MEMORY INTERFACE		TRANSFORM
16	CPUDFM1	208
24	2	3
18	3	4
19	4	207
20	5	229
22	6	228
23	7	227
25	8	230
97	9	206
92	10	205
91	11	203
98	12	204
93	13	297
95	14	299
99	15	300
89	CPUDFM16	298

(2) MA0/-MA7/

CONTROL 2		TRANSFORM
273	MA0/	231
68	1/	257
276	2/	258
69	3/	247
282	4/	246
97	5/	244
96	6/	253
95	MA7/	255

(3) PG0/-PG3/

CONTROL 2		TRANSFORM
283	PG0/	54
258	1/	254
30	2/	249
267	PG3/	232

(4) S208-S215

ALU		TRANSFORM
31	S208	94
241	09	95
232	10	293
231	11	96
18	12	97
16	13	98
225	14	99
19	S215	100

0901

(5) DELTA00-DELTA15

TRANSFORM		ALU MODULE
211	DELTA0	281
7	1	286
209	2	287
6	3	290
8	4	254
212	5	262
210	6	268
213	7	272
223	8	236
20	9	243
21	10	245
220	11	249
221	12	203
224	13	216
222	14	217
225	DELTA15	220

(6) S5-0/ - S5-7/

TRANSFORM		CONTROL 2
289	S5-0/	66
290	1/	274
90	2/	290
91	3/	275
291	4/	299
92	5/	298
292	6/	284
93	S5-7/	286

(7) S8-0 - S8-7

TRANSFORM		CONTROL 2
283	S8-0/	70
85	1/	71
285	2/	272
89	3/	269
86	4/	87
287	5/	88
288	6/	92
282	S8-7/	90

(8) ALU00/-ALU11/

ALU		TRANSFORM
86	ALU00/	36
85	03/	37
84	04/	38
57	07/	39
92	08/	68
81	ALU11/	69

(9) BUS00-BUS15

	TRANSFORM
BUS00	59
01	60
02	61
03	62
04	35
05	43
06	44
07	34
08	11
09	50
10	64
11	67
12	75
13	76
14	10
BUS15	9

(10) MM00-MM31

	TRANSFORM
MM00	28
01	31
02	27
03	26
04	25
05	24
06	23
07	22
08	55
09	56
10	57
11	58
12	63
13	65
14	66
15	70
16	40
17	41
18	49
19	48
20	47
21	46
22	45
23	42
24	77
25	78
26	79
27	80
28	81
29	82
30	83
MM31	84

Figure 5-44. 1700 Transform Interface Signals (Sheet 2 of 2)

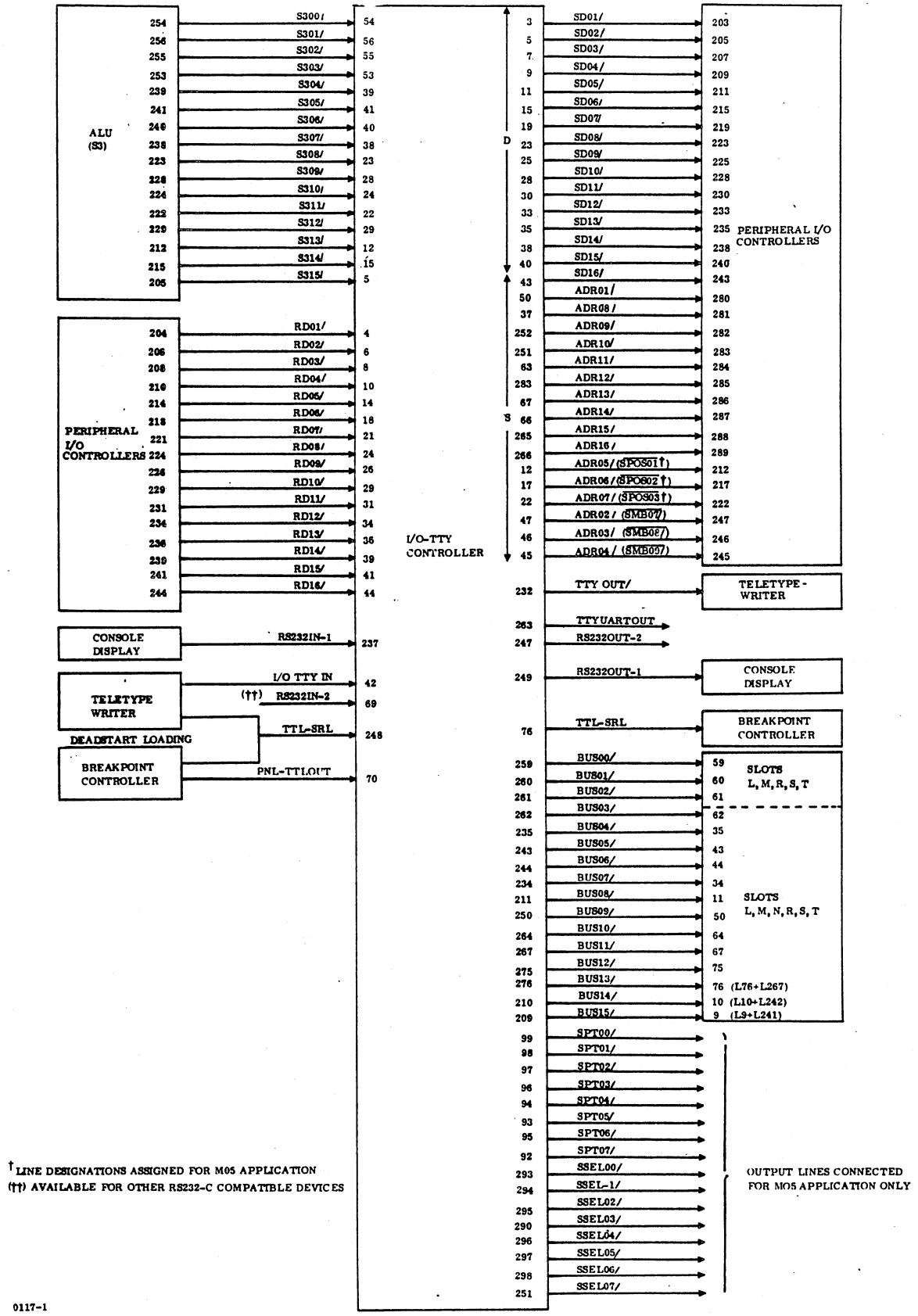


Figure 5-45. I/O-TTY Interface Signals (Sheet 1 of 2)

