

CONTROL DATA® 5600 SERIES OF MICROPROGRAMMABLE PROCESSORS

REFERENCE MANUAL



CONTROL DATA® 5600 SERIES OF MICROPROGRAMMABLE PROCESSORS

REFERENCE MANUAL

REVISION RECORD			
REVISION	DESCRIPTION		
8-25-72	Preliminary edition.		
<u> </u>			
;			
-			
Publication No.			
14232000			

©1972 by Control Data Corporation Printed in the United States of America Address comments concerning this manual to:

Control Data Corporation
Minneapolis Military Products Division
3101 East 80th Street
Minneapolis, Minnesota 55440
Attn: Publications Department

or use Comment Sheet in the back of this manual.

PREFACE

Microprogramming was proposed as early as 1951 by M. V. Wilkes as a method of generalizing the design of the control section of a digital processor. Only recently, however, have advances in the design of medium scale and large scale integrated circuits and the availability of fast, compact, and inexpensive semiconductor memories made the construction of a microprogrammable processor (MPP) feasible.

During 1970 and 1971, Control Data carried on intensive MPP development effort. This effort resulted in the present CONTROL DATA $^{\odot}$ 5600 series of MPPs, which is described in this manual.

14232000 iii/iv

CONTENTS

Section		Page
1	GENERAL DESCRIPTION	
	Introduction	1-1 1-6 1-8 1-8 1-10 1-14
2	MICROPROGRAMMABLE PROCESSOR	
	General Description	2-1
	Microinstruction Format	2-1 2-4
	Detailed Description	2-6
	Micromemory and Microcontrol Section	2-6 2-9 2-9 2-10
3	MAIN MEMORY	
	Small Plane Memory	3-1 3-3
4	INPUT/OUTPUT	
5	MICROINSTRUCTIONS	
	Microinstruction Formats Detailed Microinstruction Coding	5-1 5-5
	M Field (Bits 0 and 1)	5-5 5-6 5-12

14232000 v

CONTENTS (CONT.)

Section		Page
	B Field (Bits 10 Through 12)	5-14 5-17 5-21 5-24 5-24 5-26
6	SPECIAL FEATURES AND OPTIONS	
	Status/Mode Register	6-1
	MPP Operating Modes	6-3 6-5 6-7
	Bit Test	6-8 6-9
	Transform Specification	6-10 6-12 6-15
	Data Formatting Register Files and File Addressing Interrupts Multiple Memory Accesses Memory Protection Memory Parity Double Precision Logic Dead Start Breakpoint Micromemory Operand References Micromemory Load Option 6-23	6-15 6-15 6-16 6-18 6-19 6-20 6-20 6-21 6-22 /6-24
7	MICROINSTRUCTION TIMING	
	Microinstruction Classes	7-1 7-2
8	5600 PROCESSOR CONTROLS AND INDICATORS	

APPENDIXES

A	Powers of Two Table
В	Conversions
С	Hexadecimal Conversion Tables
D	Hexadecimal Arithmetic Tables
Е	1's Complement and 2's Complement Numbers

ILLUSTRATIONS

Figure		Page
1-1	Digital Processor Organizations	1-7
1-2	5600 System Ruggedized Enclosures	1-11
1-3	5600 Series Processor Logic Module	1-13
2-1	Simplified MPP Block Diagram	2-2
2-2	Detailed MPP Block Diagram	2-7
3-1	Basic SPM Block Diagram	3-2
3-2	SPM Stack Module	3-3
3-3	Basic LPM Block Diagram	3-4
3-4	LPM Core Board 3-5	
5-1	Microinstruction Formats	5-2
5-2	Double Precision Option Block Diagram	5-7
6-1	Sample Transform Design Sheet for 16-Bit MPP (Including Standard Transforms)	6-11
6-2	Example: 16-Bit Instruction Format	
6-3	Transform Design Example	6-12
6-4	· ·	6-13
0-4	Multi-Port Logic Block Diagram	6-18
7-1	Microinstruction Classification · · · · · · · · · · · · · · · · · · ·	7-2
8-1	16-Bit MPP Maintenance Panel	8-2

14232000 ix/x

TABLES

Table		Page
1-1 1-2	5600 Series Equipment • • • • • • • • • • • • • • • • • • •	1-1
1-2	Jood Beries Characteristics	1-2
5-1	M Field Operations · · · · · · · · · · · · · · · · · · ·	5-5
5-2	Logical Operations · · · · · · · · · · · · · · · · · · ·	5-8
5-3	Arithmetic Operations • • • • • • • • • • • • • • • • • • •	5-8
5-4	Shift Operations	5-10
5-5	Scale Operations · · · · · · · · · · · · · · · · · · ·	5-11
5-6	A Input Operations · · · · · · · · · · · · · · · · · · ·	5-12
5-7	A' Input Operations · · · · · · · · · · · · · · · · · · ·	5-13
5-8	B Codes · · · · · · · · · · · · · · · · · · ·	5-15
5-9	B' Codes · · · · · · · · · · · · · · · · · · ·	5-16
5-10	D Code Transfers · · · · · · · · · · · · · · · · · · ·	5-18
5-11	D' Code Transfers · · · · · · · · · · · · · · · · · · ·	5-19
5-12	D" Code Transfers · · · · · · · · · · · · · · · · · · ·	5-20
5-13	DD" Code Transfers • • • • • • • • • • • • • • • • • • •	5-20
5-14	T Addressing Modes · · · · · · · · · · · · · · · · · · ·	5-22
5-15	T' Addressing Modes · · · · · · · · · · · · · · · · · · ·	5-23
5-16	S Field Codes · · · · · · · · · · · · · · · · · · ·	5-24
5-17	C Code Actions · · · · · · · · · · · · · · · · · · ·	5-26
6-1	Standard MPP Bit Types	6-8
6-2	MPP Interrupt Addresses	6-17
7-1	Microinstruction Execution Times	7-1
8-1	MPP Controls and Indicators	8-3

14232000 xi/xii

GENERAL DESCRIPTION

INTRODUCTION

This manual is written to provide a general introduction to the CONTROL DATA® 5600 series of microprogrammable processors (MPPs) and core memory systems. The manual also contains information on the microprogramming of 5600 processors and the options available which allow a 5600 system to be configured to meet the requirements of a data processing application. For any specific 5600 system, this reference manual is supplemented by a separate customer engineering manual which specifies the options selected for that system and contains the information necessary for the installation, checkout, operation, and maintenance of the system.

Listed in Table 1-1 are the model numbers and the descriptions of the 5600 series equipment.

Model Number Equipment

5601 16-bit processor

5602 Power supply unit

5603 Large plane memory (LPM) system

5604 Small plane memory (SPM) system

5605 32-bit processor

TABLE 1-1. 5600 SERIES EQUIPMENT

A complete configuration identification number for a typical 16-bit processor, for example, is 560102A. 5601 designates the family of 16-bit processors, 02 identifies one specific design of the 5601 processor, and A designates the series, which is related to retrofit activity.

5600 series equipment is available in both ruggedized and militarized forms. The general characteristics of both types, plus some typical instruction execution times for 5600 processors, are given in Table 1-2.

14232000

TABLE 1-2. 5600 SERIES CHARACTERISTICS

PROCESSOR	
Туре	General purpose, microprogrammable digital processor
Organization	Register-oriented or file-oriented
Word length	Flexible: configured in 4-bit increments from 8 to 32 bits
Micromemory type	Semiconductor: read/write or read-only
Micromemory size	Configured in increments of 512 microinstructions up to 8192 microinstructions
Micromemory cycle time	80 nanoseconds
Arithmetic	Binary, with dynamic selection of 1's or 2's complement mode
MAIN MEMORY	
Requirement	Optional; required for computer configurations
Туре	 SPM: core storage, 3D, three-wire, configured in 4K x 9-bit increments (eight data, one parity) up to 16K x 18 bits
	• LPM: core storage, 2-1/2D, three-wire, configured in 16K x 9-bit increments (eight data, one parity) up to 262K x 36 bits
Cycle time	1.0 microsecond
Addressing modes	Determined by microprogram; flexible to suit application

1-2 14232000

TABLE 1-2. 5600 SERIES CHARACTERISTICS (Cont.)

INPUT/OUTPUT (I/O)	
Interfaces	• CONTROL DATA® 160-A
	• CONTROL DATA® 1700
	• CONTROL DATA® 3000 series
	Others to suit application
Controllers	• CONTROL DATA® 370 Paper Tape Reader
	• CONTROL DATA® 1711 Teletypewriter
	Others to suit application
MECHANICAL	
Basic hardware	Modular
Construction	Rack-mount enclosure standard; others to suit application
Power required	100 to 130 VAC, 47 to 440 Hz, single phase, 430 watts (typical 16-bit processor with 16K main memory)
Weight	
Processor (typical)	Approximately 50 pounds
• LPM (typical)	Approximately 70 pounds
Dimensions	
Processor enclosure (standard)	19W x 7 x 22D inches
• LPM enclosure (standard)	19W x 14H x 22D inches
1	

TABLE 1-2. 5600 SERIES CHARACTERISTICS (Cont.)

ENVIRONMENTS				
	Ruggedized	Militarized (per MIL-E-16400, Class 4)		
Temperature	• -35°C to +65°C, nonoperating	• -62°C to +75°C, nonoperating		
	• 0°C to +50°C, operating	• -5°C to +75°C, operating		
Shock	5g, 11 microseconds, perpendicular	50g, 11 microseconds, half sine wave		
Vibration	Normal transit loads	5-500 Hz, 5g, half sine wave		
Humidity	40 to 90 percent relative, no condensation	Up to 100 percent relative, with condensation		
EMI		MIL-STD-461, Class 1D		
Altitude	• 15,000 feet, nonoperating	• 50,000 feet, nonoperating		
	• 6,000 feet, operating	• 10,000 feet, operating		
Sand and dust		MIL-STD-108		
Salt spray		FED-STD-151, Method 811.1		
Acoustical noise		MIL-STD-470		
Mean time between failures:				
(16-bit processor with 8K SPM)	3200 hours	4000 hours		
Mean time to repair:				
(16-bit processor with 8K SPM)	30 minutes	30 minutes		

TABLE 1-2. 5600 SERIES CHARACTERISTICS (Cont.)

EXECUTION TIMES FOR COMPUTER EMULATOR APPLICATION

Times given are in microseconds for a typical 5601 16-bit computer emulator. Note that other configurations may result in different execution times.

Instruction	Execution Time	
<u>instruction</u>	(Microseconds)	
Load register relative	2.0	
Store register relative	2.0	
Replace add relative	3.0	
Unconditional jump relative	1.3	
File jump decrement	1.5	
Enter register	1.0	
Increase register	1.2	
Register add	1.3	
Register subtract	1.3	
Register multiply	5.2	
Register divide	9.4-11.3	
Register transfer	1.5	
Long load register*	3.0	
Long store register*	3.0	
Long multiply*	10.5-12.4	
Long add*	3.0	
Long subtract*	3.0	

^{*32-}bit instruction with 16-bit memory address

MICROPROGRAMMING CONCEPTS

The concept of microprogramming as applied to digital processor design can best be understood by examining Figure 1-1, which shows a conventional organization of a digital processor and a microprogrammable organization of the same processor. The block sizes in the figures represent the relative complexity, design difficulty, and cost of implementation for each section of the computer.

The control section of the conventional processor is the most costly and difficult element to design since the various operations necessary for execution of the instructions are relatively complex. The usual result is a one-time-only design of this section of the processor.

In contrast, the MPP has replaced the control section with two units:

- A micromemory which contains the instruction execution steps
- A simplified microcontrol section of the processor, which performs small steps in the execution of a total processor instruction

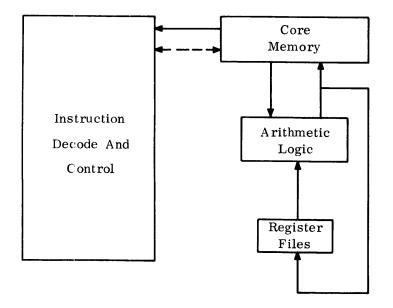
The sequencing of the steps for the execution of a complete instruction is provided by reading a sequence of microinstructions from the high speed micromemory.

The advantages of a microprogrammable approach to the design of computers and specialized digital devices are:

- Simplified design of the device
- The same hardware components can be used for different devices or applications
- Low development costs for new computer architectures and specialized digital devices

1-6 14232000

CONVENTIONAL PROCESSOR ORGANIZATION



Dotted lines are control signals.
Solid lines are instructions and data flow.

MICROPROGRAMMABLE PROCESSOR ORGANIZATION

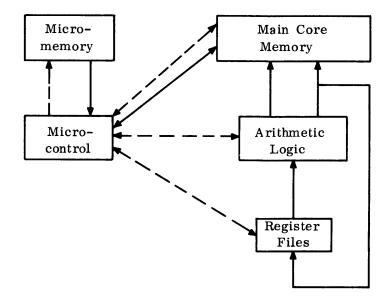


Figure 1-1. Digital Processor Organizations

EXAMPLES OF MICROPROGRAMMING APPLICATIONS

The microprogrammable approach is not limited to the design of computers, but also is used to design any digital device that has a complex control operation. Some applications are:

- Emulation of existing computers
- Design of new computers
- I/O controller and pre-processor for another computer
- Device controller, such as a disc controller and magnetic tape controller
- Specialized algorithm processors, such as a fast Fourier transform (FFT) device

5600 SERIES FUNCTIONAL DESCRIPTION

The Control Data 5600 series contains processors, core memory systems, operator interfaces, I/O interfaces, and power supplies. The main element in the series is the flexible configuration MPP. The term MPP as used in this manual refers to the microprogrammable processor with all its expanded capabilities, although a specific MPP in any given application will have only those capabilities required by that application. All MPPs share the same general logical organization at the microprogram level, but variations are provided from the following list to allow an MPP to meet the requirements of a specific application.

- Word size from 8 bits to 32 bits in 4-bit increments
- Total micromemory capacity of from 512 instructions to 8192 instructions
- None, one, or two sets of file registers
- 16, 32, 48, or 64 interrupts
- 16, 32, 48, or 64 status mode bits
- Optional double precision hardware
- Core memory storage capacity of from 4096 to 262,144 words (or more, if required), limited only by addressing
- Other options as listed later in this manual

1-8 14232000

All of the options required for a particular MPP must be specified at the time of design of the MPP, at least to the extent of specifying the minimum and maximum ranges of the options.

The architecture of the MPP allows simple emulation of many digital computers, but the architecture also allows the efficient design of other digital devices that have complicated logical requirements such as disc controllers, magnetic tape controllers, I/O channel controllers, and specialized algorithm devices.

The basic architecture provides a minimum arithmetic capability in the MPP at the microprogram level, provides the ability to transfer data between a number of registers, and allows branching decisions based on the sign or zero or nonzero condition of registers. The order of execution of these simple capabilities is under control of the microprogram, which allows the development of complicated operations by the combination of these simple operations.

The main difference between an MPP and a fixed-operation device (computer, controller, etc.) is in the organization of the control section of the MPP. In a fixed-operation computer, for example, the steps that make up a software instruction (gating a quantity from one register in the processor to another register, incrementing a counter, or initiating a memory read or write cycle) are performed in a fixed time sequence controlled by hardwired logic. In the MPP, these steps are controlled by the bits in a series of microinstructions that comprise the microprogram for that instruction. The microinstructions are executed at a rate of one microinstruction every 168 nanoseconds, so the five or six microinstructions required for a complicated software instruction can be executed in one microsecond, the normal memory cycle time. In addition, the microinstructions of the MPP provide for parallel execution of several operations during each microinstruction.

For example, during one microinstruction execution, up to four operations from the following groups can be executed at one time.

- Arithmetic, logical, shifting, or scaling operations on registers in the MPP
- Initiate a main memory read or write
- Initiate an input or output operation, control counters, and set control registers
- Select the next microinstruction to execute (based on the result of an arithmetic or logical operation) upon detection of an interrupt or the presence of some other processor condition

14232000

The microinstructions which comprise the microprogram are stored in the semiconductor micromemory, which has a cycle time of 80 nanoseconds. This micromemory is available as a read/write memory, which is used in MPPs that are under initial development of a microprogram or in applications that require dynamic changing of the microprogram. For fixed-application MPPs, the micromemory is available as a read-only memory, which has the advantage of nonvolatile storage of the microprogram.

For 5600 systems requiring core memory storage capacity, the system contains either an SPM or an LPM. The SPM is made up of pluggable modules identical in size to the processor modules. The major components of the LPM are contained on pluggable modules approximately 17 inches wide by 17 inches deep. The SPM is configured in 4096-word increments up to 16,384 words. The LPM is configured in 16,384-word increments up to 262,144 words. Both types of memory provide a basic access time of 400 nanoseconds maximum and a basic cycle time of 1.0 microsecond maximum.

5600 series I/O capabilities include Control Data 160-A, 1700, and 3300 series channel interfaces and Control Data 370 paper tape reader and 1711 teletypewriter controllers. Other interfaces and controllers are designed for specific applications.

Operator interfaces include as standard a full-function operation and maintenance panel or a basic function panel.

5600 SERIES PHYSICAL DESCRIPTION

Because the components of a 5600 system are modular, the system can be configured to accommodate almost any physical requirement. Typically, the system is packaged in one or more 19-inch rack-mountable enclosures. Figure 1-2 shows a ruggedized 16-bit 5600 system contained in two rack-mountable enclosures. The top enclosure contains the central processing unit (CPU) (including micromemory, control, arithmetic, and I/O logic) and a power supply providing +5 VDC and -5 VDC. The bottom enclosure contains the 16,384-word SPM core memory, its associated logic, and a power supply providing +10 VDC and +30 VDC. In this system, +5 VDC and -5 VDC are supplied to the logic in the main memory enclosure by the supplies in the CPU enclosure through a cable.

Each 5600 system contains a power monitoring system, which activates a processor interrupt when input AC power drops below a predetermined level. This system is designed so that the processor has time to store the contents of the working registers and register files in main memory before logic power drops below the level where reliable circuit operation is assured.

1-10 14232000





Figure 1-2. 5600 System Ruggedized Enclosures

5600 series logic for the processor and SPM is packaged on small, rugged, pluggable modules, one of which is shown in Figure 1-3. The modules are designed in general to concentrate all the logic associated with one 5600 function on one module in order to simplify system maintenance. Most 5600 systems are constructed using the standard 5600 series modules for the control, arithmetic, and memory functions; specific applications may require nonstandard logic modules for special I/O interfaces or other functions. The modules are 6.75 inches wide by 3.75 inches high, with a 112-pin connector. The logic modules plug into a connector baseplate. Interconnections among modules are provided on the connector baseplate by a computer-controlled wirewrapping process.

No special cooling provisions are needed for most 5600 system applications. In the ruggedized system shown in Figure 1-2, for example, forced air is used to cool all electronic components. Each enclosure has its own independent, forced-air blower system. These cooling systems enable each enclosure to be maintained at its proper operating temperature while undergoing maintenance. The memory enclosure draws air in through a filter on the front panel and passes it directly over the component boards; the air is then exhausted from the rear of the enclosure. The processor enclosure has a blower system that draws air from inlets at the bottom and forces it through a center plenum in the enclosure; the air then passes over the printed-circuit cards and is exhausted through the sides of the enclosure.

1-12

Figure 1-3. 5600 Series Processor Logic Module

14232000

GLOSSARY

A field In microinstruction, A field specifies source of operand to be sent

to ALU from selector 1.

A register General purpose register.

A* register Register included in systems containing hardware double precision

option.

ALU Arithmetic and logical unit. Performs arithmetic and logical

operations on two operands sent to it from the two selectors.

B field In microinstruction, B field specifies source of operand to be

sent to ALU from selector 2.

BG Bit generator. Allows a word to be sent to ALU with all zeros

except one bit set at any bit position; used for masking or

arithmetic operations.

C field In microinstruction, constant (C) field. Can contain constants,

micromemory addresses, or other codes, depending on format

of microinstruction.

CPU Central processing unit. Consists of micromemory, control

section, arithmetic section, and I/O.

D field In microinstruction, destination (D) field. Specifies destination

for results of operation performed by ALU.

Dead start Optional logic that allows read/write micromemory to be loaded

from external input device.

Emulation Process combining hardware and firmware design, by which one

processor (emulator) executes programs designed for different processor, even though one-to-one hardware correspondence does

not exist.

F field In microinstruction, function (F) field. Specifies operation to

be performed by ALU or shift or scale of A or AQ registers.

F register General purpose register.

File 1 Register file addressed by contents of K register.

File 2 Register file typically addressed by contents of N register.

Firmware General term for combination of microinstructions used in micro-

program to perform a certain operation.

I register General purpose register that can be used to hold software

instruction during execution if MPP is configured as emulator.

IC Integrated circuit.

I/O Input/output.

K register 8-bit counter that can be cleared, incremented, or decremented

under microinstruction control. Also used to address file 1.

LPM Large plane memory.

M field In microinstruction, mode (M) field specifies addressing mode to

be used to obtain next microinstruction pair from micromemory.

MA register Micromemory address register. Holds micromemory address of

current microinstruction pair.

MA transform Micromemory address transform.

MAC Memory address counter. Holds address of next sequential

microinstruction pair.

Main memory Core memory used by MPP for storage of operands, etc.

Mask register Used to control processing of internal and external interrupts.

Microinstruction 32-bit instruction from micromemory that controls all operations

throughout 5600 system.

Micromemory High speed semiconductor memory, which contains microprograms.

Microprogram Set of microinstructions stored in micromemory.

MIR Microinstruction register. Holds microinstruction being executed.

MM Micromemory.

MPP Microprogrammable processor. CPU portion of any specific

5600 system.

N register 8-bit counter that can be cleared, incremented, or decremented

under microinstruction control. Also used to address file 2.

14232000 1–15

P register General purpose register that can be used to hold main

memory address of software instruction being executed if MPP

is configured as emulator.

Program protect Optional logic which, when enabled, prevents unprotected programs

and I/O users from changing contents of protected areas of main

memory.

Q register General purpose register used in multiply and divide operations.

Q* register Register included in systems containing hardware double

precision option.

RTJ register Return jump register. Holds micromemory address, to which

control will return at completion of a subroutine.

S field In microinstruction, special (S) field specifies operation to be

performed in parallel with ALU operation.

S1, S2, etc. Selector 1, selector 2, etc.

Selector Multiplexer that allows one of several sources of data to be

selected for transfer from one location in MPP organization to

another under microinstruction control.

SM Status/mode register.

Split adder Optional process by which ALU can be functionally split into

two independent ALUs under microinstruction control.

SPM Small plane memory.

Status/mode register Contains flag bits and status/mode bits. Flag bits are set

under microinstruction control to enable certain internal MPP operations (e.g., double precision flag set enables double precision logic). Status/mode bits indicate internal or external

conditions (e.g., memory parity error).

T field In microinstruction, test (T) field specifies whether upper or

lower microinstruction of next microinstruction pair is to be

executed.

Transform matrix Selects bits from various sources in MPP organization and

translates them into micromemory address in MA register, or

transfers them to K or N register.

X register General purpose processor register.

X* register Register included in systems containing hardware double

precision option.

1-16 14232000

GENERAL DESCRIPTION

Figure 2-1 is a simplified block diagram of the 5600 series MPP. The MPP basically provides for operations on the contents of the six registers (I, P, A, X, Q, and F) and the contents of the register files. These operations are performed in the arithmetic and logical unit (ALU), and the results are transferred back to the registers or files.

The functions of the processor are controlled by the current microinstruction in the microinstruction register (MIR). The source of the microinstruction is the micromemory (MM). If the micromemory is read-only, the complete set of microprograms (called the firmware) is copied into the micromemory during manufacture. For read/write micromemories, optional dead start logic allows the firmware to be loaded into the micromemory from an external input device.

MICROINSTRUCTION FORMAT

The general microinstruction format and the number of bits in each field are illustrated below. Each field is described briefly in the following paragraphs. Section 5 of this manual contains a detailed description of the microinstruction fields and codes.

2	5	3	3	3	3	5	3
M	F	A	В	D	Т	S	С

14232000 2-1

Figure 2-1. Simplified MPP Block Diagram

The M field of the microinstruction is used to specify the mode of selecting the next microinstruction pair; it is also used to specify the interpretation of parts of the rest of the microinstruction.

The M field specifies:

- Sequential addressing, in which the next sequential microinstruction pair will be read out of micromemory
- Jump addressing, in which the C field of the current microinstruction provides the address of the next microinstruction pair
- Return jump (RTJ) addressing, in which the RTJ register contains the address of the next microinstruction pair.
- Both sequential addressing and the use of the contents of the C field to set up the K or the N register.

The T field is used to select the specific microinstruction from the next microinstruction pair to execute. In general, the following are the conditions which can be selected by the T field.

- Unconditionally, the T field can select the lower of the current microinstruction pair (this overrides the M field selection) or the upper or lower of the next microinstruction pair.
- The result of the ALU operation can be tested for negative, zero, or carryout.
- The least significant bit of the Q register can be examined (for use in a multiply algorithm).
- The zero condition of the K or N register (control register) can be examined.
- Existence of interrupts, reject on I/O reference, and general test conditions (bit test) can also be checked.

The F field specifies the arithmetic or logic operation to be performed in the ALU on the two inputs specified in the A field and the B field. The result of the operation can be shifted and transferred to the location specified by the D field. As an alternate operation, the F field specifies a shift to take place in the A or AQ register that does not use the ALU.

The A field specifies the register or information source to be gated to the output of selector 1 (S1) and thus be available as the A input to the ALU.

The B field specifies the register or information source to be gated to the output of selector 2 (S2) and thus be available as the B input to the ALU.

14232000 2-3

The D field specifies the destination of the ALU output, or alternately, the destination of information provided to S1.

The S field specifies an operation to take place in parallel with the operation in the ALU. Specifically, the S field allows read and write commands to be given to the main core memory. Additional codes in S allow expansion of the number of sources and destinations which can be specified in the A, B, and D fields.

The C field is a multipurpose field. It is normally used to specify the address of the next microinstruction pair when a jump is specified in the M field; it may also be used like the S field to specify an additional operation to take place in parallel with the operations specified in the F and S fields.

MPP ORGANIZATION

The I, P, A, Q, X, and F registers shown in the register block of Figure 2-1 are all word length registers (the word size which was selected for the particular 5600 system application). They are all general purpose, but the names of the registers have been selected to suggest the typical use of each register. The general characteristics of the registers are described in the following listing.

- The I register is unique in that information is stored in it from the output of S1 instead of from the ALU output. It is normally used for storing an emulated software instruction read from the main memory but may be used for general storage.
- The P register is a general purpose register, generally used for holding the software program address in an emulation application.
- The A register is a general purpose register that also has built-in shifting capabilities for a single word shift. The length of the shift is controlled by the count contained in the N register.
- The Q register is a general purpose register. It can be combined with the A register to form a double-word AQ register, which can be shifted by the count contained in the N register.
- The X register is a general purpose register.
- The F register is a general purpose register that is used to contain information to be stored in the file registers.

The bit generator is a specialized circuit which provides for generating a word consisting of all 0's and one bit set to 1. The bit that is set to 1 is specified either by the contents of the C field of the microinstruction or by the contents of the N register.

2-4 14232000

The MPP organization has provisions for two register files, file 1 and file 2. File 1 consists of 256 word-size registers. The selection of the register to reference in file 1 is controlled by the contents of the K register. If file 1 is omitted, a constant can be wired into the organization to be provided when file 1 is referenced. Data is stored in file 1 by referencing the file in the D field. Data is automatically transferred to the F register for storage in file 1.

File 2 consists of 32 word-length registers and is addressed by the N register. It is normally used to contain often-used constants and requires a special command in the S field to cause the contents of the F register to be written into file 2.

If the 256-word file 1 is not included in an MPP configuration, an option allows the K register to be used for addressing the 32-word file 2 (since the N register has more fixed uses than the K register); an option also allows the 32-word file 2 to be referenced in the D field as the destination for writing into the file.

The N and K registers are control registers. The 8-bit K register is primarily used to address file 1, but also is used as a counter by the firmware; it can be set, incremented, decremented, cleared, and tested for zero. The N register is an 8-bit register which is used to reference the 32-word file 2; it is used as a counter and control register for A or AQ shifts and as a repeat control when the repeat code is specified in the S field.

The status/mode (SM) register consists of a minimum of 16 bits and a maximum of 64 bits (or twice the MPP word size). SM register bits have the following uses.

- Mode bits specify the mode of operation of the MPP. Examples of mode bits are the specification of 1's or 2's complement arithmetic and enabling or disabling the interrupt system. Further details are provided in Section 5.
- Status bits report on the status of the machine. One example is a bit which is set on detection of an overflow in the ALU operation.
- Unassigned bits can be used by the microprogram to provide 1-bit storage of conditions in the microprogram.

The mask register consists of 16 to 64 bits, limited to twice the machine word size. If a bit in the mask register is set to a 1, the MPP will recognize the corresponding interrupt line. If the mask bit is set to 0, the interrupt will not be recognized by the interrupt system.

The interrupt address decoder is a special logic network that generates the interrupt line address of the highest priority active interrupt that has its corresponding bit set to 1 in the mask register. This interrupt address can be read into the ALU for an operation to determine which interrupt requires service.

The page/micromemory (P/MA) register contains the micromemory address of the current microinstruction pair.

14232000 2-5

The micromemory address counter (MAC) is a counter that is incremented by one to determine the next sequential microinstruction pair.

The return jump (RTJ) register is used when the return jump addressing mode is specified in the M field of the microinstruction. The RTJ register holds the address of the next microinstruction pair to be executed when control returns to the main microprogram.

DETAILED DESCRIPTION

The detailed organization of the MPP is shown in Figure 2-2. This diagram shows that the registers of the MPP are interconnected primarily by selectors. The selectors in the diagram are numbered from S1 to S9. A selector is a multiplexer that provides for transferring one of several inputs to an output. Selectors S1, S2, S3, S4, and S9 are word-width selectors that provide for the transfer of a complete MPP word to the output. Selectors S5, S6, and S8 are 8-bit selectors and provide for the transfer of an 8-bit quantity; selector S7 is a single bit selector for transferring one of 16 input bits to its output.

The unlabeled inputs to selectors represent uncommitted logic in the basic MPP design, which is specified by the system designer to provide an efficient solution to a specific system design problem. Arrows leaving a register or a data path represent data points that are available on the backpanel wiring of the MPP; they are candidate bits of information for connection to the uncommitted logic of the MPP. The detailed wiring of the uncommitted selector inputs is performed on the transform module, which is different for each MPP application. The details of standard transform requirements are given later in this manual.

The wiring of specific bits to the SM register and the assignment of interrupt lines represent another area of uncommitted design in the standard MPP. Details of the SM and interrupt operations are also defined later in this manual.

MICROMEMORY AND MICROCONTROL SECTION

The MPP micromemory is an 80-nanosecond IC memory with a basic memory size of 256 words. Each word is 64 bits long, containing two 32-bit microinstructions. The basic memory size can be expanded in steps of 512 microinstructions to a maximum of 8192 microinstructions. The micromemory is available in two forms: a read/write memory is available for use during program development and in applications that require the MPP to be reprogrammed or reorganized for multiple applications, and a read-only memory is available for fixed applications of the MPP. The read/write micromemory can be loaded from an external paper tape reader, or data can be written into micromemory under control of the microprogram. The firmware is copied into a read-only micromemory during manufacture.

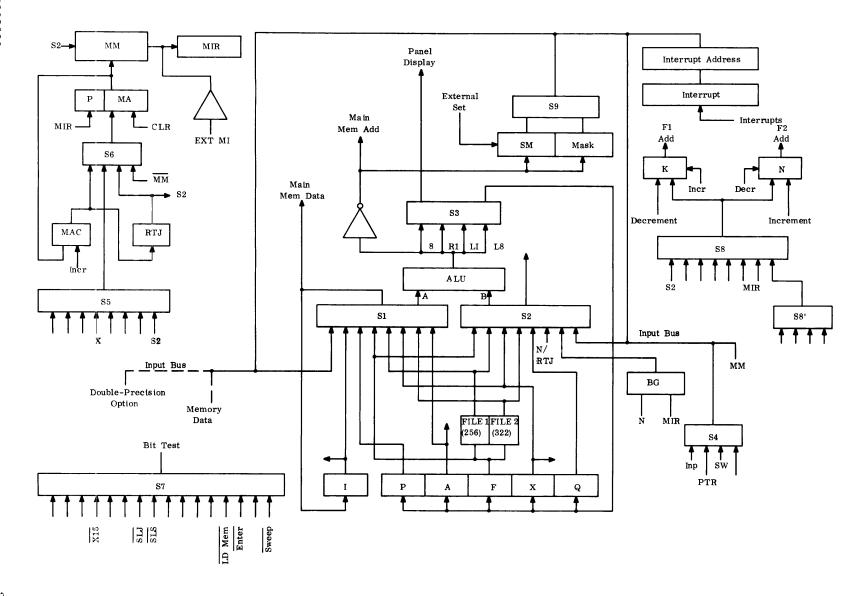


Figure 2-2. Detailed MPP Block Diagram

The micromemory is addressed by the P/MA register. The P portion of the register is a page control and consists of four bits specifying up to 16 pages of micromemory. A page consists of 256 words or 512 microinstructions. The MA register is an 8-bit register that specifies one of the 256 microinstruction pairs within the page that is to be the source of the next microinstruction. The T field of the current microinstruction is used to select the upper or the lower of the microinstruction pair as the next microinstruction to be executed. Microinstruction sequencing is designed so that no automatic overflow of addressing from the MA register to the P register occurs; any transfer of control between pages is initiated by a page jump operation, MA transform operation, or clear page operation.

PAGE/MICROMEMORY ADDRESS REGISTER

The P/MA register provides addressing for the micromemory. The micromemory is organized into pages of 256 locations, which provide for 512 microinstructions. The P portion of the register selects one of 16 pages, while the MA provides for selection one of the 256 locations within the page.

The P portion of the register can be set from a microinstruction. The micromemory address portion of the register can be set from a microinstruction, from the MAC, from the RTJ register, or from an output of the transform module. The MA portion of the register has no sequencing capability; this is provided by the MAC.