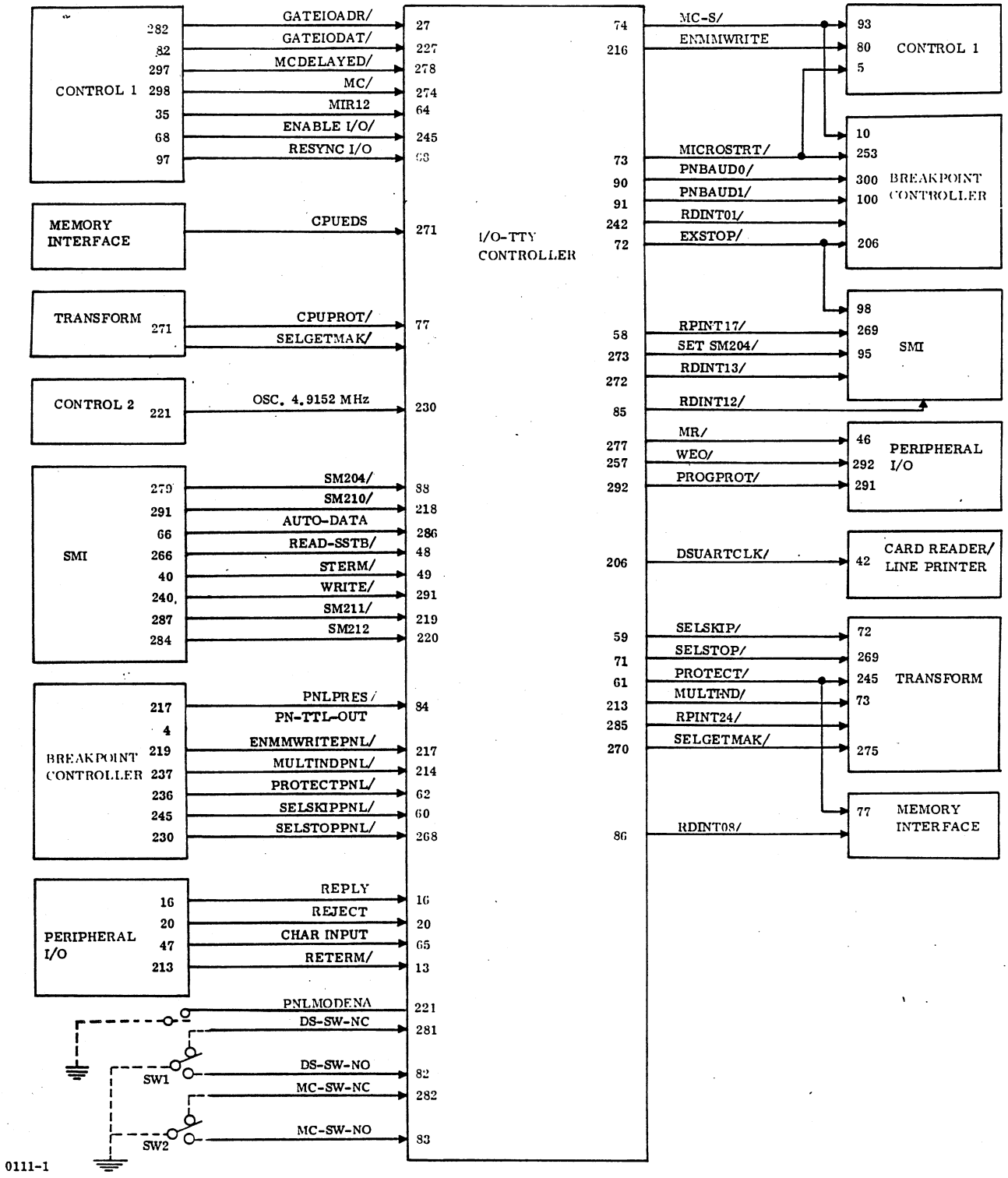


Figure 5-45. I/O-TTY Interface Signals (Sheet 2 of 2)

MAINTENANCE 6

PREVENTIVE MAINTENANCE

Preventive maintenance performed on the basic processor consists of the following:

Check the six muffin fans located at the bottom of the card cage for proper operation. Replace where required.

Remove any dust or other foreign matter that has collected on the printed-circuit boards or the cage.

Run the appropriate diagnostics to verify operational readiness.

When on-site spares are present, install them into the system and verify their operation using the diagnostics.

This preventive maintenance takes approximately one hour to perform and should be done annually, preferably as part of an emergency service call.

CALIBRATION AND ALIGNMENT

None required.

TROUBLESHOOTING

Troubleshooting can be performed using the applicable operational diagnostic system (ODS) diagnostics and the diagnostics decision logic tables (DDLTs) in the central processor field repair guide. These are effective in isolating faults to the board level. Repair should be effected by replacing the faulty board with a previously verified spare.

MAINTENANCE AIDS

The only maintenance aid required is the card extender, part number 96742400.

ON-SITE MAINTENANCE

Both emergency and preventive on-site maintenance is limited to isolating a fault to the replaceable subassembly, then effecting repair by replacing the faulty assembly with a previously tested spare.