MEMORY ADDRESS COUNTER

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

MICROINSTRUCTION REGISTER

The MIR is a 32-bit register which is used to hold the microinstruction during execution. Data is entered into the MIR from the micromemory; either the upper or the lower 32 bits of the contents of the micromemory location are gated to the MIR, based on the value of the test bit determined during the preceding microinstruction. A test bit of 0 specifies the upper microinstruction, and a test bit of 1 specifies the lower microinstruction.

RETURN JUMP REGISTER

The RTJ register is provided to capture the location of the next microinstruction pair at any time, when specified by a microinstruction. When this capture is specified, the contents of MAC is incremented and stored in the RTJ register. The contents of the RTJ register is unchanged until the next command is given to save a new address. This saving of the next instruction pair location is independent of any actual transfer of control. The output of the RTJ register can be gated to the MA register to perform the return operation or it may be read into the organization of the MPP through selector S2.

2-8 14232000

TRANSFORMS AND THE TRANSFORM MODULE

The transform module provides the MPP with the capability to select bits from various sources in the organization of the machine and transfer them to the MA register to start a sequence of microinstructions, or to transfer the bits to the K or the N register for use in counting and file addressing or shifting control. The transforms allow the MPP to extract bits from a register, place the bits in the required positions, add base address bits, and transfer the resulting data to the required registers simultaneously. The transform module also provides for definition of up to 16 bits to be available for the bit test operations. Refer to Section 6 for a detailed discussion of the transform operations.

ARITHMETIC AND LOGICAL UNIT

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

These two inputs are combined according to the function code specified in the microinstruction; the result is delivered simultaneously at the output of the ALU for possible shifting and delivery to the destination register, to the ALU buffer circuit for delivery to the SM and mask registers, and to the memory address bus and the panel displays. It is possible to ignore the output of the ALU on any instruction. The result of the ALU operation as to the sign, zero, and magnitude (by means of the carryout test) is available to the test bit logic for instruction sequencing.

Arithmetic operations may be in 1's or 2's complement arithmetic and can operate on either single precision operands or on double precision operands, using the double precision hardware available as an option. Selection of single or double precision and 1 or 2's complement mode is controlled by the microprogram which controls the states of the applicable bits in the SM register.

Also available as an option is the ability to perform split adder operations. This option allows the ALU adder to be split between any group of four bits. For example, in a 16-bit MPP, a split may be made between the upper and lower 8-bit groups or the upper 4 and lower 12 bits and would allow operations on each group to take place simultaneously. The upper portion of the adder is acted upon using 2's complement arithmetic, while the lower portion may be operated on with either 1's or 2's complement. The selection of the split adder and the arithmetic mode of the lower section is also controlled by setting or clearing bits in the SM register. The split has no effect on logical operations, since these do not involve a carry between bits.

14232000 2-9

DATA MANIPULATION ORGANIZATION

The data manipulation organization of the MPP provides for storing data in one of six working registers and two files and for selecting data to be processed through the ALU. ALU results are transferred back to one of the registers or out of the organization to control external equipment.

Several other registers are provided to control addressing of the two files, in addition to being used for sequencing and repeat control.

The primary data registers are I, P, A, F, X, and Q. The names given to the registers are descriptive of the normal use of the registers in an emulation; however, the names are not intended to limit the use of the registers.

The following paragraphs detail the connection and operation of each of the registers and some of their intended uses.

I REGISTER

The I register is a word length register used primarily to hold the emulated software instruction being executed by the MPP. The contents of the I register may be connected to the transform module to allow selected bits or portions of the instruction to be combined in the transform. It is also available as input to S1, and therefore, to the A input of the ALU.

Data is entered into the I register from the output of S1. This connection is provided so that data may be transferred from memory directly to the I register, in addition to performing some other operation on the memory data or transferring the memory data to another register through the ALU.

P REGISTER

The P register is a word length register which receives data from the ALU and whose output is provided to S1 and thus to the A input of the ALU. This is a general purpose register; however, it normally is used to contain the software instruction counter for the emulation of a computer.

A REGISTER

The A register is a word length, general purpose register which receives data from the ALU and provides output to S1 and thus to the A input of the ALU. Output from the A register is also available to the transform module and to the I/O system. In a standard I/O system, the contents of the A register is used to provide data output to the I/O system.

The A register is mechanized as a shifting register, and it can be shifted left or right without the use of the ALU. The A register may also be combined with the Q register to form a double length shifting register, which operates independently of the ALU.

2-10 14232000

F REGISTER

The F register is a word length, general purpose register which can be read into the A or B input of the ALU. This register is also used as the file entry register and contains the information written into file 1 or file 2 when these files are used as the destination of an ALU operation.

X REGISTER

The X register is a word length, general purpose register which can be read into the A or B input of the ALU. This register also has provisions to supply information to the transform module and to the I/O system.

Q REGISTER

The Q register is a word length, general purpose register which receives data from the ALU and provides output to S2 and thus to the B side of the ALU. The Q register is mechanized as a shifting register; it may be shifted left or right in conjunction with the A register, without the use of the ALU.

REGISTER FILE USAGE

File 1 and file 2 are two optional, word length, scratchpad, storage register files which are addressed by the K and N registers, respectively. The files are optional because the hardware mechanizing either or both files may be omitted from a particular MPP at the time of design.

FILE 1

File 1 is a file of 256 general purpose, word-sized registers which 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 ALU on demand.

FILE 2

File 2 is a 32-word file and is addressed by the lower five bits of the N register. It is intended primarily as a source of constants, but may be used as a general purpose, word-sized register which delivers its output to S1 and S2, and thus to the A and B side of the ALU.

BIT GENERATOR

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

14232000 2-11

SELECTOR 1

S1 provides for the selection of one of eight inputs for delivery to the ALU or for delivery to the I register if the I register is selected as a destination in the microinstruction. The selector also provides data to the main memory for the memory write command.

Input to the selector is from the following:

- Input bus
- I register
- P register
- F register
- File 1
- X register
- File 2
- A register

Output from the selector goes to the following:

- ALU
- Main memory data bus for writing
- I register

INPUT BUS TO SELECTOR 1

The input bus is one input to S1 and provides for selection of input data to be transmitted to S1. The data may come from:

- Data from the external core memory
- Output of the SM register or the interrupt mask register

SELECTOR 2

S2 provides for the selection of one of eight inputs for delivery to the B input of the ALU. In addition, the second output provides for the transfer of information from this selector to the transform board and to the micromemory.

Input to S2 is from the following:

- F register
- File 1
- X register
- File 2
- Q register
- N and RTJ registers
- Bit generator
- Input bus

INPUT BUS TO SELECTOR 2

The input bus provides for submultiplexing of data for input to S2. Inputs to the input bus are:

- Main memory
- Interrupt address
- Micromemory
- I/O system, paper tape reader, and switches from the maintenance panel

STATUS/MODE REGISTER

The SM register can consist of one to four modules. Each card provides four flag bits and 12 status/mode bits.

A flag bit can be set or reset by a set or clear flag bit command from the microinstruction. A status/mode bit or flag bit can be set or reset by transferring data into the SM register from the output of the ALU. In addition, any status/mode bit can be set by an external connection and can be cleared externally only by a master clear command.

All of the bits in the SM register can be read as an input to S1, and all bits can be changed by storing data in the SM register. These bits are general purpose, and the wiring and meaning of the bits depend on the design of a particular MPP application. Refer to Section 6 for a detailed discussion of SM register operation.

14232000 2-13

INTERRUPTS AND MASK REGISTER

The interrupts and the mask register are interconnected so that interrupts on the interrupt line will be recognized only if the corresponding bit of the mask register is set to 1. If the mask register bit is 0, the interrupts are ignored. Interrupts are provided in groups of 16, and the maximum number allowed is equal to twice the number of bits in the basic processor word.

Each interrupt line terminates in the interrupt register, where circuitry compares the interrupt with the mask bit; if the mask bit is 1, it allows the microprogram to determine that at least one interrupt is active and to determine the address or bit number of the highest priority active interrupt through the address calculator. An interrupt is normally a stable line until it is acknowledged by the MPP program. If a pulse interrupt is used, a bit is used from the SM register to create a stable interrupt.

The interrupt address network is an encoder network which develops the address of the interrupt line number of the highest priority active interrupt whose mask bit is set. This address is then read in on the B input bus to the S2 input to the B side of the ALU. This address remains until the interrupt is removed from the interrupt line or until a higher priority interrupt occurs whose mask bit is set.

Commands are available to read the mask register as an input to S1 and to write the mask register on an output of the ALU. Each bit in the mask register enables the corresponding interrupt line if it is a 1. If the mask bit is 0, an interrupt on the corresponding line is ignored.

K REGISTER

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

N REGISTER

The N register is an 8-bit counter which 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 which controls microinstruction execution for operations such as multiplication and division. It may also be used as a programmed counter, since the original value of N can be tested against zero by the microinstructions.

2-14 14232000

Some 5600 system applications, including computer emulators, require a core memory for storage of software instructions, operands, and other data. This core memory is termed the main memory to distinguish it from the processor micromemory. Two types of main memory are available for use in MPP systems. The SPM is constructed of modules that are the same size as the processor logic modules; it is used in applications where small size is important and where the storage requirement does not exceed 16,384 18-bit words. The LPM is constructed of modules approximately 17 inches square; the LPM can be configured to store up to 262,144 36-bit words for each LPM controller module.

Both types of main memory are supplied with word lengths in increments of nine bits: eight bits for data and one extra bit that can be used for parity checking or program protection, as required by the application. Both types of main memory operate with an access time of 400 nanoseconds and a cycle time of 1.0 microsecond.

SMALL PLANE MEMORY

Figure 3-1 is a block diagram of the basic SPM. Each of the functional sections is contained on one or more (depending on the storage word size) pluggable modules. Figure 3-2 is a photograph of an SPM memory stack.

The SPM is a 3D, three-wire system to minimize the number of components required and to simplify memory stack assembly. The four standard operating modes are read/restore, clear/write, read/modify/write, and destructive read. The memory control logic receives the memory request and mode signals from the processor when a micro-instruction requesting a memory operation is executed. The memory control logic sends timing and read/write mode signals to the data logic and the X/Y drive logic. The main memory address to be referenced is supplied to the X/Y drive logic from the output of the processor ALU. The data logic receives data to be written into main memory from the output of selector S1 in the processor; data read from main memory is sent by the data logic to the input bus. The memory control logic returns the memory resume signal to the processor when the memory operation is substantially complete.

The X and Y drive currents for the memory stack are controlled by the X and Y source and sink switches. The output of the voltage regulator is varied as a function of memory stack temperature so that optimum X and Y drive currents are supplied to the stack over the operating temperature range of the memory.

14232000

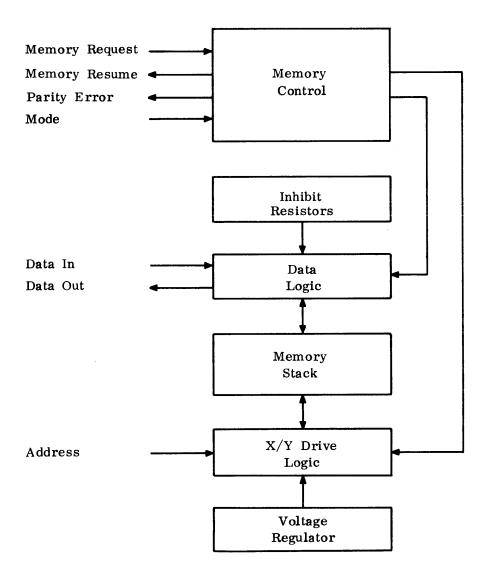


Figure 3-1. Basic SPM Block Diagram

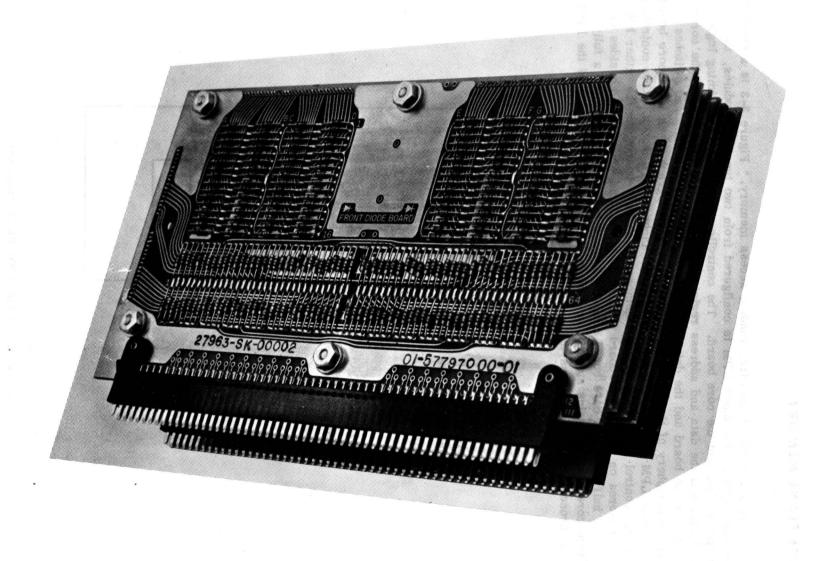


Figure 3-2. SPM Stack Module

LARGE PLANE MEMORY

The LPM is a 2-1/2D, three-wire, random access memory. Figure 3-3 is a block diagram of the basic LPM. The basic LPM is configured from two types of modules, the controller module and the core board. The controller module contains timing logic, parity logic, and the data and address registers. The core board contains the cores in the center of the board and the drivers, sense amplifiers, decoders, and switches around the periphery of the board. Figure 3-4 is a photograph of an LPM core board. Both types of LPM modules are approximately 17 inches by 17 inches. One controller module can control up to 262K of core memory. The core boards can be either half populated (cores on one side of board only) or full populated (cores on both sides of board). A half populated core board contains storage for 16K 9-bit bytes; a fully populated core board contains storage for 32K 9-bit bytes. The interface of the LPM with the processor is the same as that described for the SPM.

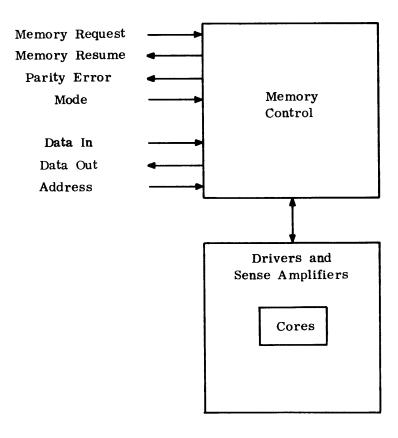


Figure 3-3. Basic LPM Block Diagram

Figure 3-4. LPM Core Board

INPUT/OUTPUT

The I/O capabilities of any 5600 system are described in the customer engineering manual for that system. The I/O logic is typically custom designed to accommodate the requirements of the specific application and may include one or more of the following options.

- Control Data 160-A channel
- Control Data 1700 channel
- Control Data 3000-series channel
- Serial channel
- Control Data 370 paper tape reader controller
- Control Data 1711 teletypewriter controller

14232000

MICROINSTRUCTION FORMATS

Each micromemory address specifies the location of two microinstructions. Each 32-bit microinstruction is divided into five main sections and is numbered from left to right as bits 0 to 31.

Bits	Function
0,1	Mode (M) field specifies format of S and C field and sequencing mode to obtain next microinstruction pair
2-15	ALU control field specifies ALU operation, sources of operands, and destination of result of operation
16-18	Test (T) field specifies method of selecting which microinstruction of next microinstruction pair to execute
19-23	Special (S) field specifies subformat selection and special operation
24-31	Constant (C) or suboperation field specifies constants, micromemory addresses, or other codes

The total instruction appears as follows:

0 1	2 15	16 18	8 19 2	3 24 31
M	ALU Control	Т	S	C

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

M	Addressing Mode	Format for bits 19 to 31
00	Return	Format 1
01	Sequential	Format 1
10	Jump	Format 2
11	Sequential	Format 3

Figure 5-1 shows all microinstruction formats. Note that format 1 and format 2 microinstructions have two subformats, which are selected by the value of bit 19.

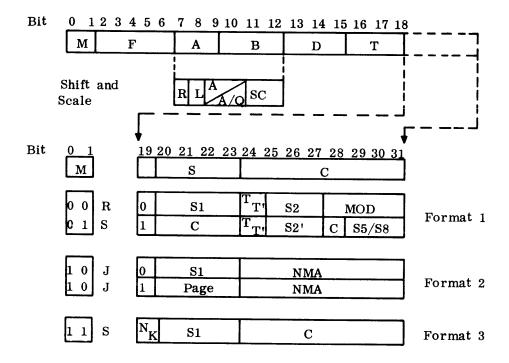


Figure 5-1. Microinstruction Formats

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

Bit	2	6	7	9	10	12	13	15
	F		A	1		В	D	

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

Bit	2		6	7	8	9	10	11	12	13	15
		F		R	L	A AG		s	C		

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

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

The codings in the S and C fields depend upon the contents of the M field. The S and C fields are coded in three formats. Format 1 is specified when the M field contains 00 (return mode) or 01 (sequential mode) as follows:

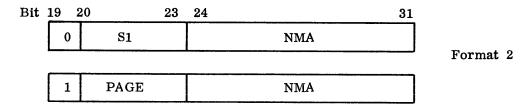
Bit	19	20	23	24	25	27	28	31
	0	S1		T T'	S2			MOD

Format 1

Bit	19	20		23	24	25	27	28	29	31
	1		C		T T'	S	2'	С	S5,	/S8

The S1 field specifies operations such as main memory read or write operations; alternate codings to be used in the A, B, and D fields; etc. The T/T' bit specifies that the code in the T field is to be interpreted as the normal T code (T/T' = 0) or as the alternate T' code (T/T' = 1). The subformat select bit, bit 19, determines whether bits 25 through 31 are to be interpreted as S2 codes or as S2' codes. The S2 code can be a constant for driving the bit generator, additional information to control the main memory read or write, or a code to initiate an I/O transfer or other operation. The S2' and S5/S8 codes are associated with transforms.

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



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

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

Bit	19	20	23	24 31	L
	N/K	S1		C]

This format allows one special operation to be performed as specified by the S1 code and also causes the eight bits of the C field to be transferred to the N register (bit 19 = 1) or to the K register (bit 19 = 0).

DETAILED MICROINSTRUCTION CODING

The following paragraphs give the detailed descriptions of the microinstruction operations and the codes for each microinstruction field. The mnemonic for each code is given for reference; the mnemonics are applicable when the microprogram assembler is used.

M FIELD (BITS O AND 1)

The M field defines the major operation taking place in the microinstruction and also specifies the type of sequencing which will be used to obtain the next instruction pair. The operations specified in the M field are listed in Table 5-1.

TABLE 5-1. M FIELD OPERATIONS

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

F FIELD (BITS 2 THROUGH 6)

The F field specifies the logical or arithmetic operation to be performed by the ALU or the shift or scale operations performed with the A and Q registers. The split adder and the double precision hardware options are described in the following paragraphs.

The split adder option allows the main ALU to be split into two independent adders. This split is activated by setting the adder split flag in the SM register. The split blocks the carry between the two portions of the adder. The upper portion of the adder always functions as a 2's complement adder; the lower portion can function as a 1's complement or as a 2's complement adder, depending upon the state of the 1's complement SM register flag. In 1's complement mode, the carryout of the lower portion is used as the end-around carry bit. In 2's complement mode, both portions of the adder act as independent 2's complement adders. The split adder has no effect on logical operations because no carry is involved in these operations.

The double precision hardware arithmetic option provides the capability to perform arithmetic on double-length operands. The double precision logic contains three additional registers (A*, Q*, X*) and a second ALU (ALU*) distinct from the main MPP elements. The A* register is unconditionally inputted to ALU*. The output of ALU* can be shifted left and right in a multiply or divide operation, and the output goes to the A* register. The X* and Q* registers are loadable only; they cannot be specified as destinations for the results from ALU*. The Q* register can be shifted during double precision multiply or divide iterations. The double precision logic, if present, is enabled when the double precision flag is set in the SM register. Figure 5-2 is a block diagram of the double precision logic.

LOGICAL OPERATIONS

The logical operations perform bit-by-bit combinations of the A input and the B input for delivery to the destination. Double precision logical operations cannot be performed on the A^* and X^* registers.

The logical operations are described in Table 5-2.

ARITHMETIC OPERATIONS

The arithmetic operators (Table 5-3) operate on either single precision operands (using the main ALU) or double precision operands (if the double precision logic is present and the double precision flag is set in the SM register). Two additional options are provided and coded in the arithmetic function code. The first option provides for a carry input to the adder (indicated by a plus sign in the microinstruction mnemonic). This is used for doing multiple precision arithmetic beyond that provided by the hardware logic. A T field code to check for a carryout of the ALU is provided to determine whether a carry into the ALU should be used on the next arithmetic operation. With the double precision bit set in the SM register, the carryin is entered in the lower bit of the double precision ALU; otherwise, the carryin is entered in the main ALU.

5-6 14232000

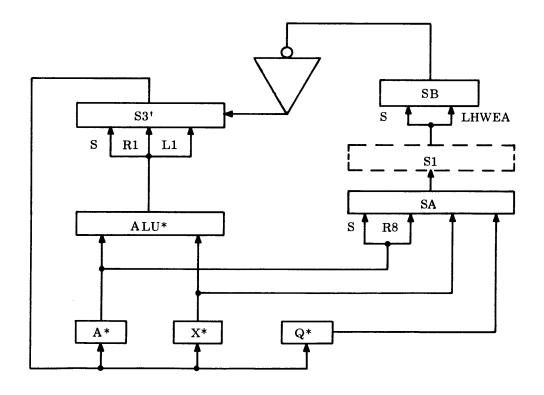


Figure 5-2. Double Precision Option Block Diagram

TABLE 5-2. LOGICAL OPERATIONS

F Code	Mnemonic	A input = 0 0 1 1 B input = 0 1 0 1
1 0000	1111011101110	Bit Result
01100	ZERO	0 0 0 0
01110	A•B	0 0 0 1
01101	A•-B	0 0 1 0
01111	Α	0 0 1 1
01000	-A•B	0 1 0 0
01010	В	0 1 0 1
01001	EOR	0 1 1 0
01011	A+B	0 1 1 1
00100	-A•-B	1 0 0 0
00110	-EOR	1 0 0 1
00101	-B	1 0 1 0
00111	A+-B	1 0 1 1
00000	-A	1 1 0 0
00010	-A+B	1 1 0 1
00001	-A+-B	1 1 1 0
00011	ONE	1 1 1 1

TABLE 5-3. ARITHMETIC OPERATIONS

F Code	Mnemonic	Operation
10100	SUB	Subtract B input from A input.
11000	ADD	Add A and B inputs.
10101	SUBT	Subtract with an overflow test.
11001	ADDT	Add with an overflow test.
10110	SUB+	Perform A + 1 - B input (1's complement only).
11010	ADD+	Perform A + B + 1.
10111	SUB+T	Perform SUB+ with an overflow test (1's complement only).
11011	ADD+T	Perform ADD+ with an overflow test.

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

SHIFT OPERATIONS

The shift operations in the F field specify a shift of the A register or the AQ register of the main MPP organization only; no shift is possible in the double precision registers from this command. The ALU is not used to perform the shift, but will perform some operation based on its decoding of the F field (which should be considered as unknown). The destination will receive this meaningless output unless an NOP is chosen for the destination of the D field.

The type of shift is determined by the coding in bits 7 to 12 of the microinstruction, and the amount of the shift is determined by the number contained in the N register. The operation examines the N register, and, if it is zero, the next microinstruction is executed. The T field codes normally used with a shift are U, L, BIT TEST, and LQL. Other T codes must be used with caution.

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

The shift conditions are:

- Shift A Shift the A register only.
- Shift AQ Shift the combined A and Q register. The Q register is considered to be the least significant bits of the combined AQ register.
- Shift left or right.
- Enter 0 Enter a 0 in the vacated bit position at the end of the register.
- Enter 1 Enter a 1 in the vacated bit position at the end of the register.
- Extend sign Extend the sign (for a right shift only).
- End-around carry Enter the bit coming off the end of the register into the vacated on position at the other end.

All shifts are performed with an F code of 11110. The type of shift is determined by bits 7 through 12 of the microinstruction. The shifts are defined in Table 5-4.

14232000

TABLE 5-4. SHIFT OPERATIONS

Bit	Code	I	
7 8 9	11 12	Mnemonic	Operation
100	0 0	AR0E	A is right shifted (N) bits, with 0 entered at most significant bit.
1 0 0	0 1	ARSE	A is right shifted (N) bits, with sign extension.
100	1 0	AREA	A is right shifted (N) bits, with end-around carry.
0 1 0	0 0	A LOE	A is left shifted (N) bits, with 0 entered as least significant bit.
010	0 1	AL1E	A is left shifted (N) bits, with 1 entered as least significant bit.
0 1 0	1 0	ALEA	A is left shifted (N) bits, with end-around carry.
101	0 0	AQR0E	AQ is right shifted (N) bits, with 0 entered as most significant bit in A.
101	0 1	AQRSE	AQ is right shifted (N) bits, with sign extension.
101	1 0	AQREA	AQ is right shifted (N) bits, with end-around carry.
0 1 1	0 0	AQL0E	AQ is left shifted (N) bits, with 0 entered at least significant bit in Q.
011	1 0	AQLEA	AQ is left shifted (N) bits, with end-around carry.

SCALE OPERATIONS

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

The scale operation is performed as follows:

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

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

All scale operations are performed with an F code of 11111. The type of shift for the scale is determined by bits 7 through 12 of the instruction. The scales are given in Table 5-5.

TABLE 5-5. SCALE OPERATIONS

Bit	Bit Code		Mnemonic	Operation	
7 8 9	11 1	12	Minemonic	Operation	
0 1 0	0	0	SL0E	A is scaled left, with 0 entered as least significant bit.	
0 1 0	0	1	SL1E	A is scaled left, with 1 entered as least significant bit.	
0 1 0	1	0	SLEA	A is scaled left, with end-around carry.	
0 1 1	0	0	SDL0E	AQ is scaled left, with 0 entered as least significant bit in Q.	
0 1 1	1	0	SDLEA	AQ is scaled left, with end-around carry.	

A FIELD (BITS 7 THROUGH 9)

The A field specifies the input to S1 and thus to the A side of the ALU. The eight A codes are expanded by eight A' codes by placing 1010 or 0111 in the S1 field.

The eight A inputs to S1 and to the A side of the ALU are indicated when the S1 field is not 0111 or 1010; the eight A codes specify inputs from the files, the registers, or main memory, according to Table 5-6.

TABLE 5-6. A INPUT OPERATIONS

A Code	Mnemonic	Operation
000	F2	Use contents of file 2 register as A source input. Current value of N register is used to address register file 2. If value of N is changed in current microinstruction, its initial value is used to reference file register.
001	P	Use contents of P register as A source.
010	I	Use contents of I register as A source.
011	x	Use contents of X register as A source.
100	Α	Use contents of A register as A source.
101	F	Use contents of F register as A source.
110	F1	Use contents of file 1 register as A source. Current value of K register is used to address register file 1. If value of K is changed in current microinstruction, initial value of K is used to reference file register.
111	MEM	Obtain data read from main core memory and use it as A source. If memory READ command was not given either one or two microinstructions before MEM input, all-zeros condition is obtained. Restriction: If B' code is used, data from B source is also used as A source, suppressing data from memory reference.

5-12

The eight A' inputs to S1 and to the A side of the ALU are indicated if the S1 field is 0111 or 1010. The A' codes specify input from the SM registers or mask registers. The A' codes are given in Table 5-7.

TABLE 5-7. A' INPUT OPERATIONS

A' Code	Mnemonic	Operation
000	SM1	Use contents of SM register 1 as A source.
001	M1	Use contents of interrupt mask register 1 as A source.
010	SM2	Use contents of SM register 2 as A source.
011	M2	Use contents of interrupt mask register 2 as A source input.
100	A*R8	Use contents of double precision A* register, shifted right eight bits with end-around carry, as A source. A* register remains unshifted.
101	A*	Use contents of double precision A* register as A source.
110	X*	Use contents of double precision X* register as A source.
111	Q*	Use contents of double precision Q* register as A source.

B FIELD (BITS 10 THROUGH 12)

The B field specifies the input to S2 and thus to the B side of the ALU. The 11 possible B codes are expanded by the seven B' codes when the S field contains 1000. The N and RTJ codes are controlled by bits 28 and 29 from the C field. Note that the placement of N and RTJ in the input is a design option; the B code discussion that follows describes the standard option, but the fields may be reversed on specification for a particular 5600 system.

The 11 B inputs to S2 and thus to the B side of the ALU are indicated if the S field is not 1000. Code 001 of the B field is expanded by the use of bits 28 and 29 of the microinstruction for enabling the N or RTJ register to S2. This use of bits 28 and 29 is independent of the other use of the C field, and thus, by judicious use of commands in the C field, the input for the N and RTJ may be used in conjunction with commands or constants in the C field. The codes for B inputs are given in Table 5-8.

The B' inputs are specified in the B field when the S field contains 1000. The codes and actions are given in Table 5-9.

5-14 14232000

TABLE 5-8. B CODES

B Code	28 29	Mnemonic	Operation
000		F2	Use contents of file 2 register as B source. Value of N register, before instruction is executed, is used to address register file 2.
001	1 1	ZERO	B source is all zeros.
001	1 0	N	Use contents of N register as B source. Since N is 8-bit register, this source uses N as upper eight bits and zeros as lower bits.
001	0 1	RTJ	Use contents of RTJ register as B source. Since RTJ is 8-bit register, upper bits are zeros and (RTJ) serves as lower eight bits.
001	0 0	N, RTJ	Use contents of N and RTJ registers as B source. These registers are combined, with N register as upper eight bits of source and RTJ as lower eight bits. All other bits (if any) are zero.
010		BG	Use contents of BG register as B source. This register has only one bit set to 1, and position of bit in BG register is specifiable. Position is specified either on value in the N register or by number in C field, depending on state of controlling SM register bit.
011		X	Use contents of X register as B source.
100		Q	Use contents of Q register as B source.
101		F	Use contents of F register as B source.
110		F1	Similar to F2, but uses the contents of file 1 register addressed by (K) as B source.
111		MEM	Obtains data read from main core memory and uses it as B source. If memory READ command was not given either one or two instructions before MEM input command, all-zeros condition is obtained. Restriction: If A' code is used, data from A source is also used as B source, suppressing memory data.

TABLE 5-9. B' CODES

B' Code	Mnemonic	Operation
000	CSW	Use setting of data switches on maintenance panel as B source.
001	CPT	Use current character from external paper tape reader as B source input. Eight-bit character read is repeated every eight bits in input word. Sequencing of reader and start/stop operation is controlled by I/O control commands. (Refer to the INP/j code in Table 5-17.)
010	INRD	Transfer word from I/O channel j, as specified by INP/j command in C field of this microinstruction. This command is normally only given after receipt of interrupt, indicating that data is available on input channel receiver. If given at any other time, it provides status of input channel receivers.
100	MMU	Transfer upper 16 bits of data from micromemory to F register in 16-bit MPP; 32-bit MPP transfers total 32-bit word. F field must make a reference to B source. C field is used to specify micromemory address. D field must be NOP.
101	MML	Same as above, but takes lower 16 bits of data in a 16-bit MPP.
110 or 111	INTA	Use contents of interrupt address register as B source. This register contains bit representation of interrupt line number of higher priority interrupt having corresponding interrupt mask bit set. If there are no such interrupts, INTA returns a value one higher than highest possible interrupt.
011	FRMT	This is a data transform which may be wired to any bits available to normal transform operations and rearrange them as input to S2. This input comes from unlabeled input to S4.

D FIELD (BITS 13 THROUGH 15)

The D field specifies the destination of information from the main organization of the MPP. There are four sources of this information as follows:

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

These destination sources are concurrently available in various combinations, as described in the following paragraphs.

All D destinations except the I register are optionally shiftable by S3 when specified by a code in the C field or by the L8EA command in the S1 field, if the alternate codings, D' or DD", are not specified. The I destination differs from the others in that the output of the A source is the input to the I register. The codes and their operations are given in Table 5-10.

The D' destinations are specified by the D field if the S1 field is set to 1001 or 1010. The codes and action are given in Table 5-11.

The D' destinations are specified by the D field if the S1 field is set to 1011. These destinations transfer data to the double precision logic from S1. The codes and actions are given in Table 5-12.

The DD" option allows the performance of an operation on A, X, or F; this changes the register, but keeps a copy of the original register in a double precision register. This is another way of getting data to the double precision registers. The DD" option is specified when the S field contains 0001. Table 5-13 lists the DD" codes and their operations.

TABLE 5-10. D CODE TRANSFERS

D Code	Mnemonic	Operation
000	NOP	Do not transfer data to any destination.
001	P	Transfer output of S3 to P.
010	I	Transfer output of S1 to I.
011	Q	Transfer output of S3 to Q.
100	F1	*Transfer output of S3 to F register and write this data in file 1 at address specified by K at completion of this instruction.
101	A	Transfer output of S3 to A.
110	X	Transfer output of S3 to X.
111	F	Transfer output of S3 to F.

*NOTE

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

TABLE 5-11. D' CODE TRANSFERS

D' Code	Mnemonic	Operation
000, 001		Not used.
010	MMU	Transfer output of S2 to upper 16 bits of micromemory in 16-bit MPP, or transfer output of S2 to 32-bit word in micromemory in 32-bit MPP.
011	\mathbf{MML}	Transfer output of S2 to lower 16 bits of micromemory location in 16-bit MPP.
100	M1	Transfer output of ALU to mask register 1.
101	SM1	Transfer output of ALU to SM register 1.
110	M2	Transfer output of ALU to mask register 2.
111	SM2	Transfer output of ALU to SM register 2.

NOTE

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

TABLE 5-12. D" CODE TRANSFERS

D'' Code	Mnemonic	Operation
000	NOP	Do not transfer data to any destination.
001	A*LHW	Transfer output of S1 to A* register, shifted left one-half word, with end-around carry.
010	X*LHW	Transfer output of S1 to X* register, shifted left one-half word, with end-around carry.
011	Q*LHW	Transfer output of S1 to Q* register, shifted left one-half word, with end-around carry.
100	NOP	Do not transfer data to any destination.
101	A*	Transfer output of S1 to A* register.
110	X*	Transfer output of S1 to X* register.
111	Q*	Transfer output of S1 to Q* register.