CAUTION

Several PWAs in the processor contain electrostatic-sensitive devices and are identified with a red solder mask. Exercise extreme care in handling to avoid damage. Common practices, such as touching a grounded surface before handling, inserting in anti-static or conductive bag for storage or transfer, repairing at only properly equipped and grounded work stations, must be strictly followed.

SPARES TESTING

All spare subassemblies must be tested upon receipt, with retesting occurring annually.

GLOSSARY

A = B
A input and B input to arithmetic/logical unit are equal.

A FIELD
In a micro instruction, the A field specifies the source of the operand to be sent to the ALU from selector S1.

A REGISTER
General-purpose register.

A′ = B′
Not used.

A* REGISTER
Register included in systems that contain hardware double-precision option.

A03 - A07
A register inputs; A03 and A07 also activate the shift right mode of cascaded A register elements.

AB
Address buffer register; macro memory address register.

ADR
Address register on I/O TTY controller.

ADR01/ THROUGH ADR16/ (ARD01 = LSB)
Transfer of address data to peripheral I/O controller; the nine highest order bits specify the device, and the seven lowest order bits specify the function.

ADT-SELECTED
Auto-data transfer selected.

$\overline{\text{AFLD02}}$
A′ field selection of mask registers M1 and M2 and double-precision A and Q registers.

ALSBSL
Entered data during A register shift left.

ALU
Arithmetic/logical unit; performs arithmetic and logical operations on two operands sent from selectors S1 and S2.

ALU-LSBDP/
Arithmetic/logical unit least significant bit of D′ field.

ALU00 - ALU15
Arithmetic/logical unit output.

ALUM
Arithmetic/logical unit mode control.

ALUS0 - ALUS3
Arithmetic/logical unit function select.

AMODES0, AMODES1
A register mode select control signals.

AMSBSR
Entered data during A register shift right.

ANSHFTCLK
Shift clock to A register to effect left shift of data.

AP
Alternate A field coding, A′; decode of MIR07 through MIR09.

AP/ + T0
Alternate A field code, A′ or time T0 select the BUS00 and BUS02 data to be applied to selector S1.

AR-ENABLED
A register enabled.

AREG00 - AREG15
A register output.

AUTO-DATA
Micro level memory transfer of data blocks to or from any device that can accommodate the ADT mode of operation; the 1700 emulator processes each data interrupt and inputs or outputs the next word of data.

B FIELD
In micro instruction, the B field specifies the source of the operand to be sent to the ALU from selector S2.

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.

BG0/ - BG7/
Bit generator output, least significant decode.

BGGP0/ - BGGP3/
Bit generator output, most significant decode.

BGGP2-32
Not used.

BGGP3-32
Not used.

$\overline{\text{BLKAB}}$
Inhibit AB fields of micro instruction.

BLKABFF
Block AB filp-flop output to AB inhibit logic.

BLKM100/
Inhibits the macro halt interrupt M100 of the transform.

BP/
Alternate B field, B′; decode of MIR20 through MIR23.

BP/ + T0
Alternate B field code B′ or time T0 select the BUS01 and BUS03 data to be applied to selector S2.

BP000/
Not used.

BP000/ - BP001/
Gating signals that enable RTJ register output to main CPU three-state bus.

BP001/
Alternate B field code to inhibit the control 2 BUS lines.

BTLW
Upper/lower micro-instruction selection signal.

BTU/
Bit test upper; selector S7 output.

BUS00 THROUGH BUS15/ (BUS00 = MSB)
Transfer of real-time clock status, TTY controller status, character data, or peripheral I/O controller data to CPU via three-state bus.

C FIELD
In micro instruction, constant (c) field. Can contain constants, micro-memory addresses, or other codes, depending on format of micro instruction.

CARRYIN-32
Carry input for 32-bit machines only.

CARRYIN3
Carry input to arithmetic/logical unit group, ALU12 through ALU15 unit.

CARRYOUT0 - CARRYOUT3
Carry output from arithmetic/logical unit units.

CHAR INPUT
Sent by the I/O controller(s) to the CPU during read operations; when the line is active, an 8-bit character code is loaded into the lower eight bits of the A register without disturbing the upper eight bits of the A register.

CHAR0
Character zero; control line to memory interface for writing 8-bit character, buts 0 through 7 (most significant bits).

CHAR1
Character one; same as CHAR0 except for bits 8 through 15 (least significant bits).

CHAR2-32
Character two, 32-bit machine only, bits 16 through 23.

CHAR3-32
Character three, 32-bit machine only, bits 24 through 31.

CLK"
Time-extend generator clock and micro-memory time generator clock; enabled only while T1 is high.

CLR-CONT+CLR-INT
Clear controller or clear interrupt; clear I/O-TTY controller or clear interrupt from I/O-TTY controller).

$\overline{\text{CLRMIR}}$
Clear micro-instruction register.

$\overline{\text{CLRPAGE}}$
Clear page register.

$\overline{\text{CLRPGN}}$
Clear both page and N register.

CLRSM114/ - CLRSM115/, CLRSM214/ - CLRSM215/
Direct clear signals of SM114-SM115 and SM214-SM215, respectively.

$\overline{\text{CMEM}}$
Character-to-memory control line enable.

CONGATEMIR/
Control gate micro-instruction register.

CONREFMEM
Control reference memory.

CONSELMNT/
Open for future application.

CPU EDS/
CPU early data strobe. Indication from memory that data has been received.

CPU MDS
CPU memory data strobe.

CPU-PROT/ (CPUPROT/)
CPU protect. A discrete signal that indicates the CPU is in a protected status and only those instructions containing a 0 in the protected bit position will be accepted. All others are rejected.

CPU-REQ
CPU memory cycle request.

CPU-SAVE/
Stack drive inhibit. Inhibits all stack drives.

CPU-SC/
CPU split cycle request to memory.