TABLE 5-13. DD" CODES

DD'' Code	Mnemonic	Operation
101	AA*	Transfer output of S3 to A register, and transfer output of S1 to A* register.
110	XX*	Transfer output of S3 to X register, and transfer output of S1 to X* register.
111	FQ*	Transfer output of S3 to F register, and transfer output of S1 to Q* register.

T FIELD (BITS 16 THROUGH 18)

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

The T field codes consist of two groups, T codes, and T' codes. Similarly to the A, B, and D fields that are extended by using the S field, the T field is always extended in the following sense. Bit 24 of format 1 microinstructions is either 0 or 1 if T or T', respectively, is specified. The T' codes are available only for the return and sequential addressing modes (format 1). The T/T' codes select the upper or lower portion of the next microinstruction pair as the next microinstruction to execute. The T codes are listed in Table 5-14; the T' codes are listed in Table 5-15.

TABLE 5-14. T ADDRESSING MODES

T Code	Mnemonic	Operation
000	*L	Execute lower microinstruction of this microinstruction pair as next microinstruction. This operation overrides M field addressing mode.
001	Ū	Execute upper microinstruction of next microinstruction pair.
010	L	Execute lower microinstruction of next microinstruction pair.
011	KZU	If initial contents of K register is zero, execute upper microinstruction or next microinstruction pair; otherwise, execute lower microinstruction of next microinstruction pair. If decrement K command is included in same microinstruction, K will contain all ones on satisfying zero test.
100	NZU	If initial contents of N register is zero, execute upper micro- instruction of next microinstruction pair; otherwise, execute lower microinstruction of the next microinstruction pair. If decrement N command is included in same microinstruction, N will contain all ones on satisfying zero test.
101	REJU	If, on command which specifies memory reference or I/O, reference is rejected, execute upper microinstruction of next microinstruction pair. If reference is successful, execute lower microinstruction of next microinstruction pair. REJU command for T field is only used in input, output, memory read, or memory write microinstructions. If it is used in any other microinstruction, lower microinstruction is
		executed.
110	NU	If sign bit of ALU output is negative on completion of this microinstruction, execute upper microinstruction of next microinstruction pair; otherwise, execute lower microinstruction of next microinstruction pair.
111	ZL	If output of ALU is zero on completion of this instruction, execute lower microinstruction of next microinstruction pair; otherwise, execute upper microinstruction of next microinstruction pair.

TABLE 5-15. T' ADDRESSING MODES

T' Code	Mnemonic	Operation
000	*L	Execute lower microinstruction of this microinstruction pair. This operation overrides M field addressing mode.
001	LQL	If, on completion of this microinstruction, least significant bit of Q is 1, execute lower microinstruction of next microinstruction pair. Otherwise, execute upper microinstruction of next microinstruction pair.
010	I/RU	If there is an interrupt or if I/O reference results in reject, execute upper microinstruction of next microinstruction pair; otherwise, execute lower microinstruction of next microinstruction pair. See REJU for reject discussion (Table 5-14).
011	INTU	If there is an interrupt and its corresponding interrupt mask bit is set, execute upper microinstruction of next microinstruction pair; otherwise, execute lower microinstruction of next microinstruction pair.
100	BTU	Bit test. Lower order four bits in C field of this micro- instruction specify a setting of bit test selector. If bit at this position is 1, execute upper microinstruction of next micro- instruction pair; otherwise, execute lower microinstruction of next microinstruction pair.
		Bit test is general purpose testing facility that allows wiring any bit of organization available on machine's backpanel to bit test selector. This wiring is defined on transform board. Refer to Section 6 for discussion of test bits.
101	LQ*L	If on completion of this microinstruction, least significant bit of Q* is 1, execute lower microinstruction of next microinstruction pair; otherwise, execute upper microinstruction of next microinstruction pair.
110	COL	Carryout lower. If as result of arithmetic operation, there is carryout of ALU, execute lower microinstruction of next microinstruction pair. Otherwise, execute upper microinstruction of next microinstruction pair.
		This instruction allows a test for carryout of ALU during multiple-precision arithmetic.
111	Z*L	Same as ZL (Table 5-14), except ALU* is tested.

SUBFORMAT SELECT BIT (BIT 19)

Bit 19 is used to select either variations in format 1 and format 2 decoding or the choice of addressing the N or K register in format 3.

S FIELD (BITS 20 THROUGH 23)

The S field of the microinstruction is used to specify a special command (including alternate codings in the A, B, and D fields), in addition to page or constant information (as required by the code in the C field). The following S codes specify actions which take place at the same time as the ALU operation specified in the F, A, B, and D fields. The codes and operations are given in Table 5-16.

TABLE 5-16. S FIELD CODES

S Code	Mnemonic	Operation			
0000	RPT	If N register is not equal to zero, selection of next micro- instruction pair is inhibited and current microinstruction pair is next microinstruction pair. N register is decremented by one. Normal T field selection applies. If N register is equal to zero, normal next microinstruction pair is used.			
0001	DD	Alternate D field coding, DD".			
0010	NOP	No operation for S field.			
0011	READ	Read memory; transmit output of ALU to main memory as address of word to be read. Instruction execution is delayed until a resume is received from memory, acknowledging command. Memory data is available for input to system in following microinstruction if following instruction is an MA transform microinstruction, or it is available two instructions after read if memory data is an operand. REJU may be included in T field to test correct memory operation.			
0100	WRITE	Write memory; transmit output of ALU to main memory as address of word to be written, and transmit output of S1 as data to be written. Instruction execution is delayed until a resume is received from memory, acknowledging command. Data is being stored in memory on completion of this instruction. REJU may be included in T field to test correct memory operation.			
0101	L8EA	Output of ALU is shifted left eight bits, end-around.			

5-24 14232000

TABLE 5-16. S FIELD CODES (Cont.)

S Code	Mnemonic	Operation			
0110	F2WR	Write data contained in F register into file 2 at address specified by contents of N register at beginning of current microinstruction. Actual writing takes place during first part of instruction.			
0111	AP	Alternate A field coding, A'.			
1000	BP	Alternate B field coding, B'.			
1001	DP	Alternate D field coding, D'.			
1010	APDP	Alternate A and D field coding, A'D'.			
1011	DPP	Alternate D field coding, D".			
1100	GATEI	Gate output of S1 to I register.			
1101	HALT	If halt bit of SM register is 1, stop operation of MPP on completion of this microinstruction. When start signal is received, continue with next microinstruction specified by addressing mode and T field. If halt bit is 0, continue with microinstruction sequencing.			
1110	RTJ	Transfer address of next sequential microinstruction pair to RTJ register. This is done regardless of actual addressing mode used in this instruction.			
1111	CLRNP	Clear N register and page register.			

C FIELD (BITS 24 THROUGH 31)

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

The codes for this field are listed in Table 5-17.

TABLE 5-17. C CODE ACTIONS

C Code	Mnemonic	Operation			
00xxxxx		xxxxx is a constant for use in driving bit generator or in any other commands using lower five bits of instruction for control.			
0100000 0100001 0100010 0100011	WRCH/0 WRCH/1 WRCH/2 WRCH/3	Write 8-bit character specified from output of S1 at memory address specified by output of ALU. See WRITE command in S field for details of operation. Character 0 is bits 0 through 7; character 1 is bits 8 through 15, etc. Remainder of word in memory is unchanged. Character is not repositioned in WRITE command.			
0100100	RMW	READ MODIFY WRITE - Perform a read of information from main memory in same manner as READ instruction. Memory system will perform a read cycle and lock up before performing write cycle. Memory must be forced to complete write cycle by issuing WRITE instruction, using same address, before it will respond to any other read operations.			
0100101	WRHW0	Write bits 0 through 15 from output of S1 at memory address specified by output of ALU. Bits 16 through 31 in memory are unchanged.			
0100111	WRHW1	Write bits 16 through 31 from output of S1 at memory address specified by output of ALU. Bits 0 through 15 in memory are unchanged.			
0101000	WRPB	WRITE PROTECT BIT - See protect system discussion (Section 6).			
011xx00	INP/j	xx is value of j, from 0 to 3; connect input channel (receivers) j to input for reading to S2 input through S4 selector. This instruction is accompanied by specification of INRD in B field of instruction to gate input information to MPP organization. REJU may be included in T field to test success of input operation.			

TABLE 5-17. C CODE ACTIONS (Cont.)

C Code	Mnemonic	Operation		
011xx10	OUT/j	xx is value of j, from 0 to 3. Perform output on channel (transmitters) j. Output data is normally obtained from A register and other control data from X register, depending on I/O organization designed. REJU may be included in T field to test success of the output.		
1000101	INCK	Increment number contained in K register by one.		
1001101	INCN	Increment number contained in N register by one.		
1000100	DECK	Decrement number contained in K register by one.		
1001100	DECN	Decrement number contained in N register by one.		
1000000	CLRK	Clear K register.		
1001000	CLRN	Clear N register.		
101xxxx	SETF/j	xxxx is value of j, from 0 to 15. Set SM register flag j to 1.		
110xxxx	CLRF/j	xxxx is value of j, from 0 to 15. Clear SM register flag j to 0.		
1110000 or 1110001	RQLXN	Destination register (P, A, F, or X) and Q register are considered as one double-length register with Q register as lower order bits. Combined register is shifted left one bit position with complement of ALU sign bit entered into lowest bit position of Q register.		
1110011	RQR1E	Shift combined destination and Q register right one bit, and enter 1 in sign position of destination register. This command is used in multiply iteration.		
1110010	RQR0E	Shift combined destination and Q register right one bit, and enter 0 in sign position of destination register. This command is used in multiply iteration.		
1110100	RL0E	Shift destination register left one bit, entering 0 in lowest bit position of the register. This operation can not be performed when Q is destination register.		
1110101	RL1E	Shift destination register left one bit, entering 1 in lowest bit position. This operation can not be performed when Q is destination register.		

TABLE 5-17. C CODE ACTIONS (Cont.)

C Code	Mnemonic	Operation			
1110110	RR0E	Shift destination register right one bit, entering 0 in sign position of register. This operation can not be performed when Q is destination register.			
11101111	RR1E	Shift destination register right one bit, entering 1 in sign position. This operation can not be performed when Q is destination register.			
1	_	m codes have a 1 in bit 19 to specify the S2' format. See sion of the transforms.			
000xxxx	TMA/j	xxxx = j, with values from 0 to 15. Obtain next microinstruction pair from address specified by MA transform selector setting j. Note that selector is eight-position, and values of j from 8 to 15 introduce additional bit which may be used to specify alternate starting point for corresponding transform from 0 to 7.			
001xxxx	TMAK/j	xxxx = j, from 0 to 15. Obtain next instruction pair from address specified by MA transform selector setting j. Also, set K register to value specified by K transform selector setting j. See TMA/j for additional information.			
010xxxx	GITMAK/j	Gate output of S1 to I register and perform TMAK/j operation. This instruction is used in RNI sequence to get instruction from memory to I register.			
011xxxx	TK/j	xxxx = j, with values from 0 to 15. Set K register to value specified by K transform selector setting j.			
100xxxx	TN/j	xxxx = j, from 0 to 15. Set N register to value specified by N transform selector setting j.			
The follow	The following two codes are used if the optional program protect logic is included.				
101xxxx	SPS	Set protect status. See protect system discussion (Section 6).			
110xxxx	CPS	Clear protect status. See protect system discussion (Section 6).			
l	1				

TABLE 5-17. C CODE ACTIONS (Cont.)

C Code	Mnemonic	Operation			
The following format 3 codings are for setting the K and N register; this format has the M field bits 0 and 1 set to 11, while bit 19 selects the register.					
Value	K = value When microinstruction bit 19 = 0, transfer C field value (bits 24 to 31) to K register and execute next sequential microinstruction pair.				
<u> </u>		When microinstruction bit $19 = 1$, transfer C field value (bits 24 to 31) to N register and execute next sequential microinstruction pair.			
1	The following format 2 codings in the C field are used to perform a jump, specified if the M field (bits 0 and 1) is 10.				
requi		Number is address of next instruction pair. If page jump is required (bit 19 = 1), S field contains page setting instead of S field code.			

This section describes the special features of the MPP and the options available for configuring a specific 5600 system. All of the options are factory installed because they involve wiring on the backpanels of the 5600 equipment enclosures. The particular features and options included in any specific 5600 system are specified in the customer engineering manual for that system.

STATUS/MODE REGISTER

The SM register allows the microprogram to control the mode of operation of the MPP and also allows the microprogram to examine the status of certain internal and external conditions. The MPP can access one or two SM registers, referred to as SM1 and SM2. Each SM register contains the same number of bits as the basic processor word; the maximum total number of SM bits in an MPP is twice the basic processor word length. The SM registers are assembled by using the required number of 16-bit SM modules; all SM modules are logically and electrically identical. All 16 bits of an SM module can be set or reset by the microprogram, by transferring information to the SM register from the output of the ALU.

The SM register bits are numbered as follows for 16-bit and 32-bit MPPs. 16-bit MPP: SM1 bits (first module) numbered 0 to 15; SM2 bits (second module) numbered 0 to 15. 32-bit MPP: SM1 bits (first module) numbered 0 to 15, SM1 bits (second module) numbered 16 to 31; SM2 bits (third module) numbered 0 to 15, SM2 bits (fourth module) numbered 16 to 31.

The individual bits of an SM module have the following additional characteristics.

Module Bit

Functional Characteristics

0 - 3

These bits are flag bits which can be set or reset by the SETF and CLRF commands in a microinstruction. These commands provide for addressing up to 16 flag bits; four flag bits are on each SM module. In a 16-bit MPP with two SM registers, bits 0-3 of SM1 are addressed as flags 0-3 and bits 0-3 of SM2 are addressed as flags 4-7.

In a 32-bit MPP with two SM registers, bits 0-3 of SM1 are flags 0-3, bits 16-19 of SM1 are flags 4-7, bits 0-3 of SM2 are flags 8-11, and bits 16-19 of SM2 are flags 12-15.

The flag bits are not cleared by a processor master clear. These bits are available as a true output from the SM module to the backpanel wiring. No external input is provided to these bits; they can be set and reset internally by the SETF and CLRF commands.

4 - 9

These bits can be set by an external signal. They can be cleared by a processor master clear. These bits are available as a true and a complement output from the SM module to the backpanel wiring. The complement output is provided with buffering.

10-14

These bits can be set by an external signal. They can be cleared by a processor master clear. These bits are available only as a complement output from the SM module to the backpanel wiring.

15

This bit has the same characteristics as bits 10-14 except that the clear input to the bit is available for connection to an external signal source. If this connection is used, the bit can be cleared by an external signal, but it then cannot be cleared by the processor master clear.

The assignment of status and mode conditions to specific bits of the SM register is a design function. The "standard" assignment of a minimum set of conditions is included later in this section after a discussion of the actual mode and status conditions.

MPP OPERATING MODES

The following MPP operating modes are options which can be incorporated into a particular 5600 system during manufacture. The operations are discussed below as controlled by a mode bit in the SM register, but if an option is not included, the associated mode bit will not be used in the SM register. In addition, if only one condition of a normal two-condition option is included, that option can be included in the wiring of the MPP but the controlling mode bit is not included in the SM register.

DOUBLE PRECISION ARITHMETIC

If this mode bit is set to 1 and the double precision hardware option is included in the MPP, the ALU and the ALU* on the double precision option module are combined to form a double-word length ALU. This double-word length ALU operates for addition and subtraction operations but not for logical operations.

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

1's COMPLEMENT

If this mode bit is set to 1, the ALU (and ALU* for double precision operations) operates in 1's complement arithmetic mode for addition and subtraction.

If this mode bit is set to 0, operations are in 2's complement arithmetic mode for addition and subtraction.

BG INPUT FROM N

If this mode bit is set to 1, the bit generator is controlled by the lower five bits of the N register.

If this mode bit is set to 0, the bit generator is controlled by the lower five bits of the microinstruction.

ADDER SPLIT

If this mode bit is set to 1, the ALU is split into two independent ALUs at the adder split point for arithmetic operations.

If this mode bit is set to 0, the ALU operates as a single word-size ALU.

DEAD START

If this mode bit is set to 1, the dead start logic is enabled.

If this mode bit is set to 0, the dead start logic is disabled. Refer to the discussion on dead start for a description of this option.

ENABLE DMA

If this mode bit is set to 1, the DMA option is enabled to allow external devices to use the DMA channel.

If this mode bit is set to 0, the DMA channel is inhibited and the MPP has exclusive use of the memory access channel.

DISABLE TIMEOUT

If this mode bit is set to 1, the internal MPP reject circuitry is disabled, which causes the MPP to delay on main memory operations and I/O operations until an actual reject signal is received. Only the I/O systems send a reject to the MPP when the requested operation cannot be performed; the memory system does not send a reject signal, so an erroneous main memory reference to a nonexistent memory location will cause a hangup in the MPP.

If this mode bit is set to 0, the internal MPP reject circuitry generates a reject signal if no response is received from the I/O system within 5 microseconds of issuing a command to the system.

ENABLE PAPER TAPE READER CLUTCH

This mode bit is included in 5600 systems that contain the dead start option. If this mode bit is set to 1, the clutch on the 370 paper tape reader will be enabled, allowing the tape to move.

If this mode bit is set to 0, the clutch on the 370 paper tape reader is disabled, stopping the paper tape motion.

ENABLE INTERRUPT

If this mode bit is set to 1, the interrupt system of the MPP is enabled and the INTU test can examine the interrupt system.

If this mode bit is set to 0, the interrupt system of the MPP is disabled and the INTU test will always receive a reply of "no interrupt present."

6-4 14232000

HALT

If this mode bit is set to 1, any microinstruction with a HALT coded in the S field stops the operation of the MPP on completion of the microinstruction.

If this mode bit is set to 0, the HALT code in the S field of any microinstruction is ignored.

This mode bit is held at 1 when the STEP INST/OFF/MICRO switch on the maintenance panel is set to STEP INST; it is set one time for each depression of the STOP switch on the maintenance panel. The microinstructions can clear this mode bit if set by the STOP switch.

OTHER MODE BITS

Specific 5600 systems also include additional mode bits used for communication with the I/O system and with external devices. For these mode bits, refer to the customer engineering manual for the specific 5600 system.

MPP STATUS

The MPP status bit assignments described in the following paragraphs are available as options for inclusion into the SM register during manufacture. The status conditions are discussed as setting a status bit, but if the status information is not required for a particular 5600 system, the bit can be deleted from the SM register. All of these status bits are set by the condition detected; the clearing of the status bit must be performed by the microprogram because none of the bits in the SM register have a clear input from the external equipment or the MPP. (Exception: bit 15 of each SM module can be wired to have a clear input.)

OVERFLOW

This status bit is set to 1 on detection of an arithmetic overflow. An arithmetic overflow is detected only if the microinstruction performing the addition is an add or subtract with overflow test and the arithmetic result is inconsistent with the sign of the operands and the arithmetic operation.

PARITY ERROR

This status bit is set to 1 if the main memory detects a parity error on data read from the memory.

REJECT

This status bit is set to 1 if a reject is generated by the I/O system or if the memory read results in a reject. The precise definition of reject is a design decision. The following options have been implemented in specific MPP systems.

- Reject if the I/O system returns a reject signal or if the internal timeout timer elapses
- Reject only if the internal timeout timer elapses on memory or I/O operations
- Reject only if the internal timeout timer elapses on I/O operations only

PROTECT FAULT

This status bit is set to 1 if the main memory protect logic detects an attempt to write into a protected area of main memory by an unprotected program. This status bit is only used by a specific implementation of memory protect.

PAPER TAPE FEED HOLE

This status bit is set to 1 if a feedhole is detected on the 370 paper tape reader. This status bit is used only on systems using the dead start option and a paper tape reader.

OTHER STATUS BITS

Other status bits are assigned to external conditions and I/O conditions depending on specific 5600 system applications. For these status bit definitions, refer to the customer engineering manual for the specific 5600 system.

6-6 14232000

STANDARD S/M REGISTER BIT ASSIGNMENTS

The following assignment of status and mode bits in the SM register is the standard assignment to maintain maintenance compatibility among 5600 systems. The bits of the SM registers are numbered as 100 to 115 for SM1 in a 16-bit MPP and 200 to 215 for SM2. A 32-bit MPP has corresponding SM bit numbers of 100 to 131 and 200 to 231. In the following list, the bit assignments which are fixed for all 5600 systems are listed without parentheses. Those standard assignments depending on options are listed in parentheses and may be omitted or reassigned to system-unique functions if the option is not included. Bits are identified as S for status and M for mode.

Bit	Type	Operation
100	M	Double precision
101	M	1's complement
102	\mathbf{M}	Bit generator input from N
103	\mathbf{M}	(Adder split)
104	\mathbf{M}	(Dead start)
105	S	(Protect fault)
106	\mathbf{M}	(Enable DMA)
107	\mathbf{M}	(Enable paper tape reader clutch)
108	S	Memory parity error
109	M	Halt
110	S	(Reject)
111	M	Enable interrupt system
112	S	Overflow
113	_	Open
114	-	Open
115	S	(Paper tape feedhole)

BIT TEST

The bit test feature of the MPP provides the microprogram with the capability of testing one of up to 16 bits, each associated with a specific condition, to determine which microinstruction to execute from the next microinstruction pair. This capability is used by placing a BTU command in the T field of the microinstruction; the lower four bits of the C field determine which bit to test. If the tested bit is 1, the upper microinstruction is executed; otherwise, the lower microinstruction is executed.

The conditions to be tested are determined by the bits wired to S7 of the MPP (as shown in Figure 2-3); the determination of the conditions to be tested and their assignment to specific S7 inputs is a design function. Conditions to be tested can be any stable signal available on the backpanel of the MPP, including the setting of certain switches on the operation and maintenance panel.

Bits that can be tested include all the bits available for use in creating transforms, switch signals from the maintenance panel, and bits associated with any specialized I/O logic. Space is available on the transform module (which includes S7 for the bit test) to implement specialized test operations. For example, specialized logic can be designed to provide a bit test output of 1 when a certain pattern of bits appears in one portion of a register in the processor.

The bit test assignments shown in Table 6-1 are standard in all MPP systems; the bits not listed in the table are available for assignment by the microprogram designer.

TABLE 6-1. STANDARD MPP BIT TESTS

Bit	Operation		
3	Execute upper microinstruction if least significant bit of X register is 0 (this bit test is used in micromemory read and write operations).		
6	Execute lower microinstruction if SEL JUMP switch on operation and maintenance panel is on or if COND 1 switch on utility panel is on.		
7	Execute lower microinstruction if SEL STOP switch on operation and maintenance panel is on or if COND 2 switch on the utility panel is on.		
13	Execute lower microinstruction if LOAD MM/OFF/MEM switch on operation and maintenance panel is set to MEM.		
14	Execute lower microinstruction if ENTER/OFF/SWEEP switch on operation and maintenance panel is set to ENTER.		
15	Execute lower microinstruction if ENTER/OFF/SWEEP switch on operation and maintenance panel is set to SWEEP.		

6-8

TRANSFORMS

The transform feature of the MPP provides the microprogram with the capability to select any pattern of bits from the registers and the data transmission paths of the MPP, to form micromemory addresses to sequence the microprogram, and to set the contents of the K and N registers. The transforms are implemented on two selectors, S5 and S8 (refer to Figure 2-3), which are located on the transform module along with the bit test S7. This configuration enables the microprogram designer to specify backpanel wiring and transform module wiring that allows one microinstruction to extract bits from a register or registers, shift the bits to the required position, add base address or constant bits, and transfer the resulting data to the K or N register or form a micromemory jump address to control the microprogram execution. Without the transform feature, the above operations would require up to five microinstructions.

S5 is an eight-position selector which is used to form micromemory addresses. Eight different micromemory address (MA) constructions can be specified. One MA transform, number 7, is fixed as the lower eight bits of the output of S2; MA transform 7 is typically used with the operation and maintenance panel but can be used by the microprogram. MA transform 3 is a standard transform that is used for addressing micromemory for read and write operations and consists of the lower eight bits of the X register less the least significant bit of X; thus, for a 16-bit MPP, MA transform 3 uses bits 7 to 14 from the X register to form a micromemory address. The complement of the least significant bit of X is used as bit test 3 to select the upper or the lower microinstruction of the microinstruction pair addressed by MA transform 3 for execution.

Selector 8 is an eight-position selector which is used to select one of eight different sources for transfer to the K or to the N register. Two of the sources are fixed. Like MA transform 7, KN transform 0 is used for MPPs that have an operation and maintenance panel; KN transform 0 is fixed as the lower eight bits of S2 to allow 8-bit quantities set into the lower eight panel DATA switches to be transferred to the K or N register through S4 and S2. KN transform 6 is fixed as the lower eight bits of the MIR, enabling microinstructions to transfer quantities directly to K or N.

TRANSFORM SPECIFICATION

The transforms are specified on an MPP transform layout sheet; a sample sheet for the layout of a 16-bit MPP with an operation and maintenance panel is shown as Figure 6-1. This sheet also allows the assignment of bit test conditions. The standard required transforms are shown on this sheet, as are the standard bit test assignments.

The transform module has provisions for wiring 72 bits to the module for use in constructing transforms. These 72 bits may be selected from:

- The output of S1
- The output of S2
- Any of the bits in the SM register
- The A register
- The X register
- The I register
- Bits 20, 21, 22, 23, and 28 from the MIR
- The memory read data bus
- Switches on the operation maintenance or utility panel

The transform module consists of a printed circuit card that contains S5, S7, and S8 in addition to other control circuitry. Space is available on the card to include a small amount of additional specialized logic for special transform or bit test functions. The transform module is wired to provide inputs from the pins to the selectors to match the transform design sheet. Normally, not all 72 signals will be used to implement a particular transform specification, so it is possible to include spare signals to the connector of the transform module in the backpanel wiring, which will enable several different transform cards to be used in the same location for different applications of a particular MPP.

6-10 14232000

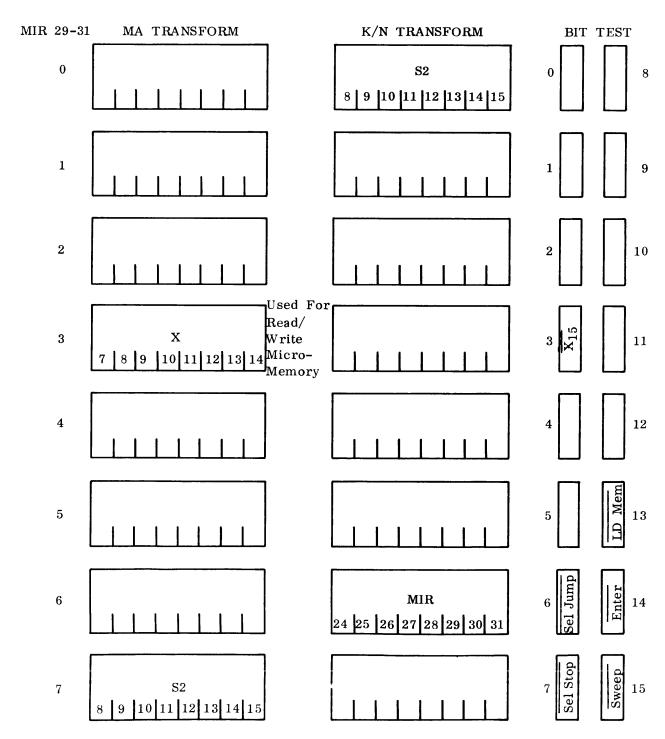


Figure 6-1. Sample Transform Design Sheet for 16-Bit MPP (Including Standard Transforms)

TRANSFORM EXECUTION

Three types of transform executions are possible in a microinstruction. These are an MA transform, K or N transform, or a combined MA transform and a K transform. The transforms are coded in the C field of a microinstruction as TMA/n for the MA transform only, as TK/n or TN/n for the K or N transform, and as TMAK/n or GITMAK/n for the combined transform; the letter n in the transform is encoded as the lower four bits of the microinstruction and is used to specify the selector position to use for S5 and S8. These same four bits are also used to specify a bit test number to be used if the T field of the instruction specifies BTU.

Execution of an MA transform uses bits 20, 21, 22, and 23 from the microinstruction (S field) to specify the page of micromemory to which to transfer control. In a two-page MPP (1024 total microinstruction memory capacity) only bit 23 is used to specify the page; bits 20, 21, and 22 are usable on the transform module to specify alternate starting locations within a page of micromemory similar to the use of bit 28 of the microinstruction in the following example.

As an example of the use of the transform, consider the emulation of a 16-bit computer by a 16-bit MPP, where the source computer has an instruction format as shown in Figure 6-2.

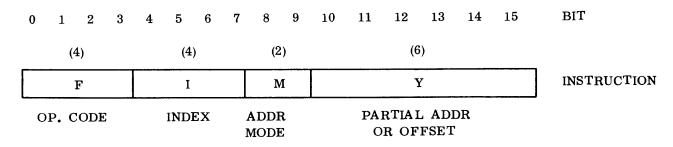


Figure 6-2. Example: 16-bit Instruction Format

F is a 4-bit operation code, I is a 4-bit index register designator, M is a 2-bit addressing mode, and Y is a 6-bit partial address or offset or a shift count in the lower five bits.

Figure 6-3 shows a transform layout that could be used in emulating a computer with this format.

It is assumed that file 1 is present and the first 16 locations will be used as index registers.

Figure 6-3. Transform Design Example

To describe the emulation process, assume that the software instruction has been read into the I register of the MPP and that the lower six bits of the instruction have also been masked off and transferred to the X register. The decoding of the operation code and the setting of the K register to specify an index register are performed by a microinstruction having TMAK/1 in the C field and BTU in the T field. The result is that the next instruction is read from a table of 16 instructions starting at location 10₈ in the micromemory. If the operation code contained in the I register is 0000, control is transferred to location 10 upper. If the operation code is 1111, control is transferred to location 17₈ lower. At the same time, the K register is set to the value of the index portion of the software instruction.

When a shift instruction is to be emulated, the shift count is transferred to the N register of the MPP to control the shift. This is accomplished by a microinstruction with a C field code of TN/2. This code transfers the lower five bits of the software instruction, contained in the I register, to the N register.

Assume that later in the execution of a software instruction it is necessary to interpret the M portion of the instruction twice for alternate actions. A microinstruction with a C field of TMA/2 results in the transfer of microprogram control to the following locations.

<u>M</u>	Micromemory location (octal)
00	40
01	44
10	50
11	54

Upper or lower microinstruction selection in the micromemory location is determined by setting the T field of the current microinstruction.

A microinstruction with a C field of TMA/10 (lower four bits of the instruction are 1010) results in a transfer of microprogram control to the following locations with upper or lower selected by the T field.

M	Micromemory location	(octal)
00	60	
01	64	
10	70	
11	74	

The above MA transform assignment allows for eight sequential microinstructions to process each M field mode of operation.

TRANSFORM EXPANSION

The design of the transform board allows for optional inclusion of an additional selector, S8' (see Figure 2-3); this provides the four additional S8' settings at the price of losing one initial S8 setting. This option provides 10 K or N transforms that can be specified by the designer. The transform numbers in the expanded version for K or N transforms are 0 (S2, lower 8 bits), 1, 2, 3, 4, 5, 8, 9, 10, and 11. Note that transform 6 is not normally specified by the designer, and transform 7 is used to extend the transforms.

An additional option allows the inclusion of two transform modules in the MPP, with dynamic selection of the module to use in executing transforms. This dynamic selection is made by setting or clearing a mode bit in the SM register. The mode bit selects one transform module if set to 0 and the other transform module if set to 1.

DATA FORMATTING

The general data transform capability, available through backpanel wiring and on the transform module, is also available for inputs of data through S4 (see Figure 2-3). The wiring to the unused S4 input can be specified to use bits from the same sources that are used for inputs to the transform module. This transmission path is enabled by using FRMT as the B field code, which gates the transformed data from the S4 input to the B input to the ALU. If the paper tape reader is not included in a 5600 system, the paper tape reader input to S4 can also be used for this data formatting. In both cases, the data transferred through S4 is word-size.

REGISTER FILES AND FILE ADDRESSING

File 1 is a 256-word file addressed by the contents of the K register, and file 2 is a 32 word file addressed by the contents of the N register. The inclusion of either or both files is optional. If a file is not present in the MPP, the microprogram designer can specify a constant to be provided when the file is addressed. For instance, the constant 1 is used in several applications. In this case, the selection of the missing file in the A or B field of a microinstruction results in the reading of the constant.

If only file 2 is included in a system, the designer can specify the use of the K register instead of the N register for addressing the file (since the N register has more non-file uses than the K register). Also in this case, the storing of data into file 1 can be controlled by the D field of the microinstruction instead of the usual S field; the storing of data into file 2 is then performed on a D field code of 100.

INTERRUPTS

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

When an interrupt is recognized, the microprogram samples the interrupt address decoder to identify the most significant interrupt. The interrupt address decoder is an asynchronous device, so the interrupt line address must be read only in the microinstruction following the interrupt test, to be sure of a correct interrupt line address. If the interrupt address is read earlier or later, there is a possibility that the address decoder output is unstable due to a newly-arrived interrupt, and a wrong address will be generated.

No standard interrupts are defined for the MPP; the use of the interrupt system and the design of interrupts are functions of the design of specialized I/O systems for the MPP. Interrupts are identified by the corresponding mask bits which are assigned to control the interrupt recognition. The bits in the mask register are identified as M100 for bit 0 of mask register 1, through the word length of the MPP system, such as M115 as bit 15 of mask register 1; M200 identifies bit 0 of mask register 2.

Interrupt addresses are generated by the interrupt address decoder according to the assignments for a 16- and 32-bit MPP given in Table 6-2.

As an example from Table 6-2, an interrupt associated with mask register 1, bit 3, for a 16-bit MPP causes an interrupt address of 12_{10} to be developed by the interrupt address decoder.