CPUREQCON/
Open for future application.

CPUWRITE
CPU write control line.

D FIELD
In micro instruction, destination (D) field. Specifies destination for results of operation performed by ALU.

D REGISTER
Data register on 1700 enhanced processor I/O card.

D01 THROUGH D16
Data bits from D register that are employed within the I/O-TTY controller.

DATAINT8/
Data interrupt (RDINT08/).

DDPP/
DD DOUBLE PRIME (DD"), alternate D field.

DEADSTART
Optional logic that allows read/write micro memory to be loaded from external input device.

DECMAC/
Micro-memory address counter (MAC) clock.

DELTA00 - DELTA15
Encode of macro instructons to selectors S1 and S2 of the arithmetic/logical unit printed wiring assembly.

DFM01 - DFM16
Data from memory bits 1 through 16; read data from memory.

DISABLE OVERFLOW
Disable overflow conditions selector.

DMAPRIOR/
When this line is true, the DMA device determined by ENAB1-5 switches has absolute priority.

DP
D'; alternate D field.

DPP/
D"; alternate D field.

(DRDY)(BEL+ESC+RES • PP)
Data ready and BEL or escape or reserved and breakpoint controller present; special character codes for selection of manual interrupt, release reserve, reserve when panel mode is selected.

DRR
Data received reset; enables loading of UART receive register.

(DS)(DIR-RT)
Data status and director.

DS-SW-NC, DS-SW-NO
When connected, provides for external loading of data into the CPU. (NC = normal closed; NO = normal open; requires momentary double-pole single-throw contact arrangement).

DS-SWNC
Deadstart switch, normal closed contact.

DS-SWNO
Deadstart switch, normal closed contact.

DSUARTCLK/
153.6 kHz clock signal which is 16 times the maximum baud rate of 9600; sent to the card reader/line printer controller.

E-REPLY
Enable reply; initiates enable of the I/O-TTY controller reply response to the CPU.

E-TTYREPLY
Enable I/O-TTY controller reply response to the CPU.

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.

$\overline{\text{EN-T0}}$
Enables even time generator.

EN-T1
Enables odd time generator.

ENA-QSHIFT
Enable A and Q register mode selection.

ENABLE DP/
Enable double precision.

ENABLE IO/
Enable input output.

ENABLE S2
Enable selector 2.

ENABLE S2-1, S2-2
Enable selector S2 lower eight bits and upper eight respectively.

ENABLE SM1/ (ENABLESM1)
Enable status mode register 1.

ENABLE SM2/ (ENABLESM2)
Enable status mode register 2.

ENABLEIO/
Enables transfer of I/O-TTY controller and peripheral I/O controller data to the three-state bus.

ENDSHIFT/SCALE
End shift and scale operations.

$\overline{\text{ENF1}}$
Enable file 1 of ALU if supplied.

ENINTGP0/ - ENINTGP3/
Enable the priority encoders for interrupt group 0 through interrupt group 3.

ENMEMBUS/
Enable memory output to CPU three-state bus.

ENMM
Enable micro memory.

ENMMLB/
Enable transform micro memory bits 16 through 31 to CPU three-state bus.

ENMMUB/
Enable transform micro memory bits 00 through 15 to CPU three-state bus.

ENMMWRITE
Provides for input of data to micro memory via control 1.

ENMMWRITEPNL
Enable micro memory write from the panel interface.

ENMMWRITEPNL/
Breakpoint controller switch condition that provides for input of data to micro memory.

ENNK/
Selector S4 select line.

ENOVERFLOW
Enable overflow latch to set SM110 bit.

ENSEC0/ - ENSEC1/
Gating signals; enable the interrupt register output to the main CPU three-state bus.

ENTESTMUX/
Enables the T field test multiplexer.

EOPSTATUS
End of page status; message termination has taken place at end of page.

EVENCLK
Delay oscillator output pulse train.

EVENPAR
Even parity.

EXSTOP/
An interrupt signal that initiates a processor macro stop (clear SM204), which causes emulation to stop.

$\overline{\text{EXTEND}}$
Signal indicates time extension is required for execution of certain micro instructions.

F FIELD
In micro instruction, the function (F) field. Specifies operation to be performed by ALU or shift or scale of A or A/Q registers.

F REGISTER
General-purpose register.

F00 - F15
Output from F register.

F200 - F215
Output from file 2.

FERROR
Framing error; character code received contained an error of start, data, parity or stop format.

FILE 1
Register file addresed by content of K register.

FILE 2
Register file typically addressed by contents of N register.

FIRMWARE
General term for combination of micro instructions used in micro program to perform a certain operation.

FLAG100 - FLAG103, FLAG200 - FLAG203
Same as status mode bit SM100-SM103 an SM200-SM203.

FLAG1C
Sets the flag bit selected by MIR30 and MIR31; decoded from MIR25 through MIR27.

FLAG2-32
Not used.

FLAG2C/
Clears the flag bit selected by MIR30 and MIR31. Decoded from MIR25 through MIR27.

FLAG3-32
Not used.

FLAGG0/
Enables the FLAG100-FLAG103 multiplexer so that one of the flag bits can be set/cleared; decoded fom MIR28 and MIR29.

FLAGG1/
Enables the FLAG200-FLAG203 multiplexer so that one of the flag bits can be set/cleared; decoded from MIR28 and MIR29.

$\overline{\text{FWREAD}}$
Read from macro memory.

$\overline{\text{FWWRITE}}$
Write to macro memory.

G1-20 - G1-22 (G2-20 - G2-22)
Interrupt priority gating signals.

GATEA-DP/
A register clock for double-precision option.

GATEA/
A register clock.

GATEAB/
CPU memory address register clock.

$\overline{\text{GATED}}$
Clock overflow flip-flop to enable setting of bit SM110.

GATEF/
F register clock.

GATEI/
I register clock.

GATEIOADER/
Input/output address strobe to I/O-TTY controller.

GATEIOADR/
A 56-nanosecond negative-going pulse that strobes D-register data to the Y register (executed by micro instruction D'=001, establishing that this is I/O address data) coincident with trailing edge transition.

GATEIODAT/
A 56-nanosecond negative-going pulse that strobes ALU data to the D register (executed by micro instruction D=000, establishing that this is I/O data) coincident with leading edge transition.

GATEIXT/
IXT register clock.

GATELBDS/
Memory lower bound address strobe.

GATEM1/
Mask register 1 clock.

GATEM2/
Mask register 2 clock.

GATEMA/
Micro memory address register clock.

GATEMEM/
CPU memory request timing clock.

GATEMIR/
Micro-instruction register clock.

GATEP/
P register clock.

GATEPAGE
Micro-memory page address register clock.

GATEPG'
Open for future application.

GATEQ-DP/
Q register clock for double-precision option.

GATERTJ
Clock return jump register.

GATESM1-1
Clock status mode register upper eight bits.

GATESM1-2
Clock status mode register lower eight bits.

GATESM1/
Status mode register 1 clock.

GATESM2-1
Clock status mode register upper eight bits.

GATESM2-2
Clock status mode register lower eight bits.

GATESM2/
Status mode register 2 clock.

GATEUBDS/
Memory upper bound address strobe.

GATEX-DP/
X register clock for double-precision option.

GATEX/
X register clock.

GEN/
Arithmetic/logical unit carry generator look ahead.

$\overline{\text{GEN0}}$ - $\overline{\text{GEN3}}$
Carry generate outputs from ALU to look-ahead carry generator.

GETMAK
Obtain the memory address of the next instruction from the K register.

HALT
Decoded from S field (MIR20 through MIR23 = 1101); stops the processor if SM109 is set.

HI-1, -2, -3
High input (common tie point 1, 2, or 3 for a logic high termination).

I REGISTER
General-purpose register that can be used to hold a software instruction during execution if the microprocessor is configured as an emulator.

I/O-TTYOUT
I/O-TTY controller output to teletypewriter.

I/OTTYIN
Transmision of teletypewriter character codes (ASCII) over a 10 mA current loop from CPU to I/O-TTY controller.

INDOOFF/
Signal generated when ALU00 through ALU15 - $00FF_{16}$.

INT STATUS
Interrupt status; indicates to CPU that interrupt exists.

$\overline{\text{INT16'}}$
Inhibit interrupt present condition to indicate error or fault condition has been detected.

INTERRUPT/
Signal indicates that an active interrupt is recognized.

INTGROUP0/ - INTGROUP7/
Group of select output of priority encoders indicating one or more interrupts within the group is active.

INTSEC0
Open for future expansion of interrupts.

ITR N≠0/
Control signal that blocks the clock of micro-memory address (MA) register during a repeat operation and N is not equal to 0.

JUMP/
Generated during jump to same page operation (MIR00 and MIR01 = 10 and MIR19 = 0) to block the clocks of the page register.

$\overline{K = 0}$
K register equals zero.

K MODE
Keyboard mode (printer connected).

K REGISTER
Eight-bit counter that can be cleared, incremented, or decremented under micro-instruction control. Also used to address file 1.

K0/ - K7/
K register output.

$\overline{KCLK}$
K register clock.

KDOWN
K register count up/down; MIR = 1 count up, MIR = 0 count down.

$\overline{KLOAD}$
Load K register.

KMODE • RDMODE
Keyboard mode and read mode; read mode is enabled and printer is connected.

$\overline{L8EA}$
Output of ALU is shifted left eight places and end around.

LOADMAC/
Load micro memory address counter.

LP-APBP
Open for future application.

$\overline{LT}$
Selection of lower micro instruction by the T field.

LT + AP + BP
Either MIR16-MIR17 = 11 (T field contains NU, Z, COL, or Z*L operation) or A' coding or B' coding.

M FIELD
In micro instruction, mode (M) field specifies addressing mode to be used to obtain next micro instruction pair from micro memory.

M100 - M115
Mask 1 register output.

M200 - M215
Mask 2 register output.

MA REGISTER
Micro-memory address register. Holds micro-memory address of current micro-instruction pair.

MA TRANSFORM
Micro-memory address transform.

MAC
Memory address counter. Holds address of next sequential micro-instruction pair.

MACRO MEMORY
Core or MOS memory used by processor for storage of operands, etc.

MAINTMODE
Not used.

MANINTSTATUS
Manual interrupt status; indicates to the CPU that a manual interrupt has occurred.

MAXFORM
Memory address transform operation.

MAXT + LTP
Either MA transform operation or MIR16-MIR17 = 11 (T field contains NU, ZL, COL, or Z*L operation).

MC-1
Master clear 1; internal I/O-TTY controller master clear derived from the applied external MC signal.

MC-S/
Master clear request condition sent to control 1 to initiate a master clear.

MC-SW-NC, MC-SW-NO
When connected, provides for initiating the master clear condition that returns CPU to the initialization routine (NC = normal closed; NO = normal open; requires momentary double-pole single-throw contact arrangement).

MC/
A discrete signal developed by the CPU whenever a processor master clear occurs; it is used to clear the I/O channels and external equipment.

MCDELAYED/
A delay of master clear until the protected areas have been inhibited from clearing operation.

MCSM
Master clear of status mode.

MCSM1 - MCSM4
Master clear applied to status mode registers 1 and 2.

$\overline{MEM}$
Initiates extend timing for macro memory request or I/O operation.

MEMCYCLE/
Generated whenever a read or write macro memory operation is decoded.

MEMPAR/
Memory parity bit to CPU.

MI-SW-NC, MI-SW-NO
When connected, provides applicaton of CPU manual interrupt (NC = normal closed; NO = normal open; requires momentary double-pole single-throw contact arrangement).

MICRO INSTRUCTION
32-bit instruction from micro memory that controls all operations throughout the computer system.

MICRO MEMORY
High-speed semiconductor memory that contains micro programs.

MICRO PROGRAM
Set of micro instructions stored in micro memory.

MICROHALT
Enable the stop logic to initiate the time generator stop condition.

MICROMODE
Micro-memory mode.

MICRORUN
Micro running.

MICROSTOP/
Micro stop.

MICROSTRT/
I/O-TTY controller logic that enables microstart when panel interface is not supplied.

MIR
Micro-instruction register; holds micro instruction being executed.

MIR00 - MIR31
Micro-instruction register output bits 00 through 31.

MIR12
Micro instruction register bit 12 provides selection response control (character input, reply, reject, RTERM) responses to the CPU via the three-state bus.

MIR31 = Q00/
MIR31 or Q00 are to enter the least significant bit position of selector S3 during shift left one place operation.

MM00 - MM31
Micro memory output bits 00 through 31.

MMGATEX
Micro memory clock to X register.

$\overline{\text{MMSTBMA}}$
Micro memory strobe of memory address register.

MMSTBPAGE
Micro memory strobe of page register.

MMWE
Micro memory write enable for lower and upper 16 bits.

MMWL16/
Micro memory write lower 16 bits.

MMWRITE
Enable micro memory write to X register.

MMWU16/
Micro memory write upper 16 bits.

MR/
Originates in the CPU as a master clear and clears I/O controllers and external equipment.

MULTIND/
Panel simulation selection of indirect addressing capabilities.

MULTINDPNL/
Breakpoint controller switch that provides for jump from the present address to auxiliary addresses.

$\overline{\text{N = 0}}$
N register content not equal to zero.

N = 0/
N register equals 0.

N REGISTER
Eight-bit counter that can be cleared, incremented, or decremented under micro-instruction control; also used to address file 2.

N SHIFT
Clock of N register during shift/scale operation.

N0/ - N7/
N register output.

$\overline{\text{N3-1}}$
Enable file 2 of ALU.

$\overline{\text{NCLK}}$
Clock N register.

$\overline{\text{NLOAD}}$
Load N register.

ODDCLK
Clock input pulse chain to time generator.

OSC 4.9152 MHz
Fundamental oscillator frequency source from which the baud rate generator, timing generator, and real-time clock rates are derived.

P REGISTER
General-purpose register that can be used to hold macro-memory address of software instruction being executed if processor is configured as an emulator.

PE
Parity error; parity bit indicates received character code word is incorrect.

P00 - P15
Output from P register to selector S1.

PG0 - PG3
Micro-memory page address register output.

PN-TTL-OUT
Provide for transmission of keyboard character code via TTL levels to the breakpoint controller.

PNBAUD00/, PNBAUD1/
Baud rate clock frequency selection for the panel interface.

PNL-BAUD0, PNL-BAUD1
Baud rate selection for breakpoint controller baud rate generator.

PNL-TTLOUT
Breakpoint controller TTL level output signal.

PNLKEYINT/
Breakpoint controller keyboard (RDINT12/).

PNLMODENA
Allows entering into the reserve (panel) mode (by pressing the escape button on the TTY/CD) when high, and prevents entering into the reserve (panel) mode when low. When no connection is made to this signal, entering into the reserve mode is allowed.

PNLPRES
Breakpoint controller present; indicates whether the breakpoint controller is inserted.

POWER FAIL/
Signal generated by power supply whenever it detects power interruption of more than a half cycle.

PROG-PROT/
Indicates to I/O controllers that I/O instruction is protected.

PROP0/ - PROP1/
Carry propogate generated from look-ahead carry generator.

PROP01/ - PROP31/
Carry propagates generated from arithmetic/logical units.

PROP01/generated from ALU00 through 03 unit
PROP11/generated from ALU04 through 07 unit
PROP21/generated from ALU08 through 11 unit
PROP31/generated from ALU12 through 15 unit

PROT
Protect; condition requires that the response to the CPU affect a protected area of memory.

PROTECT/ (PROT-PNL1)
Protect switch to the transform.

PROTECTPNL/
Enables CPU protect system when panel interface is present.

PS
Page storage register.

PSIM-DRR
Panel simulation data receive reset; reset to UART receive register to enable input of new character code.

PSIM-ECHO
Breakpoint controller (panel simulation) echo; selection of echo response to output device in a panel mode operation.

PSIM-TBRL
Panel simulation transmit buffer register load; loading of UART transmit buffer register during panel simulation mode.

PUA10-08
Pull up (high) location A10 pin 8.

PUB3-01
Pull up (high) condition applied to ALU registers.

PUH10-08
Pull up (high) location H10 pin 8.

PUL3-05
Pull up (high) condition applied to ALU registers.

PUL3-07
Pull up (high) condition applied to ALU registers.

PULL-UP
High condition applied to clear input of RTJ register and count enable of MAC counter.

Q REGISTER
General-purpose register used in multiply and divide operations.

Q* REGISTER
Register included in systems that contain hardware double-precision option.

Q-LSB-DP
Not used.

Q00 - Q15
Q register output bits 00 through 15.

QLSBSL
Data to enter during Q register shift left.

QMODES0 - QMODES1
Q register mode control.

QMODES0', QMODES1'
Q-register mode select control signals.

QMSBSR
Data to enter during Q register shift right.

QSHFTCLK
Shift clock to Q register to effect left shift of data.

RD01/ THROUGH RD16/ (RD01 = LSB)
Transfer of peripheral I/O controller data and status to CPU three-state bus.

RDINT001/ - RDINT15/
Data interrupts at the micro interrupt level.

RDINT01/
Interrupts CPU whenever TTY is ready to transfer a byte of data during auto-data transfer.

RDINT08/
Micro level interrupt that occurs every 3.33 milliseconds when real-time clock is enabled.

RDINT12/
Micro level interrupt that occurs whenever a panel key is pressed during panel simulation operation.

RDINT13
Breakpoint controller transmit buffer register empty interrupt (PNLTBRINT).

RDINT13/
Interrupt to the SMI to indicate the TTY/CD controller UART transmit buffer register is empty during panel simulation.

READ
Command from CPU requesting an input response.

READ-SSTB/
A strobe signal for the M05 peripheral controller; occurs twice with each execution of set/sample input/output (SIO) command; occurs once near the beginning and once near the end of an output data operation.

READCYCLE
Signal generated whenever a read macro memory or read modify write operation is decoded.

READMODE
Indicates the read mode condition of I/O controllers has been enabled.

REJECT
If the specified operation cannot be performed by the I/O when the read or write signal appears, a reject response is sent to the CPU within 10 microseconds. If no I/O response (reply or reject) occurs within 13 microseconds after the request, the emulator generates an internal reject.

RELEASED
Provide for release of reserved condition, panel mode.

REPLY
Indicates to the CPU that the associated I/O device is available to send or receive data.

Write event sequence:

1. CPU bus (internal A/Q channel) transfer 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 microseconds later.
3. The A/Q channel drops the write signal when the reply is received.
4. Absence of a write signal drops the reply.

Read event sequence:

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

RESERVED
Provides for breakpoint controller panel mode operation.

RESTART
Restart odd/even time generator.

$\overline{\text{RESUME}}$
Resume operations.

RESYNC I/O
General-purpose 56-nanosecond strobe (same as T4) to resync the response signals with the CPU timing by strobing the signals into a 74175 D-type latch.

ROM
Read-only memory.

RPINT16/ - RPINT31/
Program interrupts at the macro interrupt level.

RPINT17/
Macro level signal that signals end-of-operation alarm, or manual interrupt.

RPINT24/
Macro level signal that signals termination of the real-time clock operation.

$\overline{\text{RPT}}$
Repeat operation if N is not equal to 0.

RR1 THROUGH RR8
Receiver register data bits 1 through 8; character code data bits from UART to the CPU and internal I/O-TTY controller logic.

RS232IN-1, RS232IN-2
Transmission of CD or breakpoint controller character codes (ASCII) to CPU via I/O-TTY controller; signals must conform with EIA standard RS232-C and CCITT recommendation V24.

RS232OUT-1
Transmission of charactr codes from CPU to CD or breakpoint controller; signals must conform to EIA standard RS232-C and CCITT recommendation V24.

RTCSEL
Real-time clock selected; enables real-time clock logic.

RTERM
Receive termination.

RTJ
Return jump register; holds micro memory address for return of control after completion of subroutine.

RTJ0 - RTJ7
Return jump register outputs.

S FIELD
In micro instruction, special (S) field specifies operation to be performed in parallel with ALU operation.

S1, S2
Selector 1, selector 2, and so forth.

S100 - S115
Selector S1 output.

$\overline{\text{S100}}$ - $\overline{\text{S115}}$
Selector S1 inputs to ALU.

S1S0-1 - S1S2-1, S1S0-2 - S1S2-2
Selection code signals for A input to ALU from selector S1.

S1S0/ - S1S2/
Selector indicates that the processor is in adder split mode and a subtract operation is decoded in the F field (MIR04 = 1).

S2 = KN
Selector S2 contents equals contents of K and N registers.

S2 = SHIFT
Signal generated whenever MIR25-MIR27 - 111 indicating a shift operation via the arithmetic/logical unit is decoded in the C field.

S200 - S215
Selector S2 output.

$\overline{\text{S200}}$ - $\overline{\text{S215}}$
Selector S2 inputs to ALU.

S2S0-1 - S2S2-1, S2S0-2 - S2S2-2
Selection code signals for B input to ALU from selector S2.

S2S0/ - S2S2/
Selector S2 select lines.

S3*S0, S3*S1
S3 shift operation control.

$\overline{\text{S300}}$ - $\overline{\text{S315}}$
Selector S3 outputs fed back to P, A, F, X, and G registers.

S300/ THROUGH S315/
Data input lines from CPU arithmetic logic unit (ALU) S3.

S3S0 - S3S1
Selector S3 select lines.

S3S0-1, S3S1-1
Selection code signals for selector S3.

S5-0/ - S5-7/
Selector S5 output.

S5-S0 - S5-S3
Selector S5 select line.

S6CONS1
Select signal required to generate selector S6 select lines S6S0 and S6S1.

S8 ENABLE
Selector S8 enable line.

S8-0 - S8-7
Selector S8 output.

SAMPINTS/
Clock for interrupt holding flip-flops.

$\overline{\text{SAVEAB}}$
Save A and B of micro instruction.

$\overline{\text{SAVET}}$
Save T field of micro instruction.

SC+SHIFT
Scale or shift operation.

SCALE
Scale operation.

SD01/ THROUGH SD16/ (SD01/ = LSB)
Transfer of director functions or data to peripheral I/O controllers.

SELECTOR
Multiplexer that allows one of several sources of data to be selected for transfer from one location in processor organization to another under micro instruction control.

SELGETMAK/
Micro level signal used for step mode during panel simulation.

SELSKIP/
Selects the skip condition fo the selective skip instruction.

SELSKIPNL/
Selective skip from breakpoint controller.

SELSTOP/
When active, causes the CPU to halt when a selective stop instrucion is encountered.

SELSTOPPNL/
Selective stop from breakpoint controller.

SELTTYSTATUS
Select TTY status; enables the I/O-TTY controller status to be placed on the CPU three-state bus.

SET SM204/
Initiates the deadstart routine.

SETSM104/ - SETSM115/
Set status mode register 1 bits 04 through 15.

SETSM204/
Set status mode bit 204 (deadstart loading).

SETSM204/ - SETSM215/
Set status mode register 2 bits 04 through 15.

SHIFTCLK
Shift clock.

SM
Status/mode register.

SM104 - SM115
Status mode register 1 output.

SM113/ MM/
Designates read from the micro memory at the address determined by the N/K register.

SM204 - SM215
Status mode register 2 output.

SM204/
Micro-level signal that selects deadstart baud rate selection for the panel interface.

SM210
Status mode bit 210 (initiate panel simulation data ready reset for UART).

SM211/
Status mode bit 211 (initiate panel simulation transmit buffer loading of UART).

SM212/
Status mode bit 212 (initiate panel simulation panel mode).

SMB07/ THROUGH SMB09
Transfer of mode bits to M05 device. Applicable to M05 set/sample peripheral controllers only.

SPLIT ADDER
Optional process by which ALU can be functionally split into two independent ALUs under micro instruction control.

SPOS01/ THROUGH SPOS03/
Bits that spcify the position of the M05 device on the port; up to eight multiple devices may be on the same port. Applicable to M05 set/sample peripheral controllers only.

SPT0 THROUGH SPT7
Send port line. Designations applicable to M05 set/sample capability only when custom-wired.

SPT00/ THROUGH SPT07/
M05 port selection signals (hexadecimal code selection of port address lines); the active line is determined by the E bits of the Y-register address word. These lines are custom-wired to the associated M05 controller port at time of installation. Applicable to M05 set/sample peripheral controllers only.

SSEL0 THROUGH SSEL7
Send select line. Designations applicable to M05 set/sample capability only when custom-wired.

SSEL00/ THROUGH SSEL07/
M05 peripheral device selection in accordance with ADT operations. Applicable to M05 set/sample peripheral controllers only.

STATUS/MODE REGISTER
Contains flag bits and status mode bits. Flag bits are set under micro-instruction control to enable certain internal processor operations. Status/mode bits indicate internal or external conditions.

STERM/
A signal that the emulator generates when the last transfer of a current automatic data transfer (ADT) opertion is in progress.

STOP
Inhibits time generator.

STROBE-D
D-register strobe to check data from CPU ALU to D register flip-flop.

(SUB)(ADDER SPLIT)/
Signal indicates tha the processor is in adder split mode and a subtract operation is decoded in the F field (MIR04 = 1).

T = 0/
The T field of micro instructon equals zero.

T FIELD
In micro instruction, test (T) field specifies whether the upper or lower micro instruction of the next micro-instruction pair is to be executed.

T0 - T7
Processing timing.

T1, T2, T3
Timing pulses produced by internal timing generator.

T2ODDCLK/
Odd clock trigger available at time T2.

T3/ • MM/
Memory enable at time T3.

T4 • T5/
Gate memory instruction register at time T4.

TBREMPTY (TREMPTY)
The UART transmit buffer register is empty.

TERM
Terminates RTC sequence and TTY busy status.

TERMINATE SCALE
Terminate scale operations.

TRANSFORM MATRIX
Selects bits from various sources in processor organization and translates them into micro-memory address in MA register, or transfers them to K or N register.

TTL-SRL
TTL serial input.

TTL-SRL-1
Transmission of MOS I/O device data to the I/O-TTY controller from the CPU.

TTL-SRL-2 (TTL-SRL2/)
TTL serial output. Transmission of MOS data from I/O-TTY controller to the CPU.

TTY-ADT-INT/
I/O-TTY controller auto-data transfer interrupt (RDINT01).

TTY-DRR
TTY data receive reset; initiates the UART data receive reset.

TTY-TBRL
TTY transmit buffer register load; initiates loading of the UART transmit buffer register.

TTYSEL
TTYY select; initiates the response from the I/O-TTY controller.

TTYUARTOUT
Transmission of teletypewriter character codes (ASCII) over a 20 mA current loop from I/O-TTY controller to teletypewriter.

UART-IN
Universal asynchronous receiver transmitter input; input to the UART from the teletypewriter, CD, or breakpoint controller.

(W=0)(EQ=1)
W equals zero and E equals one; conditions derived from Y register address bits.

WAITRESUM
Inhibit resume.

WE0/
W equals zero (Y-register W field hexadecimal value of an address word). This signal is present whenever bits 12 through 16 of the address in the Y register are all low.

WEF1
Write/read enable for file 1; high = write, low = read.

WEF2
Write/read enable for file 2; high = write, low = read.

WRITE
Initiates an output transfer of one data word from the ALU A register to the I/O-TTY controller D register.

WRITE32BIT/
Available for 32-bit machines.

WT-MODE
The write mode condition of peripheral I/O controllers has been enabled.

X REGISTER
General-purpose processor register.

X* REGISTER
Register included in systems that contain hardware double-precision option.

X00 - X15
Output from the X register.

XFORMPAGE
Transform strobe of page register.

XFORMSTBMA
Transform strobe of memory address register.

XTBLKT4/
This signal disables the D field decoders at normal time T4 during a GITMAK/xt transform operation since the destination register is already gated by CPU-MDS.

(XXXX X111)
RR1, RR2, RR3 are high (1s) and RR4 through RR7 are immaterial.

Y01 through Y16
Address word bits from Y register used by internal I/O-TTY controller logic.

1.2288 MHz
Basic clock frequency for generating the real-time clock 3.3 millisecond time pulses.

COMMENT SHEET

MANUAL TITLE: ®CDC Basic Micro-Programmable Processor
Hardware Reference/Maintenance Manual

PUBLICATION NO.: 39451400 REVISION: E

NAME: _____

COMPANY: _____

STREET ADDRESS: _____

CITY: _____ STATE: _____ ZIP CODE: _____

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

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

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Technical Publications Department
2401 North Fairview Avenue
St. Paul, Minnesota 55113

CUT ALONG LINE

FOLD

FOLD

03377