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

6-16 14232000

TABLE 6-2. MPP INTERRUPT ADDRESSES

	Interrupt Address		
Mask Bit	16-Bit MPP	32-Bit MPP	
M100	15	31	
M101	14	30	
•	•	•	
•	•	•	
•	•	•	
M115	0	16	
M116		15	
M117		14	
•		•	
•		•	
•		•	
M131		0	
M200	31	63	
M201	30	62	
•	•	•	
•	•	•	
•	•	•	
M215	16	48	
M216		47	
•		•	
•		•	
•		•	
M231		32	
		<u> </u>	

A design option in the interrupt system provides for leaving any 16-bit group of interrupts active constantly, regardless of the setting of the enable interrupt system mode bit in the SM register. In use, this option allows a set of interrupts to remain always active while the remaining interrupts may be enabled or disabled by the enable interrupt SM register bit.

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

MULTIPLE MEMORY ACCESSES

Every MPP main memory has one standard memory access port to interface with the processor. This port can be shared between the processor and an external memory user by means of the DMA logic in the processor. Optionally, the main memory can be supplied with logic for up to three additional memory access ports, to allow memory access for more than one processor or other external memory users. Access to the memory through these additional ports can be on an equal-priority rotating basis, or on a hardwired primary/secondary priority basis. Figure 6-4 shows the maximum configuration allowed by the optional multi-port logic, where up to four users (processors or other equipment) can access up to four main memories.

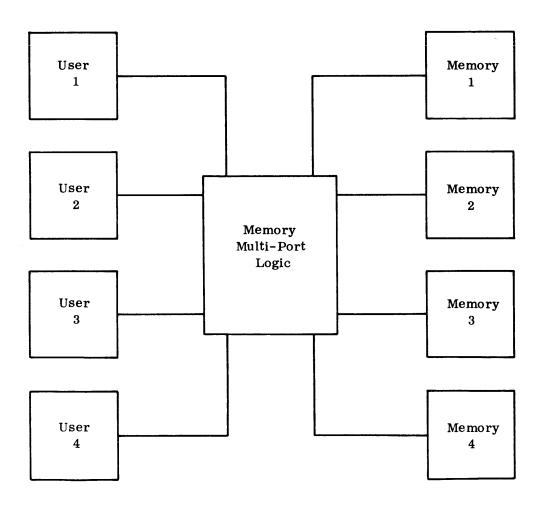


Figure 6-4. Multi-Port Logic Block Diagram

MEMORY PROTECTION

Memory protection is a special system design problem for each application that requires it. However, the following system has been implemented in one version of the MPP. Each location in main memory has associated with it a protect bit. If this bit is 0, the location is unprotected and any user can read or write in that location. If the bit is 1, the location is protected and any user can read the word, but only a user with protect status can write into the location. Reading or writing the location does not change the protect bit. The memory control unit is designed to handle the above logical decisions. The MPP is provided with microinstruction commands to control the protect system. These commands are:

- Write protect bit. This command is coded in the C field of a microinstruction as WRPB and writes the least significant bit of the output of S1 as the protect bit at the main memory address specified by the output of the ALU.
- Set protect status. This command is coded in the C field of a microinstruction as SPS and sets the protect status for the MPP so it can write in protected locations of main memory.
- Clear protect status. This command is coded in the C field of a microinstruction as CPS and clears the protect status for the MPP; any attempt to write in a protected location will result in a rejection of the operation by the memory and will cause the protect fault bit in the SM register to be set. The microprogram will then reflect the protect violation back to the software instructions being emulated.

MEMORY PARITY

The memory parity feature is a design option for the MPP. The modular main core memory provides storage in 9-bit increments; thus a 16-bit MPP has two extra bits per word and a 32-bit MPP has four extra bits. One or more of the extra bits may be assigned to the main memory parity function. When storing a word, the memory generates the correct parity bit or bits and stores the parity along with the word. On a read of the word from memory, the parity is checked by the memory control logic and, if a parity error is detected, the parity error bit in the SM register is set for handling by the microprogram.

DOUBLE PRECISION LOGIC

The double precision option has been explained earlier in this manual. If the double precision option is not incorporated into an MPP, the commands used to reference the A^* , X^* , and Q^* registers are available to reference specialized I/O registers, if included in the system.

DEAD START

The dead start option consists of the dead start module, which contains a read-only micromemory. The dead start micromemory contains storage for either 64 or 512 microinstructions plus the necessary logic to write into the read/write micromemory. The microprogram included in the read-only memory is prepared to meet the specific 5600 system requirements and is inserted in the memory at manufacture.

The determination of whether to execute from the dead start micromemory or the regular processor micromemory is controlled by the setting of the dead start mode bit in the SM register. If this bit is 1, the MPP references the dead start micromemory to read the microinstructions but references the regular micromemory to read or store micromemory words. If the dead start mode bit is 0, all operations are performed with the regular micromemory.

The dead start mode bit is set to 1 by any of the following conditions:

- 1) Power on master clear if the operation and maintenance panel is locked out
- 2) Setting the LOAD MM/OFF/MEM switch on the maintenance panel to MM
- 3) By the microprogram in regular micromemory setting the dead start mode bit in the SM register

For conditions 1 and 2, a master clear is assumed and operation of the dead start microprogram starts at location 0 in the dead start read-only micromemory.

For condition 3, the microinstruction following the microinstruction that sets the dead start mode bit to 1 is executed from the regular micromemory. Then the next microinstruction is executed from the dead start micromemory at the last regular micromemory address plus 1.

Conversely, if the dead start bit is cleared to 0 by the microprogram in the dead start micromemory, one additional microinstruction is executed in the dead start memory before returning to the regular micromemory.

When executing from the dead start memory, page jump instructions do not transfer control out of the dead start micromemory but will change the page register for referencing regular micromemory.

6-20 14232000

BREAKPOINT

The optional breakpoint switches on the operation and maintenance panel and the breakpoint logic allow microprogram execution to be stopped on either main memory references or on micromemory references.

For main memory references, the BREAKPOINT switch is set to MEM and the thumbwheel switches are set to the main memory address to be monitored. When a reference is made to the monitored address, microprogram execution halts upon completion of the microinstruction following the microinstruction initiating that reference.

For an MPP in a computer emulator application, microprogram execution can be halted after a software instruction is read from main memory. A typical microprogram segment for a software instruction read operation is:

F	A	В	D	S/C
ADD+	P	ZERO	Р	READ
А • В	ME M	F2	X	GITMAK/X

In this case, microprogram execution is halted after the software instruction from main memory is read into the I and X registers and after the micromemory address has been updated by the MA transform.

When data is stored into the monitored main memory address, the breakpoint logic halts microprogram execution after the microinstruction following the store microinstruction.

For an operand read from main memory, the operand is normally lost from main memory when microprogram execution halts on the completion of the microinstruction following the read microinstruction. However, the operand should be available in the registers of the MPP. It is necessary to perform a restart to continue from this condition.

To monitor micromemory references, the BREAKPOINT switch is set to MM and the thumbwheel switches are set in the following manner. The left-most thumbwheel is ignored. The next thumbwheel is used to specify a micromemory page and upper or lower microinstruction. The two right-most thumbwheels specify the address of the microinstruction pair in the selected micromemory page.

The setting of the thumbwheel that specifies page and upper or lower microinstruction is determined by taking twice the page number and adding 0 for an upper microinstruction or 1 for a lower microinstruction. Thus, to breakpoint at the lower microinstruction of the microinstruction pair at address AB in page 1 (described as octal 653 lower or hexadecimal 1AB lower), the page select thumbwheel is set to 3 and the two right-most thumbwheels are set to AB. The complete thumbwheel setting is 03AB.

As a further example:

- Upper location 3, setting 0003
- Lower location 3, setting 0103

MICROMEMORY OPERAND REFERENCES

The MPP has the capability of transferring information between the micromemory and the registers of the processor. The transfer of data from micromemory to registers is used to obtain constants from the micromemory for use by the microprogram. The transfer from the registers to the micromemory is possible only if the micromemory is a read/write micromemory and if the MM ENABLES/WRITE switch on the maintenance panel is on. This transfer is primarily used for loading a read/write micromemory using a dead start program, but also can be used to utilize micromemory as temporary storage.

Micromemory is addressed as one to 16 pages of 256 words each, where each word is 64 bits and is divided into an upper 32-bit word and a lower 32-bit word. A 32-bit or larger MPP can reference 32-bit micromemory words by specifying page, address, and upper or lower microinstruction. For 16- to 28-bit MPPs, only 16 bits can be transferred at a time and an additional specification of the upper or lower 16 bits of each 32-bit micromemory half word is required to address all bits of micromemory.

The standard minimum transform for the MPP assigns the micromemory addressing function to the X register in which a 9-bit number in X specifies the micromemory word address in the upper eight bits and specifies the upper/lower 32-bit word selection as the lower order bit of X.

Instructions for micromemory operand references may be in an upper or lower micro-instruction; the next microinstruction executed following the referencing microinstruction is always the upper microinstruction of the next sequential location. The page of the operand is the current setting of the page portion of the P/MA register.

For execution from the regular micromemory, this means that the microinstructions referencing a micromemory operand must reside in the same page as the operand. Execution from the dead start micromemory allows setting of the page register to reference all of the regular micromemory for operand references.

6-22 14232000

The following microinstructions read and write operands from micromemory. The read microinstruction results in the transfer of the 16- or 32-bit operand to the F register. The write microinstruction transfers 16 or 32 bits from the register specified in the B field to the micromemory. The X register contains the address for micromemory as described previously.

M	F	A	В	D	Т	S	С	
01	-В		MMU	NOP	BTU	B'	3	Read MM
01	В		$\left\{\begin{matrix} \mathrm{F} \\ \mathrm{F1} \\ \mathrm{F2} \\ \mathrm{Q} \end{matrix}\right\}$	MMU	вти	D'	3	} Write MM

For a 32-bit MPP, all 32 bits are transferred to or from a 32-bit micromemory half word. For a 16- to 28-bit MPP, the least significant 16 bits of the register are involved in the transfer. To reference the lower 16 bits of the 32-bit micromemory half word, the code MML replaces MMU.

MICROMEMORY LOAD OPTION

A paper tape reader micromemory load module is available as an option in the MPP. This option consists of one pluggable module which replaces the dead start option and provides logic to load the read/write micromemory from the Control Data 370 paper tape reader under maintenance panel control. Using this option, micromemory is loaded by setting the MM ENABLES/WRITE switch on and the LOAD MM/OFF/MEM switch to LOAD MM. Then the MASTER CLEAR and START switches are depressed. The punched paper tape (in special load format) mounted on the 370 paper tape reader is loaded into the micromemory.

14232000 6-23/6-24

MICROINSTRUCTION CLASSES

The basic MPP microinstruction execution time is 168 nanoseconds. Some microinstructions have longer execution times to allow certain operations to be completed. The microinstructions have been grouped according to execution times as types A, B, C, D, and E as shown in Table 7-1.

TABLE 7-1. MICROINSTRUCTION EXECUTION TIMES

Microinstruction Class	Execution Time in Nanoseconds
A	168
В	224
C	280
D	336
E (shift microinstruction)	168 + 55n (where n is number of shifts)

The classification of microinstructions is shown in Figure 7-1; the factors which increase microinstruction timing are: using 1's complement arithmetic, performing an add or subtract, performing an MA transform, performing tests, and using alternate A- or B-field codings. For example, the instruction

is identified as entry 0111 in the figure (a class D microinstruction) for 2's complement arithmetic or as entry 1111 (again in class D) for 1's complement arithmetic.

The current microprogram assembler provides an automatic classification of instructions on the assembly listing; class A microinstructions are not identified, but classes B, C, D, and E are identified for timing purposes.

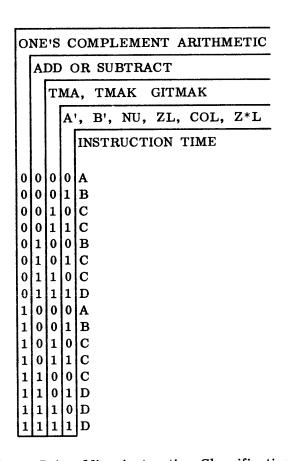


Figure 7-1. Microinstruction Classification

MICROINSTRUCTION AND MEMORY TIMING

Analysis of a microprogram for execution time starts by classifying each of the microinstructions as type A, B, C, D, or E. This is done by using the microinstruction classification table or by examining the assembler output listing.

The sequence of microinstructions to be timed is written down and the main memory read or write microinstructions are marked. The total time required is then calculated from memory command to memory command. Each read or write command adds 300 nanoseconds to the total time. The times are added, starting with the microinstruction after the read or write command, and then 300 nanoseconds are added for the following read or write command.

If the total time, including the 300-nanosecond add-on for read or write, is less than the memory complete cycle time (1 microsecond, or 1,000 nanoseconds), take 1 microsecond as the execution time for that path. If the total execution time is greater than 1 microsecond, the execution time for that segment is the calculated time.

As an example, calculate the execution time for the microprogram segment consisting of microinstruction types

C A C A C

where a main memory read is included in the first and the third C type microinstruction. Starting with the microinstruction after the first read, going through the next read microinstruction, and writing down the times, thus:

Instruction	Time	
Α	168	
C	280	
Α	168	
\mathbf{c}	280	
Memory read delay	300	
Total	1,196 nanoseconds	3

Since the calculated execution time is greater than the 1-microsecond main memory cycle time, the calculated execution time is the actual execution time.

14232000 7-3/7-4

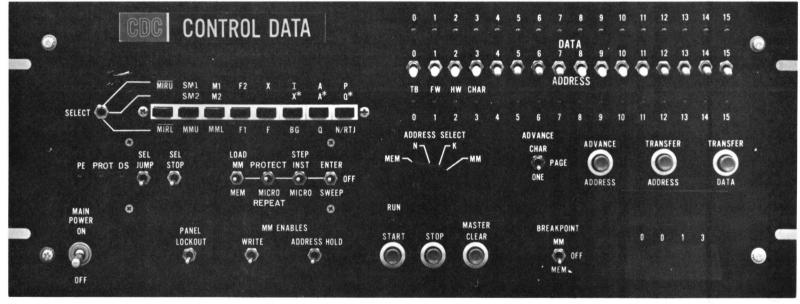


Figure 8-1. 16-Bit MPP Maintenance Panel

5600 PROCESSOR CONTROLS AND INDICATORS

Figure 8-1 shows a typical 16-bit 5600 processor operation and maintenance panel. The panel is optional with some 5600 systems; when included, it allows the operator to observe and manually control many processor functions during both normal and maintenance operations. Listed Table 8-1 is each panel control and indicator and its function for the typical 16-bit MPP. Note that controls and indicators can be added, deleted, or functionally changed to suit the requirements of a specific application; refer to the system customer engineering manual for exceptions to the following list.

8-1

8

TABLE 8-1. MPP CONTROLS AND INDICATORS

Panel Nomenclature	Device	Function
DATA 0 to 15	Indicator lights	Displays current binary data as determined by positions of SELECT switch and pushbuttons.
DATA 0 to 15	Toggle switches	Specify input data or address. Up position = 1; down position = 0 .
		When TRANSFER DATA switch is depressed, number set in DATA switches is transferred to register specified by SELECT switch and pushbuttons.
		When TRANSFER ADDRESS switch is depressed, number set in DATA switches is transferred to address register specified by ADDRESS SELECT switch.
ADDRESS 0 to 15	Indicator lights	Indicate address being referenced. Address displayed is specified by ADDRESS SELECT switch. Indicators at bit positions 0, 1, 2, and 3 indicate state of test bit (TB), full word (FW), half word (HW), character (CHAR) signals. The FW, HW, and CHAR indicators comprise a character counter that indicates which two 8-bit characters are addressed (not applicable for MPPs containing dead start option).
ADDRESS SELECT	Rotary switch	Specifies address for display on ADDRESS indicators and for transfer address or advance address operations.
		• MEM: main memory
		• N: N register
		• K: K register
		MM: micromemory address register

TABLE 8-1. MPP CONTROLS AND INDICATORS (Cont.)

Panel Nomenclature	Device	Function
ADDRESS SELECT (Cont.)	Rotary switch	Contents of N is displayed on DATA indicators 0-7. When MEM is specified, address of last main memory read or write operation is displayed on ADDRESS indicators (only for MPPs with breakpoint option). Contents of K and MM address register is displayed on ADDRESS indicators.
ADVANCE	Toggle switch	Specifies that MM address will be advanced by a character (CHAR), by an MM page (PAGE), or by one MM instruction (ONE) when ADVANCE ADDRESS is depressed. (CHAR position not used in MPPs with dead start option.)
ADVANCE ADDRESS	Pushbutton	Causes address specified by ADDRESS SELECT to advance by one when depressed, for N and K. Causes address to advance by one character, one page, or one instruction for MM. MEM can not be advanced by depressing ADVANCE ADDRESS pushbutton.
TRANSFER ADDRESS	Pushbutton	Transfers quantity set into DATA switches to N, K, or MM, as specified by ADDRESS SELECT switch. TRANSFER ADDRESS has no effect when ADDRESS SELECT specifies MEM.
TRANSFER DATA	Pushbutton	Transfers quantity set into DATA switches to destination specified by SELECT switch and pushbuttons, if destination can accept data from maintenance panel (refer to SELECT).
SELECT	Toggle switch and pushbuttons	
• MIRU, MIRL		Specifies that complement of quantity set into DATA switches is to be transferred to MIR upper portion (bits 0-15) or to MIR lower portion (bits 16-31) when TRANSFER DATA is depressed. Contents of MIR is not displayed on DATA indicators.

TABLE 8-1. MPP CONTROLS AND INDICATORS (Cont.)

Panel Nomenclature	Device	Function
• SM1, SM2, MMU		Specifies status/mode register 1 (SM1), SM2, or upper half of micromemory location (MMU) for display or data transfer.
		NOTE
		Data can be transferred to MM only if MPP has a read/write MM and MM ENABLES/WRITE switch is in up position.
• M1, M2, MML		Specifies mask register 1 (M1), M2, or lower half of MM location (MML) for display or data transfer.
• F2, F1		Specifies register file 2 (F2) or F1 for display or data transfer.
• X, F		Specifies X or F register for display or data transfer.
● I, X*, BG		Specifies I or X* register for display or data transfer. Specifies bit generator (BG) for display. Output of BG controlled by microinstruction, or by lower five bits of N register if flag 3 of SM1 is set.
• A, A*, Q		Specifies A, A*, or Q register for display or data transfer.
● P, Q*, N/RTJ		Specifies P or Q* register for display or data transfer. When N/RTJ is selected, contents of N is displayed on DATA indicators 0-7 and contents of RTJ is displayed on DATA indicators 8-15. N can be loaded by transfer address operation if ADDRESS SELECT is set to N. RTJ is alterable only by execution of microinstruction.
PE	Indicator light	Indicates a main memory parity error (PE) when lighted.
PROT	Indicator light	Indicates MPP is in protect mode (systems containing optional program protect logic).

TABLE 8-1. MPP CONTROLS AND INDICATORS (Cont.)

Panel Nomenclature	Device	Function
Tanoz Homonozawa 10	2007.00	
DS	Indicator light	Indicates MPP is in dead start (DS) mode.
SEL JUMP	Toggle switch	In up position, interpreted by firmware to indicate selective jump is required.
SEL STOP	Toggle switch	In up position, interpreted by firmware to indicate selective stop is required.
LOAD MM/OFF/MEM	Toggle switch	Used for selecting micromemory (MM) or main memory (MEM) for loading. DS indicator lights when set to LOAD MM. Loading starts when START switch is depressed.
PROTECT/OFF/ MICRO REPEAT	Toggle switch	Enables optional protect system when set to PROTECT. When set to MICRO REPEAT, instruction currently in MIR register is repeated continuously when START is depressed.
STEP INST/OFF/ MICRO	Toggle switch	Specifies that each time START is depressed, program execution will halt after executing one software instruction (INST) or after executing one microinstruction (MICRO).
ENTER/OFF/SWEEP	Toggle switch	Switch whose position can be sensed by microinstructions.
PANEL LOCKOUT	Toggle switch	In up position, disables main function switches except MAIN POWER. SELECT and ADDRESS SELECT still operable for display purposes.
MAIN POWER ON/OFF	Toggle switch	Controls AC power input to processor enclosure.
	•	
	·	

TABLE 8-1. MPP CONTROLS AND INDICATORS (Cont.)

Panel Nomenclature	Device	Function
MM ENABLES	Toggle switches	
• WRITE		In up position, enables writing into read/write micromemory through DATA and TRANSFER DATA switches.
• ADDRESS HOLD		In up position, prevents MIR from being cleared when MASTER CLEAR depressed. Allows microprogram to be started at location specified by MM address register.
START	Pushbutton	Starts program execution.
RUN	Indicator light	Indicates processor is running when lighted.
STOP	Pushbutton	Stops program execution when next micro-instruction containing a halt code is executed.
MASTER CLEAR	Pushbutton	Clears MPP logic circuits.
BREAKPOINT MM/ OFF/MEM	Toggle switch	Specifies that processor is to halt at micromemory address (MM) or at main memory address (MEM), in systems containing breakpoint option.
Breakpoint	Thumbwheel switches	Specify hexadecimal address at which processor is to stop, if enabled by BREAKPOINT MM/OFF/MEM switch.

POWERS OF TWO TABLE

```
2" " 2"
                                     0
                                         1.0
                                          0.5
                                          0.125
                                          0.062 5
                                          0.031 25
                               32
                                         0.015 625
                                          0.007 812 5
                              128
                             256
                                         0.003 906 25
                              512
                                          0 001 953 125
                                         0 000 976 562 5
                           1 024
                                     10
                                          0.000 488 281 25
                                    11
                          2 048
                                          0.000 244 140 625
                           4 096
                                    12
                         8 192
16 384
                                         0 000 122 070 312 5
0 000 061 035 156 25
                         32 768
                                         0.000 030 517 578 125
                         65 536
                                         0 000 015 258 789 062 5
                                   16
                        131 072 17
262 144 18
                                         0.000 007 629 394 531 25
0.000 003 814 697 265 625
                        524 288
                                   19 0 000 001 907 348 632 812 5
                                          0 000 000 953 674 316 406 25
                                         0 000 000 476 837 158 203 125
0 000 000 238 418 579 101 562 5
                     2 097 152 21
4 194 304 22
                     8 388 608 23 0 000 000 119 209 289 550 781 25
                                         0 000 000 059 604 644 775 390 625
0 000 000 029 802 322 387 695 312 5
0 000 000 014 901 161 193 847 656 25
0 000 000 007 450 580 596 923 828 125
                    16 777 216
                    33 554 432 25
                  134 217 728 27
                  268 435 456
                                          0 000 000 003 725 290 298 461 914 062 5
                                          0 000 000 001 862 645 149 230 957 031 25
                  536 870 912
                                         0 000 000 000 931 322 574 615 478 515 625
0 000 000 000 465 661 287 307 739 257 812 5
                1 073 741 824 30
               2 147 483 648
               4 294 967 296
                                          0 000 000 000 232 830 643 653 869 628 906 25
                                          0 000 000 000 116 415 321 826 934 814 453 125
0 000 000 000 058 207 660 913 467 407 226 562 5
               8 589 934 592
              17 179 869 184
                                   34
              34 359 738 368
                                          0.000 000 000 029 103 830 456 733 703 613 281 25
            137 438 953 472 37
274 877 906 944 38
                                         0 000 000 000 007 275 957 614 183 425 903 320 312 5
0 000 000 000 003 637 978 807 091 712 951 660 156 25
                                         0.000 000 000 001 818 989 403 545 856 475 830 078 125
             549 755 813 888 39
                                          0 000 000 000 000 909 494 701 772 928 237 915 039 062 5
                                          0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
          2 199 023 255 552 41
          4 398 046 511 104 42
                                         0 000 000 000 000 227 373 675 443 232 059 478 759 765 625
0 000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
          8 796 093 022 208 43
                                          0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
         17 592 186 044 416 44
                                          0 000 000 000 000 000 084 34 18 868 808 104 809 941 406 25
0 000 000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
0 000 000 000 000 000 114 210 854 715 202 003 717 422 485 351 562 5
0 000 000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
        35 184 372 088 832 45
         70 368 744 177 664 46
       140 737 488 355 328 47
                                          0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625
       281 474 976 710 656 48
                                          0 000 000 000 000 001 776 356 839 400 250 464 677 810 668 945 312 5
       562 949 953 421 312 49
       125 899 906 842 624
    2 251 799 813 685 248 51
                                          0 000 000 000 000 000 444 089 209 850 062 616 169 452 667 236 328 125
                                          0 000 000 000 000 000 222 044 604 925 031 308 084 726 333 618 164 062 5
     4 503 599 627 370 496 52
                                          0 000 000 000 000 000 111 022 302 462 515 654 042 363 166 809 082 031 25 0 000 000 000 000 000 055 511 151 231 257 827 021 181 583 404 541 015 625
     9 007 199 254 740 992
    18 014 398 509 481 984
                                     54
   36 028 797 018 963 968
                                          0 000 000 000 000 000 027 755 575 615 628 913 510 590 791 702 270 507 812 5
                                         0 000 000 000 000 000 000 013 877 787 807 814 456 755 295 395 851 135 253 906 25
0 000 000 000 000 000 000 006 938 893 903 907 228 377 647 697 925 567 626 953 125
0 000 000 000 000 000 000 003 469 446 951 953 614 188 823 848 962 783 813 476 562 5
0 000 000 000 000 000 000 001 734 723 475 976 807 094 411 924 481 391 906 738 281 25
   72 057 594 037 927 936
                                     56
  144 115 188 075 855 872
288 230 376 151 711 744
                                     58
  576 460 752 303 423 488
1 152 921 504 606 846 976 60 0000 000 000 000 000 867 361 737 988 403 547 205 962 240 695 953 369 140 625
```

CONVERSIONS

TABLE B-1. HEXADECIMAL DIGITS

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	Α	1010
11	В	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Integer Conversion

1) Convert Binary to Hexadecimal

Binary number = $1101 \ 0011 \ 0100$

Group binary number in groups of four starting from the right. Write hexadecimal equivalent for each group of four.

1101 0011 0100

D 3 4

1101 0011 0100
$$_2 = D34_{16}$$

2) Convert Hexadecimal to Binary

Hexadecimal number = D3AE5

Write 4-bit binary equivalent for each hexadecimal digit.

 $D3AE5_{16} \equiv 1101 \ 0011 \ 1010 \ 1110 \ 0101_2$

3) Convert Hexadecimal to Decimal

Hexadecimal number = D34

Make successive cumulative multiplications from left to right, adding units position.

$$D34_{16} \equiv 3380_{10}$$

4) Convert Decimal to Hexadecimal

Decimal number = 3380

Divide and collect remainder in reverse order.

Fractional Conversion

1) Convert Binary to Hexadecimal

Binary Number = 0 1010 1011 1100 (13-bit machine)

a) Assume that the binary point is between first and second bit.

Sign bit =
$$0.1010 \ 1011 \ 1100$$

b) Group in groups of four from the binary point.

0.1010 1011 1100

A B C

01010101111100
$$_2 = .ABC_{16}$$

2) Convert Hexadecimal to Binary

Hexadecimal number = .E3A

Write 4-bit binary equivalent of each hexadecimal digit.

.E 3 A
.1110 0011 1010
.E3A₁₆
$$\equiv$$
 0.1110 0011 1010₂ (13-bit machine)

3) Convert Hexadecimal to Decimal

Hexadecimal number = .8A7

a) Assume integer number and use integer technique.

$$8 = 8$$

$$\frac{x16}{128}$$

$$A = \frac{+10}{138}$$

$$\frac{x16}{2208}$$

$$7 = \frac{+7}{2215}$$

b) Divide result by 16^n (n is the number of fraction positions).

$$16^3 = 4096$$

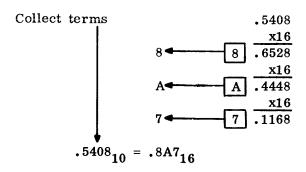
$$\begin{array}{c|c}
 .540771 \\
4096 \overline{\smash{)}2215.000000}
\end{array}$$

$$.8A7_{16} = .540771_{10}$$

4) Convert Decimal to Hexadecimal

Hexadecimal number = 0.5408

Collect integer parts of product in order of calculation.



HEXADECIMAL CONVERSION TABLES

нех.	DEC.	HEX.	DEC.	нех.	DEC.	н	EX.	DEC.	HEX.	DEC.	
1	1	10	16	100	256	100	00	4096	10000	65536	
2	2	20	32	200	512	200	00	8192	20000	131072	
3	3	30	48	300	768	300	00	12288	30000	196608	
4	4	40	64	400	1024	4000		16384 40000		262144	
5	5	50	80	500	1280	500	5000 20480 50000		50000	327680	
6	6	60	96	600	1536	600		24576	60000	393216	
7	7	70	112	700	1792	700		28672	70000	458752	
8	8	80	128	800	2048	800		32768	80000	524288	
9	9	90	144	900	2304	900		36864	90000	589824	
A	10	A0	160	A00	2560	A00		40960	A0000	655360	
В	11	В0	176	В00	2816	В00		45056	B0000	720896	
C	12	CO	192	C00	3072	C00		49152	C0000	786432	
D	13	DO	208	D 0 0	3328	D00		53248	D0000	851968	
E	14	EΟ	224	E00	3584	EOC		57344	E0000	917504	
F	15	FO	240	F00	3840	FOC	00	61440	F0000	983040	
HEX.		DEC.		HEX.	DEC	•		HEX.	DEC.		
100000		1048576	1	000000	16777	216		10000000	26843	15456	
200000		2097152	2	000000	33554432			20000000	536870912		
300000		3145728	3	000000	50331648		ľ	30000000	805306368		
400000		4194304	4	000000	67108864			40000000	1073741824		
500000		5242880	1	000000	83886	080	50000000		134217	7 280	
600000		6291456		000000	100663			60000000	1610612736		
700000		7340032		000000	117440512			70000000	187904	8192	
800000		8388608		000000	134217728			80000000	214748	3648	
900000		9437184		000000	150994944		1	90000000	2415919104		
A00000		10485760		000000	167772			A000000	2684354560		
B00000		1534336		000000	184549			В0000000	295279		
C00000		2582912		000000	201326			C0000000	322122		
D00000 E00000		13631488 14680064		000000	218103			D0000000	348966		
F00000		5728640		000000	234881	-		E0000000	375809		
100000		. 5 / 20040			251658	240		F0000000	402653	1840	

14232000 C-1/C-2

ADDITION

	0	1	2	3	4	5	6	7	8	9	A	В	С	D	Ε	F
0	0	-	2	3	4	5	6	7	8	9	A	В	C	٥	Ε	F
1	-	2	3	4	5	6	7	8	9	A	В	С	٥	E	F	10
2	2	3	4	5	6	7	8	•	A	В	U	D	E	Ł	9	11
3	3	4	5	6	7	8	9	4	8	U	۵	Ε	£	9	Η	12
4	4	5	6	7	8	9	A	8	С	۵	Ε	F	9	Ξ	12	13
5	5	6	7	8	9	Α	В	U	٥	Ε	F	10	=	12	13	14
6	6	7	8	9	A	В	O	D	Ε	F	10	H	12	13	14	15
7	7	8	9	Α	В	С	D	Ε	F	ō	н	12	13	14	15	16.
8	8	9	Α	В	С	D	Ε	F	10	Ξ	12	13	14	5	16	17
9	9	Α	В	С	٥	Ε	F	9	Ξ	12	13	14	15	16	17	18
A	A	В	С	D	Ε	F	10	Ξ	12	13	14	15	16	17	18	19
В	В	С	D	Ε	F	10	ш	12	13	14	15	16	17	18	19	IA
С	С	D	Ε	F	10	ш	12	13	14	15	16	17	18	19	IA	IB
D	۵	Ε	F	īO	н	12	13	14	15	16	17	18	19	IA	IB	ıc
Ε	E	F	10	11	12	13	14	15	16	17	18	19	1A	18	IC	10
F	F	10	Ξ	12	13	14	15	16	17	18	19	IA	18	IC	ID	ΙE

MULTIPLICATION

1'S COMPLEMENT AND 2'S COMPLEMENT NUMBERS

E

In the MPP, arithmetic operations can be performed on numbers expressed either in 1's complement or 2's complement form. In 2's complement form, the negative of a number, A, is defined by 2^n -A, where N is the number of bits in the processor word. For example, in a 5-bit processor, the number -6 is expressed in 2's complement form as:

$$\begin{array}{rrr}
100000 & (2^5) \\
-00110 & (-6) \\
\hline
11010 & (-6 in 2's complement form)
\end{array}$$

In 1's complement form, the negative of a number, A, is defined by 2^n-1-A . The number -6 is expressed in 1's complement form as:

11111
$$(2^{5}-1)$$

 -00110 (-6)
11001 $(-6 \text{ in 1's complement form})$

The numbers from -15 to 15 in 1's complement and 2's complement form are contained in the following listing. Note that there are two representations for zero in 1's complement form; either may result, depending on the particular computational sequence.

14232000 E-1

<u>Decimal</u>	1's Complement	2's Complement
15	01111	01111
14	01110	01110
13	01101	01101
12	01100	01100
11	01011	01011
10	01010	01010
9	01001	01001
8	01000	01000
7	00111	00111
6	00110	00110
5	00101	00101
4	00100	00100
3	00011	00011
2	00010	00010
· <u>1</u>	00001	00001
0	00000	00000
0	11111	_
-1	11110	11111
-2	11101	11110
-3	11100	11101
-4	11011	11100
- 5	11010	11011
-6	11001	11010
- 7	11000	11001
-8	10111	11000
-9	10110	10111
-10	10101	10110
-11	10100	10101
-12.	10011	10100
-13	10010	10011
-14	10001	10010
-15	10000	10001
	<u>.</u>	

E-2 14232000

COMMENT SHEET

MANUAL TITLE	
PUBLICATION NO.	REVISION
FROM:	NAME:
	BUSINESS ADDRESS:

COMMENTS:

This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number references and fill in publication revision level as shown by the last entry on the Record of Revision page at the front of the manual. Customer engineers are urged to use the TAR.

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Minneapolis Military Products Division 3101 East 80th Street Minneapolis, Minnesota 55440

Attn: Publications Department

FOLD

FOLD



e____

CORPORATION

CORPORATE HEADQUARTERS, 8100 34th AVE. SO., MINNEAPOLIS, MINN. 55440 SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD