60429103

CONTROL DATA
CORPORATION

CDC®

AE121-B	MATRIX ALGORITHM PROCESSOR
AE125-B	MATRIX ALGORITHM PROCESSOR
AT304-A	SQUARE ROOT UNIT
AT305-A	DIVIDE UNIT
AT306-A	ADD/SUBTRACT UNIT
AT307-A	MULTIPLY UNIT
AT397-A	ENHANCED-FFT ADD/SUBTRACT UNIT
AT400-A	FFT ENHANCEMENT
BB318-A	8K MEMORY MODULE
BB320-A	24K MEMORY EXPANSION
GD803-A	POWER SUPPLY EXPANSION

GENERAL DESCRIPTION
OPERATION AND PROGRAMMING
THEORY OF OPERATION

Volume 1 of 3

HARDWARE MAINTENANCE MANUAL

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

REVISION RECORD

REVISION	DESCRIPTION
A (10-15-76)	Manual released.
B (12-15-77)	Manual revised; includes Field Change Order 38344. Pages iii, iv, 2-1, 2-2, 2-37 in volume 1; iii, iv, 4-2, sheet PWR of section 5 in volume 2; iii, iv, 8-48, and 8-49 in volume 3 are revised.
C (3-31-78)	Manual revised; includes Engineering Change Order 37207 and Field Change Order 37233. Pages iii, iv in volume 1; iii, iv, sheets PPL, PPM, ADD1 3.1, ADD1 3.2, ADD2 3.1, A/D2 3.7, BUS 3.1, BUS 3.4, CHN1 3.1, CHN1 3.5, CT7 3.6, CT7 3.10, CT9 3.4, ECS1 3.1, ECS1 3.4, ECS1 3.7, MUL1 3.3 in volume 2; and pages iii, iv, and 8-13 in volume 3 are revised.
D (10-24-78)	Manual revised; includes Publications Change Order 39518. Front cover, title page, iv, v, 2-31 in volume 1; front cover, title page, iv, v, sheet MAP 1.5 in volume 2; front cover, title page, iv, v, and 8-33 in volume 3 are revised. Pages ADD1 3.1A, sheet ADD1 3.1A, ADD2 3.1A, sheet ADD2 3.1A, ADD2 3.2A, and sheet ADD2 3.2A in volume 2; and 8-34 in volume 3 are added.
Publication No. 60429103	

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

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4201 North Lexington Avenue
St. Paul, Minnesota 55112

or use Comment Sheet in the back of this manual.

MANUAL TO EQUIPMENT LEVEL CORRELATION SHEET

SHEET 1 OF 2

		EQUIPMENTS					
MANUAL REV	FCO OR ECO	AE121-B	AT304-A	AT305-A	AT306-A	AT307-A	BB318-A
A	Released	B01	A01	A01	A01	A01	A01
B	FCO38344	B02	A01	A01	A01	A01	A01
C	ECO37207	B03	A01	A01	A01	A01	A01
	FCO37233	B04	A01	A01	A02	A02	A01

MANUAL TO EQUIPMENT LEVEL CORRELATION SHEET (Cont'd)

SHEET 2 OF 2

MANUAL REV	FCO OR ECO	EQUIPMENTS					
		BB320-A	GD803-A	AE125-B	AT397-A	AT400-A	
A	Released	A01	A01	-	-	-	
B	FCO38344	A01	A01	-	-	-	
C	ECO37207	A02	A01	-	-	-	
	FCO37233	A02	A01	-	-	-	
D	ECO39518	-	-	B01	A01	A01	

PREFACE

This manual provides hardware maintenance information for the CDC® AE121-B/AE125-B Matrix Algorithm Processor (MAP) and its related options. Volume 1 of the manual contains equipment descriptions, hardware reference information, and operational theory. Volume 2 contains installation procedures, diagrams, and maintenance information. Volume 3 contains parts data and wire lists.

One of the components of MAP is the NCR M63-2-STD cassette transport. NCR manual MS-413 provides training and field service information for this transport.

The MAP hardware maintenance manual is intended for use mainly by customer engineers. The following table defines the user and purpose of supplementary and related documents. Control Data assumes no responsibility for the contents of the uncontrolled documents listed in the table.

NOTES

1. The equipments described in this manual were previously documented on a quotation for special equipment (QSE) basis rather than on an equipment configurator (EC) basis. Therefore, when requesting additional parts data or wire lists (as described in volume 3, sections 8 and 9), please include with your request the appropriate EC code(s) rather than a QSE number.
2. The AT400-A FFT Enhancement may be installed in the AE121-B MAP and used with modified MAP controlware to reduce processing time for Fast Fourier Transform (FFT) algorithms. The AE125-B MAP includes the FFT enhancement and is equivalent to an AT400-A-equipped AE121-B. In this manual, the words (enhanced-FFT MAP) refer either to an AE125-B or to an AT400-A-equipped AE121-B.
3. The AT397-A Enhanced-FFT Add/Subtract Unit should be used when adding an additional add/subtract unit to an enhanced-FFT MAP.

Document	Type	Part No.	User	Purpose
CYBER 70 Models 72/73/74 6000 Computer Systems I/O Specifications	Controlled	60352500	Customer engineer	Describes the communication link between MAP and the host computer.
ECL 10,000 Series Circuit Description Manual	Controlled	60417700	Customer engineer	Describes the ECL logic elements used in MAP and the symbols which represent them.
TTL Microcircuits Manual	Controlled	60406000	Customer engineer	Describes the TTL logic elements used in MAP and the symbols which represent them.
6000 MAP III Controlware ERS	Uncontrolled	12104400	System programmer	Defines the external interface to MAP internal controlware.
6000 MAP III Assembler ERS	Uncontrolled	12104300	Controlware programmer	Describes the assembler used to generate internal controlware.
MAP III Command Diagnostics ERS	Uncontrolled	12104200	Customer engineer	Describes diagnostics used to test the MAP control unit.
MAP III Memory Test (QMM) ERS	Uncontrolled	12104100	Customer engineer	Describes diagnostics used to test MAP data storage.
MAP III Test Functional Units (TFU) ERS	Uncontrolled	22836600	Customer engineer	Describes diagnostics used to test MAP functional units.
MAP III Systems Confidence Test (QM3) ERS	Uncontrolled	12103900	Customer engineer	Describes diagnostics used to confirm normal MAP operation.

CONTENTS

VOLUME 1

VOLUME 2

VOLUME 3

FIGURES

TABLES

INTRODUCTION

MAP is a microprogrammable, high-speed matrix processor that accepts operand matrices from a host computer system, processes them according to an algorithm selected by the computer, and then returns results to the computer. Since MAP is microprogrammable, its potential algorithm repertoire is unlimited. The microprogrammable nature of MAP also reduces the amount of hardware change required to interface the unit with different types of computers.

MAP CONFIGURATIONS

MAP is available in configurations to suit different processing requirements. Figure 1-1 illustrates the basic MAP and indicates the type and quantity of options that can be attached to it.

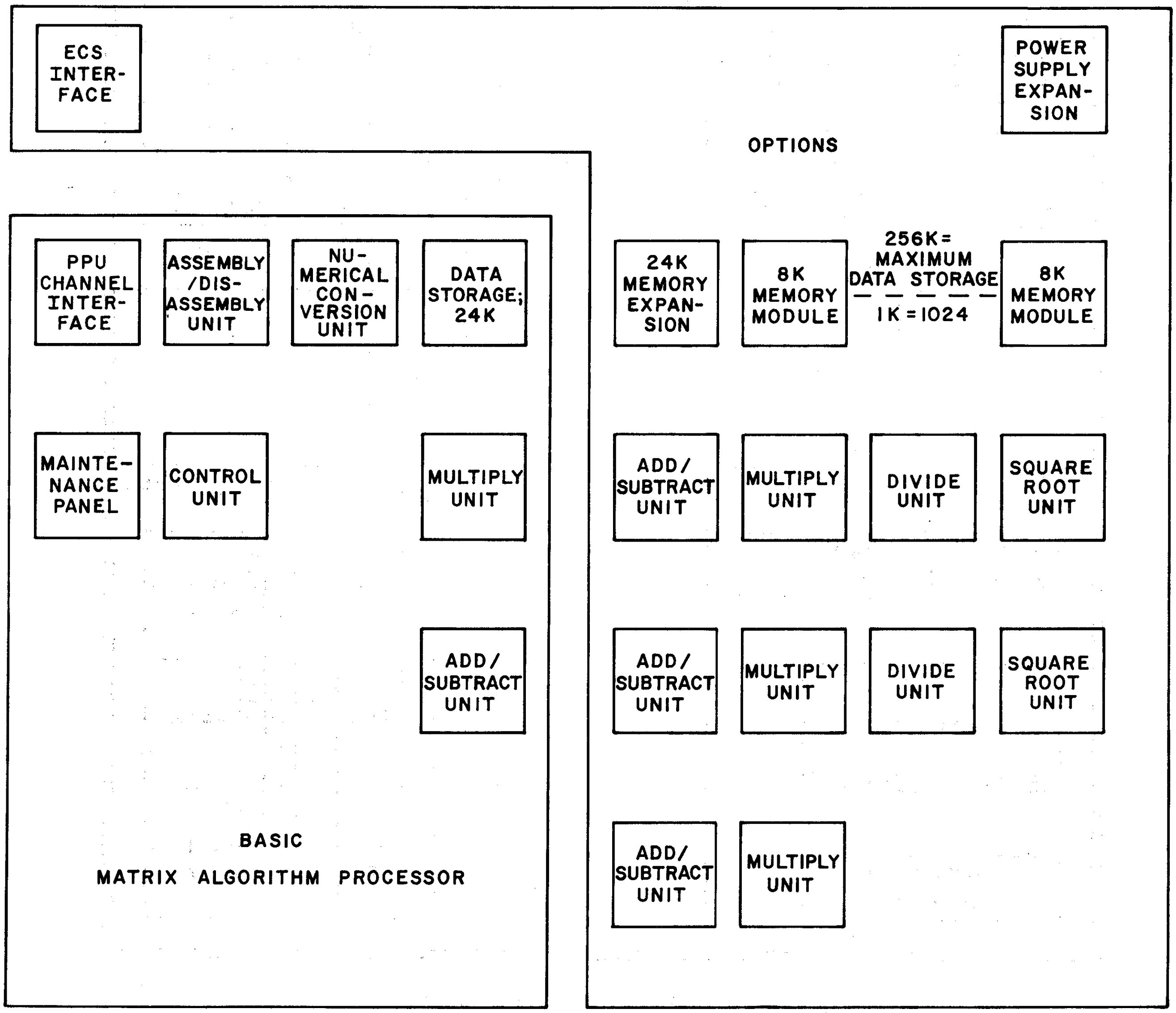


Figure 1-1. Basic MAP and Related Options

FUNCTIONAL DESCRIPTION

MAP consists of several functional components as illustrated in figure 1-2. The following paragraphs provide a brief description of each component and a basic operational description of MAP.

INTERFACES

MAP has a peripheral processor unit (PPU) channel interface, an optional extended core storage (ECS) interface, and an off-line load interface. Each interface performs the level shifting and signal conversion required to connect MAP with a particular device.

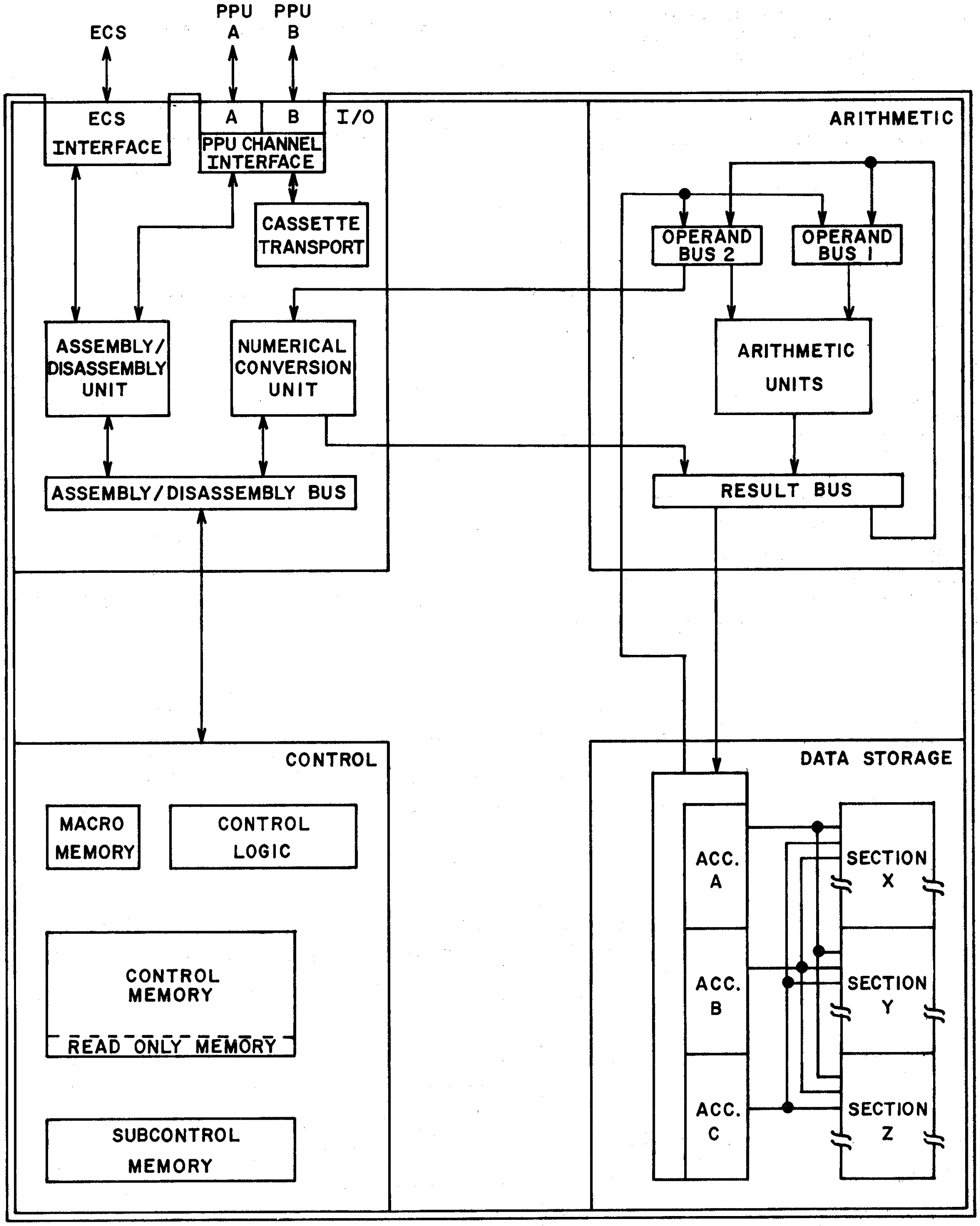


Figure 1-2. MAP Functional Components

PPU Channel Interface

The PPU channel interface has two accesses. Each access can connect MAP to one PPU channel from a CDC 6000 series, lower CDC CYBER 70 series, or CDC CYBER 170 series computer system. Although MAP can accept data and parameters from a PPU channel, the PPU channel is used primarily for transfer of function and status words.

ECS Interface

This optional interface connects MAP with ECS via an ECS controller. The ECS interface is used to transfer data and controlware between the computer system and MAP.

Off-line Load Interface

The off-line load interface permits MAP controlware or diagnostics to be loaded from or dumped to a cassette transport mounted on the maintenance panel.

CONTROL UNIT

The control unit processes microcode to control MAP input/output (I/O) and arithmetic activity. It consists of three random access memories with associated control, four register files, an arithmetic logical unit (ALU), and a group of control registers.

Control Unit Memories

Each control unit memory has its own address counter and parity checker. Parity is generated when a word is written and checked when the word is read.

Macro Memory

The macro memory is a general-purpose memory containing 1024 18-bit words. It stores parameters for arithmetic operations, macro level instructions, intermediate data and addresses, and comparison data for diagnostic purposes.

Control Memory

The control memory stores microinstructions and has a word size of 60 bits. It contains 2048 words of read/write memory and 256 words of read-only memory (ROM).

Subcontrol Memory

The subcontrol memory stores 72-bit words containing control bits for arithmetic units and busses. It contains 1024 words.

Register Files

Control unit register files are designated A, B, C, and K. Each file contains 16 18-bit registers. Files A, B, and C are used as general-purpose registers and to determine addresses for the three data storage accesses. File K is used as a general-purpose register file and as a counter file during I/O or arithmetic operations.

ALU

The ALU performs arithmetic or logical operations on two 18-bit source quantities and transfers results to selected destinations. Sources include all register files, macro memory, the channel function register, and channel status registers. Destinations include macro memory, the channel function register, and various control registers.

Control Registers

The control unit contains several registers for controlling different types of activity. These registers include instruction registers for the control and subcontrol memories, a microprogram address register, a macro memory address pointer register, a programmed status register, ECS limit and current address registers, and control registers for the numerical conversion and assembly/disassembly units.

MAINTENANCE PANEL

The maintenance panel contains two octal displays, an octal keyboard, a cassette transport, control switches, and status indicators. The panel permits an operator to perform the following functions.

- Load and dump control unit memories
- Monitor and enter numbers into various control unit registers
- Start and stop microprogram execution
- Monitor arithmetic results

Refer to section 2 for complete descriptions of maintenance panel controls, indicators, and procedures.

ASSEMBLY/DISASSEMBLY UNIT

This unit accepts 12-bit words from the PPU channel interface or 60-bit words from the ECS interface and places the words in a 72-bit format for presentation to the assembly/disassembly bus. Conversely, the assembly/disassembly unit accepts 72-bit quantities from the assembly/disassembly bus and disassembles them into 12-bit words for the PPU channel interface or into 60-bit words for the ECS interface. Controlware determines the operating characteristics of the assembly/disassembly unit by loading control words into two assembly/disassembly control registers.

NUMERICAL CONVERSION UNIT

This unit converts external floating- and fixed-point formats to the format used by MAP and converts the MAP format to external floating- and fixed-point formats. The numerical conversion unit can also extend or complement signs and adjust exponents.

Before using the numerical conversion unit, the control unit issues a radix point adjust word and a numerical conversion control word. These two words control the conversion process as data passes through the numerical conversion unit.

DATA STORAGE

Storage Sections

MAP data storage is divided into three sections: X, Y, and Z. Each section has its own address and control logic. A priority network connects three accesses (A, B, and C) to data storage. The priority network resolves conflicts when two or three accesses attempt to address locations in the same section. In the absence of conflicts, the sectional nature of data storage permits two operands to be read and a result stored during one microinstruction time.

Section Sizes

Each section of basic data storage contains 8K† 32-bit words. There are also two parity bits per word (one parity bit per 16-bit half-word). Sections X and Y can each be expanded to 96K in 8K increments. Section Z can be expanded to 64K in 8K increments. The maximum data storage size is 256K.

Section Switches

A set of switches indicates the size of each section. The count formed by the switches also defines the upper portion of the first word address of the next section (sections X and Y) or the upper portion of the first word address of the out-of-range portion of data storage (section Z).

ARITHMETIC UNITS

MAP has provision for four types of arithmetic units: add/subtract, multiply, divide, and square root. The basic MAP contains one add/subtract unit and one multiply unit. The maximum configuration contains four add/subtract units, four multiply units, two divide units, and two square root units.

Each arithmetic unit has one or two operand feeder registers and a result register. Microinstructions control arithmetic operations by routing operands and results on busses between data storage and arithmetic units. Multiply and add/subtract units are interconnected by an additional nonbus path to facilitate execution of certain types of algorithms.

BUSSES

MAP has a 72-bit assembly/disassembly bus, two 32-bit operand busses, and a 32-bit result bus. Microinstructions control the origin and destination(s) of bus data. Figure 1-2 illustrates how the four main busses interconnect MAP components. The control unit contains several 18-bit busses for distribution of ALU results and memory addresses.

BASIC OPERATIONAL DESCRIPTION

Typical MAP operations involve the following steps.

- Function issue
- Interface reservation
- Function identification
- Function processing

Function Issue

A PPU functions MAP by sending a valid function on one of the PPU channels to which MAP is connected. A MAP function consists of a 3-bit function code plus 9 additional bits which may be used individually for function definition or which may be combined to form a parameter. Since MAP is the only device on a channel, no equipment select code is required or used.

Interface Reservation

The MAP PPU channel interface contains two accesses. Only one access at a time can control the PPU channel interface. A PPU can cause an access to gain control of (reserve) the PPU channel interface in two ways. First, when the PPU channel interface is unreserved, the first access to receive a function gains control of the interface. Second, in case of a channel hangup, a PPU can issue a release opposite channel function to disconnect the opposite channel and gain control of the interface.

Function Identification

MAP functions fall into two categories, hardware functions and controlware functions. The PPU channel interface processes hardware functions, and the control unit processes controlware functions. The PPU channel interface identifies the category of a function by examining the function code. Function codes 0 through 6 are the controlware function codes; function code 7 indicates a hardware function.

† 1K=1024

Function Processing

The ability of MAP to process a function depends upon the category of the function and upon the reserved status of the PPU channel interface. When the PPU channel interface is unreserved or reserved to the requesting access, MAP begins function processing immediately upon receipt of any function. When the PPU channel interface is reserved to the opposite access, MAP responds to and executes only hardware functions which do not interfere with activity currently being performed for the opposite access. MAP does not respond to other functions until the opposite access releases the PPU channel interface.

Hardware Function Processing

MAP hardware functions are concerned with channel/access control and with status replies to the requesting PPU. These functions are processed by hardware in the PPU channel interface.

Controlware Function Processing

MAP controlware functions are used to initialize the control unit. When the PPU channel interface detects a controlware function, the interface sends the function to the control unit. Function codes 0 through 6 correspond to addresses 0 through 6 in the ROM portion of the control memory. The contents of each of these seven addresses can be the first word address of a controlware function routine. When the control unit receives a function, it fetches the appropriate first word address from the ROM and begins executing the selected function routine.

The routine for loading control unit memories executes from the ROM. Dump, algorithm, and data I/O routines execute from the read/write portion of control memory.

At the completion of any controlware function, the control unit halts and waits for the next function.

PHYSICAL DESCRIPTION

MAJOR COMPONENT IDENTIFICATION

Figures 1-3 and 1-4 illustrate major MAP components.

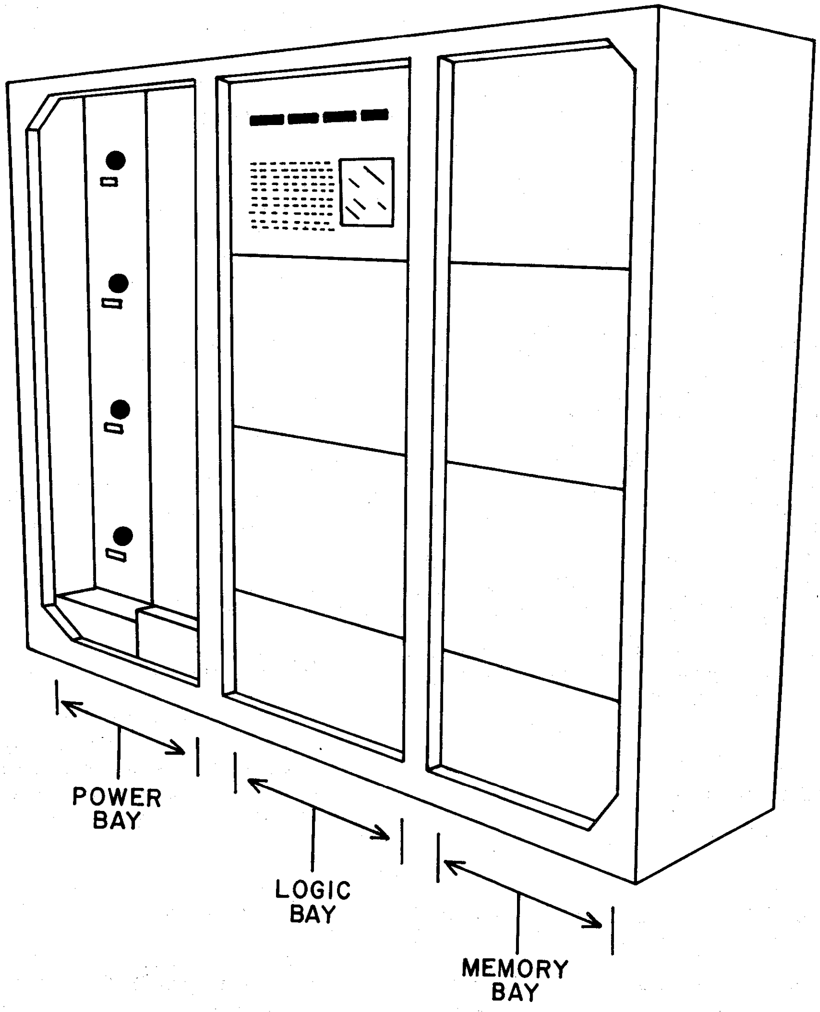


Figure 1-3. MAP Front View

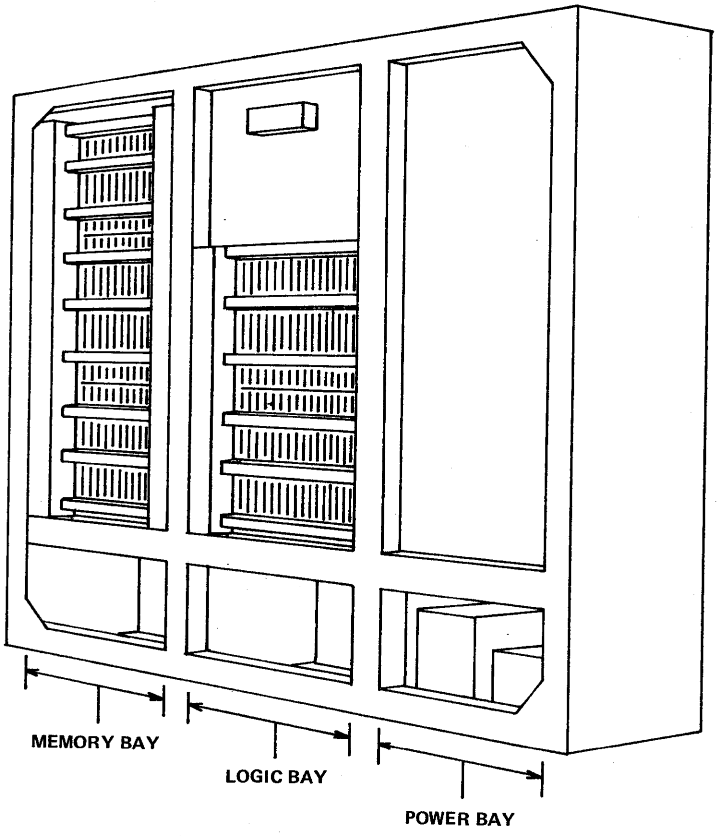


Figure 1-4. MAP Rear View

CABINET

MAP can be housed in a double-bay or triple-bay cabinet. The triple-bay cabinet is used when MAP contains more than 24K of data storage.

The cabinet contains a power bay and one or two logic bays. The power bay contains an adjustable power supply, a circuit breaker, and a ripple test point for each logic voltage. The first logic bay contains the maintenance panel, the control unit, 24K of data storage, and the arithmetic units. An optional second logic bay contains from 24K to 232K of additional data storage.

Blowers in the base of the cabinet cool power supplies and 180-paks by circulating refrigerated air obtained from underneath the false floor.

POWER SUPPLY/DISTRIBUTION

Power Supplies

MAP has power supplies to supply five dc voltages: -12 volts, -5.2 volts, -2.2 volts, +5 volts, and +12 volts. The logic and memory bays each use -5.2 volts and -2.2 volts at high amperage while the maintenance panel uses the other three voltages at low amperage. The -5.2 volt and -2.2 volt power supplies operate from 400-Hz three-phase power; the other power supplies operate from 50/60-Hz single-phase power.

Protection

The power supply/distribution system contains sensors which disconnect 400-Hz power when the cabinet air temperature reaches 130°F (55°C) or when one of the logic or memory blowers fails. In addition, each high amperage power supply is protected from current overload by its own circuit breaker.

DC Voltage Adjustment

Each high amperage power supply has a variable ratio transformer which permits the output voltage to be adjusted for margin testing. A meter and voltage select switch permit convenient monitoring of high amperage power supply output voltages.

Distribution

A network of insulated bus bars and wires distributes logic voltages to the logic and memory bays. A cable assembly carries power from the three low amperage power supplies to the maintenance panel.

Power Supply Expansion Option

This option adds two additional power supplies to the power bay. The option must be installed when MAP data storage exceeds 96K or when any one of the data storage sections exceeds 32K. The power supply expansion option permits data storage to be increased to a maximum of 256K.

LOGIC CHASSIS

A MAP logic chassis consists of a printed circuit backpanel, a pak guide assembly, a complement of 180-paks, and a hinged front cover.

The backpanel provides a printed circuit ground plane and connector/wirewrap stakes which mate with 180-pak edge connectors. Twisted-pair wires carry logic signals between 180-paks.

The pak guide assembly consists of two planes of rails welded to a frame. The rails align 180-pak connectors with backpanel connectors as the pak mates with the backpanel.

The front cover provides a seal during normal operation so that cooling air is forced to exit at the top of the cabinet.

180-PAK

Figure 1-5 illustrates a typical 180-pak. Except for the maintenance panel and cassette, all MAP logic is contained in 180-paks. A 180-pak is an epoxy board containing power and ground planes and providing 180 16-pin integrated circuit (chip) locations in a 9 x 20 matrix. A 288-pin edge connector interfaces the 180-pak to the backpanel. Wires glued to the surface of the board interconnect chips. (Data storage 180-paks use etched conductors rather than glued wires to interconnect chips.) Cams provide mechanical advantage when installing or removing the board from the logic chassis.

CAUTION

> To prevent damage to the pak guide assembly, 180-paks must be fully inserted or removed. Do not use the pak guide assembly as a temporary storage place for 180-paks.

INTEGRATED CIRCUITS

All MAP logic is constructed of ECL 10,000 series chips except for portions of the maintenance panel and cassette which use TTL chips. ECL elements are encapsulated in 16-pin (or 24-pin) dual inline packages. A small printed circuit assembly is used to interface each 24-pin dual inline package with two chip locations on a 180-pak.

MAINTENANCE PANEL

The maintenance panel is a large printed circuit board with switches, indicators, and the cassette transport mounted on the front side. The back side contains foil paths and logic to interface the maintenance panel and cassette transport with the MAP logic chassis.

SPECIFICATIONS

Tables 1-1 through 1-4 provide physical specifications for MAP. Figure 1-6 illustrates cabinet dimensions, door swings, and cable cutouts.

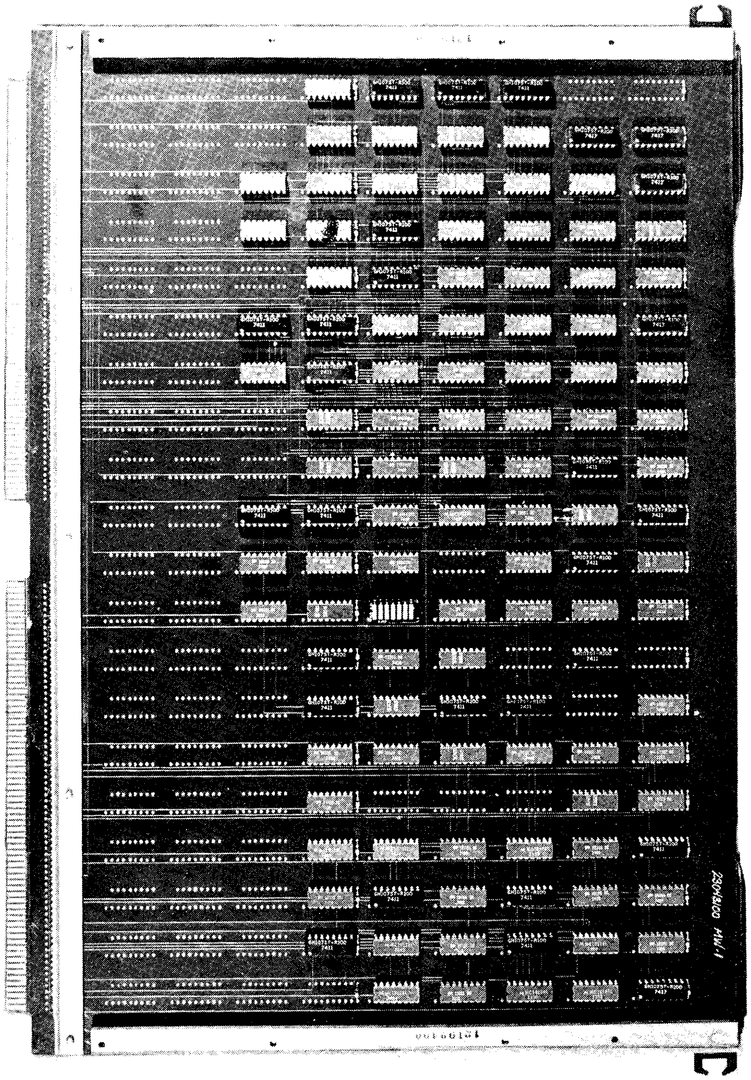


Figure 1-5. Typical 180-Pak

TABLE 1-1. PHYSICAL CHARACTERISTICS

	Height		Width		Depth		Weight	
Equipment	In.	Cm	In.	Cm	In.	Cm	Lb	Kg
2-bay MAP	76	193	55.3	40.5	25	63.5	1700	770
3-bay MAP	76	193	82.8	210.3	25	63.5	2500	1135

TABLE 1-2. ELECTRICAL DATA

Equipment	AC Voltage	Freq. (Hz)	Phase	Amps/Phase (max)	kva (max)
MAP with 96K of data storage	120/208	400	3	26	11
	120	50/60	1	10	2
MAP with 256K of data storage	120/208	400	3	50	20
	120	50/60	1	15	3

TABLE 1-3. ENVIRONMENTAL RESTRICTIONS

	Nonoperating		Operating		
Parameter	Minimum	Maximum	Minimum	Nominal	Maximum
Temperature	-30°F -34°C	150°F 66°C	40°F 5°C	75°F 24°C	90°F 32°C
Temperature gradient	-	-	-	-	20°F/hr 11°C/hr
Relative humidity (no condensation)	5%	95%	10%	-	90%
Altitude	-	-	-	-	6000 ft 1.85 km

TABLE 1-4. HEAT DISSIPATION

Equipment	Heat Dissipation (max)
2-bay MAP	19,380 Btu/hr 4880 Kg-cal/hr
3-bay MAP	35,250 Btu/hr 7850 Kg-cal/hr

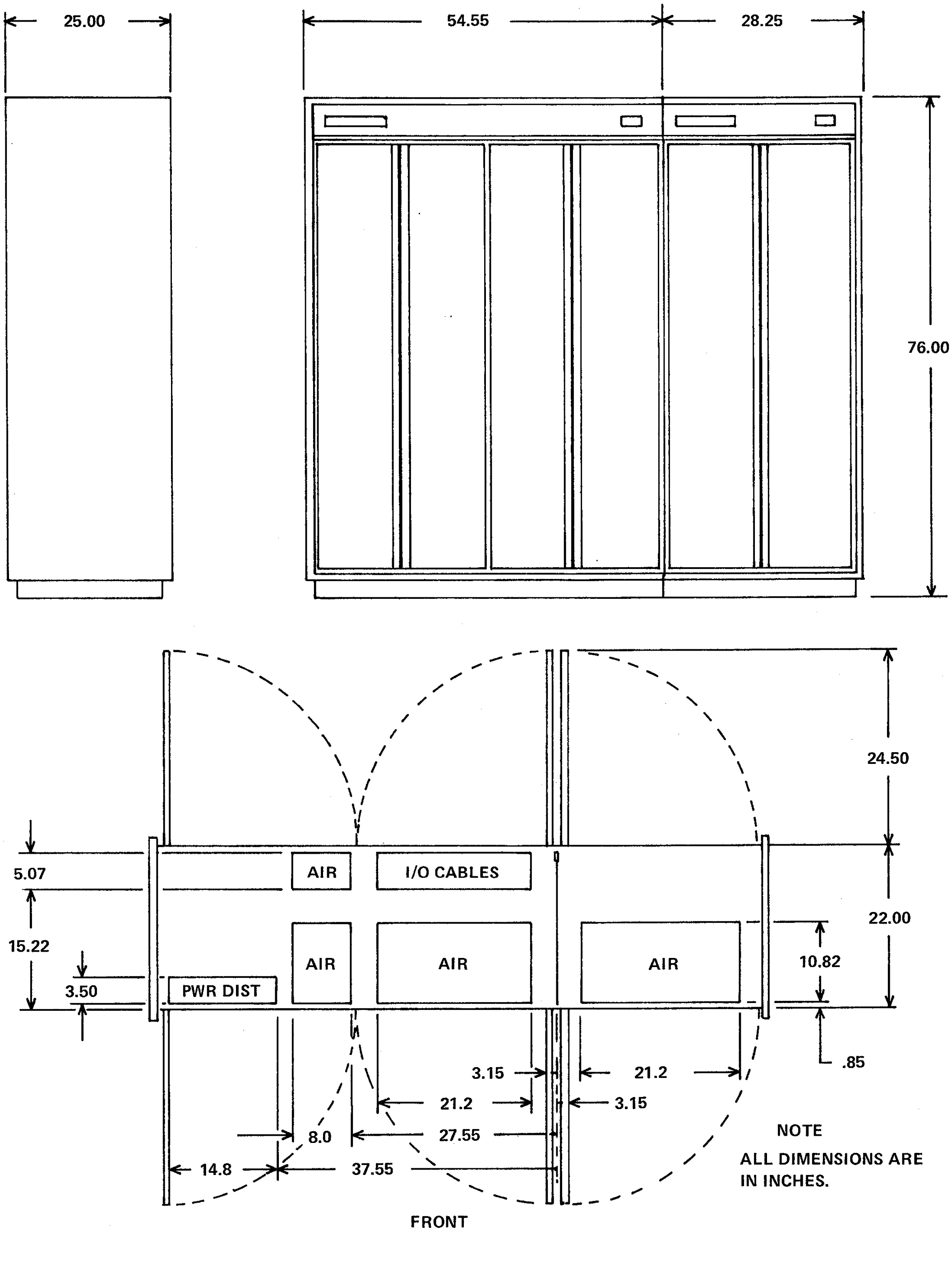


Figure 1-6. MAP Cabinet Dimensions

INTRODUCTION

This section describes MAP controls and indicators, provides normal operating procedures, and provides hardware reference information.

OPERATION

CONTROLS AND INDICATORS

Power Controls

Figure 2-1 shows the locations of power controls. Each power supply in the power bay has its own circuit breaker and voltage adjust knob. The cassette power supplies mounted behind the maintenance panel are nonadjustable.

Master power controls include a 60HZ DISCONNECT switch and a 400HZ DISCONNECT circuit breaker. The switch controls blower power while the circuit breaker controls power used in the airflow sensor and thermostat circuits. When both the switch and circuit breaker are OFF, no power can leave the power distribution box.

The percent meter located in the lower left of the power bay indicates how much a selected power supply deviates from its nominal voltage. A rotary switch adjacent to the meter selects the power supply to be monitored.

Maintenance Panel

Figure 2-2 illustrates the maintenance panel. The panel contains a 24-digit data display, a 6-digit address display, a cassette transport, and 106 switches/indicators.

Data Display

This display reflects the quantity on the 72-bit assembly/disassembly (A/D) bus. When other than 72-bit quantities are displayed, the lower portion of the display conatins the quantity, and the upper portion of the display is filled with sevens.

Address Display

This display reflects a selected status word or a selected address counter or register.

Cassette Transport

This device provides the capability of loading and dumping control unit memories while MAP is off-line.

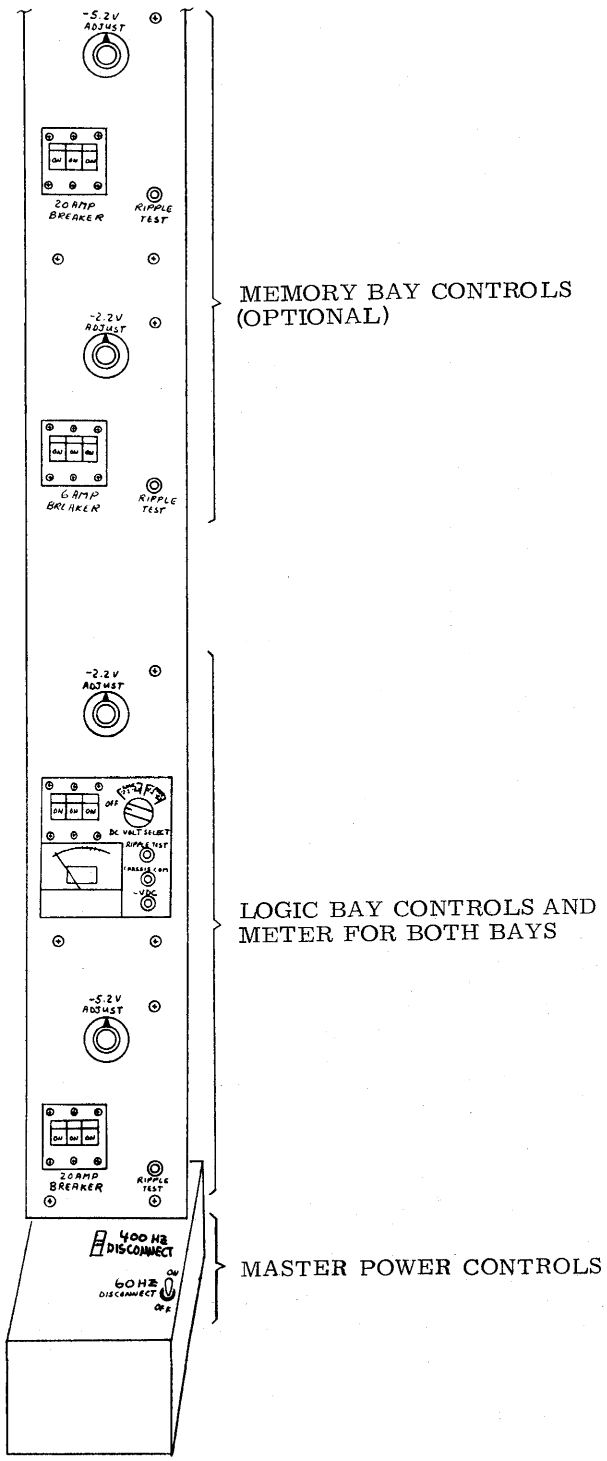


Figure 2-1. Power Controls

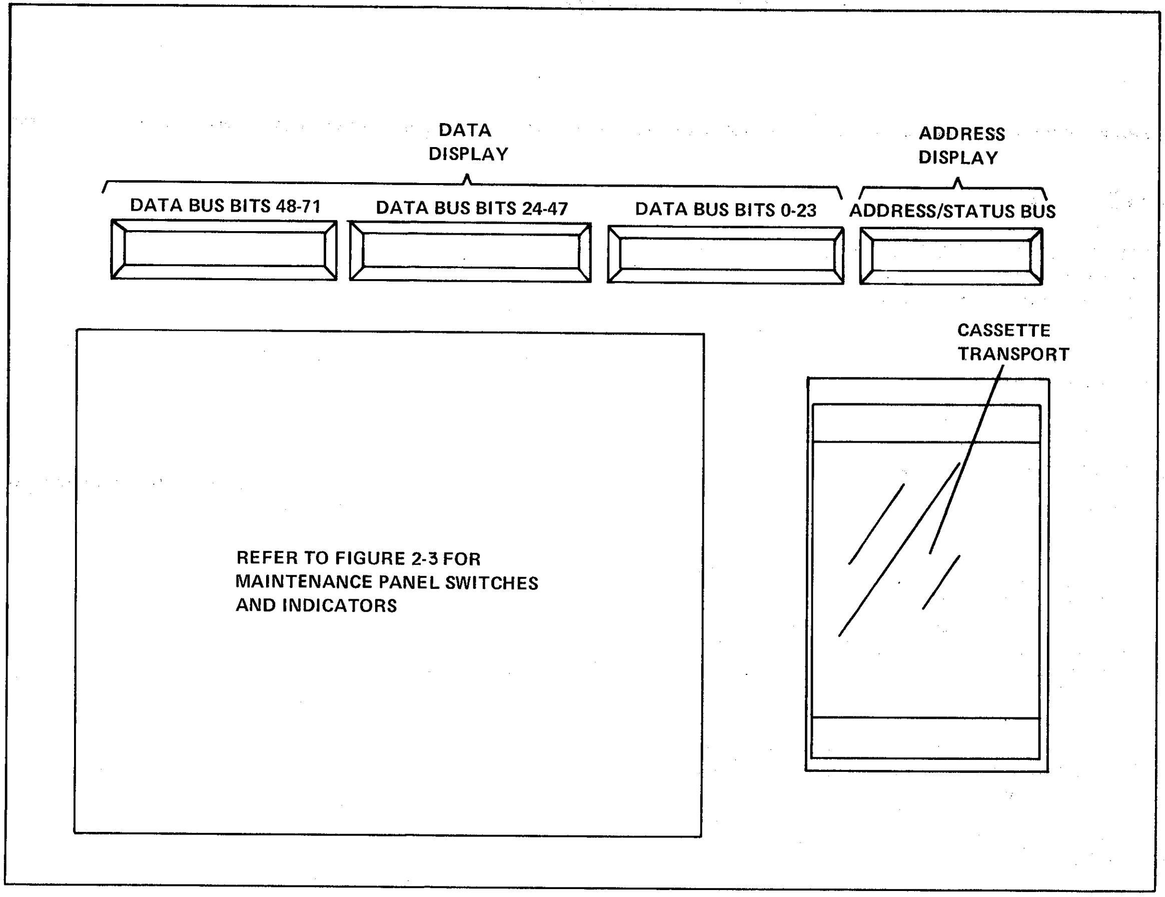


Figure 2-2. Maintenance Panel

Switches/Indicators

MAP switches and indicators are divided into three color groups. Green is associated with address selection or status, amber with data selection or status, and red with control. Figure 2-3 illustrates the 11 categories of switches/indicators on the maintenance panel. Table 2-1 describes the function of each switch/indicator.

OPERATING PROCEDURES

The procedures in this section cover normal MAP operations. Refer to section 5 for off-line autoload procedures and other maintenance panel procedures. Refer to section 4 for the initial power-on procedure.

Power Up

1. Ensure that each circuit breaker in the power bay is in the ON position.
2. Turn the 60HZ DISCONNECT switch to the ON position. The blowers will operate and logic power will come up.
3. Ensure that the cassette ENABLE switch/ indicator on the maintenance panel is not lighted. If the indicator is lighted, press it to clear cassette enable and enable the PPU channel.
4. MAP is autoloaded from a PPU by transferring controlware from the PPU to MAP. MAP is ready to execute functions when the controlware transfer completes.

Power Down

Turn the 60HZ DISCONNECT switch and the 400 HZ DISCONNECT circuit breaker to the OFF position.

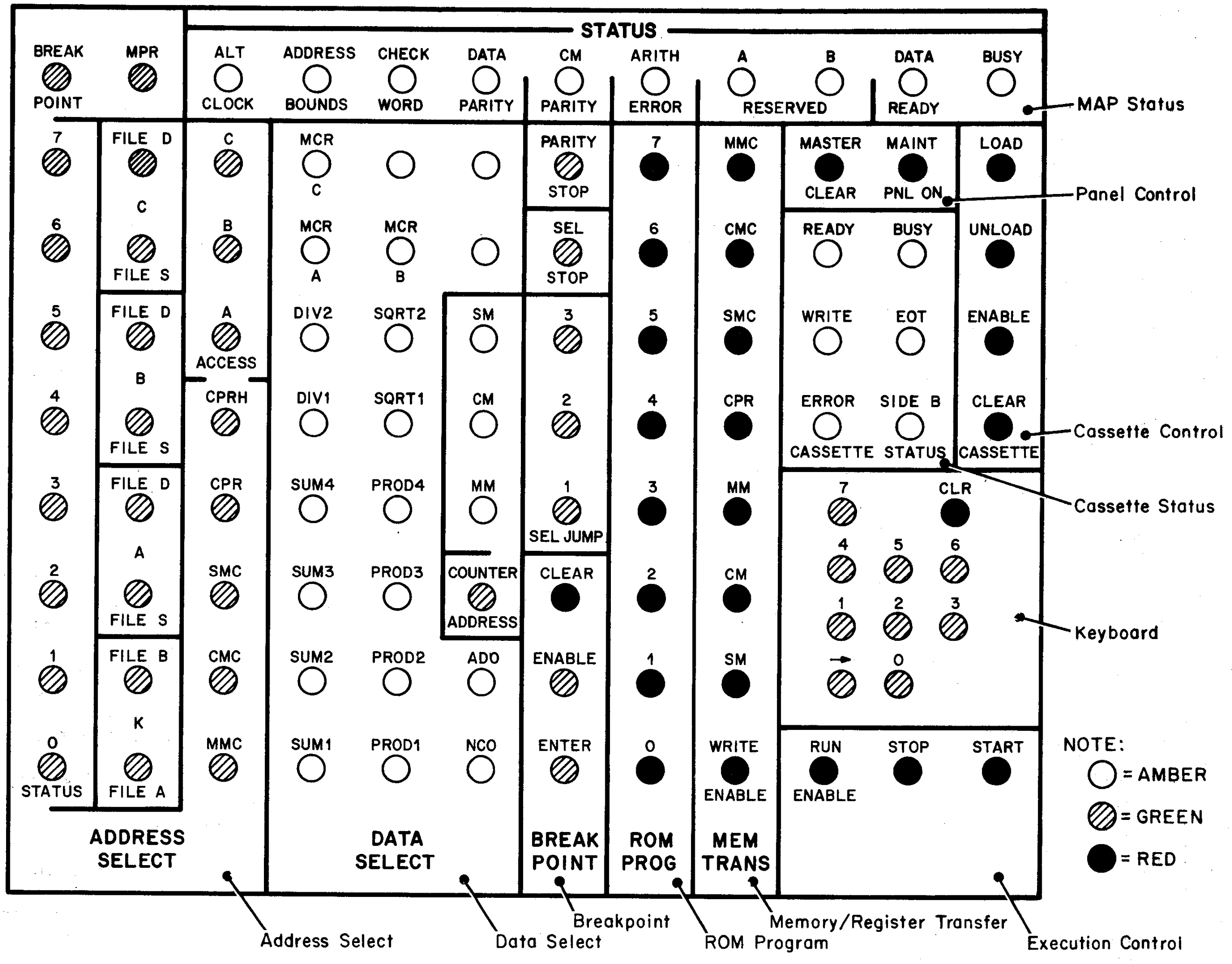


Figure 2-3. Maintenance Panel Switch Categories

TABLE 2-1. MAINTENANCE PANEL SWITCH FUNCTIONS

Category	Nomenclature	Device	Function
MAP status	ALT CLOCK	Indicator	Indicates MAP is running under control of a clock with a frequency different from the normal clock frequency.
MAP status	ADDRESS BOUNDS	Indicator	Indicates MAP has attempted to access an out-of-range address in data storage.
MAP status	CHECKWORD	Indicator	Indicates detection of a check word error during loading of control unit memories.
MAP status	DATA PARITY	Indicator	Indicates detection of a parity error in data storage.
MAP status	CM PARITY	Indicator	Indicates detection of a parity error in one of the three control unit memories.
MAP status	PARITY STOP	Switch/indicator	Toggles stop on parity error capability. When illuminated, MAP stops upon detection of a parity error in one of the three control unit memories.

TABLE 2-1. MAINTENANCE PANEL SWITCH FUNCTIONS (Cont'd)

Category	Nomenclature	Device	Function
MAP status	ARITH ERROR	Indicator	Indicates arithmetic underflow, overflow, or a divide fault or square root fault.
MAP status	A, B, RESERVED	Indicator	Indicates which access (if any) of the channel interface has control of MAP.
MAP status	DATA READY	Indicator	Indicates there is a data word in the channel interface.
MAP status	BUSY	Indicator	Indicates MAP is executing microinstructions.
Panel control	MASTER CLEAR	Switch	Stops controlware execution and clears the control unit; does not clear any registers.
Panel control	MAINT PNL ON	Switch/indicator	Toggles maintenance panel enable. Lights when panel is enabled. When not illuminated, all other switches except SEL STOP and SEL JUMP 1, 2, 3, are disabled.
Cassette control	LOAD	Switch/indicator	Initiates cassette motion toward loadpoint. Lights when tape is at loadpoint.
Cassette control	UNLOAD	Switch/indicator	Initiates a rewind to loadpoint when tape is past loadpoint. Initiates a rewind to tape leader when tape is at loadpoint. Lights when cassette may be removed.
Cassette control	ENABLE	Switch/indicator	Toggles cassette enable/channel disable. Lights when cassette is enabled and PPU data channel is disabled.
Cassette control	CLEAR	Switch	Clears cassette control logic and error conditions.
Cassette status	READY	Indicator	Indicates cassette is ready to perform a read/write operation.
Cassette status	BUSY	Indicator	Indicates cassette is performing a read/write operation.
Cassette status	WRITE	Switch/indicator	Toggles cassette read/write mode. Lights when cassette is in write mode.
Cassette status	EOT	Indicator	Indicates cassette end-of-tape condition.
Cassette status	ERROR	Indicator	Indicates a cassette error condition.
Cassette status	SIDE B	Indicator	Indicates cassette side B is positioned under read/record heads.
Keyboard	0 through 7	Switch	Left-shifts the assembly/disassembly output (ADO) register one octal digit and enters the selected digit into the lowest order digit of the ADO register.
Keyboard	→	Switch	Right-shifts the ADO register one octal digit. The shift is end-off.
Keyboard	CLR	Switch	Clears the ADO register.
Execution control	START	Switch/indicator	Initiates control memory instruction execution at the address in the control program register (CPR). Lights when an instruction has been executed.

TABLE 2-1. MAINTENANCE PANEL SWITCH FUNCTIONS (Cont'd)

Category	Nomenclature	Device	Function
Execution control	STOP	Switch	Halts instruction execution or continuous memory transfers.
Execution control	RUN ENABLE	Switch/indicator	Enables continuous instruction execution or memory transfers when held down at the time the START switch or one of the memory transfer switches is pressed. Lights during instruction execution or continuous memory transfers.
Memory/ register transfer	WRITE ENABLE	Switch/indicator	Toggles transfer read/write mode. Lights when transfers from the A/D bus to memories/registers (write control memory operations) are enabled.
Memory/ register transfer	MMC	Switch/indicator	Transfers the A/D bus to the macro memory counter (MMC). Lights to indicate that the quantity has been transferred.
Memory/ register transfer	CMC	Switch/indicator	Transfers the A/D bus to the control memory counter (CMC). Lights to indicate that the quantity has been transferred.
Memory/ register transfer	SMC	Switch/indicator	Transfers the A/D bus to the subcontrol memory counter (SMC). Lights to indicate that the quantity has been transferred.
Memory/ register transfer	CPR	Switch/indicator	Transfers the A/D bus to the control memory program address register (CPR). Lights to indicate that the quantity has been transferred.
Memory/ register transfer	MM	Switch/indicator	Transfers the A/D bus to the macro memory (MM) address specified by MMC and then increments MMC. Lights to indicate that the quantity has been transferred.
Memory/ register transfer	CM	Switch/indicator	Transfers the A/D bus to the control memory (CM) address specified by CMC and then increments CMC. Lights to indicate that the quantity has been transferred.
Memory/ register transfer	SM	Switch/indicator	Transfers the A/D bus to the subcontrol memory (SM) address specified by SMC and then increments SMC. Lights to indicate that the quantity has been transferred.
ROM program	0 through 7	Switch/indicator	Initiates continuous instruction execution (if RUN ENABLE switch is pressed) at one of the eight starting addresses in the read-only memory (ROM) portion of control memory. Switches 0 through 7 correspond to addresses 7410_8 through 7417_8, respectively.
Breakpoint	ENTER	Switch	When held down, enables the breakpoint register to be entered from the keyboard. When ENTER is pressed, the keyboard → and CLR switches have no effect on the breakpoint register, but they affect the ADO register.

TABLE 2-1. MAINTENANCE PANEL SWITCH FUNCTIONS (Cont'd)

Category	Nomenclature	Device	Function
Breakpoint	ENABLE	Switch/indicator	Toggles breakpoint mode. Lights to indicate breakpoint mode. A breakpoint halt occurs when the quantity in the address display is equal to the breakpoint register. The halt occurs after the instruction causing the breakpoint condition.
Breakpoint	CLEAR	Switch	Clears the breakpoint register.
Selective jump	SEL JUMP 1,2,3	Switch/indicator	Toggles a jump condition between true and false. Lights when the condition is true. The three selective jump conditions are used in conjunction with the jump control (4_8) instruction.
Selective stop	SEL STOP	Switch/indicator	Toggles enable selective stop. Lights when selective stop is enabled. Selective stop is used in conjunction with the jump control (4_8) instruction.
Data select	MCR, A,B,C	Switch/indicator	Enables one of the memory catching registers (MCR) to the A/D bus. Lights to identify the quantity in the data display.
Data select	DIV 1,2	Switch/indicator	Enables one of the divide unit result registers to the A/D bus. Lights to identify the quantity in the data display.
Data select	SUM 1,2,3,4	Switch/indicator	Enables one of the add/subtract unit result registers to the A/D bus. Lights to identify the quantity in the data display.
Data select	SQRT 1,2	Switch/indicator	Enables one of the square root unit result registers to the A/D bus. Lights to identify the quantity in the data display.
Data select	PROD 1,2,3,4	Switch/indicator	Enables one of the multiply unit result registers to the A/D bus. Lights to identify the quantity in the data display.
Data select	SM	Switch/indicator	Enables the contents of the subcontrol memory address specified by the SM address mux to the A/D bus. Lights to identify the quantity in the data display. When the COUNTER ADDRESS indicator is on, SMC provides the address. Otherwise, microinstruction bits 7 through 16 form the address.
Data select	CM	Switch/indicator	Enables the contents of the control memory address specified by the CM address mux to the A/D bus. Lights to identify the quantity in the data display. When the COUNTER ADDRESS indicator is on, CMC provides the address. Otherwise, CPR provides the address.
Data select	MM	Switch/indicator	Enables the contents of the macro memory address specified by the MM address mux to the A/D bus. Lights to identify the quantity in the data display. When the COUNTER ADDRESS indicator is on, MMC provides the address. Otherwise, the macro pointer adder provides the address.

TABLE 2-1. MAINTENANCE PANEL SWITCH FUNCTIONS (Cont'd)

Category	Nomenclature	Device	Function
Data select	COUNTER ADDRESS	Indicator	Indicates that the quantity in the data display is the contents of an address supplied by one of the three memory counters (CMC, SMC, or MMC). This indicator is forced on when the MASTER CLEAR switch is pressed.
Data select	ADO	Switch/indicator	Enables the assembly disassembly output register (ADO) to the A/D bus. Lights to identify the quantity in the data display.
Data select	NCO	Switch/indicator	Enables the numerical control unit output register (NCO) to the A/D bus. Lights to identify the quantity in the data display.
Address select	BREAK POINT	Switch/indicator	Enables the breakpoint register to the address display. Lights to identify the quantity in the address display.
Address select	MPR	Switch/indicator	Enables the macro memory pointer register (MPR) to the address display. Lights to identify the quantity in the address display.
Address select	STATUS 0 through 7	Switch/indicator	Enables one of the eight status words to the address display. Lights to identify the quantity in the address display. Status words are described under 7020_8 Select Status Words in this section.
Address select	ACCESS A, B, C	Switch/indicator	Transfers the A or B access adder or the C access adder or bit inversion network to the address display. Lights to identify the quantity in the address display. The displayed address is the result of the last operands loaded into the adder feeder registers. If the current micro-instruction selects bit inversion, the displayed address is from the bit inversion network rather than from the C access adder.
Address select	CPRH	Switch/indicator	Enables the control memory program address holding (CPRH) register to the address display. Lights to identify the quantity in the address display.
Address select	CPR, SMC, CMC, MMC	Switch/indicator	Enables the selected register or counter to the address display. Lights to identify the quantity in the address display.

HARDWARE REFERENCE INFORMATION

This part describes MAP functions, macro memory instructions, control memory instructions, and the subcontrol memory instruction. Throughout this part, bit 0 refers to bit 2^0, bit 1 refers to bit 2^1, and so forth.

FUNCTIONS

MAP responds to one hardware function ($7XXX_8$ codes) and four controlware functions (codes $10XX_8$, $20XX_8$, 3000_8, and $400X_8$). The hardware function is not sensitive to changes in controlware. Although controlware function characteristics are determined by microinstruction routines, the routines for three of the four controlware functions reside in the ROM portion of control memory. These three functions are also insensitive to controlware changes. Only the load from ECS ($400X_8$) function is part of the basic internal controlware and thus sensitive to internal controlware changes.

Function Format

Figure 2-4 shows the function format. Bits 9 through 11 are a three-bit operation code. Bits 0 through 8 are used for additional function definition or as a parameter. Since MAP is the only device on a PPU channel, no equipment select code is required or used.

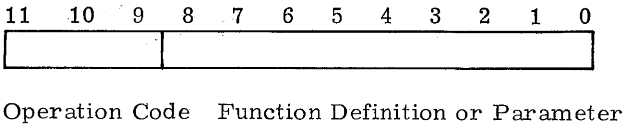


Figure 2-4. Function Format

PPU Channel Interface Reservation

The PPU channel interface contains two accesses. Only one access at a time can control the PPU channel interface, although the access not in control can relay status from MAP to a PPU. A PPU can cause an access to gain control of (reserve) the PPU channel interface in one of two ways. First, when the PPU channel interface is unreserved, the first access to receive a function gains control of the interface. Second, in case of a channel hangup, a PPU can issue a release opposite channel (7004_8) function to disconnect the opposite channel and gain control of the PPU channel interface. To prevent channel hangups, the first function of a sequence should be a select status words (7020_8) function and the last function should be a release channel (7002_8) function.

Function Processing

MAP's ability to process a function depends upon the type of function (hardware or controlware), the reserved status of the PPU channel interface, and MAP's busy status.

When the PPU channel interface is unreserved or reserved to the requesting access and MAP is not busy (that is, not executing microinstructions), MAP begins function processing immediately upon receipt of any function.

When the PPU channel interface is unreserved or reserved to the requesting access and MAP is busy, MAP responds to and executes only hardware functions. MAP retains a controlware function arriving at this time but does not respond to or execute the function until MAP becomes not busy.

When the PPU channel interface is reserved to the opposite access, MAP responds to and executes only select status words (7020_8) and release opposite channel (7004_8) functions.

Function Reply

MAP acknowledges receipt of a function by returning an inactive to the requesting PPU. Although the PPU channel goes inactive, the PPU channel interface remains reserved to the requesting PPU, providing the function caused interface reservation or the interface was already reserved to the requesting PPU.

MAP responds to valid hardware functions within 1 microsecond and to valid controlware functions (assuming MAP is not busy) within 5 microseconds. When it receives an invalid function, MAP sends a function reply and then halts. To prevent a possible PPU channel hangup (caused by the PPU driver pass counter disconnecting the channel just after a MAP function reply disconnect), a controlware function should be sent only when MAP is not busy.

Checkword

All parameter and data transfers between a PPU and MAP must be accompanied by a 12-bit checkword. A checkword follows the last data word of a transfer and is formed by adding (endoff) all of the data words of the transfer together and taking the twos complement of the result. The following example illustrates checkword formation for a hypothetical two-word transfer from a PPU to MAP.

First word	6005_8
Second word	3402_8
Sum (dropping carry)	1407_8
Two's complement	6371_8 = checkword

MAP halts if it detects a checkword error after a transfer from a PPU.

Note that functions from a PPU to MAP and status transfers from MAP to a PPU are not accompanied by checkwords.

Hardware Functions

Table 2-2 lists MAP hardware functions. These functions may be sent individually or any number of them may be combined and sent simultaneously. To combine functions, take the number formed by the lower three digits of each function, add the numbers, and restore the 7_8 operation code. For example, the 7020_8 (select status words) and 7040_8 (select inactive on EOT) functions can be sent together with a 7060_8 function.

7001_8 Function Clear

This function halts and clears the MAP control unit but has no effect on registers, data storage, or error status.

TABLE 2-2. MAP HARDWARE FUNCTIONS

Function	Name	PPU Channel Interface Reservation Required for Execution
7001_8	Function clear	Yes
7002_8	Release channel	Yes
7004_8	Release opposite channel	No
7010_8	Clear error status	Yes
7020_8	Select status words	No
7040_8	Select inactive on EOT	Yes
7100_8	Start execution	Yes
7200_8	Enable active	Yes

7002_8 Release Channel

This function cancels the PPU channel interface's reservation to the functioning PPU channel. The PPU channel interface will then be reserved to the next PPU to send a function.

7004_8Release Opposite Channel

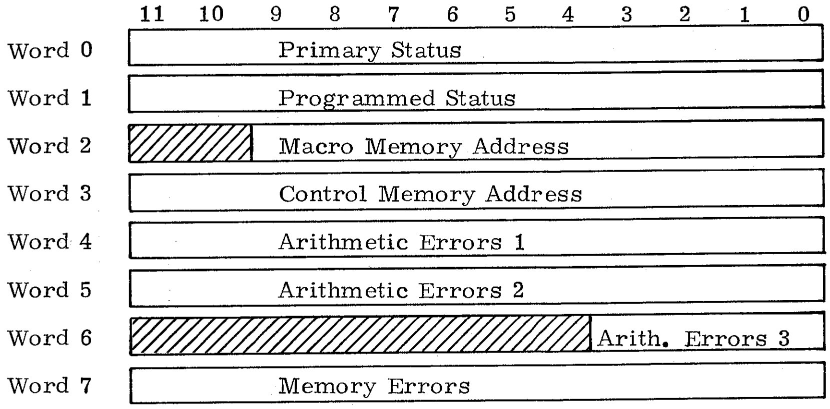
This function cancels the PPU channel interface's reservation to the opposite PPU channel. The PPU channel interface will then be reserved to the next PPU to send a function. The 7004_8 function should be used only in case of a channel hangup.

7010_8 Clear Error Status

This function clears all error bits in status words 0, 4, 5, 6, and 7. When the internal controlware has suspended macro processing because of an error condition, a 7010_8 function reinitiates macro processing.

7020_8 Select Status Words

This function causes MAP to send an eight-word status block to the requesting PPU. After sending a 7020_8 function, the PPU should activate the channel, input the desired number of status words, and then disconnect the channel. MAP considers the 7020_8 function complete when the PPU disconnects the channel, no matter how many words have been accepted by the PPU. Figure 2-5 shows the status block format.



	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	Primary Status											
Word 1	Programmed Status											
Word 2	▨	▨	Macro Memory Address									
Word 3	Control Memory Address											
Word 4	Arithmetic Errors 1											
Word 5	Arithmetic Errors 2											
Word 6	▨	▨	▨	▨	▨	▨	▨	▨	Arith. Errors 3			
Word 7	Memory Errors											

Figure 2-5. Status Block Format

Word 0, Primary Status

Bit 0, Ready
The MAINT PNL ON and cassette ENABLE switches on the maintenance panel are both off.

Bit 1, Busy
The control unit is executing microinstructions.

Bit 2, Access A Reserved
The PPU channel connected to access A has control of the PPU channel interface.

Bit 3, Access B Reserved
The PPU channel connected to access B has control of the PPU channel interface.

Bit 4, Function Busy
MAP has received a controlware function but has not yet processed the function.

Bit 5, Arithmetic Error
At least one of the bits in word 4, word 5, or word 6 is set.

Bit 6, Control Memory Parity Error
MAP detected a parity error in control memory, subcontrol memory, or macro memory.

Bit 7, Data Memory Parity Error
MAP detected a parity error in data storage.

Bit 8, Checkword Error
The data and accompanying checkword received by MAP over the PPU channel do not agree.

Bit 9, Address Out of Bounds
MAP attempted to access an address larger than the largest data storage address available.

Bit 10, ECS Error
The ECS controller returned abort, field length error, or ECS parity error status to MAP.

Bit 11, Data Ready
A word is in the PPU channel interface. During transfers to MAP, data ready sets when the PPU sends a word to the PPU channel interface and clears when the control unit accepts the word. During transfers from MAP, data ready sets when the control unit sends a word to the PPU channel interface and clears when the PPU accepts the word.

Word 1, Programmed Status

The format of this word depends upon the internal controlware currently executing in MAP. The basic internal controlware provided by Control Data uses the following format.

Bits 0 through 5, Macro Code
The macro code for the current or last executed macro.

Bits 6 through 11, User ID Flag
The user ID flag for the current or last executed macro. ID flag uses are described under $40XXXX_8$ Execute Macro in this section.

Word 2, Macro Memory Address

Bits 0 through 9, Macro Pointer Register
The current contents of the macro memory pointer register (MPR). The PPU can use this number to determine the macro memory instruction being executed.

Bits 10, 11 Unused

Word 3, Control Memory Address

This word reflects the contents of the 12-bit control memory program address register (CPR).

Word 4, Arithmetic Errors 1

Bit 0, Adder 1 Overflow
Bit 1, Adder 2 Overflow
Bit 2, Adder 3 Overflow
Bit 3, Adder 4 Overflow
The result exponent exceeds 77_8 (positive overflow) or 277_8 (negative overflow).

Bit 4, Adder 1 Underflow
Bit 5, Adder 2 Underflow
Bit 6, Adder 3 Underflow
Bit 7, Adder 4 Underflow
The result exponent is less than 101_8 (positive underflow) or 301_8 (negative underflow).

Bit 8, Divide 1 Overflow
A negative result exponent occurred when the A operand exponent was positive and the B operand exponent was negative. The result itself can be positive or negative.

Bit 9, Divide 1 Underflow
A positive result exponent occurred when the A operand exponent was negative and the B operand exponent was positive. The result itself can be positive or negative.

Bit 10, Divide 1 Indefinite

Divide unit 1 attempted to divide 0 by 0. The result is positive if the signs of the operand 0s are alike and negative if the signs of the operand 0s are different. The result coefficient is 77777777_8.

Bit 11, Divide 1 Fault

Divide unit 1 attempted to divide a nonzero operand by 0. The sign of the result is the same as the sign of the nonzero operand. The result coefficient is 77777777_8.

Word 5, Arithmetic Errors 2

Bit 0, Multiply 1 Overflow

Bit 1, Multiply 2 Overflow

Bit 2, Multiply 3 Overflow

Bit 3, Multiply 4 Overflow

A negative result exponent occurred when both the A and B operand exponents were positive. The result itself can be positive or negative.

Bit 4, Multiply 1 Underflow

Bit 5, Multiply 2 Underflow

Bit 6, Multiply 3 Underflow

Bit 7, Multiply 4 Underflow

A positive result exponent occurred when both the A and B operand exponents were negative. The result itself can be either positive or negative.

Bit 8, Divide 2 Overflow

A negative result exponent occurred when the A operand exponent was positive and the B operand exponent was negative. The result itself can be positive or negative.

Bit 9, Divide 2 Underflow

A positive result exponent occurred when the A operand exponent was negative and the B operand exponent was positive. The result itself can be positive or negative.

Bit 10, Divide 2 Indefinite

Divide unit 2 attempted to divide 0 by 0. The result is positive if the signs of the operand 0s are alike and negative if the signs of the operand 0s are different. The result coefficient is 77777777_8.

Bit 11, Divide 2 Fault

Divide unit 2 attempted to divide a nonzero operand by 0. The sign of the result is the same as the sign of the nonzero operand. The result coefficient is 77777777_8.

Word 6, Arithmetic Errors 3

Bit 0, Square Root 1 Fault

Bit 1, Square Root 2 Fault

A negative operand was supplied to the indicated square root unit. The positive square root of the negative operand is in the root register when the operation completes.

Bit 2, Numerical Conversion Overflow

Floating-point: A postconversion exponent exceeded the range determined by the current NCC and RPA words.

Fixed-point: A negative result occurred when a preconversion exponent was subtracted from the RPA word.

Bit 3, Numerical Conversion Underflow

Floating-point: A postconversion exponent is below the range determined by the current NCC and RPA words.

Fixed-point: A result greater than 37_8 occurred when a preconversion exponent was subtracted from the RPA word.

Bits 4 through 11, Unused

Word 7, Memory Errors

Bit 0, Access A Parity Error

Bit 1, Access B Parity Error

Bit 2, Access C Parity Error

MAP detected a parity error in the indicated data storage access.

Bit 3, Subcontrol Parity Error

Bit 4, Control Parity Error

Bit 5, Macro Parity Error

MAP detected a parity error in the indicated control unit memory.

Bit 6, Access A Range Fault

Bit 7, Access B Range Fault

Bit 8, Access C Range Fault

The indicated access attempted to read from or write into an address larger than the largest data storage address available.

Bit 9, ECS Parity Error

Bit 10, ECS Abort

Bit 11, ECS Field Length Error.

The ECS controller returned the indicated signal to MAP.

7040_8 Select Inactive on EOT

This function enables the PPU channel interface to send an inactive signal to (that is, disconnect) the functioning PPU channel when one of the following conditions occurs.

- A transfer from MAP is in process, the control unit has sent an end of transfer (EOT) signal to the PPU channel interface, and the PPU channel has accepted all words of the transfer, including the checkword.
- A transfer to MAP is in process, the control unit has sent an EOT signal to the PPU channel interface, and the PPU channel interface has received a word from the PPU channel.
- A PPU channel has accepted the eighth status word.

The inactive remains enabled until the PPU sends a $7XXX_8$ function without a 040_8 component.

7100_8 Start Execution

This function initiates microinstruction execution at the address currently in the control memory program address register (CPR). The 7100_8 function is provided for diagnostic use.

7200_8, Enable Active

This function enables the control unit to send an active signal to the PPU channel. The 7200_8 function has no effect when used with the basic internal controlware provided by Control Data because this controlware never attempts to activate a PPU channel.

The active remains enabled until the PPU sends a $7XXX_8$ function without a 200_8 component.

Controlware Functions

Table 2-3 lists MAP controlware functions. Except for $20XX_8$ functions, controlware functions from one category may be combined and sent together. For example, the 1001_8 (start macro memory instruction execution) and 1002_8 (load macro memory) functions can be sent together with a 1003_8 function.

TABLE 2-3. MAP CONTROLWARE FUNCTIONS

Category	Function	Name
Load controlware from PPU channel	1001_8 1002_8 1004_8 1010_8	Start macro memory instruction execution Load macro memory Load subcontrol memory Load control memory
Dump controlware to PPU channel	2001_8 2002_8 2004_8 2010_8	Transfer register files to macro memory Dump macro memory Dump subcontrol memory Dump control memory
Change execution	3000_8	Change macro memory instruction execution
Load controlware from ECS	4002_8 4004_8	Load subcontrol memory from ECS Load control memory from ECS

A parameter block is associated with each controlware function. Each parameter block must be accompanied by a checkword.

All controlware functions except load from ECS (400X8) are executed from the ROM portion of control memory and thus are not affected by changes in internal controlware.

Controlware Function Issue

The following sequence outlines PPU activity associated with sending a controlware function.

1. Send a 7020_8 (select status words) function and input the status block.
2. Examine the status block for the following conditions.
   a. Appropriate PPU channel access reserved? (Word 0, bits 2 and 3) If not, go to 1.
   b. Function busy? (Word 0, bit 4) If yes, go to 1.
   c. MAP busy? (Word 0, bit 1) If yes, go to 1.
   d. Any errors? (Word 0, bits 5 through 10) If yes, go to an error routine.

3. Send the controlware function to MAP.
4. Wait for the function reply.
5. Activate the PPU channel and send the parameter block associated with the controlware function. Generate the checkword as the transfer proceeds.
6. Send the checkword after the last data word of the transfer and then disconnect the channel.
7. If the function sequence is complete, send a 7002_8 (release channel) function to make the PPU channel interface available to either PPU channel.

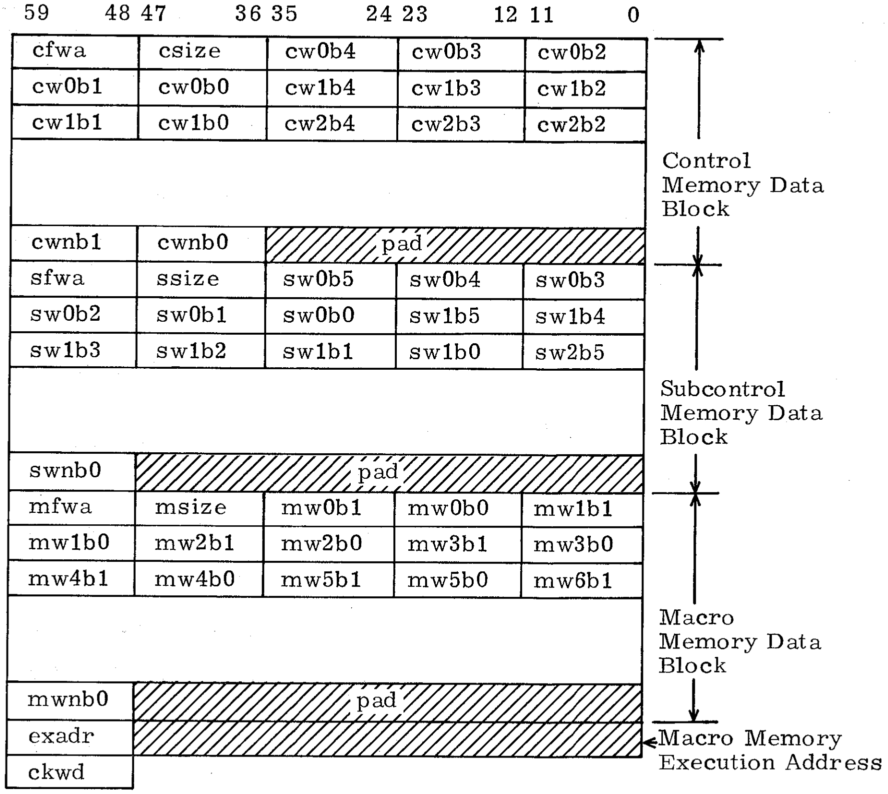
$10XX_8$ Load Controlware From PPU Channel

These four functions may be used singly or combined to accomplish one or more of the following tasks.

1001_8	Starts macro memory instruction execution at the address provided in the accompanying parameter block.
1002_8	Loads external controlware from the PPU channel into macro memory as specified in the accompanying parameter block.
1004_8	Loads internal controlware from the PPU channel into subcontrol memory as specified in the accompanying parameter block.
1010_8	Loads internal controlware from the PPU channel into control memory as specified in the accompanying parameter block.

Figure 2-6 shows the format of the parameter block associated with $10XX_8$ functions. The block is illustrated in 60-bit format because that is the way the block is output from the MAP assembler post-processor. Each PPU word is one 12-bit byte of the block. Striped fields are zero-filled.

The final byte of the transfer, ckwd, is generated by the PPU during the transfer. MAP halts immediately after the transfer if it detects a checkword error.



59 48	47 36	35 24	23 12	11 0
cfwa	csize	cw0b4	cw0b3	cw0b2
cw0b1	cw0b0	cw1b4	cw1b3	cw1b2
cw1b1	cw1b0	cw2b4	cw2b3	cw2b2
cwnb1	cwnb0	pad		
sfwa	ssize	sw0b5	sw0b4	sw0b3
sw0b2	sw0b1	sw0b0	sw1b5	sw1b4
sw1b3	sw1b2	sw1b1	sw1b0	sw2b5
swnb0	pad			
mfwa	msize	mw0b1	mw0b0	mw1b1
mw1b0	mw2b1	mw2b0	mw3b1	mw3b0
mw4b1	mw4b0	mw5b1	mw5b0	mw6b1
mwnb0	pad			
exadr	▨			
ckwd				

Control Memory Data Block
Subcontrol Memory Data Block
Macro Memory Data Block
Macro Memory Execution Address

Figure 2-6. $10XX_8$ Function Parameter Block

Portions of the parameter block must either be omitted or be replaced with zero-fill when not all of the $10XX_8$ functions are combined together. When the 1001_8 function is not part of the combination, the macro memory execution address (exadr) and its four bytes of zero-fill must be omitted from the parameter block. When the 1002_8, 1004_8, or 1010_8 function is not part of the combination, the macro memory, subcontrol memory, or control memory data block, respectively, must be replaced with five bytes of zero-fill.

Mnemonics in figure 2-6 are defined as follows:

cfwa First control memory address to be loaded.

csize Number of 60-bit control memory words to be loaded.

cwibj Byte j of control memory word i. Bytes are numbered right to left, 0 through 4.

sfwa First subcontrol memory address to be loaded.

ssize Number of 72-bit subcontrol memory words to be loaded.

swibj Byte j of subcontrol memory word i. Bytes are numbered right to left, 0 through 5.

mfwa First macro memory address to be loaded.

msize Number of 18-bit macro memory words to be loaded.

mwibj Byte j of macro memory word i. Bytes are numbered right to left, 0 and 1. An 18-bit macro memory word is right-justified in the 24 bits formed by bytes 0 and 1. The leftmost six bits of byte 1 are zero-filled.

exadr Macro memory execution address.

pad From zero to four zero-filled bytes.

ckwd Checkword.

$20XX_8$ Dump Controlware to PPU Channel

These four functions perform the following tasks.

2001_8 Transfers the contents of control unit register files to macro memory as shown in figure 2-7.

Register	Macro Memory Address
K0	1700_8
↓	↓
K15	1717_8
A0	1720_8
↓	↓
A15	1737_8
B0	1740_8
↓	↓
B15	1757_8
C0	1760_8
↓	↓
C15	1777_8

Figure 2-7. Register File/Macro Memory Transfer

2002_8 Dumps external controlware from macro memory to the PPU channel as specified in the accompanying parameter block.

2004_8 Dumps internal controlware from subcontrol memory to the PPU channel as specified in the accompanying parameter block.

2010_8 Dumps internal controlware from control memory to the PPU channel as specified in the accompanying parameter block.

The 2001_8 and 2002_8 functions may be combined and sent as a 2003_8 function. In this case, MAP transfers the register files to macro memory before proceeding with the dump. The 2003_8 function is the only combination of $20XX_8$ functions allowed.

Figure 2-8 shows the format of the parameter block associated with 2002_8, 2004_8, and 2010_8 functions. A parameter block should not be sent with a 2001_8 function.

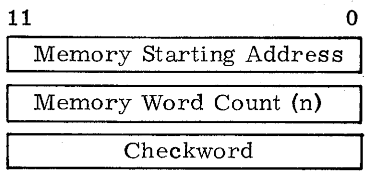


Figure 2-8. $20XX_8$ Function Parameter Block

After MAP has accepted the parameter block, the PPU should activate the PPU channel and prepare to receive a block of 12-bit words. The size of the block is determined by the memory being dumped and the memory word count (n) as follows:

Function	Memory Dumped	Block Size (12-bit words)
2002_8	Macro memory	2n+1
2004_8	Subcontrol memory	6n+1
2010_8	Control memory	5n+1

The added word in each block is the checkword.

When macro memory is being dumped, each macro memory word is placed in the rightmost 18 bits of each 24-bit pair of PPU words.

Figure 2-9 shows the block format for each type of memory dump.

Mnemonics and byte numbering for figure 2-9 are the same as those used for figure 2-6.

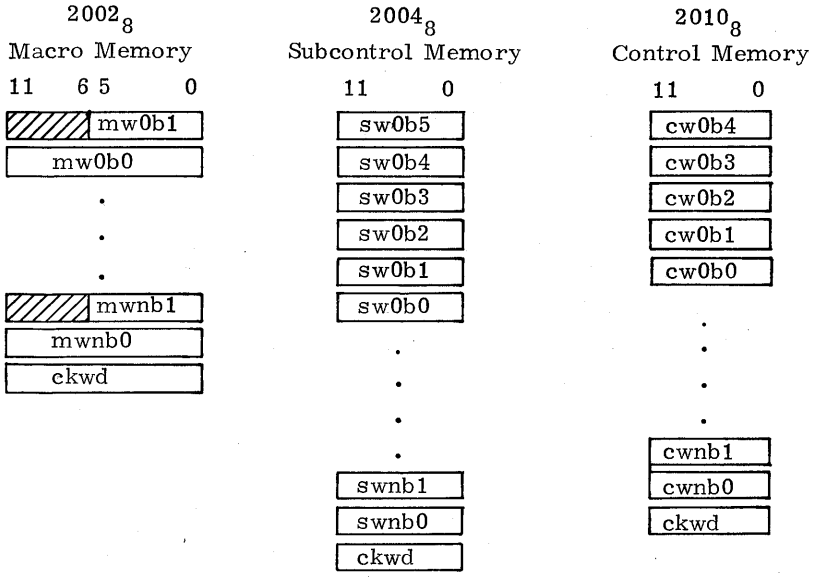


Figure 2-9. Memory Dump Block Formats

3000_8 Change Execution

This function initiates external controlware execution at the macro memory address provided in the accompanying parameter block. Figure 2-10 shows the parameter block associated with the 3000_8 function.

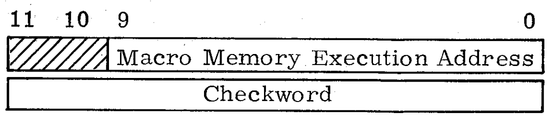


Figure 2-10. 3000_8 Function Parameter Block

$400X_8$ Load Controlware From ECS[†]

These two functions may be used singly or combined to accomplish one or both of the following tasks.

4002_8	Loads internal controlware from ECS into subcontrol memory as specified in the accompanying parameter block.
4004_8	Loads internal controlware from ECS into control memory as specified in the accompanying parameter block.

Figure 2-11 shows the parameter block associated with the $400X_8$ function.

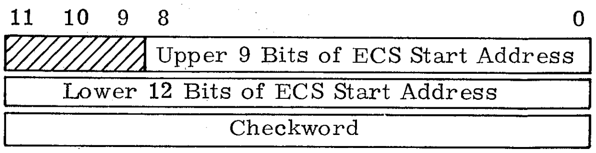

11 10 9 8 0
Upper 9 Bits of ECS Start Address
Lower 12 Bits of ECS Start Address
Checkword

Figure 2-11. $400X_8$ Function Parameter Block

Figure 2-12 shows the format of the internal controlware block stored in ECS. If subcontrol memory is to be loaded alone (4002_8 function), the ECS start address should be set to the first word address (in ECS) of the subcontrol memory data block.

Mnemonics and byte numbering for figure 2-12 are the same as those used for figure 2-6.

MACRO MEMORY INSTRUCTIONS

These 18-bit instructions are the elements used to generate external controlware. Macro memory instructions are actually pseudo instructions whose characteristics are determined by microinstructions executing from the ROM portion of control memory. Table 2-4 lists macro memory instructions.

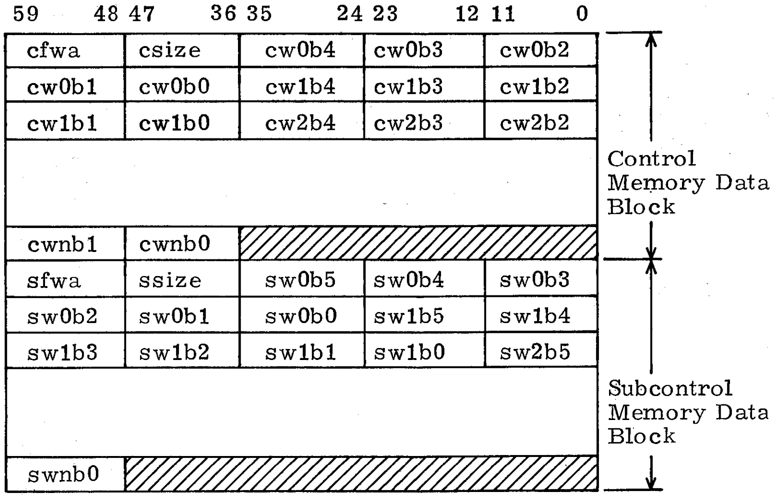
020000_8 No Operation

This instruction transfers control to the instruction in the next macro memory address.

$04XXXX_8$ Return Jump

This instruction allows control to transfer from one macro memory routine to another and to then return to the original routine. Figure 2-13 shows the return jump instruction format.

[†] MAP can execute $400X_8$ functions only when appropriate portions of the basic internal controlware are resident in control and subcontrol memory.



59 48	47 36	35 24	23 12	11 0
cfwa	csize	cw0b4	cw0b3	cw0b2
cw0b1	cw0b0	cw1b4	cw1b3	cw1b2
cw1b1	cw1b0	cw2b4	cw2b3	cw2b2
cwnb1	cwnb0	▨	▨	▨
sfwa	ssize	sw0b5	sw0b4	sw0b3
sw0b2	sw0b1	sw0b0	sw1b5	sw1b4
sw1b3	sw1b2	sw1b1	sw1b0	sw2b5
swnb0	▨	▨	▨	▨

Control Memory Data Block
Subcontrol Memory Data Block

Figure 2-12. ECS Format For Internal Controlware

TABLE 2-4. MACRO MEMORY INSTRUCTIONS

Instruction	Name
020000_8	No operation
$04XXXX_8$	Return jump
100000_8	Halt
$20XXXX_8$	Jump
$40XXXX_8$	Execute macro

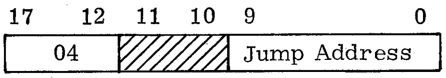


Figure 2-13. Return Jump Instruction Format

The return jump instruction takes the current macro memory program address, adds 1 to it, places the result in a jump ($20XXXX_8$) instruction, stores the jump instruction at the jump address, and transfers control to the instruction at jump address plus 1. The routine associated with the jump address returns control to the original routine by executing a jump to the jump address.

100000_8 Halt

This instruction stops control memory instruction execution and thus stops macro memory instruction execution. The halt instruction is used to terminate macro memory instruction strings. MAP remains stopped until a PPU sends a new function.

$20XXXX_8$ Jump

This instruction transfers control to the jump address. Figure 2-14 shows the jump instruction format.

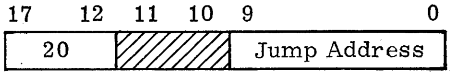


Figure 2-14. Jump Instruction Format

$40XXXX_8$ Execute Macro

This instruction transfers control to one of 77_8 possible macros in control memory. When it has completed processing the selected macro, the internal controlware returns control to the instruction at macro memory address P+2, assuming that the execute macro instruction was at address P. Figure 2-15 shows the execute macro instruction and its associated parameter word.

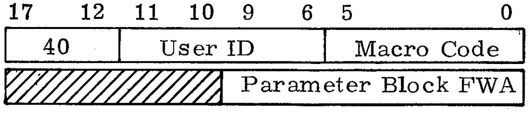


Figure 2-15. Execute Macro Instruction Format

Execute macro instruction fields and parameter word are defined as follows:

Macro code — Number ranging from 0 to 77_8 which identifies the macro to be executed.

User ID — Number ranging from 0 to 77_8 which has several possible uses:

- Specific macro identifier
- Sequence counter for a group of macros executed together
- Interruptible macro string identifier

Parameter block FWA — First word address in macro memory of the parameter block associated with this macro.

CONTROL MEMORY INSTRUCTIONS

These 60-bit instructions (called microinstructions) are the primary elements used to generate internal controlware. Table 2-5 lists control memory instructions. Mnemonics used in the instruction descriptions are defined in table 2-1. Parentheses indicate the contents of a register or counter.

TABLE 2-5. CONTROL MEMORY INSTRUCTIONS

Octal Operation Code	Name
0	Control processor data transfer
1	I/O data transfer
2	Interregister
3	Arithmetic '
4	Jump control
6	Arithmetic index load
7	File modify

0 Control Processor Data Transfer

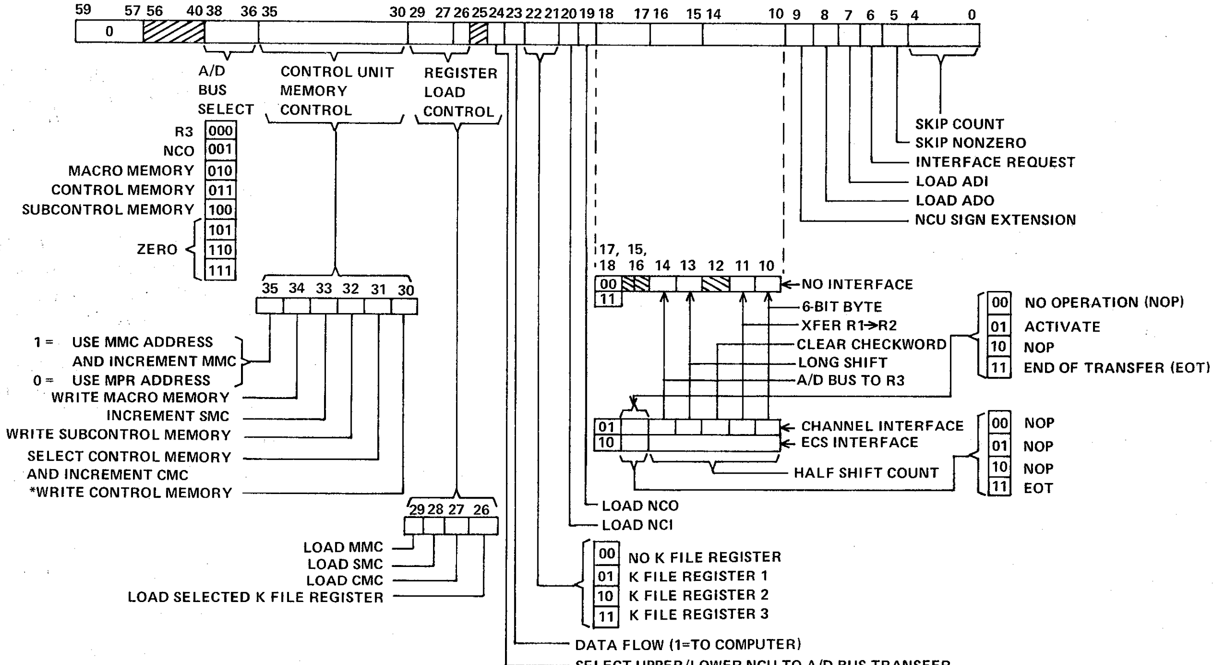
This microinstruction controls activity associated with data transfers between a PPU channel or ECS and macro memory, control memory, or subcontrol memory. Figure 2-16 shows the format of the 0 microinstruction.

Bits 0 Through 4 (Skip Count)

These bits form a 5-bit twos complement number which is added to CPR if the K file register specified by bits 21 and 22 satisfies the condition specified by bit 5. The 5-bit number provides a plus 15, minus 16 skip capability for generating microinstruction loops.

Bit 5 (Skip Nonzero)

When bit 5 is 1 and the K file register specified by bits 21 and 22 is nonzero, the skip count is added to CPR. When bit 5 is 0 and the K file register specified by bits 21 and 22 is zero, the skip count is added to CPR. When neither condition is present, CPR increments by 1 and the selected K file register decrements by 1.



*DO NOT USE THIS BIT WITHOUT HAVING THE CONTROL MEMORY SELECT BIT SET ALSO.

Figure 2-16. 0 Control Processor Data Transfer Microinstruction

Bit 6 (Interface Request)

When bits 17 and 18 form the codes 10_2 or 01_2, bit 6 enables the transfer of one 12-bit word to or from a PPU channel or one 60-bit word to or from the buffer in the ECS interface. Microinstruction execution is delayed until the word is transferred. When bits 17 and 18 form the codes 00_2 or 11_2, bit 6 has no effect.

Bits 7 and 8 (A/D Load Control)

These bits enable loading of various registers in the A/D unit. The registers loaded are a function of bits 7, 8, 17, 18, and 23 as shown in table 2-6.

TABLE 2-6. A/D UNIT REGISTER LOADING

Bit					Load Register		
18,	17	23	8	7	R1, R2	R3	R6, R7
0	1	0	0	1	1	0	0
0	1	0	1	0	0	1	0
0	1	0	1	1	1	1	0
0	1	1	0	1	0	1	0
0	1	1	1	0	1	0	0
0	1	1	1	1	1	1	0
1	0	0	0	1	0	0	0
1	0	0	1	0	0	1	0
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	0
1	0	1	1	0	0	0	1
1	0	1	1	1	0	1	1

Bit 9 (Numerical Conversion Sign Extension)

This bit causes the numerical conversion unit to extend the sign bit of the numerical conversion input (NCI) register to 32 bits. When bit 9 and bit 19 (load NCO) are 1, all 32 bits of the numerical conversion output (NCO) register assume the state of the sign bit of NCI.

Bits 10 through 18 (Interface Select Field)

Bits 17 and 18 of this field specify the interface as follows:

Bit 18	17	Interface
0	0	No interface
0	1	PPU channel interface
1	0	ECS interface
1	1	No interface

Bits 10 through 16 are interpreted according to the selected interface.

1. No interface or PPU channel interface (codes 00_2, 01_2, 11_2)

Bit 10 Selects 6-bit byte if 1; 8-bit byte if 0.

Bit 11 Transfers R1 to R2.

Bit 12 Clears checkword accumulator. When the PPU channel interface is selected and bit 6 (interface request) is set in the same instruction, the checkword accumulator is cleared before the data word is transferred.

Bit 13 When this bit is 1, the A/D unit enables the high order byte of R3 to the lower order byte position of R1. When bit 13 is 0, the A/D unit enables the byte whose high order bit is located at bit 31 of R3 to the lower order byte position of R1.

Bit 14 Transfers the quantity on the A/D bus to R3. This bit disables the effect of bit 13.

Bits 15 and 16 Form a 2-bit code which is interpreted as follows:

Bit 16	15	Selected Activity
0	0	No operation.
0	1	Activate PPU channel if enabled by function.
1	0	No operation.
1	1	End of transfer. Test PPU checkword (PPU write) or send MAP checkword (PPU read).

2. ECS interface (code 10_2)

Bits 10 through 14 Form a 5-bit shift count which is multiplied by 2 and applied to the left and right shifters in the A/D unit. As shown in figure 2-17, the shift count specifies the number of bit positions between a selected 32-bit path and the highest order bit of R5/R4 or R7/R6.

Bits 15 and 16 Form a 2-bit code which is interpreted as follows:

Bit 16	15	Selected Activity
0	0	No operation.
0	1	No operation.
1	0	No operation.
1	1	Send EOT signal to ECS interface.

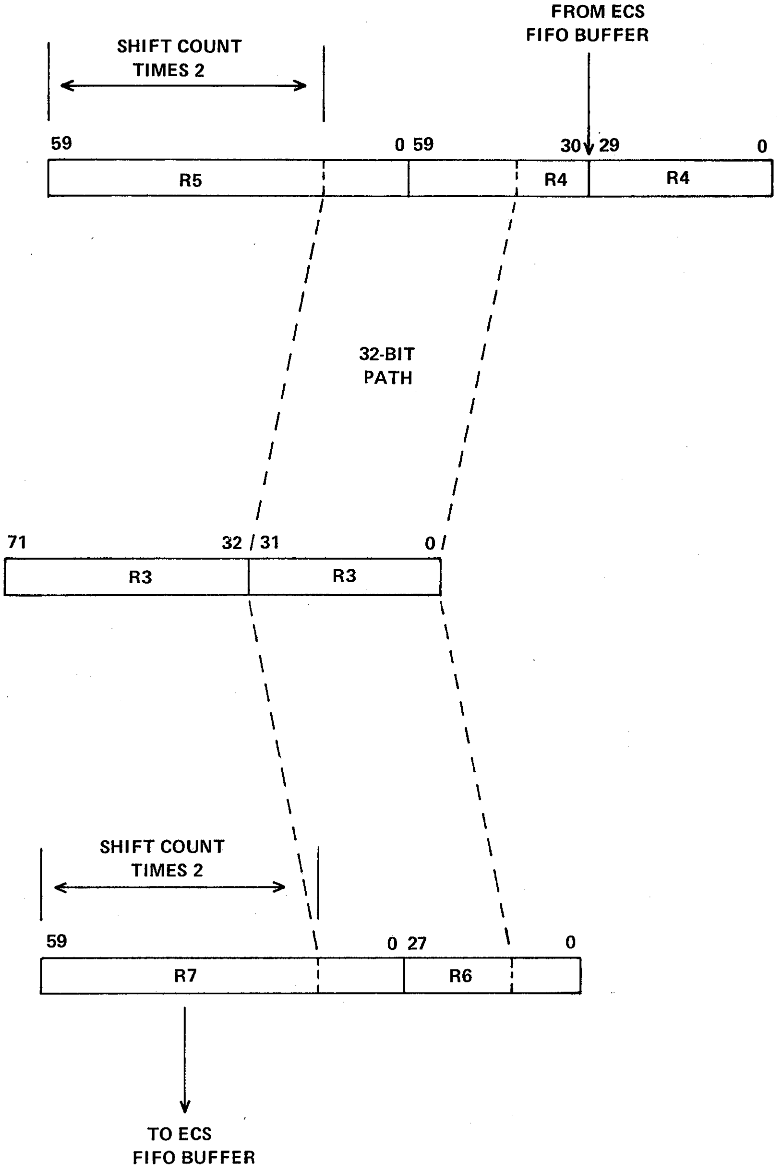


Figure 2-17. Shift Count

Bit 19 (Load NCO)

This bit loads the numerical conversion unit result into NCO.

Bit 20 (Load NCI)

When bit 7 of the NCC word is 1 (external to internal conversion), this bit loads bits 0 through 31 of the A/D bus into NCI. When NCC bit 7 is zero (internal to external conversion), this bit loads operand bus 2 into NCI. Control memory cannot be sent directly to NCI via the A/D bus.

Bits 21 and 22 (K File Select)

These bits select K file register 1, 2, or 3 for loading or for testing and decrementing. If both bits are 0, no register is selected and the test result is forced to be nonzero. The selected register decrements to 777777_8 after zero is detected.

Bit 23 (Data Flow)

When 1, this bit selects data transfer from MAP. When 0, bit 23 selects data transfer to MAP.

Bit 24 (Select Upper NCU to A/D Bus)

This bit sends numerical conversion unit bits 16 through 31 to bits 0 through 15 of the A/D bus and to bits 16 through 31 of the A/D bus if the code formed by microinstruction bits 36 through 38 is 001_2.

Bit 25 (Unused)

Bit 26 (Load K File Register)

This bit transfers bits 0 through 17 of the A/D bus to the K file register selected by bits 21 and 22.

Bit 27 (Load CMC)

This bit transfers bits 0 through 11 of the A/D bus to CMC.

Bit 28 (Load SMC)

This bit transfers bits 0 through 9 of the A/D bus to SMC.

Bit 29 (Load MMC)

This bit transfers bits 0 through 9 of the A/D bus to MMC.

Bits 30 through 35 (Memory Control Field)

This field controls address information and reading from and writing into the three control unit memories.

Bit 30 Enables bits 0 through 59 of the A/D bus to be written in control memory at the address enabled by bit 31. This bit must not be 1 unless bit 31 is also 1.

Bit 31 Reads from or writes into the control memory address specified by CMC and then increments CMC by 1.

Bit 32 Enables bits 0 through 71 of the A/D bus to be written in the subcontrol memory address specified by SMC. When bit 32 is 0, the subcontrol memory address specified by SMC is read.

Bit 33 Increments SMC by 1 after the current address has been read or written.

Bit 34 Enables bits 0 through 17 of the A/D bus to be written in the macro memory address specified by bit 35. When bit 34 is 0, the macro memory address specified by bit 35 is read.

Bit 35 Reads from or writes into the macro memory address specified by MMC and then increments MMC by 1. When bit 35 is 0, MPR is used for the address. MPR is not altered.

NOTE

The 0 microinstruction should not be used for direct memory to memory transfers via the A/D bus.

Bits 36 through 38 (A/D Bus Select)

These bits form a 3-bit code which selects the quantity to be gated to the A/D bus. Unused bits on the A/D bus are zero-filled. When the quantity on the A/D bus is to be used, it must be used during the instruction containing the code for that quantity.

NOTE

When a memory being written is also gated to the A/D bus, the quantity on the bus becomes zero and the quantity written in the memory is indeterminate.

Codes and selected quantities are as follows:

Bit			Quantity Selected
38	37	36	To A/D Bus
0	0	0	R3 (A/D output register)
0	0	1	Numerical conversion output register
0	1	0	Selected macro memory cell
0	1	1	Selected control memory cell
1	0	0	Selected subcontrol memory cell
1	0	1	Zero
1	1	0	Zero
1	1	1	Zero

Bits 40 through 56 (Unused)

Bits 57 through 59 (Operation Code)

These bits form the octal digit 0.

1_8 Input/Output Data Transfer

This microinstruction controls activity associated with data transfers between a PPU channel or ECS and MAP data storage. Figure 2-18 shows the format of the 1_8 microinstruction. A 3_8 arithmetic microinstruction and its associated subinstruction must be used to specify bus codes for operand bus 2 and the result bus.

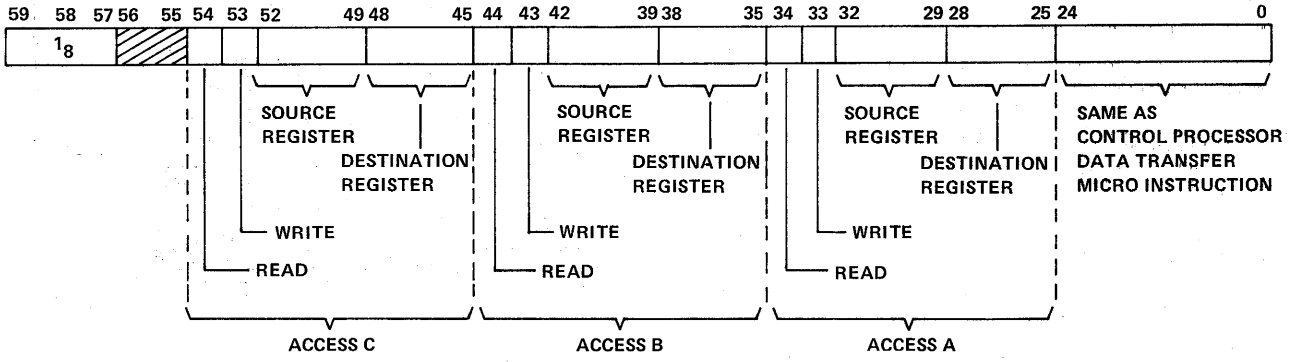
Bits 0 Through 24

These bits perform exactly as bits 0 through 24 of the 0 control processor data transfer microinstruction.

Bits 25 Through 34

This field controls data storage access A.

Bits 25 through 28 — Form a 4-bit number which selects one of the 16 registers in the access A destination file to contain the access A data storage address. The final contents of the selected destination register is obtained by adding the initial contents of the selected destination register to the contents of the selected access A source file register.



NOTE: WHEN BOTH THE READ AND WRITE BITS FOR AN ACCESS ARE SET, THE DESTINATION FILE REGISTER IS CHANGED BUT NO DATA STORAGE REQUEST OCCURS FOR THAT ACCESS.

Figure 2-18. 1_8 Data Transfer Microinstruction

Bits 29 through 32 — Form a 4-bit number which selects one of the 16 registers in the access A source file. The contents of this register is added to the initial contents of the selected access A destination file register.

Bit 33 — Write. Enables a write operation for access A.

Bit 34 — Read. Enables a read operation for access B. When both the read and write bits are set, the destination file register is changed, but no data storage request occurs for that access.

Bits 35 through 44 (Access B Field)

This field controls data storage access B. Bit functions are the same as the functions described for access A, except they apply to access B.

Bits 45 through 54 (Access C Field)

This field controls data storage access C. Bit functions are the same as the functions described for access A, except they apply to access C.

Bits 55 and 56 (Unused)

Bits 57 through 59 (Operation Code)

These bits form the octal digit 1.

2_8 Interregister

This microinstruction manipulates control unit registers and macro memory. It is used to monitor status, update parameters, interpret macro memory instructions, and perform off-line diagnostics. Figure 2-19 shows the format of the 2_8 microinstruction.

Bits 0 through 18 (Immediate Operand or Macro Memory Address)

This field permits an immediate operand or the contents of a macro memory address to be applied to the A input of the ALU. The field also permits a macro memory address to be specified as a destination for an ALU result quantity. Figure 2-19 shows the way bits 16, 17, and 18 specify the immediate operand or the method of address formation.

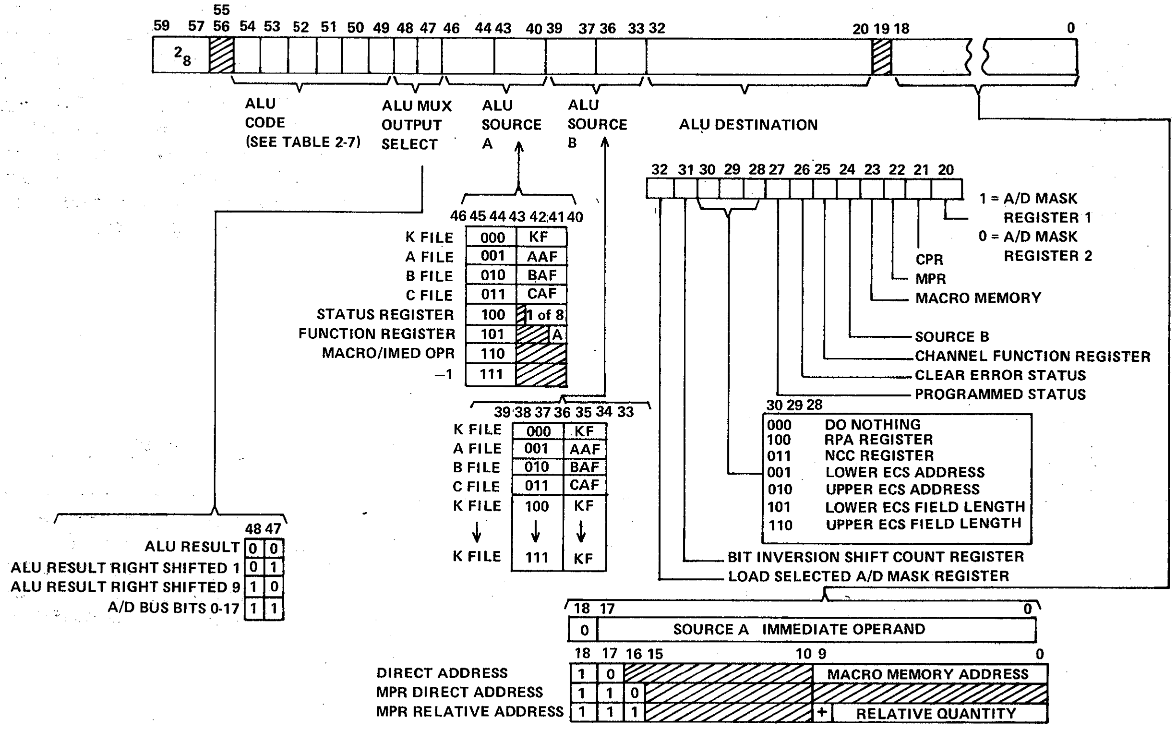


Figure 2-19. 2_8 Interregister Microinstruction

When an immediate operand is specified, bits 0 through 17 are the immediate operand.

When direct addressing is specified, bits 0 through 9 are the macro memory address.

When MPR direct addressing is specified, MPR supplies the macro memory address.

When MPR relative addressing is specified, bits 0 through 9 form a 10-bit twos complement number which is added to MPR to form the macro memory address.

Bit 19 (Unused)

Bits 20 through 32 (ALU Result Destination Field)

This field selects one or more destinations for the ALU result output and provides for clearing error status.

Bit 20 1 selects A/D word 1 mask register; 0 selects A/D word 2 mask register. Bit 32 loads the selected mask register with the ALU result.

Bit 21 Loads CPR with bits 0 through 11 of the ALU result.

Bit 22 Loads MPR with bits 0 through 9 of the ALU result.

Bit 23 Loads the macro memory address specified by bits 0 through 18 with the ALU result.

Bit 24 Loads the source B register selected by bits 33 through 39 with the ALU result.

Bit 25 Loads the PPU channel function register with bits 0 through 11 of the ALU result.

Bit 26 Sets status words 4, 5, 6, and 7 to zero. Since the status words are cleared at the end of the microinstruction, one of the status words can be used as a source A input during the microinstruction.

Bit 27 Loads the programmed status register with bits 0 through 11 of the ALU result

Bits 28 through 30 Form a 3-bit code which loads one of the six registers listed in figure 2-19 with bits 0 through 11 of the ALU result. The NCC register, if selected, receives bits 0 through 10 of the ALU result.

Bit 31 Loads the bit inversion shift count register with bits 0 through 3 of the ALU result.

Bit 32 Loads the A/D mask register selected by bit 20 with the ALU result.

Bits 33 through 39 (ALU Source B Field)

This field permits one of the A, B, C, or K file registers to be specified as the B input to the ALU.

Bits 33 through 36 Form a 4-bit number which selects one of the 16 registers in the file selected by bits 37 through 39.

Bits 37 through 39 Form a 3-bit code which specifies the file to provide the selected register. Codes 0, 1, 2, and 3 select files K, A, B, and C, respectively; codes 4 through 7 each select file K.

Bits 40 through 46 (ALU Source A Field)

This field permits a quantity from one of eight categories to be specified as the A input to the ALU. Interpretation of bits 40 through 43 depends upon the code formed by bits 44 through 46.

Codes 0 through 3 Select files K, A, B, or C, respectively. Bits 40 through 43 form a 4-bit number which specifies one of the 16 registers in the selected file.

Code 4 Selects the status register specified by bits 40 through 42. Status register 0 holds word 0, register 1 holds word 1, and so forth. Status words are described under 7020_8 Select Status Words in this section.

Code 5 Selects the PPU channel function register. Bit 40 indicates to the PPU channel interface that the function has been processed.

Code 6 Selects the quantity specified by bits 0 through 18.

Code 7 Selects -1 as source A.

Bits 47 and 48 (ALU Output Select)

This 2-bit code specifies the form of the quantity designated as the ALU result. Codes and ALU result quantities are as follows:

Bit		ALU Result
17	47	Quantity
0	0	ALU result
0	1	ALU result right-shifted one bit position
1	0	ALU result right-shifted nine bit positions, end-around
1	1	Bits 0 through 17 of the A/D bus. The quantity on the A/D bus is the quantity selected by the last 0 microinstruction.

Bits 49 through 54 (ALU Operation Code Field)

This field selects the operation to be performed by the ALU. Table 2-7 indicates the effect of each bit in the field.

Bits 55 and 56 (Unused)

Bits 57 through 59 (Operation Code)

These bits form the octal digit 2.

3_8 Arithmetic

This microinstruction and its associated subcontrol memory instruction (called a subinstruction) control MAP arithmetic operations. Figure 2-20 shows the format of the 3_8 microinstruction.

Bits 0 Through 5

These bits perform exactly as bits 0 through 5 of the control processor data transfer microinstruction, except that a 4-bit field (bits 21 through 24) rather than a 2-bit field is provided for K file register selection.

Bit 6 (Inhibit K Decrement)

This bit prevents the K file register specified by bits 21 through 24 from being decremented.

Bits 7 through 16 (Subinstruction Address)

This 10-bit number selects the subinstruction to be executed during the next microinstruction time.

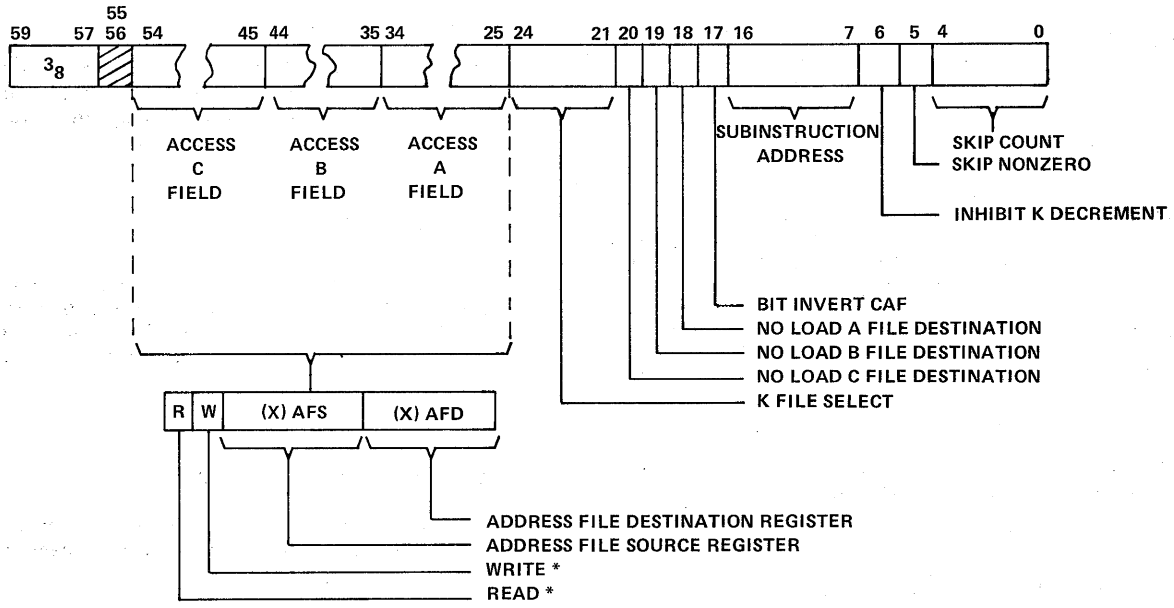
TABLE 2-7. ALU OPERATION CODES

Bit 52	Bit 51	Bit 50	Bit 49	Bit 53=1, Bit 54=x	Bit 53=0, Bit 54=0
0	0	0	0	$F=\overline{A}$	F=A
0	0	0	1	$F=\overline{A}+\overline{B}$	F=A plus $(A\cdot\overline{B})$
0	0	1	0	$F=\overline{A}+B$	F=A plus $(A\cdot B)$
0	0	1	1	F=logical 1	F=A times 2
0	1	0	0	$F=\overline{A}\cdot\overline{B}$	F=(A+B) plus 0
0	1	0	1	$F=\overline{B}$	F=(A+B) plus $(A\cdot\overline{B})$
0	1	1	0	$F=A\overline{\forall}B$	F=A plus B
0	1	1	1	$F=A+\overline{B}$	F=A plus (A+B)
1	0	0	0	$F=\overline{A}\cdot B$	$F=(A+\overline{B})$ plus 0
1	0	0	1	$F=A\forall B$	F=A minus B minus 1
1	0	1	0	F=B	$F=(A+\overline{B})$ plus $(A\cdot B)$
1	0	1	1	F=A+B	F=A plus $(A+\overline{B})$
1	1	0	0	F=logical 0	F=minus 1 (2's comp)
1	1	0	1	$F=A\cdot\overline{B}$	$F=(A\cdot\overline{B})$ minus 1
1	1	1	0	$F=A\cdot B$	$F=(A\cdot B)$ minus 1
1	1	1	1	F=A	F=A minus 1

+ = logical OR A = source A operand

. = logical AND B = source B operand

$\forall$ = exclusive OR F=result of ALU operation

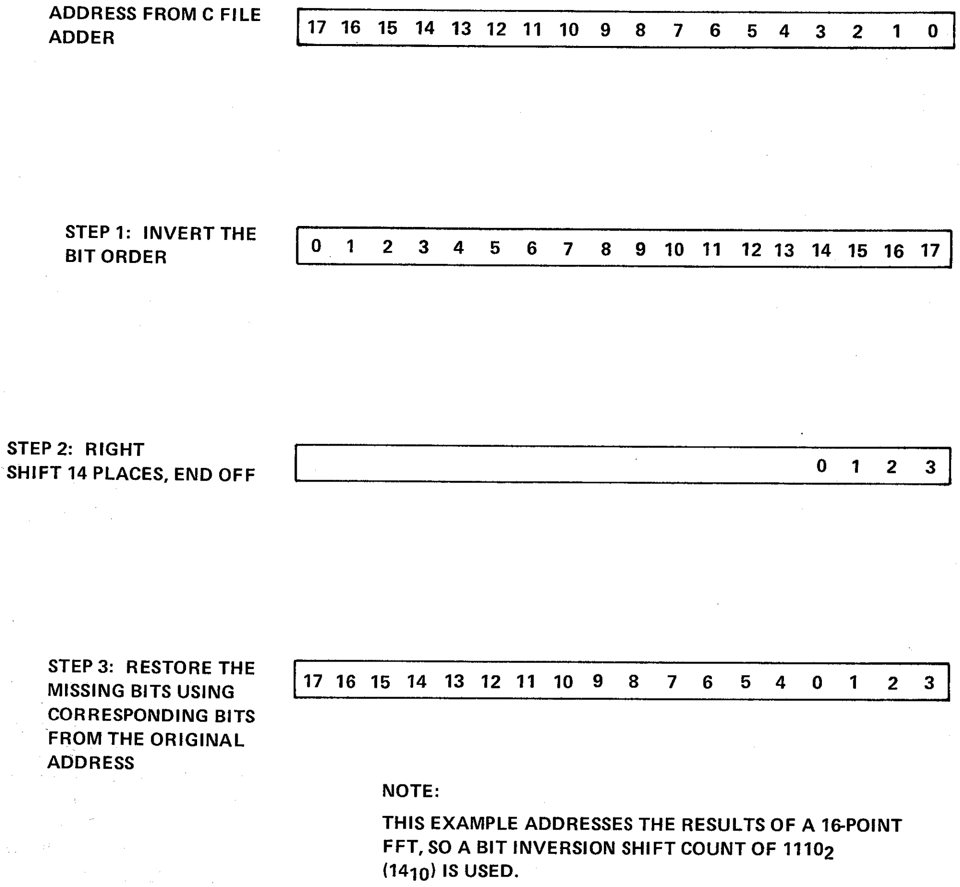


*BOTH BITS SET CAUSES FILE CHANGE BUT NO DATA STORAGE REQUEST.

Figure 2-20. 3_8 Arithmetic Microinstruction

Bit 17 (Bit Invert CAF)

This bit routes the data storage address from the selected C file source register through an inversion network before sending the address to data storage. The inversion network is used for addressing the results of a fast Fourier transform (FFT). Figure 2-21 shows the operation of the inversion network. Since the bit inversion shift count (loaded by the 2_8 microinstruction) is limited to four bits, an 8-point FFT is the smallest transform that can be addressed by the bit inversion network.


ADDRESS FROM C FILE ADDER
17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
STEP 1: INVERT THE BIT ORDER
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
STEP 2: RIGHT SHIFT 14 PLACES, END OFF
0 1 2 3
STEP 3: RESTORE THE MISSING BITS USING CORRESPONDING BITS FROM THE ORIGINAL ADDRESS
17 16 15 14 13 12 11 10 9 8 7 6 5 4 0 1 2 3

NOTE:

THIS EXAMPLE ADDRESSES THE RESULTS OF A 16-POINT FFT, SO A BIT INVERSION SHIFT COUNT OF 1110_2 (14_{10}) IS USED.

Figure 2-21. Bit Inversion Network Operation

Bit 18 (No Load A File Destination)

This bit prevents the new access A address from being written into the selected A file destination register. The new address is formed by adding the contents of the selected A file source register to the initial contents of the selected A file destination register.

Bit 19 (No Load B File Desintation)

This bit has the same effect on access B as bit 18 has on access A.

Bit 20 (No Load C File Destination)

This bit has the same effect on access C as bit 18 has on access A.

Bits 21 through 24 (K File Select)

These four bits form a number which selects one of 16 registers in the K file to be tested for zero, and if bit 6 is 0, to be decremented by one. The selected register decrements to 777777_8 after zero is detected.

Bits 25 through 54 (Data Storage Access Control Fields)

These bits perform exactly as bits 25 through 54 of the 1_8 I/O Data Transfer microinstruction with the following exceptions.

- Since the subinstruction specified by bits 7 through 16 controls the routing of operands and results, read and write operations selected by the arithmetic microinstruction do not complete until the subinstruction has executed. Thus, a data storage operation requested during one microinstruction time completes at the end of the next microinstruction time.
- Each of the accesses has a no load bit to prevent entry of the new address into the selected access file register.
- When the bit inversion network is enabled (bit 17=1) and the no load C file destination bit (bit 20) is 0, the manipulated address is loaded into the selected C file destination register as well as being sent to data storage.

Bits 55 and 56 (Unused)

Bits 57 through 59 (Operation Code)

These bits form the octal digit 3.

4_8 Jump Control

This microinstruction controls branch and halt operations. Figure 2-22 shows the format of the 4_8 microinstruction.

Bits 0 through 12 (Jump Address Field)

This field contains a 12-bit direct or relative jump address (bits 0 through 11) and a relative/direct select bit (bit 12). When bit 12 is 1 and the jump condition is met, the jump address is interpreted as a twos complement number and added to CPR. When bit 12 is 0 and the jump condition is met, the jump address is entered into CPR.

Bit 13 (Jump on False)

When this bit is 1, a jump is executed if the selected condition is false. When bit 13 is 0, a jump is executed if the selected condition is true.

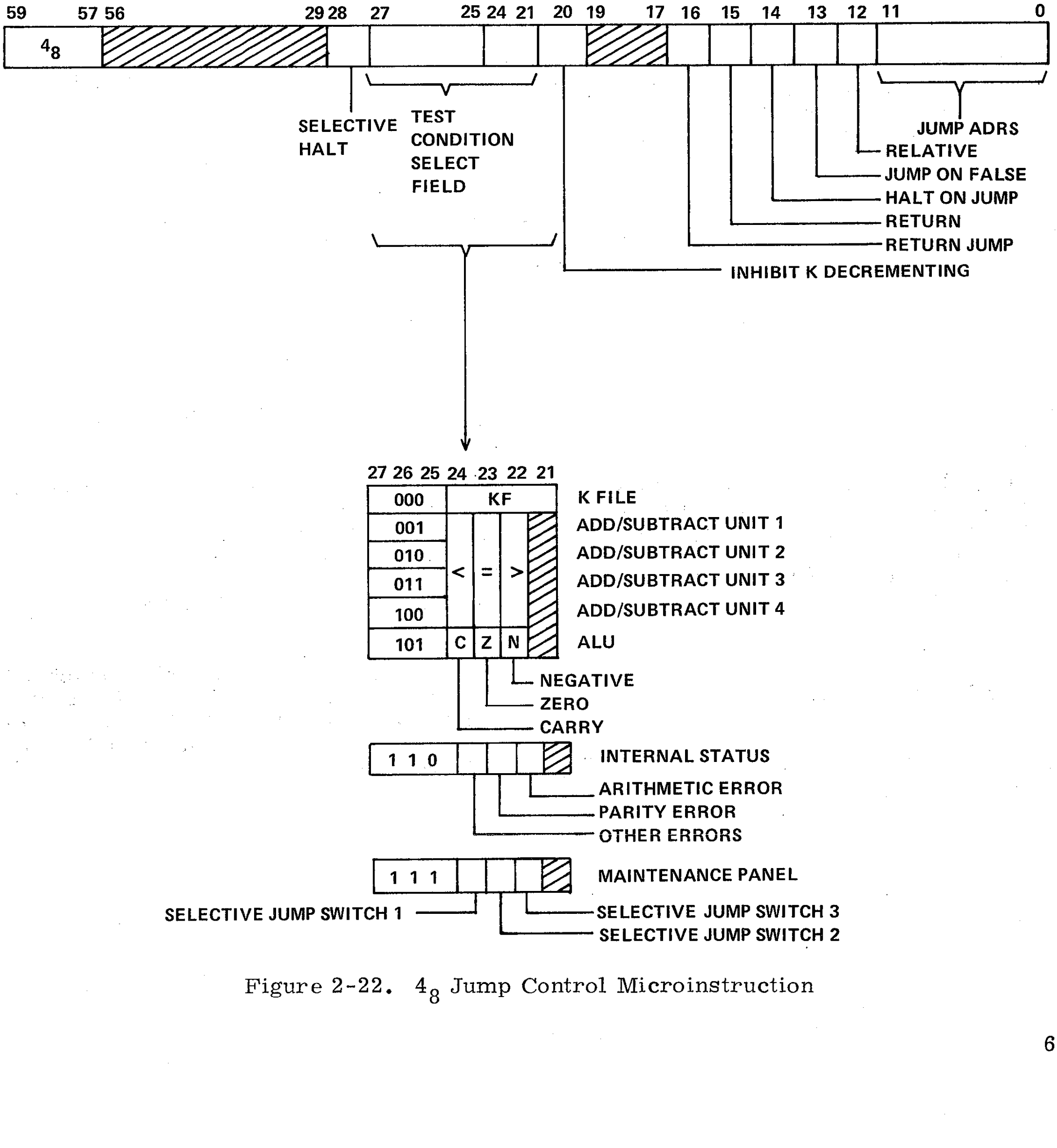


Figure 2-22. 4_8 Jump Control Microinstruction

Bit 14 (Halt on Jump)

When this bit is 1 and the jump condition is met, the jump address is entered into CPR and microinstruction execution stops.

Bit 15 (Return)

When this bit is 1 and the jump condition is met, CPRH transfers to CPR and microinstruction execution continues at the new address. When bit 15 and bit 16 (return jump) are both 1, CPRH is used as the jump address before CPR plus 1 transfers to CPRH.

Bit 16 (Return Jump)

When this bit is 1 and the jump condition is met, CPR plus 1 transfers to CPRH.

Bits 17 through 19 (Unused)

Bit 20 (Inhibit K Decrement)

When bits 21 through 27 specify a K file register to be tested for zero, bit 20 prevents the selected register from being decremented after the test.

Bits 21 through 27 (Test Condition Select Field)

This field permits items from one of eight categories to be specified as the jump test condition(s). Interpretation of bits 21 through 24 depends upon the code formed by bits 25 through 27.

Code 0 Tests for zero the K file register specified by bits 21 through 24. The selected K file register decrements by one after the test, providing bit 20 (inhibit K decrement) is 0.

Codes 1 through 4 Select add/subtract units 1 through 4, respectively, to be tested for the condition(s) specified by bits 22 through 24. Condition and bit assignments are as follows:

Bit	Condition To Be Tested†
22	A greater than B
23	A equal to B
24	A less than B

Code 5 Tests the ALU negative, zero, and carry flags specified by bits 22, 23, and 24, respectively. ALU flags are set each time a 2_8 interregister microinstruction is executed. The test is true if the ALU result is negative, if the ALU result is zero, or if the ALU generated a carry.

Code 6 Tests for arithmetic errors, parity errors, and other errors specified by bits 22, 23, and 24, respectively. The arithmetic error test is true when one or more bits in status registers 4, 5, or 6 are set. The parity error test is true when a parity error is detected in any of the control unit memories, data storage, or ECS. The other error test is true when an error not tested by bit 22 or bit 23 has been detected.

Code 7 Tests maintenance panel SEL JUMP switches 3, 2, and 1 specified by bits 22, 23, and 24, respectively. If the specified switch is on, the test is true.

Bit 28 (Selective Halt)

When this bit is 1 and the maintenance panel SEL STOP switch is on, microinstruction execution stops.

Bits 29 through 56 (Unused)

Bits 57 through 59 (Operation Code)

These bits form the octal digit 4.

6_8 Arithmetic Index Load

This microinstruction and its associated subinstruction are used to load the access A, B, or C register files from data storage via the numerical conversion unit. Figure 2-23 shows the format of the 6_8 microinstruction. Figure 2-24 shows the path between data storage and the selected access register.

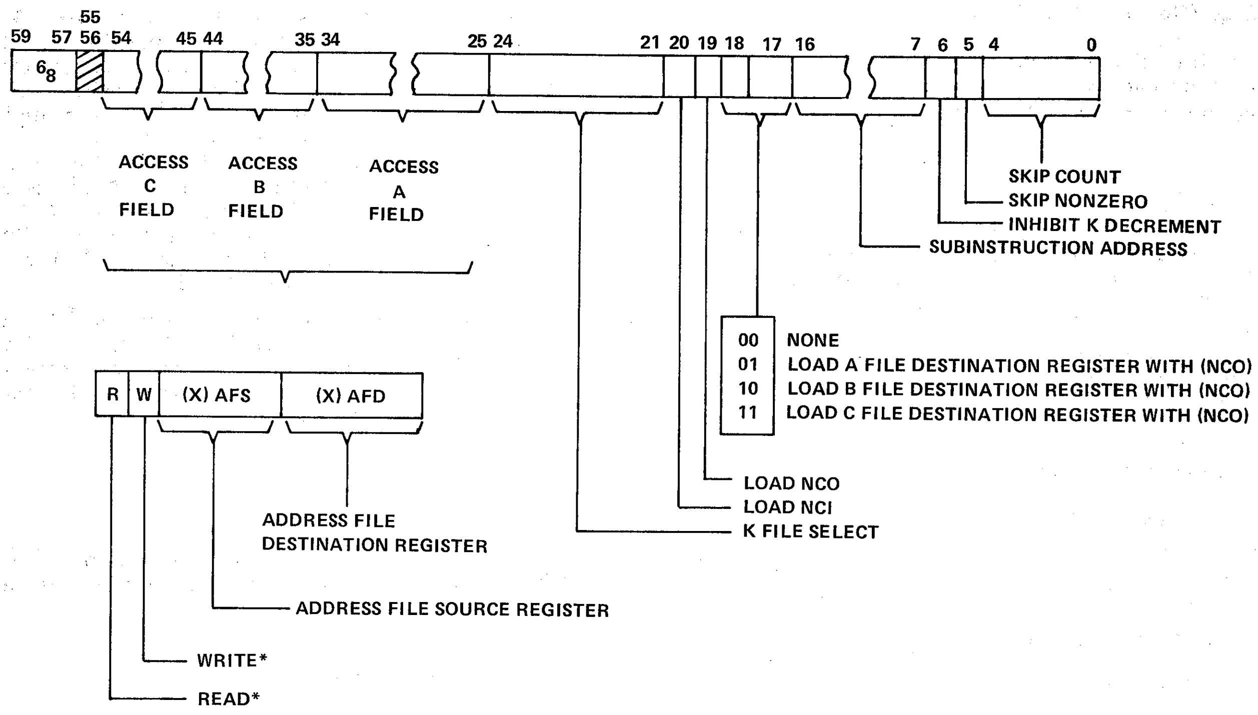
Bits 0 through 16

These bits perform exactly as bits 0 through 16 of the 3_8 arithmetic microinstruction.

Bits 17 and 18 (Address File Select Code)

This code selects none or one of the registers in the A, B, or C files to receive the contents of the NCO register. Figure 2-23 shows the relationship between codes and files.

† The A operand is the operand from operand bus 2 or the product from the associated multiply unit. The B operand is the output of the selected add/subtract unit or the operand from operand bus 1.



*BOTH BITS SET CAUSES FILE CHANGE BUT NO DATA STORAGE REQUEST

Figure 2-23. 6_8 Arithmetic Index Load Microinstruction

Bits 19 and 20

These bits perform exactly as bits 19 and 20 of the 0 control processor data transfer microinstruction.

Bits 21 through 54

These bits perform exactly as bits 21 through 54 of the 3_8 arithmetic microinstruction

Bits 55 and 56 (Unused)

Bits 57 through 59 (Operation Code)

These bits form the octal digit 6.

7_8 File Modify

This microinstruction initializes the A, B, C, and K register files. All four files can be operated upon during one microinstruction time. Figure 2-25 shows the format of the 7_8 microinstruction.

Bits 0 through 24 (K File Load Field)

This field permits an immediate operand or the contents of a macro memory address to be loaded into one of the K file registers. Bits 21 through 24 form a 4-bit number which selects one of the upper 15 K file registers to be loaded. When register 0 is selected, no register is loaded. Figure 2-25 shows the way bits 16, 17, and 18 specify the immediate operand or the method of address formation.

When an immediate operand is specified, bits 0 through 17 are the immediate operand.

When direct addressing is specified, bits 0 through 9 are the macro memory address.

When MPR direct addressing is specified, MPR supplies the macro memory address.

When MPR relative addressing is specified, bits 0 through 9 form a 10-bit twos complement number which is added to MPR to form the macro memory address.

Bits 19 and 20 of this field are unused.

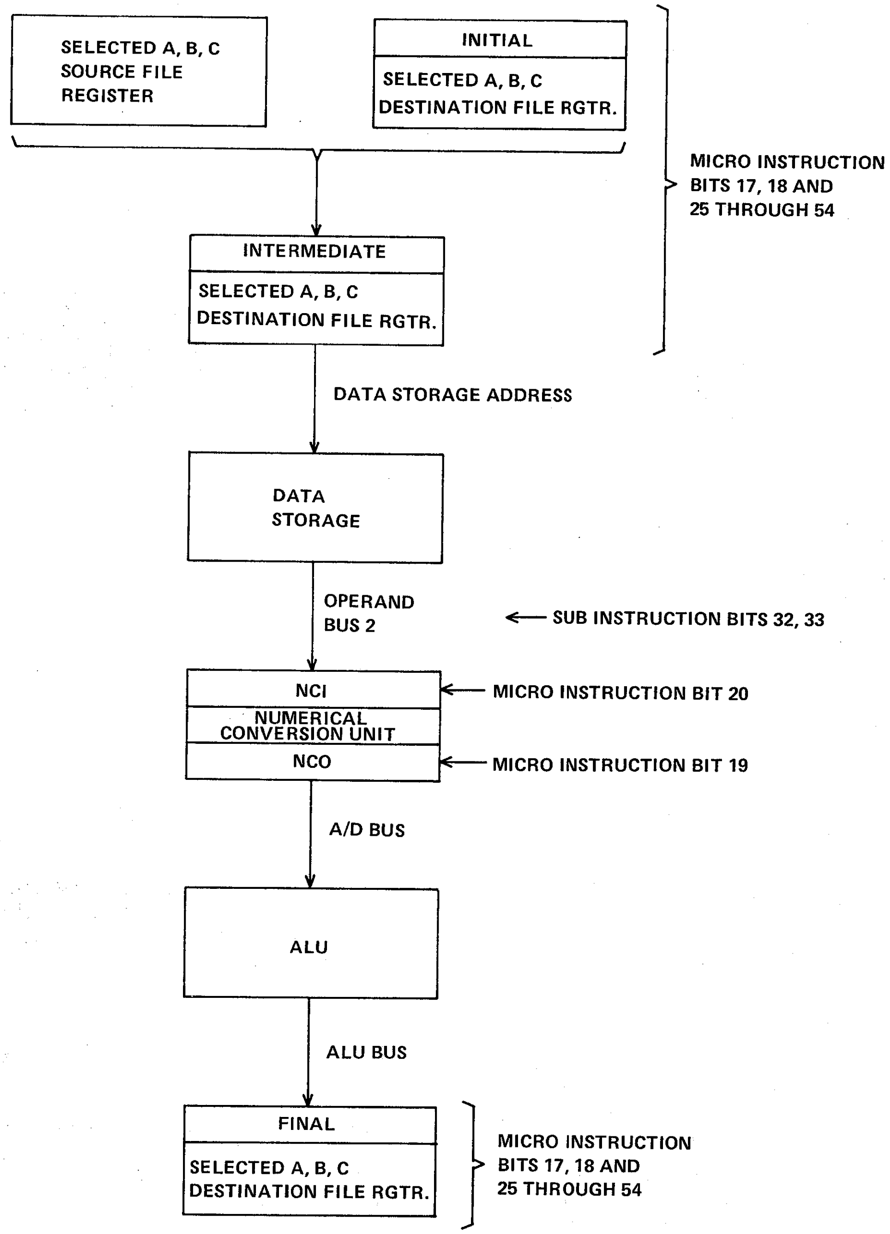


Figure 2-24. Arithmetic Index Load Data Path

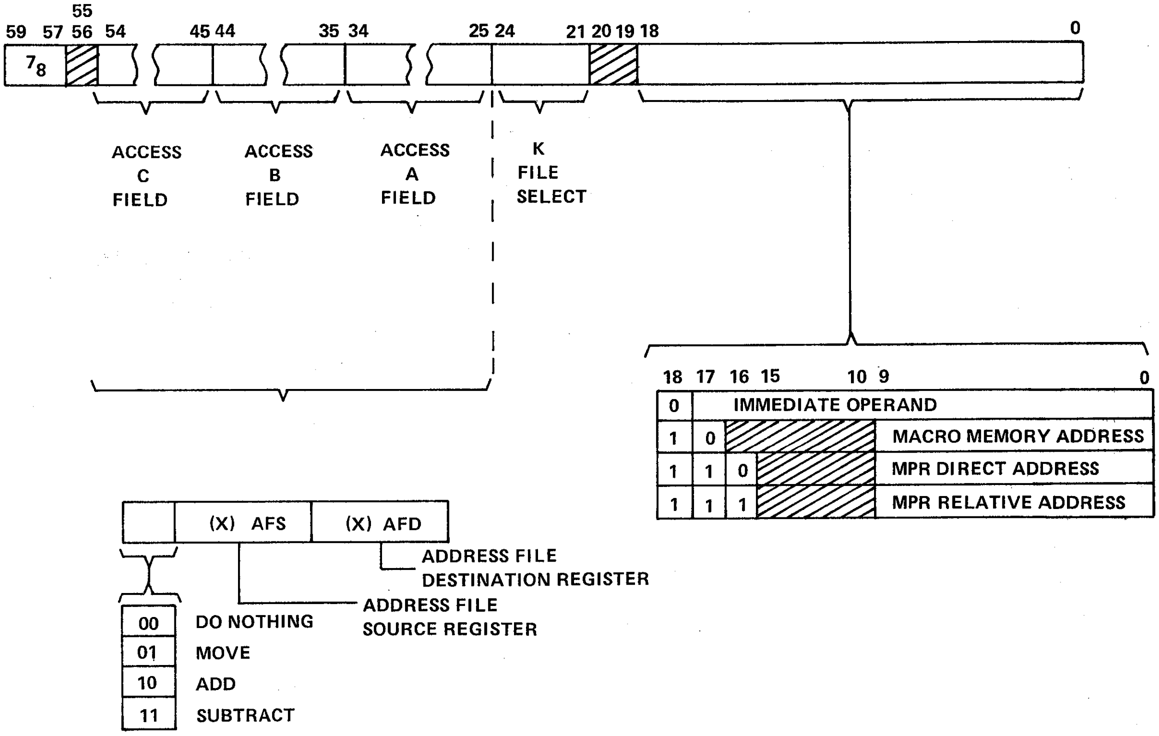


Figure 2-25. 7_8 File Modify Microinstruction

Bits 25 through 34 (Access A Field)

This field controls interregister operations within the A file.

Bits 25 through 28 — Form a 4-bit number which specifies the A file destination register for the interregister operation.

Bits 29 through 32 — Form a 4-bit number which specifies the A file source register for the interregister operation.

Bits 33 and 34 — Form a 2-bit code which specifies the A file interregister operation as follows:

Bit 34	33	Interregister Operation
0	0	Do nothing.
0	1	Move the contents of the source register to the destination register.
1	0	Add the contents of the source and destination registers and store the sum in the destination register.
1	1	Subtract the contents of the source register from the contents of the destination register and store the result in the destination register.

Bits 35 through 44 (Access B Field)

This field controls interregister operations within the B file. Bit functions are the same as the functions described for access A, except they apply to access B.

Bits 45 through 54 (Access C Field)

This field controls interregister operations within the C file. Bit functions are the same as the functions described for access A, except they apply to access C.

Bits 55 and 56 (Unused)

Bits 57 through 59 (Operation Code)

These bits form the octal digit 7.

SUBCONTROL MEMORY INSTRUCTION

This 72-bit instruction (called a subinstruction) is used with 3_8 arithmetic and 6_8 arithmetic index load microinstructions. The subinstruction provides for simultaneous control of operand bus 1, operand bus 2, the result bus, and all arithmetic units. Figure 2-26 shows the format of the subinstruction.

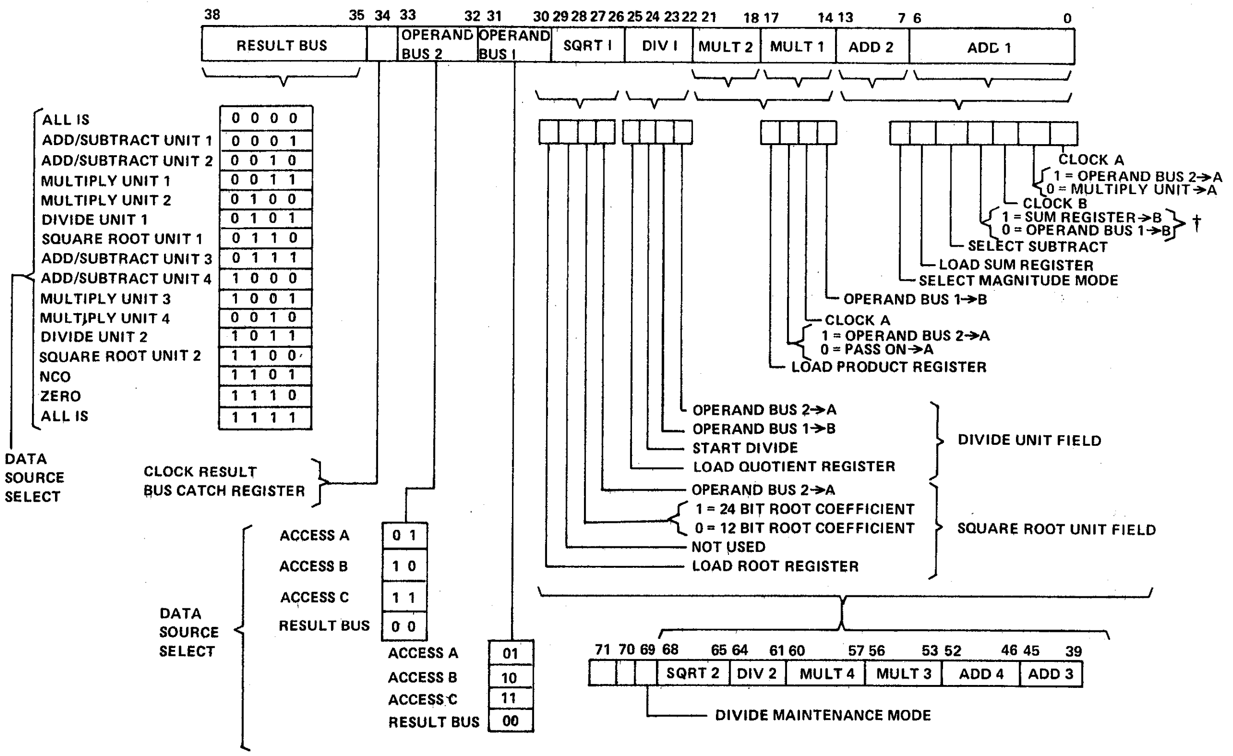


Figure 2-26. Subinstruction

Bits 0 through 6 (Add/Subtract Unit 1 Field)

Bit 0 (Clock A)

This bit loads the A feeder register with the operand selected by bit 1.

Bit 1 (Enable Bus 2 To A)

When 1, this bit enables operand bus 2 to the A feeder register. When 0, this bit enables the product from the multiply unit associated with this adder to the A feeder register.

Bit 2 (Clock B)

This bit loads the B feeder register with the operand selected by bit 3.

Bit 3 (Enables Adder Output To B)

When 1, this bit enables the output of this adder's sum register to the B feeder register. When 0, this bit enables operand bus 1 to the B feeder register.

Bits 2 and 3 (Enhanced-FFT MAPs Only)

These bits form a 2-bit code which controls the B feeder register as follows:

Bit 3	 2	Operation
0	0	Do nothing.
0	1	Load B feeder register with contents of operand bus 1.
1	0	Clear B feeder register.
1	1	Load B feeder register with output of this adder's sum register.

Bit 4 (Subtract)

When 1, this bit enables a subtract operation. When 0, this bit enables an add operation.

† For enhanced-FFT MAPs, these bits are defined as follows:
00 Do nothing
01 Operand bus 1→B
10 Clear B
11 Sum register→B

Bit 5 (Load Sum Register)

This bit loads the sum register with the result generated during the previous microinstruction time.

Bit 6 (Select Magnitude Mode)

This bit causes the add/subtract unit to compare the absolute magnitudes of the operands in the A and B feeder registers. The result of the compare is returned to the control unit for sample by the 4_8 jump control microinstruction.

Bits 7 through 13 (Add/Subtract Unit 2 Field)

Bits 7 through 13 control add/subtract unit 2 exactly as bits 0 through 6 control add/subtract unit 1.

Bits 14 through 17 (Multiply Unit 1 Field)

Bit 14 (Load B From Bus 1)

This bit loads the B feeder register with the operand from operand bus 1 and then initiates the multiply operation.

Bit 15 (Clock A)

This bit loads the A feeder register with the operand selected by bit 16.

Bit 16 (Enable Bus 2 To A)

When 1, this bit enables operand bus 2 to the A feeder register. When 0, this bit enables the A pass-on input (from the previous multiply unit) to the A feeder register. The A pass-on input is zero for multiply unit 1.

Bit 17 (Load Product Register)

This bit loads the product register with the result generated during the previous microinstruction time.

Bits 18 through 21 (Multiply Unit 2 Field)

Bits 18 through 21 control multiply unit 2 exactly as bits 14 through 17 control multiply unit 1.

Bits 22 through 25 (Divide Unit 1 Field)

Bit 22 (Load A From Bus 2)

This bit loads the A feeder register with the operand from operand bus 2.

Bit 23 (Load B From Bus 1)

This bit loads the B feeder register with the operand from operand bus 1.

Bit 24 (Start Divide)

This bit starts the divide operation. Operand A is divided by operand B.

The quotient register can be loaded five microinstruction times after the divide operation is started.

Bit 25 (Load Quotient Register)

This bit loads the quotient register with the result of the divide operation.

Bits 26 through 29 (Square Root Unit 1 Field)

Bit 26 (Load A From Bus 2 and Go)

This bit loads the operand from operand bus 2 into the A feeder register and then initiates the square root operation. The root register can be loaded after four or eight microinstruction times, depending upon whether bit 27 selects a 12-bit or 24-bit result, respectively.

Bit 27 (Select 24-Bit Result)

When 1, this bit instructs the square root unit to generate a root with 24 significant bits in the coefficient. When 0, bit 27 instructs the square root unit to generate a root with 12 significant bits in the coefficient.

Bit 28 (Unused)

Bit 29 (Load Root Register)

This bit loads the root register with the result of the square root operation.

Bits 30 and 31 (Bus 1 Source Select)

These bits form a 2-bit code which selects a data source for operand bus 1 as shown in figure 2-26.

Bits 32 and 33 (Bus 2 Source Select)

These bits form a 2-bit code which selects a data source for operand bus 2 as shown in figure 2-26.

Bit 34 (Load Result Bus Catching Register)

This bit loads the quantity selected by bits 35 through 38 into the result bus catching register.

Bits 35 through 38 (Result Bus Source Select)

These bits form a 4-bit code which selects a data source for the result bus as shown in figure 2-26. Codes 0 and 15 force one-fill on the result bus. Code 14 forces zero-fill on the result bus.

Bits 39 through 45 (Add/Subtract Unit 3 Field)

Bits 39 through 45 control add/subtract unit 3 exactly as bits 0 through 6 control add/subtract unit 1.

Bits 46 through 52 (Add/Subtract Unit 4 Field)

Bits 46 through 52 control add/subtract unit 4 exactly as bits 0 through 6 control add/subtract unit 1.

Bits 53 through 56 (Multiply Unit 3 Field)

Bits 53 through 56 control multiply unit 3 exactly as bits 14 through 17 control multiply unit 1.

Bits 57 through 60 (Multiply Unit 4 Field)

Bits 57 through 60 control multiply unit 4 exactly as bits 14 through 17 control multiply unit 1.

Bits 61 through 64 (Divide Unit 2 Field)

Bits 61 through 64 control divide unit 2 exactly as bits 22 through 25 control divide unit 1.

Bits 65 through 68 (Square Root Unit 2 Field)

Bits 65 through 68 control square root unit 2 exactly as bits 26 through 29 control square root unit 1.

Bit 69 (Divide Maintenance Mode)

This bit replaces the divide clock in both divide units with the main MAP clock. This allows a diagnostic routine to examine the quotient register seven microinstruction times after the start of the divide operation.

Bits 70 and 71 (Unused)

INTRODUCTION

This section describes MAP architecture, the MAP internal data format, PPU channel/6640 interface signals, and major functional components within MAP.

Some of the functional component descriptions use abridged microinstruction sequences to show typical operations of a component. Mnemonics appearing in these microinstruction sequences are defined in the 6000 MAP III Assembler ERS listed in the preface. References to diagrams use the sheet numbering system described under Key to Symbols in section 6.

MAP ARCHITECTURE

Processing flexibility and processing speed are the two major objectives which influenced MAP design. The microprogrammability and macro string storage capability of MAP permit flexibility in arithmetic algorithms and data format conversion. The following architectural elements are included in MAP to enhance processing speed.

- MAP's triple access 3-section data storage permits three storage accesses to proceed simultaneously. For many types of algorithms, this permits two operands to be read and a result stored during the same microinstruction time.
- Multiple arithmetic units permit arithmetic operations to proceed in a parallel and/or overlapping manner.
- High speed multiply and add/subtract units provide results in a single microinstruction time. This feature concentrates processing speed in the arithmetic units used most often by typical arithmetic algorithms. In addition, a nonbus data path connects multiply unit outputs with add/subtract unit inputs to facilitate sum of product calculations.
- The optional ECS interface supports I/O with the host computer at a rate much higher than a PPU data channel. This option reduces the ratio of I/O time to arithmetic processing time.

The emitter coupled logic (ECL) and large circuit boards used in MAP provide the short propagation delay times and high packaging density necessary for the most effective implementation of MAP architecture.

INTERNAL DATA FORMAT

Figure 3-1 shows the MAP internal data format.

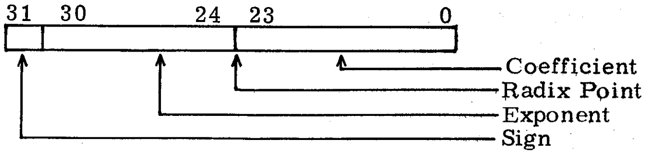


Figure 3-1. MAP Internal Data Format

The format consists of a 24-bit normalized coefficient, a 7-bit, biased, twos complement exponent, and a sign bit using sign and magnitude representation. The exponent represents powers of 2.

The range for positive normalized internal floating-point MAP numbers is from 2^{-100_8} to $0.77777777_8 \cdot 2^{77_8}$. The smallest and largest decimal numbers (seven significant digits) in this range are $0.5421011 \cdot 10^{-19}$ and $0.9223371 \cdot 10^{19}$. All positive fixed-point numbers not exceeding 2^{24} (16,777,216) can be represented exactly in internal format.

Figure 3-2 shows the MAP number spectrum in octal form.

Exponent Power	Floating-Point Number	Value
	OVERFLOW	
2^{77_8}	077.4	XXXXXX
↕	↕	↕
2^3	003.4	= 4
2^2	002.4	= 2
2^1	001.4	= 1
2^0	000.4	= 1/2
2^{-1}	177.4	= 1/4
2^{-2}	176.4	= 1/8
2^{-3}	175.4	= 1/16
2^{-4}	174.4	= 1/32
↕	↕	↕
2^{-77_8}	101.4	= XXXXXX
	UNDERFLOW	

zero = 000.0000000
100.XXX is indeterminate

Figure 3-2. MAP Number Spectrum (Octal)

INTERFACE SIGNALS

PPU CHANNEL INTERFACE SIGNALS

These signals connect MAP with each of two PPU channels. All signals on this interface are 25-nanosecond pulses which are synchronized with 1-MHz and 10-MHz clock signals also present on the interface.

Signals From PPU Only

1 MHz Clock/ 10 MHz Clock
MAP uses these signals to synchronize all communication with a PPU.

Function
This signal identifies the accompanying 12 data signals as a function code.

Master Clear
The PPU sends this signal to clear MAP just after a computer system deadstart.

Bidirectional PPU Channel Interface Signals

Data
These 12 signals carry data, function, and status words between the PPU and MAP.

Full
This signal indicates to the receiver that the word formed by the 12 data signals can be accepted.

Empty
This signal indicates to the transmitter that the receiver has accepted the word accompanying the last full signal.

Active
The transmitter sends this signal to initiate or resume communication with the receiver. An active signal accompanies each function signal.

Inactive
MAP replies to a function signal (and its associated function code) with an inactive signal. An inactive signal may also be used by either the PPU or MAP to terminate communication.

CONTROLLER INTERFACE SIGNALS

These signals connect MAP with an ECS controller. All signals on this interface are 25-nanosecond pulses which are synchronized with the 10-MHz clock signal also present on the interface.

Signals From MAP

ECS Address
These 24 signals accompany each request to provide the ECS start address to the controller.

ECS Write Data
These 60 signals transfer 60-bit words from MAP to the controller at a maximum rate of 10 MHz.

Store Bit
This signal accompanies a request signal to initiate a write ECS operation. In the absence of a store bit signal, the controller performs an ECS read operation upon receipt of a request signal.

Request
This signal initiates the ECS operation selected by the store bit signal. Each request transfers a full or partial 480-bit ECS record.

Write Go
This signal accompanies the first 60-bit word sent from MAP to the controller.

Signals From Controller

10 MHz Clock
MAP uses this signal to synchronize all communication with the controller. This signal should be adjusted at the controller so that MAP requests arrive at the controller at t35.

ECS Read Data
These 60 signals transfer 60-bit words from the controller to MAP at a maximum rate of 10 MHz.

Read Go
This signal accompanies the first 60-bit word sent from the controller to MAP.

Abort
This signal occurs about 200 nanoseconds after an accept signal when the controller is unable to perform the requested ECS operation.

Parity Error
This signal indicates that a parity error was detected in the current 480-bit ECS record.

COMPONENT THEORY

Section 1 of this manual provides a brief functional description of each MAP component. The purpose of the component theory in this section is to provide an expanded functional description of these components. Refer to appropriate diagrams and associated backup text in section 6 for details relating to the hardware implementation of MAP components.

ECS INTERFACE

This optional interface transfers 60-bit words between an ECS controller and the A/D unit. The ECS interface contains an ECS address counter, an ECS field length register, transmitters/receivers, a first-in first-out (FIFO) buffer that can store up to six 480-bit ECS records, and ECS control logic. The FIFO buffer and data receivers/transmitters are located on the ECS2 paks at LB26 and LB27. The rest of the ECS logic is located on the ECS1 pak at LB28.

ECS Address Counter

Before any ECS operation, controlware uses inter-register (ROP) microinstructions to load the ECS address counter with the ECS start address. The counter increments as each 60-bit word is received from or transmitted to the controller.

The 24-bit output of the ECS address counter feeds the ECS address transmitters. MAP sends the ECS address to the controller with each ECS request.

ECS Field Length Register

Before any ECS operation, controlware uses ROP microinstructions to load the 18-bit ECS field length register with the upper 18 bits of the highest legal ECS address for the current operation. MAP compares the current ECS address with the field length register value before each ECS request. When the current address exceeds the field length address, MAP generates ECS field length error status (word 7, bit 11) but does not stop microinstruction execution. However, for the rest of the transfer, MAP either receives zero-fill (read) or transmits nothing (write).

Transmitters/Receivers

The same type of transmitters/receivers are used in the ECS interface and in the PPU channel interface. Transmitters use a gated driver coupled to a pulse transformer to convert MAP logic levels to the sine waveform required for the CDC 6000/CYBER coaxial transmission scheme. Likewise, receivers use a pulse transformer coupled to a differential receiver and gated flip-flop to convert the coaxial waveform to MAP logic levels.

First In First Out (FIFO) Buffer

This buffer consists of three 16-word by 60-bit circular files which buffer data between the A/D unit and the controller.

FIFO buffer control logic contains two address counters. The input counter starts at 0 and increments each time a 60-bit word from the A/D unit or controller is written in the buffer. The output counter starts at 0 and increments each time a 60-bit word from the buffer is read from the buffer to the controller or the A/D unit. FIFO buffer control logic also contains full/empty logic to prevent buffer words from being overwritten before they are read.

The FIFO buffer can exchange 60-bit words with the controller at the ECS transfer rate of 10 MHz. The maximum transfer rate between the A/D unit and the buffer is one 60-bit word per microinstruction time.

ECS Control Logic

This logic controls ECS read or write operations according to the interface request bit (bit 0) the interface select field (bits 10 through 18) and the data flow bit (bit 22) of control processor data transfer (PLOAD, PDUMP) and I/O data transfer (DLOAD, DDUMP) microinstructions. Section 5 contains detailed timing charts for typical ECS operations. The following sequences provide a microinstruction oriented overview of typical ECS operations. Note that one microinstruction moves one 60-bit word between the FIFO buffer and R4/R5 or R6/R7. Transfers between the FIFO buffer and the controller are started and stopped by microinstruction, but are asynchronous to microinstruction execution.

Load From ECS Sequence

This sequence shows microinstruction activity associated with transfers from ECS to R3. Each group of eight 60-bit words from the FIFO buffer is unpacked into 15 32-bit words. DLOADE microinstructions are used because the transfer is ultimately to data storage. For transfers to control unit destinations, PLOADE microinstructions would be used. Figure 3-3 shows the contents of R4/R5 during this sequence. Assume that K file register 1 contains the number of 32-bit words to be transferred.

Step	Microinstruction	Description
1.	UJPR + 2	Jump to step 3.
2.	UJPD LEEXIT	The word count is satisfied. Jump to routine LEEXIT to terminate the transfer.
3.	DLOADE, 0, NW, SO, ZR1, -1	Initiate or maintain an ECS read operation, transfer R4 to R5 (the FIFO buffer automatically fills R4), transfer 32 bits from R5 to R3 with no (2 x 0) shift, skip to step 2 if the word count is zero, and decrement the word count.
4.	DLOADE, 16, SO, NCI, ZR1, -2	Transfer 32 bits from R3 to NCI, transfer 32 bits from R5/R4 to R3 with a 32-bit (2 x 16) right-shift, skip to step 2 if the word count is zero, and decrement the word count.

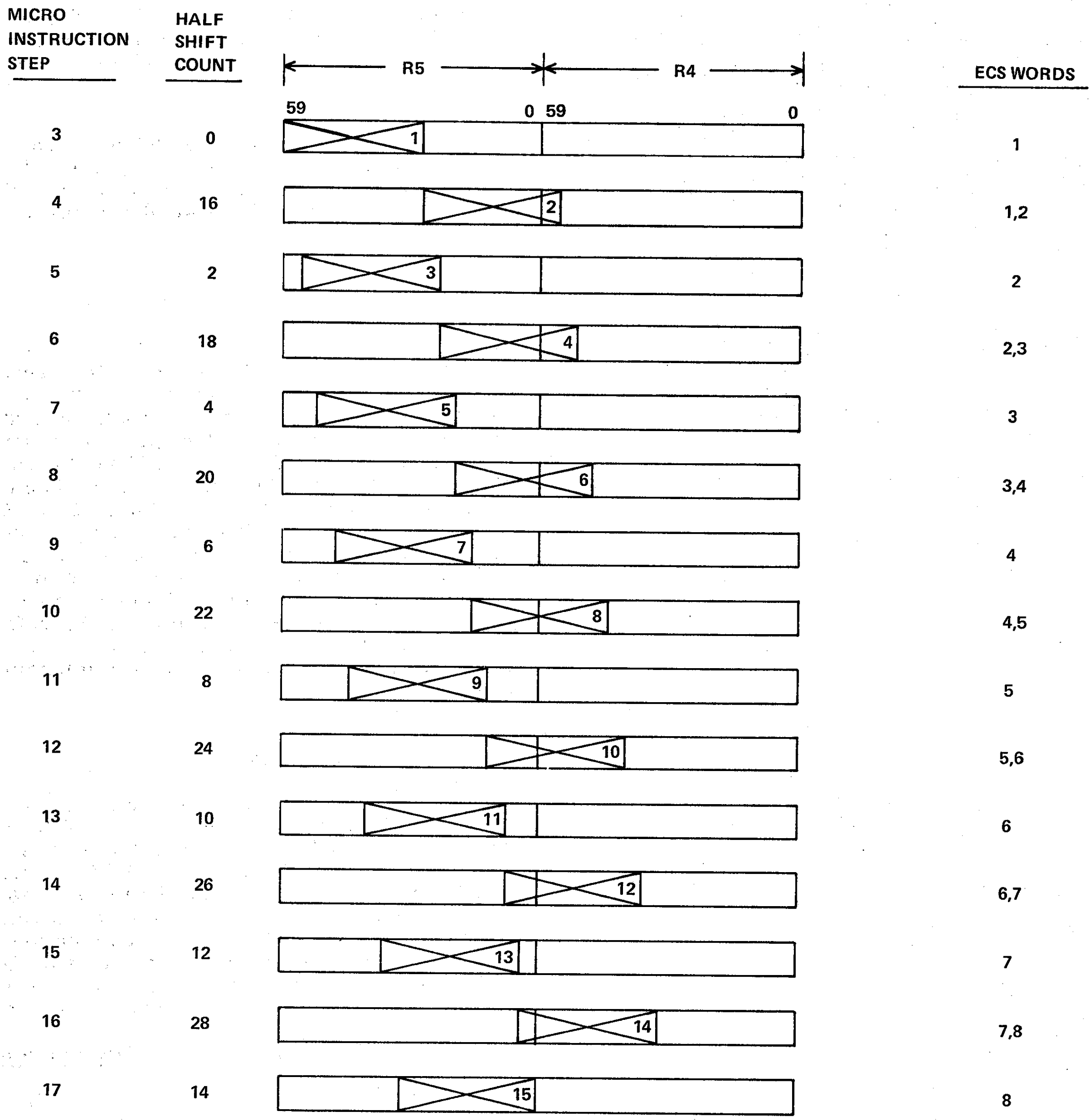


Figure 3-3. R4, R5 Contents During Unpacking of Eight 60-Bit Words Into 15 32-Bit Words

NOTE

Additional microinstruction mnemonics are required to transfer data from the numerical conversion unit to the desired destination, but these mnemonics are not shown in this sequence.

Step	Microinstruction	Description
5.	DLOADE, 2, NW, SO, NCI, ZR1, -3	Transfer 32 bits from R3 to NCI, transfer R4 to R5, transfer 32 bits from R5 to R3 with a 4-bit (2 x 2) right-shift, skip to step 2 if the word count is zero, and decrement the word count.

Step	Microinstruction	Description
6.	DLOADE 18, SO, NCI, ZR1, -4	Transfer 32 bits from R3 to NCI, transfer 32 bits from R5/R4 to R3 with a 36-bit (2 x 18) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
7.	DLOADE 4, NW, SO, NCI, ZR1, -5	Transfer 32 bits from R3 to NCI, transfer R4 to R5, transfer 32 bits from R5 to R3 with an 8-bit (2 x 4) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
8.	DLOADE 20, SO, NCI, ZR1, -6	Transfer 32 bits from R3 to NCI, transfer 32 bits from R5/R4 to R3 with a 40-bit (2 x 20) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
9.	DLOADE 6, NW, SO, NCI, ZR1, -7	Transfer 32 bits from R3 to NCI, transfer R4 to R5, transfer 32 bits from R5 to R3 with a 12-bit (2 x 6) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
10.	DLOADE 22, SO, NCI, ZR1, -8	Transfer 32 bits from R3 to NCI, transfer 32 bits from R5/R4 to R3 with a 44-bit (2 x 22) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
11.	DLOADE 8, NW, SO, NCI, ZR1, -9	Transfer 32 bits from R3 to NCI, transfer R4 to R5, transfer 32 bits from R5 to R3 with a 16-bit (2 x 8) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
12.	DLOADE 24, SO, NCI, ZR1, -10	Transfer 32 bits from R3 to NCI, transfer 32 bits from R5/R4 to R3 with a 48-bit (2 x 24) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
13.	DLOADE 10, NW, SO, NCI, ZR1, -11	Transfer 32 bits from R3 to NCI, transfer R4 to R5, transfer 32 bits from R5 to R3 with a 20-bit (2 x 10) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
14.	DLOADE 26, SO, NCI, ZR1, -12	Transfer 32 bits from R3 to NCI, transfer 32 bits from R5/R4 to R3 with a 52-bit (2 x 26) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
15.	DLOADE 12, NW, SO, NCI, ZR1, -13	Transfer 32 bits from R3 to NCI, transfer R4 to R5, transfer 32 bits from R5 to R3 with a 24-bit (2 x 12) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
16.	DLOADE 28, SO, NCI, ZR1, -14	Transfer 32 bits from R3 to NCI, transfer 32 bits from R5/R4 to R3 with a 56-bit (2 x 28) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
17.	DLOADE 14, NW, SO, NCI, ZR1, -15	Transfer 32 bits from R3 to NCI, transfer R4 to R5, transfer 32 bits from R5 to R3 with a 28-bit (2 x 14) right-shift, skip to step 2 if the word count is zero, and decrement the word count.
18.	UJPD LELOOP	Jump to step 3 (address LELOOP) to process the next 15 32-bit words of the transfer.

NOTE

Since it is possible for the controller to supply 60-bit words faster than microinstructions can process them, it is possible for the FIFO buffer to contain up to 48 words when this sequence terminates.

For this reason, the user must increase the field length of ECS read arrays by at least 48 to prevent field length errors.

Unload to ECS Sequence

This sequence shows microinstruction activity associated with transfers from R3 to ECS. Each group of 15 32-bit MAP words is packed into eight 60-bit words. DDUMPE microinstructions are used because the transfer is from data storage. For transfers from control unit sources, PDUMPE microinstructions would be used. Figure 3-4 shows the contents of R6/R7 during this sequence. Assume that K file register 1 contains the number of 32-bit words to be transferred minus 1.

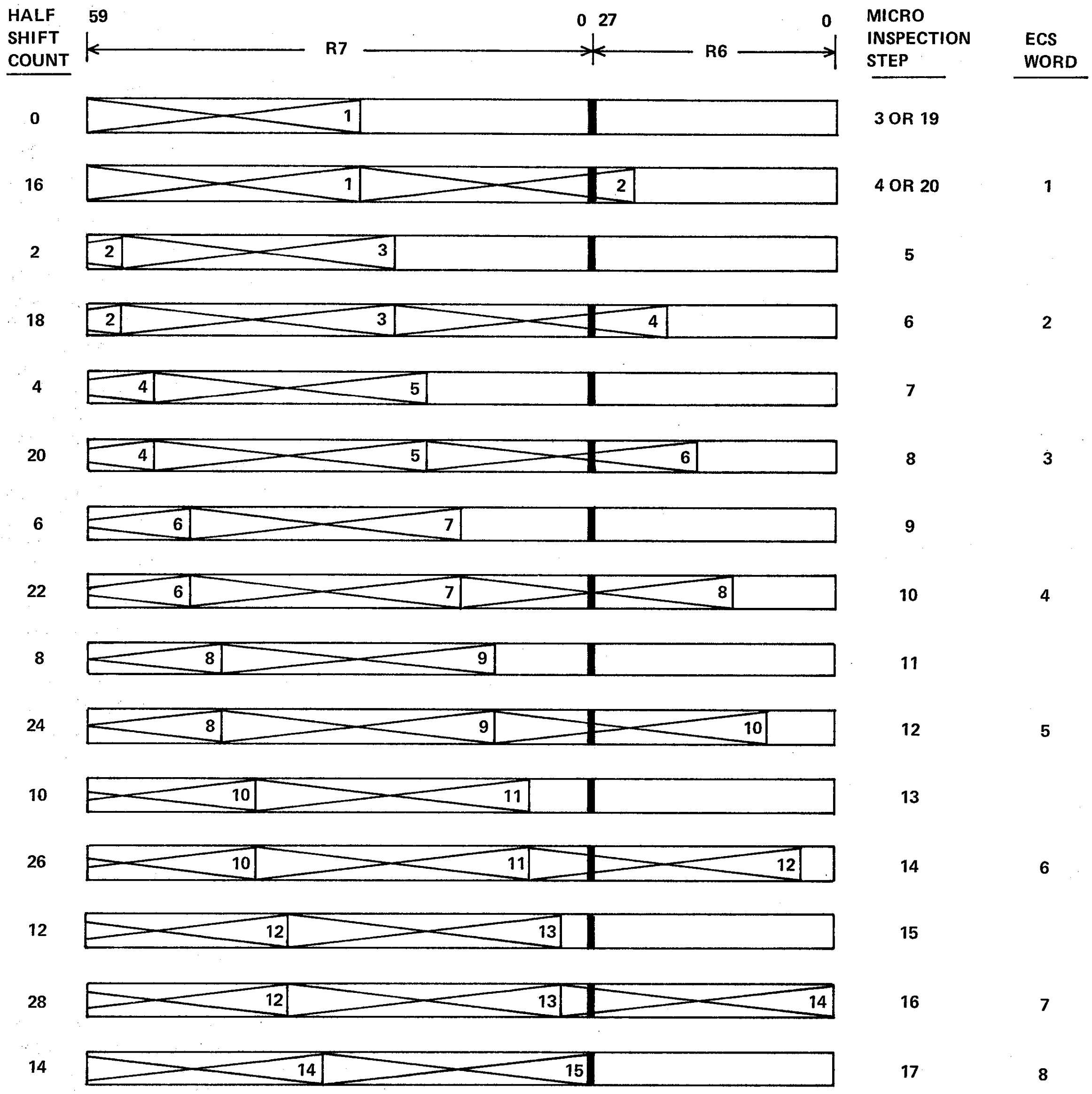


Figure 3-4. R6/R7 Contents During Packing of 15 32-Bit Words Into Eight 60-Bit Words

Step	Microinstruction	Description
1.	UJPR +2	Jump to step 3.
2.	UJPD DEEXIT	The word count is satisfied. Jump to step 22 (address DEEXIT) to terminate the transfer.

NOTE

Additional microinstruction mnemonics are required to transfer data into the numerical conversion unit, but these mnemonics are not shown in this sequence.

Step	Microinstruction	Description
3.	DDUMPE 0, NCO, SI, SO, ZR1, -1	Each of these microinstructions transfers 32 bits from NCO to R3, transfers 32 bits from R3 to R7/R6, skips to step 2 if the word count is zero, and decrements the word count. Each microinstruction with an NW mnemonic transfers 60 bits from R7 to the FIFO buffer. The half shift count in each microinstruction determines the placement of each 32-bit word in R7/R6.
4.	DDUMPE 16, NCO, SI, SO, ZR1, -2	
5.	DDUMPE 2, NCO, SI, SO, NW, ZR1,-3	
6.	DDUMPE 18, NCO, SI, SO, ZR1, -4	
7.	DDUMPE 4, NCO, SI, SO, NW, ZR1, -5	
8.	DDUMPE 20, NCO, SI, SO, ZR1, -6	
9.	DDUMPE 6, NCO, SI, SO, NW ZR1, -7	
10.	DDUMPE 22, NCO, SI, SO, ZR1, -8	
11.	DDUMPE 8, NCO, SI, SO, NW, ZR1, -9	
12.	DDUMPE 24, NCO, SI, SO, ZR1, -10	
13.	DDUMPE 10, NCO, SI, SO, NW, ZR1, -11	
14.	DDUMPE 26, NCO, SI, SO, ZR1, -12	
15.	DDUMPE 12, NCO, SI, SO, NW, ZR1, -13	
16.	DDUMPE 28, NCO, SI, SO, ZR1, -14	
17.	DDUMPE 14, NCO, SI, SO, NW, ZR1, -15	

Step	Microinstruction	Description
18.	DDUMPE 0, ZR1, +4	Skip to step 22 if the word count is zero, (word count was a multiple of 15), and decrement the word count.
19.	DDUMPE 0, NCO, SI, SO, NW	These microinstructions replace steps 3 and 4 in the main loop.
20.	DDUMPE 16, NCO, SI, SO, ZR1, +2	
21.	UJPD UELOOP	Jump to step 5 (address UELOOP) to process the next 15 32-bit words of the transfer.
22.	DDUMPE 0, NW, EOT	Send the last 60 bits of the transfer from R7 to the FIFO buffer and terminate the transfer.

NOTE

Since it is possible for the controller to accept 60-bit words faster than microinstructions can supply them, it is possible for the FIFO buffer to be empty when the microinstruction containing the end of transfer code occurs. In this case, MAP sends eight zero-filled 60-bit words to the controller. Thus, the user must increase the field length of ECS write arrays by at least eight to prevent field length errors.

PPU CHANNEL INTERFACE

This interface transfers 12-bit words between PPU channels and the A/D unit, processes hardware functions, and provides a data path between the maintenance panel cassette transport and the A/D unit. The computer side of the PPU channel interface contains a dual access switch which allows either of two PPU channels to reserve the interface on a one-at-a-time basis.

The PPU channel interface consists of control logic on the CHN1 pak at LB30 and data transfer logic on the CHN2 pak at LB29.

Section 2 describes MAP functions and PPU channel data transfers from a PPU programming standpoint.

PPU Channel Control

This logic controls the dual access switch and directs the transfer of 12-bit words between MAP and a PPU.

Dual Access Switch

The dual access switch establishes a data path between one of two PPU channels and the A/D unit. When the PPU channel interface is unreserved, the first PPU channel to issue a function gains access to the data path. The path remains reserved to the PPU channel until the channel issues a release channel function or until the opposite channel issues a release opposite channel function.

Load From PPU Channel Sequence

This sequence shows microinstruction activity associated with transfers from a PPU channel to R3. Each group of eight 12-bit PPU words is packed into three 32-bit data storage words. DLOADC microinstructions are used because the transfer is ultimately to data storage. For transfers to control unit destinations, PLOADC microinstructions would be used. Figure 3-5 shows the data paths used between registers R1, R2, and R3 during PPU channel load and dump operations. Assume that K file register 1 contains the number of 32-bit words to be transferred minus 1.

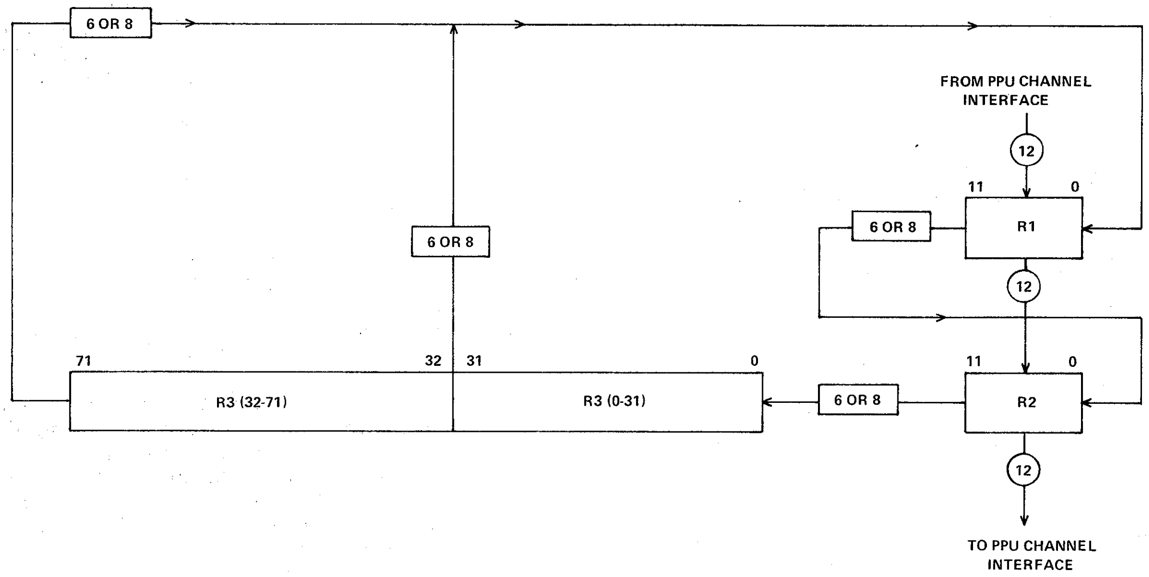


Figure 3-5. R1/R2/R3 Data Paths

Step	Microinstruction	Description
1.	DLOADC NW, TR, SI	Load a 12-bit word from the PPU channel to R1 and do a 12-bit transfer from R1 to R2.
2.	DLOADC NW, SI, SO, B8	Load a 12-bit word from the PPU channel to R1 and shift the first 8 bits of a 32-bit word from R1/R2 to R3.
3.	DLOADC SI, SO, B8	Shift the second 8 bits to R3.
4.	DLOADC SI, SO, B8	Shift the third 8 bits to R3.
5.	DLOADC NW, TR SI, NZ1, +2	Load a 12-bit word from the PPU channel to R1, do a 12-bit transfer from R1 to R2, skip to step 7 if the word count is nonzero, and decrement the word count.
6.	UJPD LASTWD	Jump to step 22 (address LASTWD) to process the last word.

Step	Microinstruction	Description
7.	DLOADC NW, SI, SO, B8	Load a 12-bit word from the PPU channel to R1 and shift the fourth 8 bits from R1/R2 to R3.
8.	DLOADC NCI, SI, SO, B8	Transfer R3 to NCI and shift the first 8 bits of the next 32-bit word from R1/R2 to R3.

NOTE

Additional microinstruction mnemonics are required to transfer data from the numerical conversion unit to the desired destination, but these mnemonics are not shown in this sequence.

Step	Microinstruction	Description
9.	DLOADC SI, SO, B8	Shift the second 8 bits to R3.
10.	DLOADC NW, TR, SI	Load a 12-bit word from the PPU channel to R1 and then do a 12-bit transfer from R1 to R2.
11.	DLOADC NW, SI, SO, B8, NZ1, +2	Load a 12-bit word from the PPU channel to R1, shift the third 8 bits to R3, skip to step 13 if the word count is nonzero, and decrement the word count.
12.	UJPD LASTWD	Jump to step 22 (address LASTWD) to process the last word.
13.	DLOADC SI, SO, B8	Shift the fourth 8 bits to R3.
14.	DLOADC NCI, SI, SO, B8	Transfer R3 to NCI and shift the first 8 bits of the next 32-bit word from R1/R2 to R3.
15.	DLOADC NW, TR SI	Load a 12-bit word from the PPU channel to R1 and do a 12-bit transfer from R1 to R2.
16.	DLOADC NW, SI, SO, B8	Load a 12-bit word from the PPU channel to R1 and shift the second 8 bits to R3.
17.	DLOADC SI, SO, B8, NZ1, +2	Shift the third 8 bits to R3, skip to step 19 if the word count is nonzero, and decrement the word count.
18.	UJPD LASTWD	Jump to step 22 (address LASTWD) to process the last word.
19.	DLOADC SI, SO, B8	Shift the fourth 8 bits to R3.
20.	DLOADC NCI	Transfer R3 to NCI
21.	UJPD LCLOOP	Jump to step 1 (address LCLOOP) to process the remaining words in the transfer.
22.	DLOADC SI, SO, B8	Shift the fourth 8 bits of the last 32-bit word to R3.
23.	DLOADC NCI, EOT	Transfer R3 to NCI and send an end of transfer signal to the PPU channel interface to initiate checkword processing.

Unload to PPU Channel Sequence

This sequence shows microinstruction activity associated with transfers from R3 to a PPU channel. Each group of three 32-bit data storage words is unpacked into eight 12-bit PPU words. DDUMPC microinstructions are used because the transfer is from data storage. For transfers from control unit sources, PDUMPC microinstructions would be used. Figure 3-5 shows the data paths used in this sequence. Assume that K contains the number of 32-bit words to be transferred minus 1.

NOTE

Additional microinstruction mnemonics are required to fetch data and enable it to R3, but these mnemonics are not shown in this sequence.

Step	Microinstruction	Description
1.	DDUMPC LR3, SI	Load a 32-bit word into R3.
2.	DDUMPC SI, SO, B8	Shift the first 8 bits of a 32-bit word from R3 to R1/R2.
3.	DDUMPC SI, SO, B8	Shift the second 8 bits of the 32-bit word from R3 to R1/R2.
4.	DDUMPC SI, SO, B8	Shift the third 8 bits of the 32-bit word from R3 to R1/R2.
5.	DDUMPC NW, TR, SI, SO, B8	Send a 12-bit word from R2 to the PPU channel, do a 12-bit transfer from R1 to R2, and shift the fourth 8 bits of the 32-bit word from R3 to R1/R2.

Step	Microinstruction	Description
6.	DDUMPC NW, LR3, SI	Send a 12-bit word from R2 to the PPU channel and load a 32-bit word into R3.
7.	DDUMPC SI, SO, B8	Shift the first 8 bits of the 32-bit word from R3 to R1/R2.
8.	DDUMPC SI, SO, B8	Shift the second 8 bits from R3 to R1/R2.
9.	DDUMPC NW, TR, SI, SO, B8, NZ1, +2	Send a 12-bit word from R2 to the PPU channel, shift the third 8 bits from R3 to R1/R2, skip to step 11 if the word count is nonzero, and decrement the word count.
10.	UJPD UEOT	Jump to step 20 (address UEOT) to terminate the transfer.
11.	DDUMPC NW, SI, SO, B8	Send a 12-bit word from R2 to the PPU channel and shift the fourth 8 bits from R3 to R1/R2.
12.	DDUMPC LR3, SI	Load a 32-bit word into R3.
13.	DDUMPC SI, SO, B8	Shift the first 8 bits of the 32-bit word from R3 to R1/R2.
14.	DDUMPC NW, TR SI, SO, B8	Send a 12-bit word from R2 to the PPU channel, do a 12-bit transfer from R1 to R2, and shift the second 8 bits from R3 to R1/R2.
15.	DDUMPC NW, SI, SO, B8, NZ1, +2	Send a 12-bit word from R2 to the PPU channel, shift the third 8 bits from R3 to R1/R2, skip to step 17 if the word count is nonzero, and decrement the word count.
16.	UJPD UEOT	Jump to step 20 (address UEOT) to terminate the transfer.
17.	DDUMPC SI, SO, B8	Shift the fourth 8 bits from R3 to R1/R2.
18.	DDUMPC NW, TR, LR3, SI, SO	Send a 12-bit word from R2 to the PPU channel, do a 12-bit transfer from R1 to R2, and load a 32-bit word into R3.
19.	DDUMPC NW, SI, SO, B8, NZ1, -16	Send a 12-bit word from R2 to the PPU channel, shift the first 8 bits of the next 32-bit word from R3 to R1/R2, skip to step 3 if the word count is nonzero, and decrement the word count.
20.	DDUMPC EOT	Send the checkword to the PPU channel.

Hardware Function Processing

All hardware functions except the select status words (7020_8) function merely generate a single pulse or set an enabling flip-flop. This activity is shown on diagram sheet CHN1 3.1.

The select status words function proceeds as follows:

1. The PPU issues a 7020_8 function over a PPU channel connected to MAP.
2. MAP receives the function on CHN2 3.1 (access A) or CHN2 3.2 (access B) and loads the 12-bit function into the appropriate holding register. The appropriate select status latch on CHN1 3.1 sets.
3. Logic on CHN1 3.1 and CHN1 3.2 replies to the function by sending an inactive signal to the PPU channel.
4. When it detects the inactive PPU channel, the PPU activates the channel and enters an input instruction sequence for the impending status transfer.
5. Logic on CHN1 3.3 responds to the active and empty channel by enabling the status word counter and triggering status logic on CT9 3.4 and CT8 3.6.
6. The status word counter feeds A feeder bus select logic on CT8 3.6. This logic enables each 12-bit status word to the A feeder bus at the appropriate time. The multiplexers and transmitters on CHN2 3.6 and CHN2 3.7 transfer the status words from the A feeder bus to channel A or channel B, respectively. Data sources for each of the status words are as follows:

Status Word	Bits	Sheet
0	0	CT9 3.5
0	1, 4	CT9 3.6
0	2, 3	CHN1 3.1
0	5, 6, 7, 9, 10	CT1 3.5
0	8	CHN1 3.3
0	11	CHN1 3.4
1	0-11	CT4 3.6
2	0-11	CT4 3.5
3	0-11	CT5 3.8
4	0-11	CT1 3.5
5	0-11	
6	0-11	
7	0-11	

7. The logic on CHN1 3.3 sends a full to the PPU channel with each status word (no checkword is sent). When the appropriate enable inactive flip-flop on CHN1 3.1 is set, the logic on CHN1 3.2 sends an inactive signal to the PPU channel after the PPU has accepted the eighth word. Otherwise, the PPU disconnects the channel to terminate the status operation.

Controlware Function Processing

All controlware functions cause MAP to begin microinstruction execution at the ROM address determined by bits 9, 10, and 11 of the function code. The following sequence applies to all controlware functions.

1. The PPU issues a controlware function over a PPU channel connected to MAP. The logic on CHN2 3.1 and CHN2 3.2 interprets all nonhardware functions as controlware functions.
2. MAP receives the function on CHN2 3.1 (access A) or CHN2 3.2 (access B) and loads the function into the appropriate holding register.
3. When a previously issued controlware function is not currently being executed, logic on CHN1 3.1 transfers the function from the holding register through the mux on CHN2.3 to the function register. If a previously issued controlware function is being executed, the function register is not loaded until execution completes.
4. Logic on CHN1 3.1 and CHN1 3.2 replies to the function by sending an inactive signal to the PPU channel.
5. The function busy flip-flop on CHN1 3.1 feeds run control logic on CT9 3.6 to generate an initial start pulse. This pulse loads bits 9, 10, and 11 of the function register into bits 0, 1, and 2 of CPR and initiates microinstruction execution at CPR. Bits 3 through 5 of CPR are zero-filled, and bits 6 through 11 are one-filled.
6. The rest of the microinstruction operation is controlled by the selected ROM routine.

ASSEMBLY/DISASSEMBLY UNIT

Figure 3-6 shows the major components of and paths through the A/D unit. Table 3-1 summarizes the functions of registers R1 through R7. Previous sequences in this section show the use of the indicated registers during typical I/O operations as follows:

Sequence	Registers Used
Load from PPU channel sequence	R1, R2, R3
Unload to PPU channel sequence	R1, R2, R3
Load from ECS sequence	R3, R4, R5
Unload to ECS sequence	R3, R6, R7

A/D Mask Words

A/D mask words are two 18-bit patterns which control the loading of R3. Figure 3-7 shows the mask word format.

TABLE 3-1. ASSEMBLY/DISASSEMBLY UNIT REGISTER FUNCTIONS

Register	Width	Operations Used For	Function
R1	12 bits	Load from PPU, unload to PPU, cassette load/dump	Accept data from PPU channel, R3, or cassette. Shift or transfer data to R2.
R2	12 bits	Load from PPU, unload to PPU, cassette load/dump	Accept data from R1. Shift data to R3 or transfer data to PPU channel or cassette.
R3	72 bits	All A/D operations	Accept data from R2, R4/R5, or A/D bus. Shift data to R1. Transfer data to R6/R7 or A/D bus.
R4	60 bits	Load from ECS	Accept data from FIFO buffer. Transfer data to R5 and R3.
R5	60 bits	Load from ECS	Accept data from R5. Transfer data to R3.
R6	28 bits	Unload to ECS	Accept data from R3. Shift data to R7.
R7	60 bits	Unload to ECS	Accept data from R3 and R6. Transfer data to FIFO buffer.

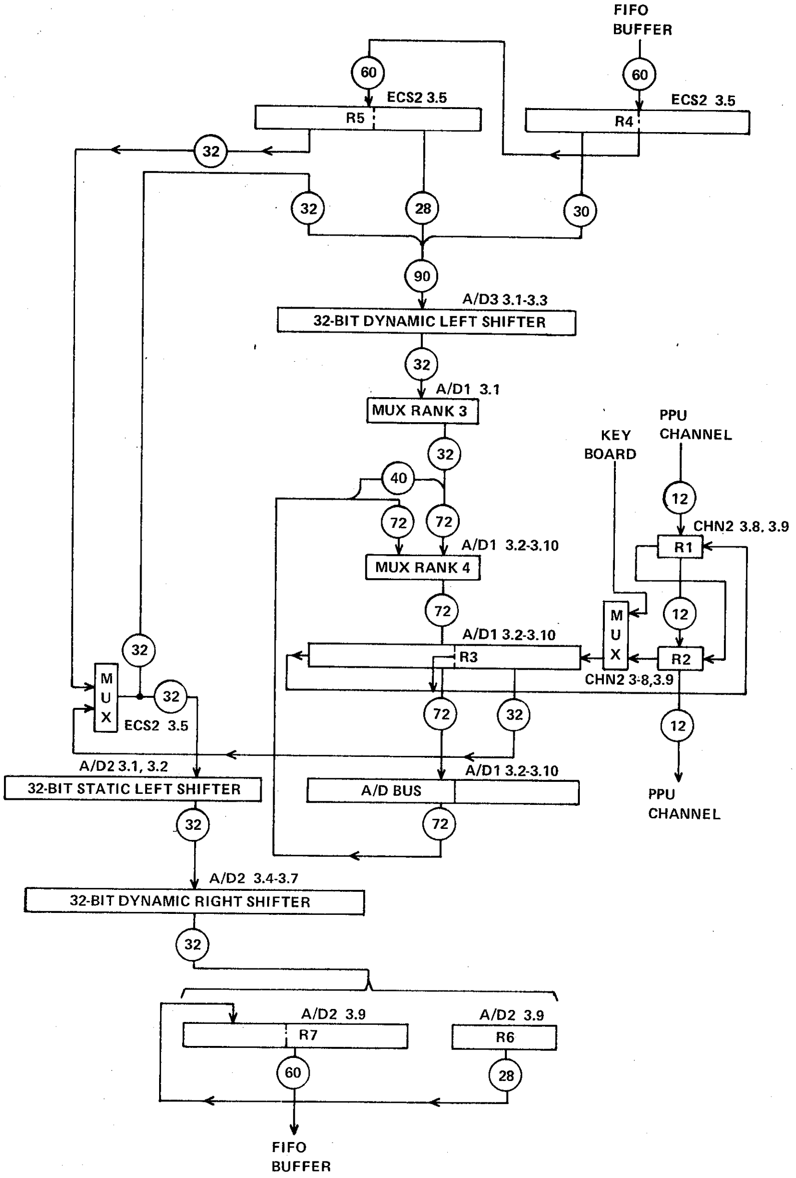


Figure 3-6. A/D Unit

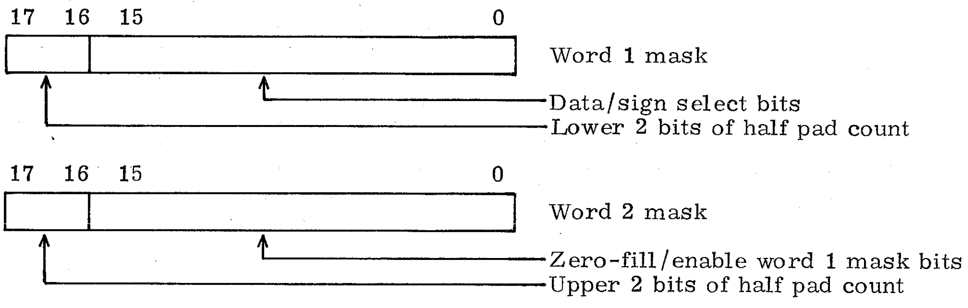

17
16
15
0
Word 1 mask
Data/sign select bits
Lower 2 bits of half pad count
17
16
15
0
Word 2 mask
Zero-fill/enable word 1 mask bits
Upper 2 bits of half pad count

Figure 3-7. A/D Mask Words

The relationship between mask bits and the R3 bits they control is:

Mask Bit	R3 Bits
0	0,1
1	2,3
2	4,5
3	6,7
4	8,9
5	10,11
6	12,13
7	14,15
8	16,17
9	18,19
10	20,21
11	22,23
12	24,25
13	26,27
14	28,29
15	30,31

Four rules govern the significance of bits 0 through 15 of mask words 1 and 2.

- Any 1 in the word 2 mask enters zeros in the controlled R3 bit positions.
- Any 0 in the word 2 mask permits the corresponding word 1 mask bit to control R3 entry.
- Any 1 in the word 1 mask enters data in the controlled R3 bit positions, providing the corresponding bit in the word 2 mask is 0.
- Any 0 in the word 1 mask enters the sign in the controlled R3 bit positions, providing the corresponding bit in the word 2 mask is 0. The sign is defined as the higher order bit of the two R3 bits controlled by the highest order 1 bit in the word 1 mask.

Bits 16 and 17 of each mask word when taken together form a half pad count. MAP doubles the half pad count to determine the number of bit positions between R3,bit 31, and the highest order R3 bit (between bit 1 and bit 31) to be transferred to ECS.

Keyboard A/D Operations

These operations include entering data from the keyboard to R3 or the breakpoint register, backspacing R3 with the → switch, and clearing R3.

R3 or Breakpoint Entry

Pressing one of the octal digit switches on the keyboard causes maintenance panel logic to send the encoded value of the switch and a key down activity code to pak CT9. If the BREAKPOINT ENTER switch is held down, the octal digit enters the lower three bits of the breakpoint shift register on CT9 3.2. Otherwise, the octal digit enters the lower three bits of R3 on A/D1 3.1. The contents of either the breakpoint shift register or R3 shift left three bit positions as each octal digit is entered. When either register is filled, further shifts are end-off.

R3 Backspace

Instead of right-shifting R3 when the → switch is pressed, a counter on CT10 3.9 controls backspace operations by left-shifting the 96-bit circular register formed by R1, R2, and R3 93 bit positions (31 R3 clocks). After the backspace, the initial lower three bits from R1 become the final upper three bits of R3.

R3 Clear

Pressing the CLR switch causes the following.

- Word 2 mask is disabled to force zeros to R3 bits 0 through 31.
- Mux rank 4 is inhibited to force zeros to R3 bits 32 through 71.
- R3 is then clocked.

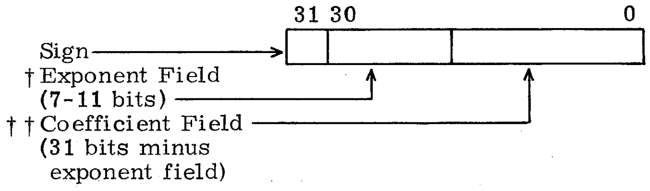
NUMERICAL CONVERSION UNIT

The numerical conversion unit accepts a 32-bit number from the A/D bus or operand bus 2, performs a conversion, and supplies the result to the A/D bus or the result bus. The unit is capable of performing the following types of conversions.

NOTE

Internal refers to the internal MAP format. External refers to formats used by the computer(s) to which MAP is connected.

1. External fixed-point (ones complement, twos complement, or left-justified sign/magnitude) to internal.
2. External floating-point to internal. Figure 3-8 shows the format to which the external word must adhere.
3. Internal to external fixed-point
4. Internal to external floating-point
5. Internal sign extension
6. Internal sign inversion
7. Internal exponent adjustment
8. No conversion



†The exponent may represent powers of 2, 4, or 16 and must be biased by $2^n/2$ where n is the number of bits in the exponent.

††The coefficient may be ones complement, twos complement, or sign/magnitude.

Figure 3-8. External Floating-Point Format

Conversion Control Words

Before initiating a numerical conversion operation, controlware uses ROP microinstructions to load a numerical conversion control (NCC) word and a radix point adjust (RPA) word into the numerical conversion unit.

NCC Word

This word identifies the external format, specifies the conversion direction, and provides for disabling conversion or inverting the sign during internal to external conversions. Figure 3-9 shows bit fields in the NCC word.

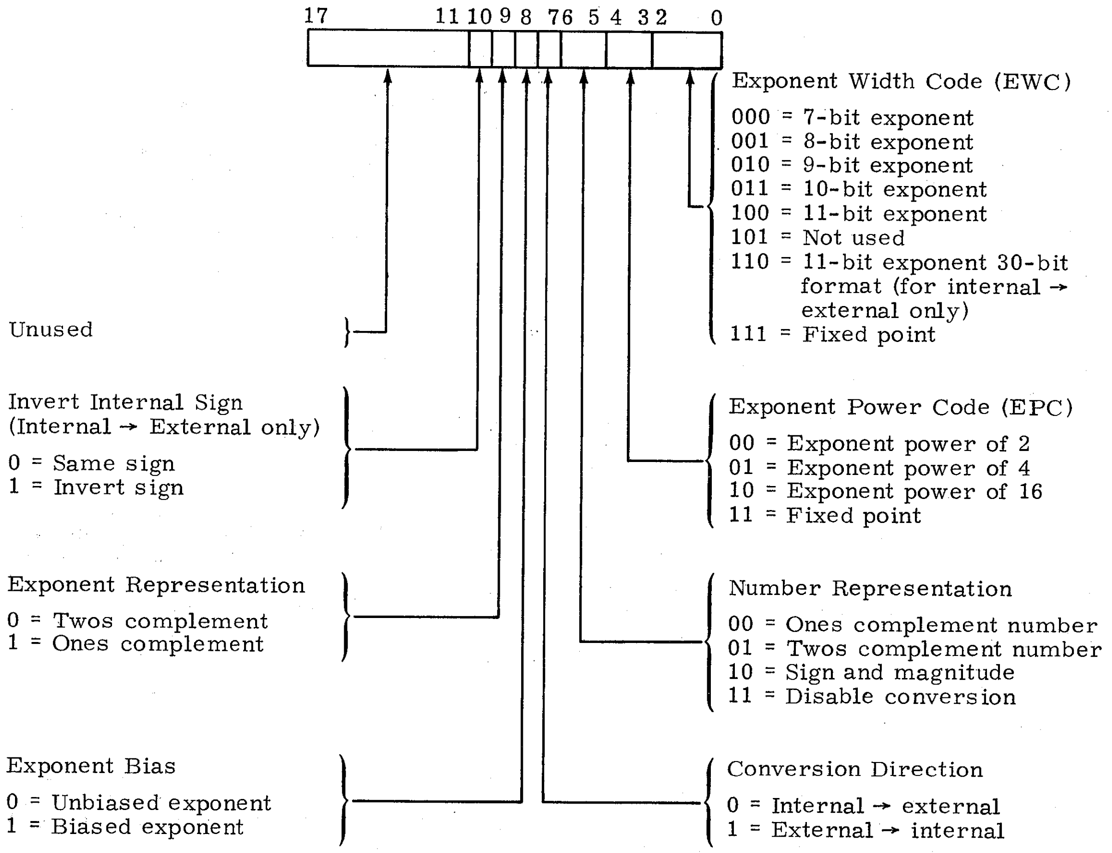


Figure 3-9. NCC Word

RPA Word

The RPA word (figure 3-10) contains a 12-bit twos complement number which normally is used to correlate internal and external exponents during conversions.

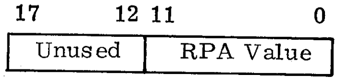


Figure 3-10. RPA Word

The RPA value is 0 when the radix point of an external floating-point number is immediately to the left of the most significant bit of the coefficient. If the radix point of an external floating-point number is to the right of the most significant bit of the coefficient, the displacement to the right equals the absolute value of the RPA word. In this case, external to internal conversions require a positive value, and internal to external conversions require a negative value. External fixed-point numbers always require a positive RPA value, regardless of conversion direction. For fixed-point conversion, the RPA value equals the width (excluding the sign bit) of the external fixed-point number.

The RPA value can also be used to adjust a set of numbers outside the range of the MAP internal format to fit within the range of the internal format. For example, RPA values of 7160_8(external to internal) and 0620_8(internal to external) permit MAP to perform add/subtract operations on 6000 series floating-point numbers between 2601_8 and 2777_8. In this case, MAP treats all positive numbers less than or equal to 2600_8 as 0.

Numerical Conversion Errors

The numerical conversion unit checks for overflow and underflow during all conversion operations. Numerical conversion overflow and underflow conditions are described under 7020_8 Select Status Words (Word 6, Arithmetic Errors 3) in section 2.

Conversion Sequences

Figures 3-11 through 3-14 show simplified paths through the numerical conversion unit for various types of conversions. For all conversions, the numerical conversion output register can be loaded one microinstruction time after the appropriate input register has been loaded.

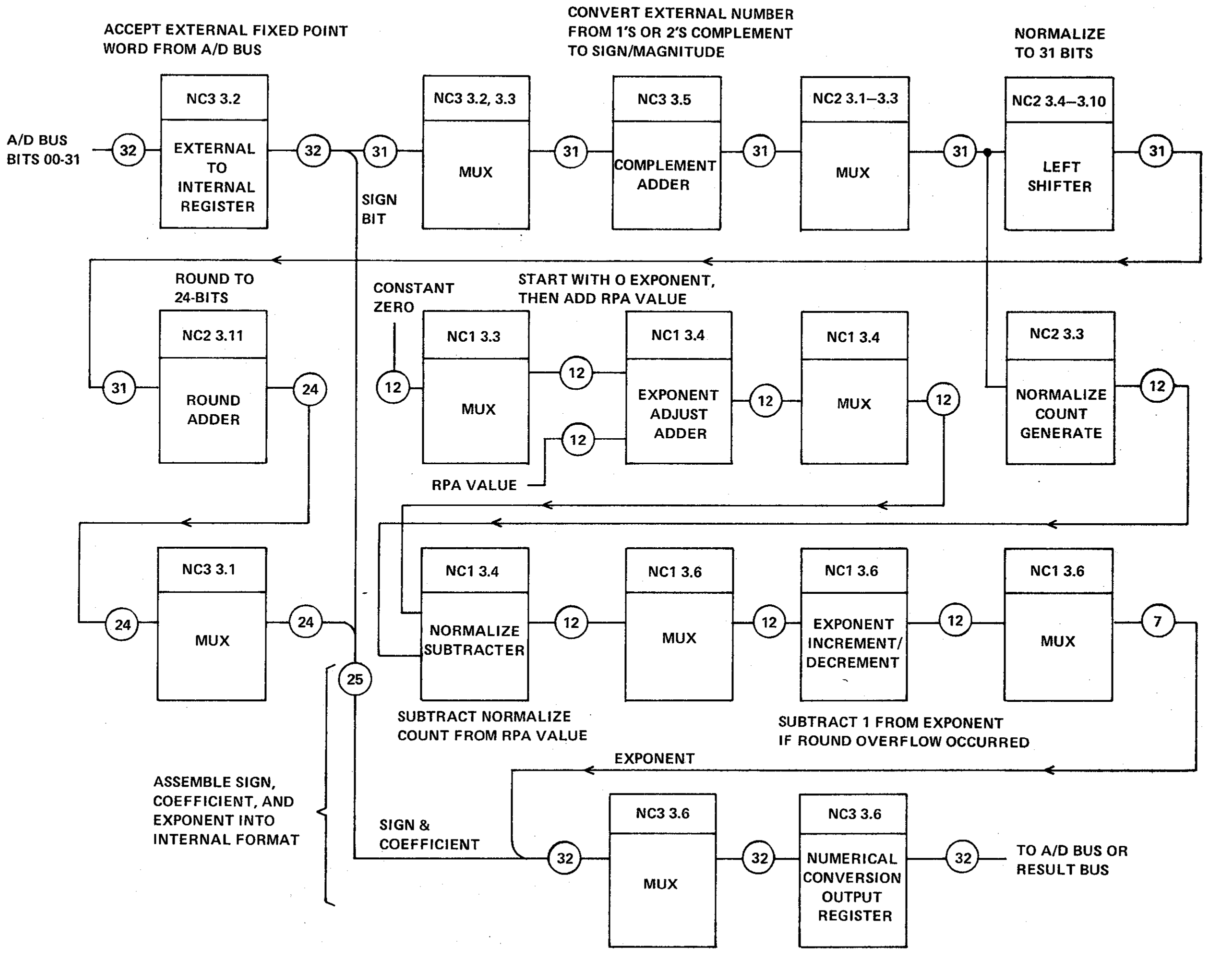


Figure 3-11. External Fixed-Point to Internal

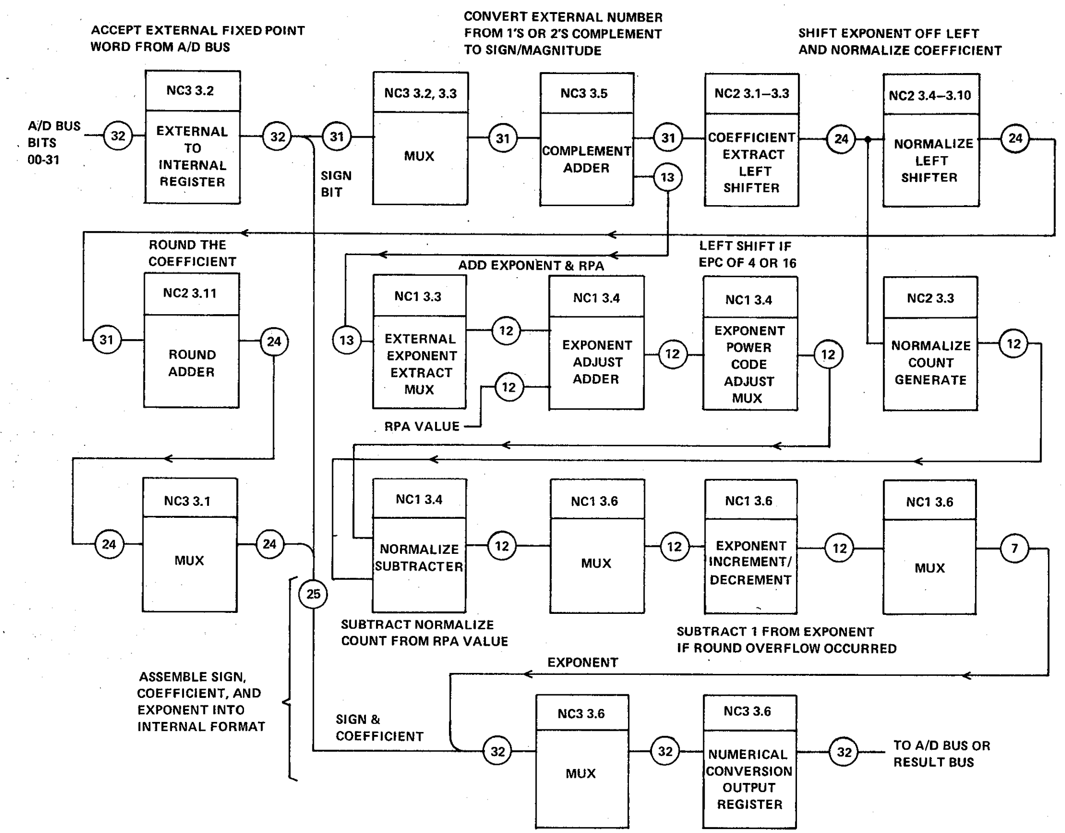

ACCEPT EXTERNAL FIXED POINT WORD FROM A/D BUS
CONVERT EXTERNAL NUMBER FROM 1'S OR 2'S COMPLEMENT TO SIGN/MAGNITUDE
SHIFT EXPONENT OFF LEFT AND NORMALIZE COEFFICIENT
A/D BUS BITS 00-31
32
NC3 3.2
EXTERNAL TO INTERNAL REGISTER
32
31
SIGN BIT
NC3 3.2, 3.3
MUX
31
NC3 3.5
COMPLEMENT ADDER
31
13
NC2 3.1–3.3
COEFFICIENT EXTRACT LEFT SHIFTER
24
NC2 3.4–3.10
NORMALIZE LEFT SHIFTER
24
ROUND THE COEFFICIENT
NC2 3.11
31
ROUND ADDER
24
ADD EXPONENT & RPA
13
NC1 3.3
EXTERNAL EXPONENT EXTRACT MUX
12
12
RPA VALUE
NC1 3.4
EXPONENT ADJUST ADDER
12
LEFT SHIFT IF EPC OF 4 OR 16
NC1 3.4
EXPONENT POWER CODE ADJUST MUX
12
NC2 3.3
NORMALIZE COUNT GENERATE
12
NC3 3.1
24
MUX
24
25
NC1 3.4
NORMALIZE SUBTRACTER
12
NC1 3.6
MUX
12
NC1 3.6
EXPONENT INCREMENT/ DECREMENT
12
NC1 3.6
MUX
7
SUBTRACT NORMALIZE COUNT FROM RPA VALUE
SUBTRACT 1 FROM EXPONENT IF ROUND OVERFLOW OCCURRED
EXPONENT
ASSEMBLE SIGN, COEFFICIENT, AND EXPONENT INTO INTERNAL FORMAT
SIGN & COEFFICIENT
32
NC3 3.6
MUX
32
NC3 3.6
NUMERICAL CONVERSION OUTPUT REGISTER
32
TO A/D BUS OR RESULT BUS

Figure 3-12. External Floating-Point to Internal

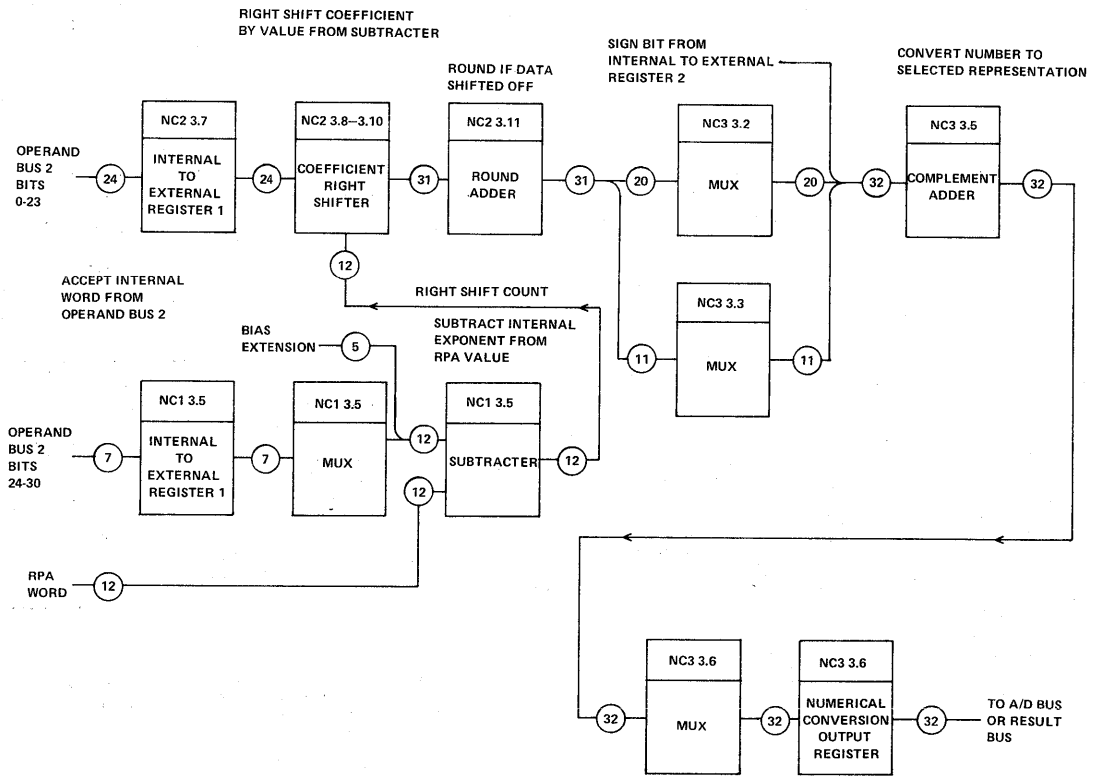
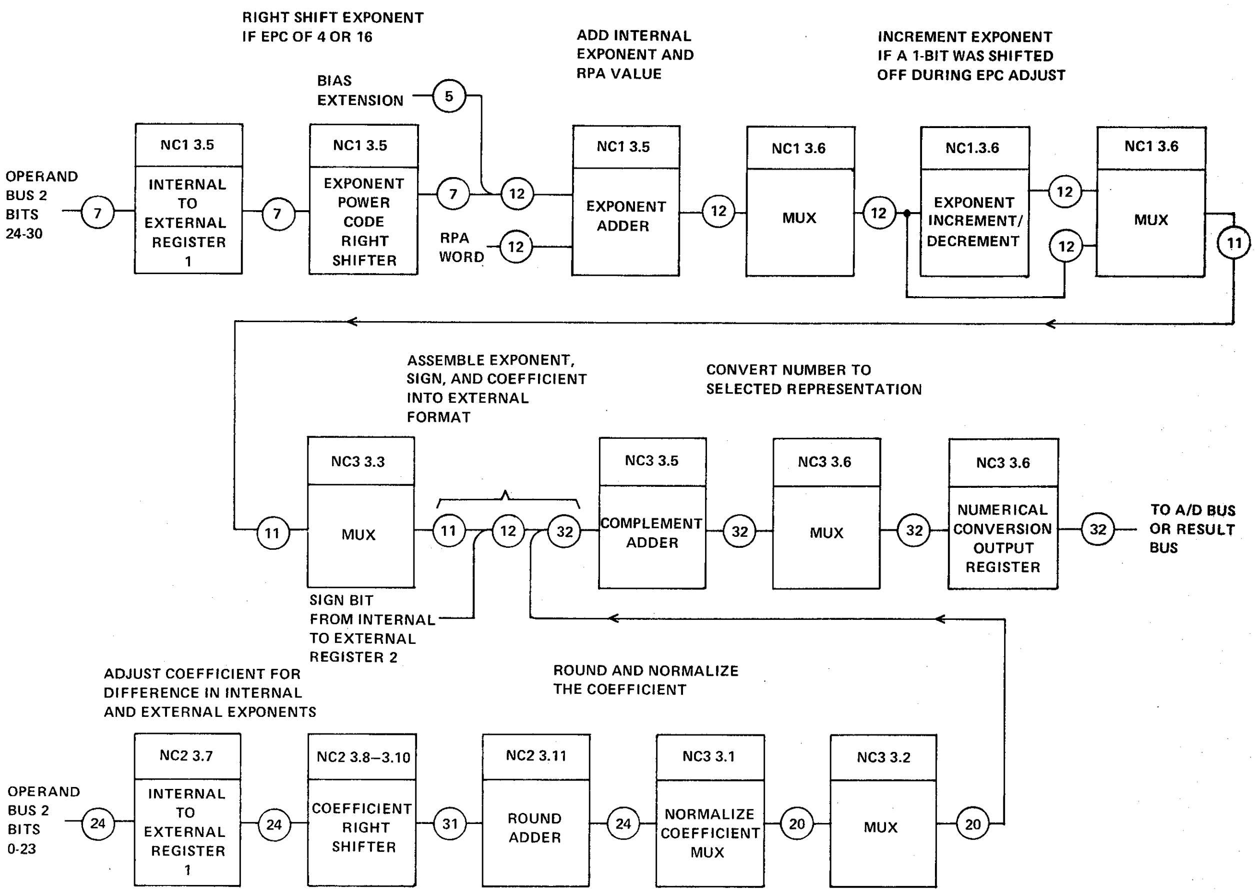

RIGHT SHIFT COEFFICIENT BY VALUE FROM SUBTRACTER
ROUND IF DATA SHIFTED OFF
SIGN BIT FROM INTERNAL TO EXTERNAL REGISTER 2
CONVERT NUMBER TO SELECTED REPRESENTATION
OPERAND BUS 2 BITS 0-23
24
NC2 3.7
INTERNAL TO EXTERNAL REGISTER 1
24
NC2 3.8–3.10
COEFFICIENT RIGHT SHIFTER
31
NC2 3.11
ROUND ADDER
31
20
NC3 3.2
MUX
20
32
NC3 3.5
COMPLEMENT ADDER
32
12
RIGHT SHIFT COUNT
ACCEPT INTERNAL WORD FROM OPERAND BUS 2
BIAS EXTENSION
5
SUBTRACT INTERNAL EXPONENT FROM RPA VALUE
11
NC3 3.3
MUX
11
OPERAND BUS 2 BITS 24-30
7
NC1 3.5
INTERNAL TO EXTERNAL REGISTER 1
7
NC1 3.5
MUX
12
12
NC1 3.5
SUBTRACTER
12
RPA WORD
12
32
NC3 3.6
MUX
32
NC3 3.6
NUMERICAL CONVERSION OUTPUT REGISTER
32
TO A/D BUS OR RESULT BUS

Figure 3-13. Internal To External Fixed-Point



RIGHT SHIFT EXPONENT IF EPC OF 4 OR 16
ADD INTERNAL EXPONENT AND RPA VALUE
INCREMENT EXPONENT IF A 1-BIT WAS SHIFTED OFF DURING EPC ADJUST
BIAS EXTENSION
5
OPERAND BUS 2 BITS 24-30
7
NC1 3.5
INTERNAL TO EXTERNAL REGISTER 1
7
NC1 3.5
EXPONENT POWER CODE RIGHT SHIFTER
7
12
RPA WORD
12
NC1 3.5
EXPONENT ADDER
12
NC1 3.6
MUX
12
NC1.3.6
EXPONENT INCREMENT/ DECREMENT
12
12
NC1 3.6
MUX
11
ASSEMBLE EXPONENT, SIGN, AND COEFFICIENT INTO EXTERNAL FORMAT
CONVERT NUMBER TO SELECTED REPRESENTATION
11
NC3 3.3
MUX
11
12
32
NC3 3.5
COMPLEMENT ADDER
32
NC3 3.6
MUX
32
NC3 3.6
NUMERICAL CONVERSION OUTPUT REGISTER
32
TO A/D BUS OR RESULT BUS
SIGN BIT FROM INTERNAL TO EXTERNAL REGISTER 2
ADJUST COEFFICIENT FOR DIFFERENCE IN INTERNAL AND EXTERNAL EXPONENTS
ROUND AND NORMALIZE THE COEFFICIENT
OPERAND BUS 2 BITS 0-23
24
NC2 3.7
INTERNAL TO EXTERNAL REGISTER 1
24
NC2 3.8–3.10
COEFFICIENT RIGHT SHIFTER
31
NC2 3.11
ROUND ADDER
24
NC3 3.1
NORMALIZE COEFFICIENT MUX
20
NC3 3.2
MUX
20

Figure 3-14. Internal to External Floating-Point

CONTROL UNIT

Diagram sheet MAP 1.2 in section 6 shows the interconnection of the 10 paks which comprise the control unit. The following paragraphs describe the major functional entities within the control unit.

Macro Memory

Figure 3-15 shows macro memory, its associated address logic, and the paths which connect macro memory with the rest of the control unit.

Macro memory is a 19-bit (18 data bits and 1 parity bit), 1024-word, semiconductor read/write memory used to store external controlware. External controlware consists of one or more routines (called macro strings) which call selected control memory routines into execution. Section 2 describes the macro memory instruction set, and the user reference manual listed in the preface supplies procedures for generating macro strings.

Macro memory receives data from the ALU bus and provides data to the A feeder bus.

Macro Pointer Register (MPR)

This 10-bit register is loaded by microinstruction from the lower 10 bits of the ALU bus to provide a base address for macro memory. The interregister (2_8) microinstruction provides for loading MPR and for using MPR as a direct address or part of a relative address.

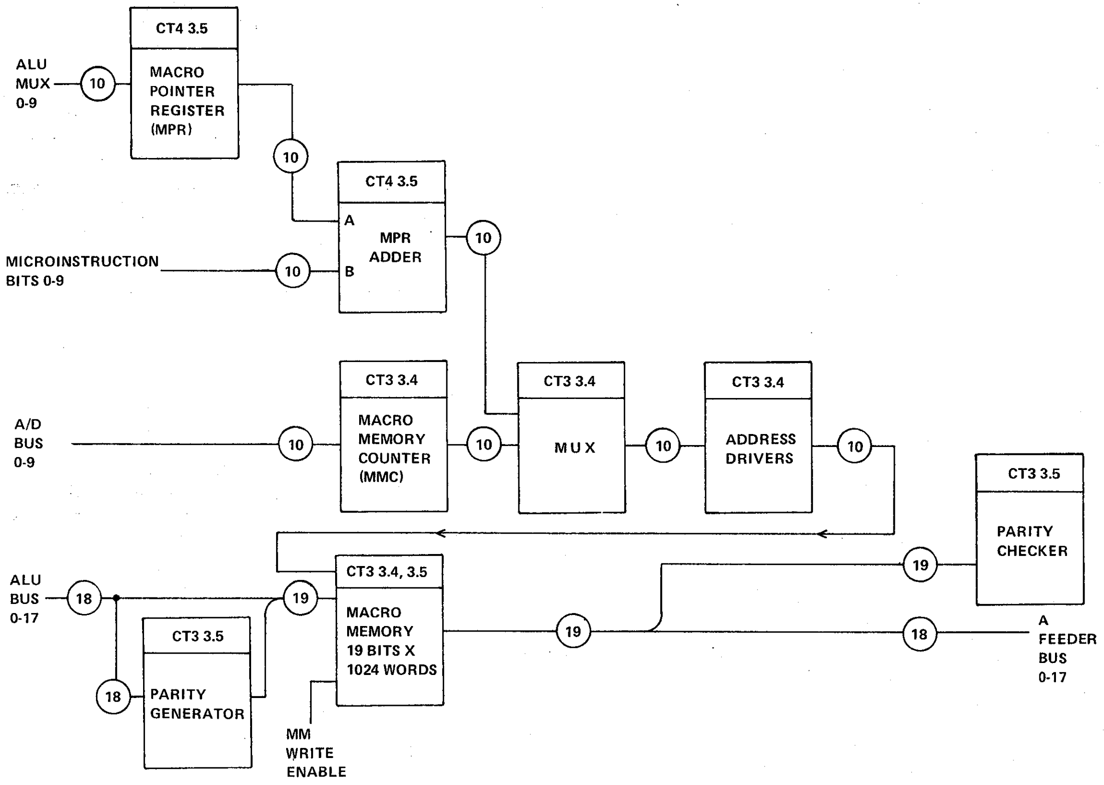


Figure 3-15. Macro Memory and Associated Logic

MPR Adder

This adder handles all macro memory addresses except those coming from the macro memory counter. The three types of noncounter addresses are as follows:

Address Type	Source
Direct	Microinstruction bits 0 through 9
MPR direct	Macro pointer register
MPR relative	MPR plus microinstruction bits 0 through 9

Macro Memory Counter (MMC)

This 10-bit up counter provides macro memory addresses when macro memory is being loaded from or unloaded to the A/D bus. MMC is preset from the lower 10 bits of the A/D bus and increments each time a maintenance panel operation or control processor data transfer (0) microinstruction transfers a word to or from macro memory.

Parity Generator/Checker

The macro memory parity generator appends an odd parity bit to each word written in macro memory. The parity checker examines each word read from macro memory for a parity error. When the parity checker detects an error, the macro memory parity error latch on CT9 3.2 sets.

Control Memory

Figure 3-16 shows control memory, its associated address logic, and the paths which connect control memory with the rest of the control unit.

Control memory consists of two 1024-word banks of semiconductor read/write memory and a 256-word semiconductor read-only memory (ROM). Word length is 60-bits with the read/write portion having an odd parity bit for a total of 61 bits.

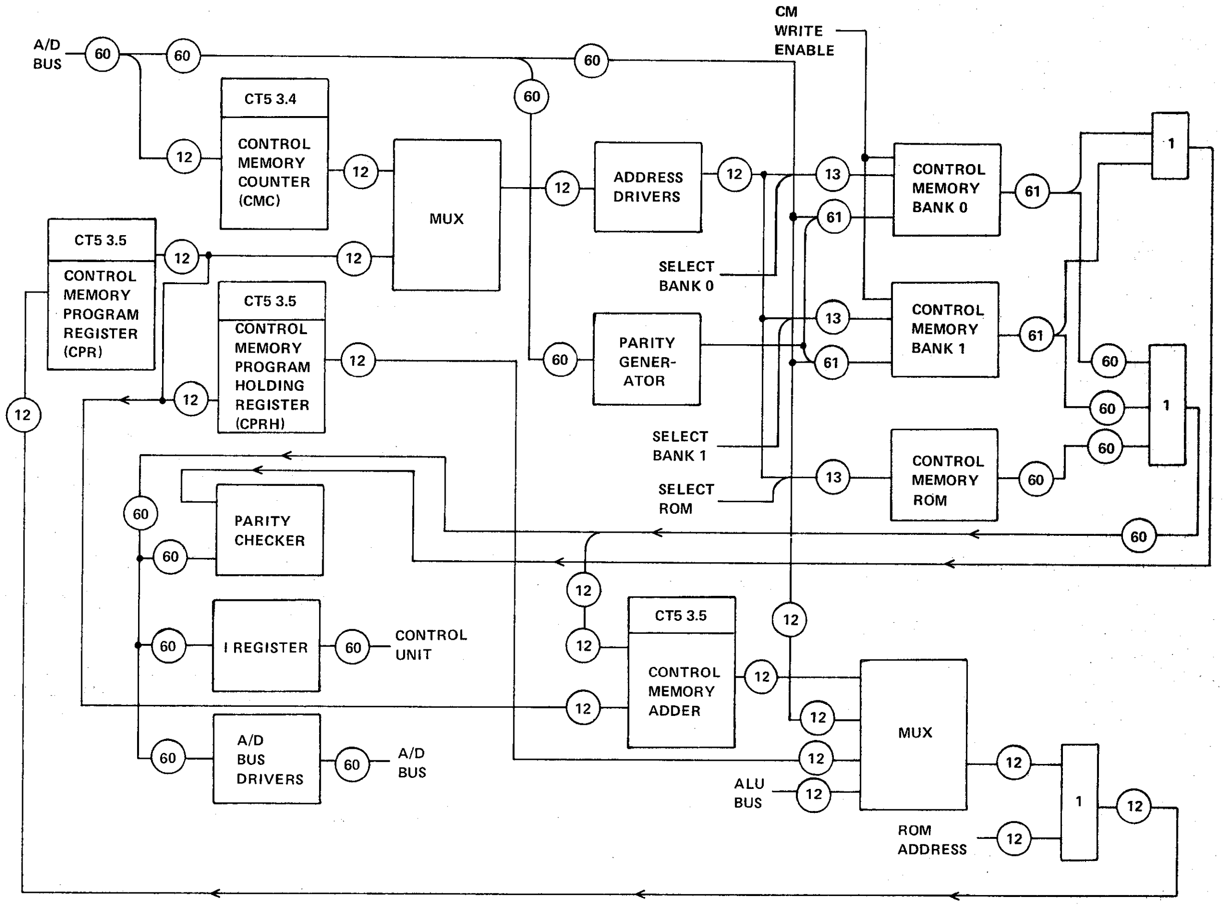


Figure 3-16. Control Memory and Associated Logic

Control memory and subcontrol memory together store internal controlware. This type of controlware consists of routines which respond to PPU channel functions, transfer data to and from ECS, and perform calculations. Section 2 describes the control memory instruction set, and the 6000 MAP III Assembler ERS listed in the preface describes the assembler used to generate internal controlware.

Control memory receives data from the A/D bus and provides data to the microinstruction (I) register and the A/D bus.

Control Memory Program Address Register (CPR)

This 12-bit register/counter holds the address of the next microinstruction to be executed, and increments when the microinstruction is executed. CPR can be loaded from the following sources.

CPR Address Source	Operation
Control memory adder	Microinstruction skip/jump
A/D bus	Maintenance panel load CPR
CPRH	Return jump
ALU bus	Interregister operation
ROM address	Controlware function or maintenance panel ROM program initiation

Control Memory Counter (CMC)

This 12-bit up counter provides control memory addresses when control memory is being loaded from or unloaded to the A/D bus. CMC is preset from the lower 12 bits of the A/D bus and increments each time a maintenance panel operation or control processor data transfer (0) microinstruction transfers a word to or from control memory.

Control Memory Program Address Holding Register (CPRH)

When a jump control (4_8) microinstruction selects a return jump,CPRH accepts the CPR address plus 1. This quantity is reentered in CPR when a jump control microinstruction with a return bit occurs.

Parity Generator/Checker

The control memory parity generator appends an odd parity bit to each word written in bank 0 or 1. The parity checker examines each word read from control memory for a parity error. Since words from the ROM portion of control memory do not have parity bits, parity error detection is disabled for ROM requests. When the parity checker detects an error in data from the read/write portion of control memory, the control memory parity error latch on CT9 3.2 sets.

Control Memory Adder

During execution of jump control (4_8) microinstructions, this adder passes a direct address from control memory bits 0 through 11 to the CPR input mux, or adds (twos complement) a relative address from control memory bits 0 through 11 to the current CPR address and sends the result to the CPR input mux.

I Register

This register holds the microinstruction currently being executed.

A/D Bus Drivers

These drivers place the currently selected control memory word on the A/D bus.

Subcontrol Memory

Figure 3-17 shows subcontrol memory, its associated address logic, and the paths which connect subcontrol memory with arithmetic units and the rest of the control unit.

Subcontrol memory is a 73-bit (72 data bits and 1 parity bit), 1024-word, semiconductor read/write memory used to store the portion of internal controlware which controls arithmetic units, operand busses, and the result bus. Section 2 describes the subcontrol memory instruction (called a subinstruction). Each subinstruction is associated with one or more arithmetic (3_8) or arithmetic index load (6_8) microinstructions.

Subcontrol memory receives data from the A/D bus and provides data to the subinstruction (SI) register and A/D bus drivers.

Subcontrol Memory Counter (SMC)

This 10-bit up counter provides subcontrol memory addresses when subcontrol memory is being loaded from or unloaded to the A/D bus. SMC is preset from the lower 10 bits of the A/D bus and increments each time a maintenance panel operation or control processor data transfer (0) microinstruction transfers a word to or from subcontrol memory.

Parity Generator/Checker

The subcontrol memory parity generator appends an odd parity bit to each word written in subcontrol memory. The parity checker examines each word read from subcontrol memory for a parity error. When the parity checker detects an error, the subcontrol memory parity error latch on CT9 3.2 sets.

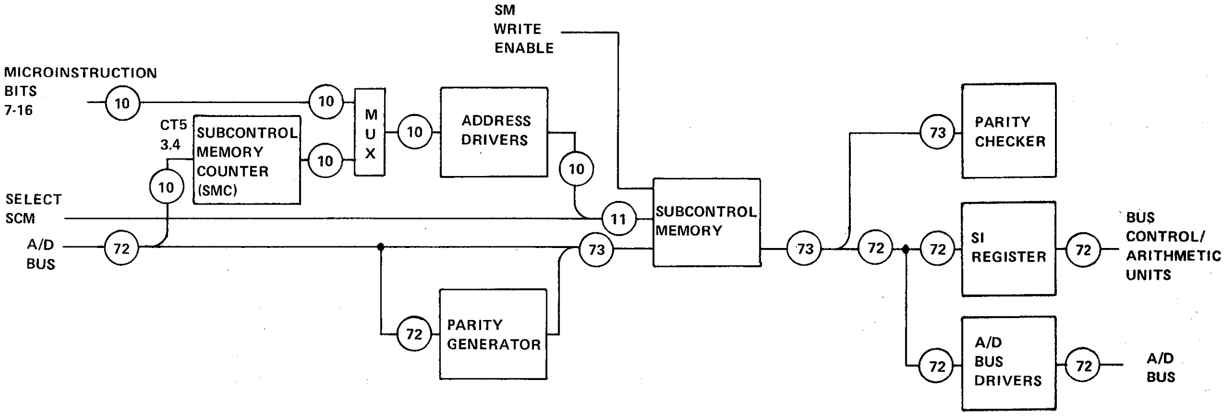


Figure 3-17. Subcontrol Memory and Associated Logic

SI Register

This register holds the subinstruction currently being executed.

A/D Bus Drivers

These drivers place the current subinstruction on the A/D bus.

Clocks

Figure 3-18 shows the MAP clock network. A pair of tuned wire oscillators on pak CT8 provide the normal clock and a margin clock which is about 10 percent faster than the normal clock. Rocker switches enable one oscillator or the other to the master clock fanout on pak CT2. Section 5 contains complete clock switch information.

Clock Fanout

The master clock fanout distributes the switch selected oscillator signal throughout MAP as signal CLKx. One of the CLKx signals feeds a shift register (on pak CT8) which generates T0 through T3. T3 feeds the BUS paks to generate T4 sync (T4S) signals. T3 also feeds logic which generates the primary T4 sync-no conflict (T4SC) signal. T4SC is distributed throughout MAP by fanouts on the bus paks.

Clock Signal Relationship

Each microinstruction time has five clock times, labeled T0 through T4. T0 is defined as the first clock pulse following each T4S or T4SC signal. Clock times are generated independently on each pak from CLK and T4S or T4SC signals. Clock times generated from T4SC occur only in the absence of data storage conflicts or skip next clock (SNC) signals. Clock times generated from T4S occur regardless of data storage conflicts or SNC signals.

Diagram CT8 3.4 includes a timing chart showing the relationship of MAP clock signals.

ALU/Data Storage Address

Figure 3-19 shows the major components involved in control unit arithmetic and logical operations. ALU logic is on pak CT4, access logic is primarily on pak CT6, and section address logic is on pak CT7.

ALU Logic

This circuitry consists of two, 18-bit, 16-word register files, an 18-bit ALU with feeder registers, a right-shift mux, and three busses.

K file registers are entered by microinstruction from the ALU bus. They feed the A feeder bus (K file A) and the B feeder bus (K file B). A and B feeder registers accept quantities from the two feeder busses and provide stable inputs to the ALU. A mux between the A feeder register and the ALU permits microinstruction bits 0 through 17 to be the ALU A input.

The ALU consists of a rank of 10181 ALU chips with associated carry look-ahead circuitry. A 4 x 1 mux attached to the ALU output permits A/D bus bits 0 through 17 to be placed on the ALU bus and also provides for one-place and nine-place end-around right-shifts of the ALU result.

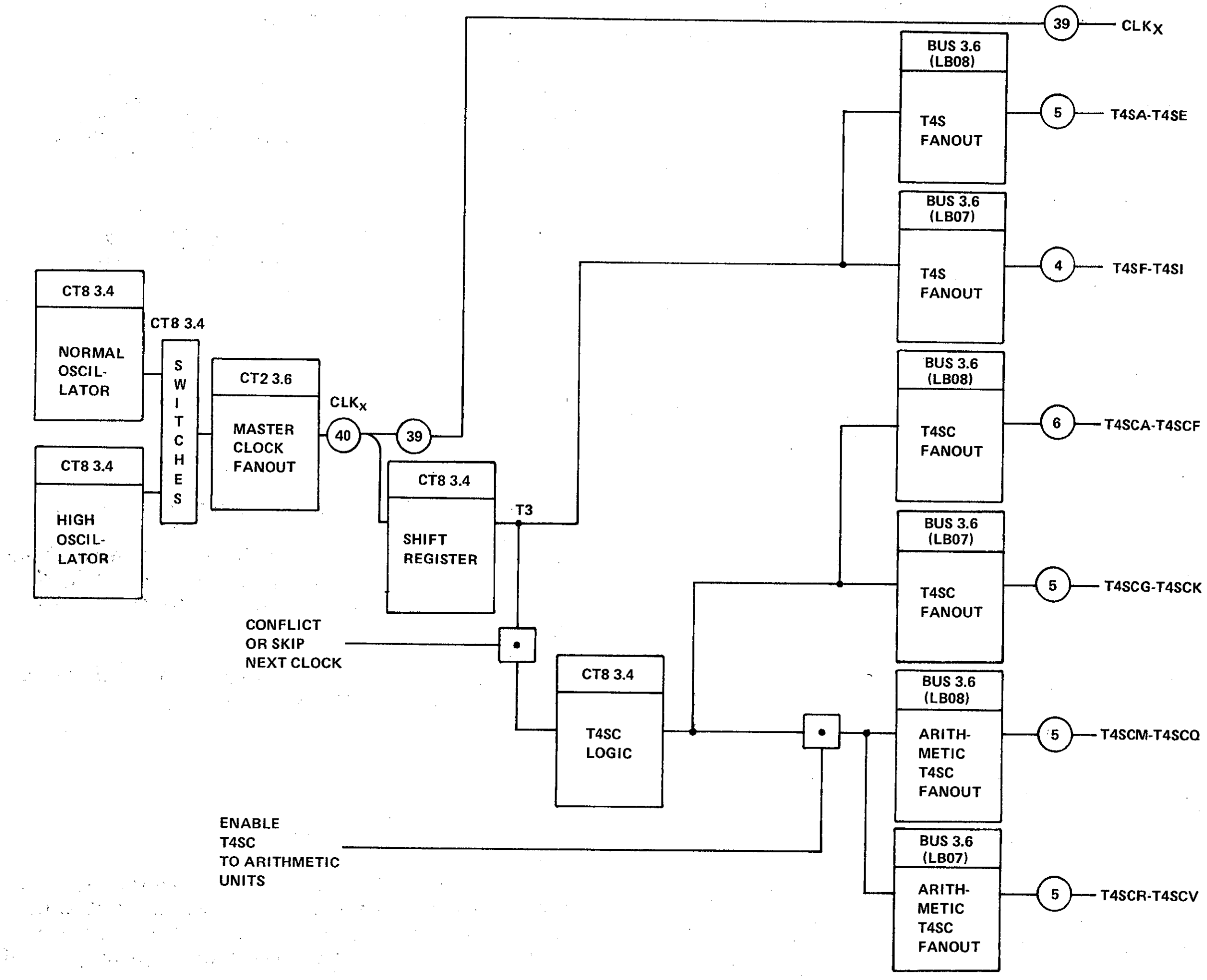


Figure 3-18. MAP Clock Network

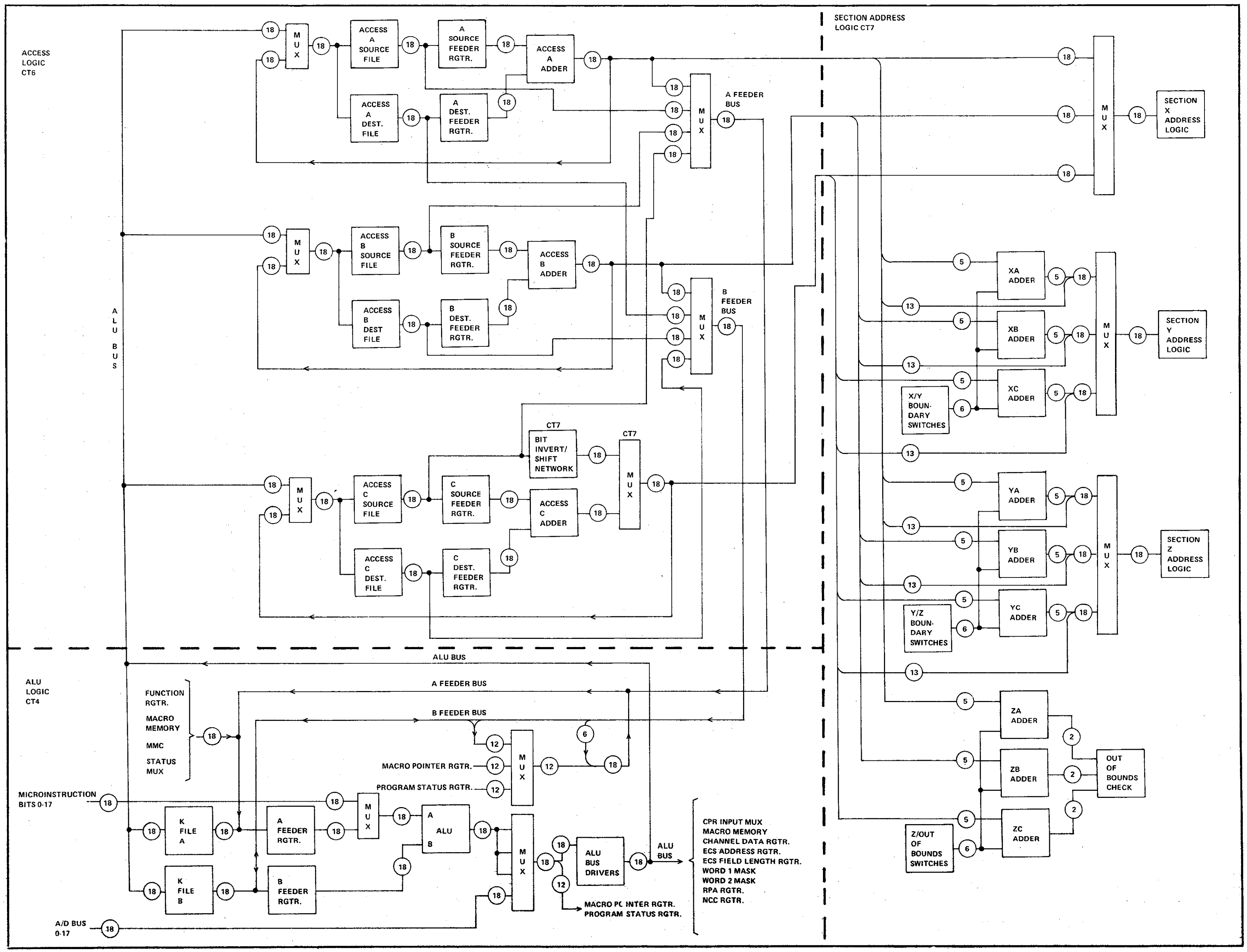


Figure 3-19. ALU/Data Storage Address Logic

The A and B feeder busses permit a variety of quantities to be gated to the ALU. The ALU bus distributes ALU results throughout the control unit, PPU channel interface, ECS interface, A/D unit, and numerical conversion unit. ALU input sources, result destinations, and operation types are controlled by the interregister (2_8) microinstruction which is described in section 2.

Access Logic

This logic is capable of generating three data storage addresses during each microinstruction time. Each access contains two 18-bit, 16-word register files and an 18-bit adder with feeder registers. The C access also contains a bit invert/shift network used for addressing the results of fast Fourier transform (FFT) calculations.

For accesses A and B, addresses supplied to section address logic come directly from the access adders. For access C, the address comes from a mux connected to the bit invert/shift network and the access C adder. The following microinstructions control address formation by access logic.

Operation Code	Microinstruction
1	I/O data transfer
3	Arithmetic
6	Arithmetic index load
7	File modify

The interregister (2_8) microinstruction provides for moving quantities from access logic to the ALU on the A and B feeder busses and for moving ALU results to access logic on the ALU bus. Refer to section 2 for descriptions of all of these microinstructions.

Section Address Logic

This circuitry determines the data storage section associated with each access address, resolves section conflicts, and converts access addresses to section addresses.

The three sets of boundary switches/adders examine incoming access addresses to determine the section associated with each address. Up to three access addresses can be handled each microinstruction time, provided the addresses are for different sections. When two or three access addresses are for the same section, two or three microinstruction times, respectively, are required to satisfy the data storage requests. During the time conflict data storage requests are being processed, most other MAP activity is suspended to maintain correspondence between operands and microinstructions.

The address logic for each section converts each access address to an 8 K-block address and a unique block select signal. The block select signal selects one of the STO pak pairs in the section, and the 8K block address selects one of the 8,192 cells in the STO pak pair.

Cassette

The cassette transport is mounted on the maintenance panel and is used for loading and dumping off-line diagnostic programs and data. When the CASSETTE ENABLE switch is active, the channel portion of the PPU channel interface is disabled, and data can transfer to or from the cassette transport.

Cassette Data Format

Figure 3-20 shows the character and record formats used by the cassette transport. Logic on pak CT10 determines the start and confirms the end of a record by checking for preamble and postamble patterns, respectively. The binary check character (BCC) provides the equivalent of logitudinal parity for a record. The BCC makes the total number of 1 bits odd in each of the eight bit positions of the data block.

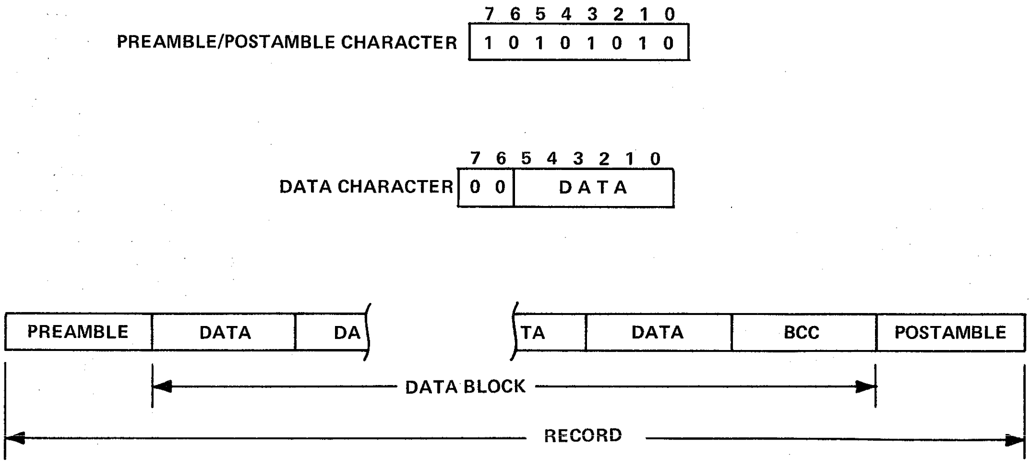
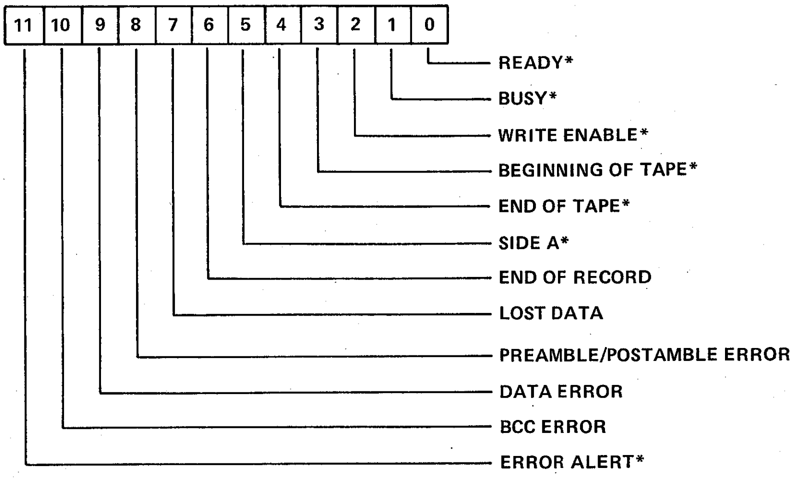


Figure 3-20. Cassette Data Formats

Cassette Status

When the CASSETTE ENABLE switch is active, cassette status can be transferred to the program status register by interregister (2_8) microinstruction and then displayed on the maintenance panel by pressing the STATUS 1 switch. Figure 3-21 shows the cassette status word format.



*ALSO DISPLAYED DIRECTLY ON MAINTENANCE PANEL

Figure 3-21. Cassette Status Word

Ready — Indicates power is applied to the cassette transport and its lid is closed.

Busy — Indicates tape is in motion.

Write Enable — Indicates the write enable plug on the cassette tape package is in place.

Beginning of Tape — Indicates the loadpoint hole is positioned at the photodetector.

End of Tape — Indicates the end of tape hole is positioned at the photodetector.

Side A — Indicates tape side A is positioned over the read/write heads.

End of Record — Indicates no data was received from the cassette transport for 166 microseconds during a read or read-after-write operation.

Lost Data — Indicates during a read operation that data was received from the cassette transport before the last 12-bit word was sent to the control unit. During a write operation, lost data indicates that 2.67 milliseconds have elapsed without either the cassette transport accepting a 12-bit data word or the control unit issuing an end of transfer signal. Lost data remains set until a master clear, panel clear, or clear errors signal is received, or until a new record is begun.

Preamble/ Postamble Error — Indicates either the preamble or the postamble received did not conform to the format shown in figure 3-20. A master clear, panel clear, or clear errors signal clears preamble/postamble error.

Data Error — Indicates bit 6 or bit 7 of a data word was not 0. A master clear, panel clear, or clear errors signal clears data error.

BCC Error — Indicates the BCC received at the end of a record did not cause the BCC being generated for that record to equal zero. A master clear, panel clear, or clear errors signal clears BCC error.

Error Alert — Indicates at least one of the cassette error status bits (bits 7, 8, 9, or 10) is active.

Microprogram Tape Rewind

A microprogram can rewind the cassette to loadpoint by entering $7xxx_8$ into the function register with an interregister (2_8) microinstruction.

Load From Cassette Operation

This operation transfers 12-bit words from the cassette transport to the control unit by way of A/D unit registers R1/R2 and the PPU channel interface. For this operation to proceed, the CASSETTE ENABLE switch must be active and control processor data transfer (1_8) microinstructions must select data flow from the computer. Each microinstruction interface request transfers one 12-bit data word to the control unit. The data transfer rate from the cassette is one 12-bit word every 2.67 milliseconds. When a data bit has not been received from the cassette for 166 microseconds, cassette control logic on pak CT10 generates an end of record signal and checks the BCC and postamble. If an interface request is pending when the end of record signal occurs, the load operation continues nonstop through the next record.

Unload to Cassette Operation

This operation transfers 12-bit words from the control unit to the cassette transport by way of A/D unit registers R1/R2 and the PPU channel interface. For this operation to proceed, the CASSETTE ENABLE switch must be active and control processor data transfer (1_8) microinstructions must select data flow to the computer. Each microinstruction interface request transfers one 12-bit data word to cassette control logic on pak CT10. The data transfer rate to the cassette is one 12-bit word every 2.67 milliseconds. Cassette control logic generates lost data status and sends a zero-filled 12-bit word to the cassette if the control unit fails to provide a data word within 2.67 milliseconds. When a microinstruction end of transfer signal occurs, cassette control logic writes the BCC and postamble, generates a record gap, and then stops tape motion.

Maintenance Panel

The maintenance panel permits an engineer to perform a variety of maintenance tasks on MAP. Section 2 describes maintenance panel displays, indicators, and switches. Section 5 provides maintenance panel procedures and section 6 diagram sheets PNLxx depict maintenance panel logic. This subsection describes some of the maintenance panel design characteristics.

Transistor-Transistor Logic

The maintenance panel uses transistor-transistor logic (TTL) because of TTL's compatibility with switches, displays, indicators, and the cassette transport. ECL/TTL transmitters and receivers interface maintenance panel logic and cassette transport logic with the control unit.

Light-Emitting Diodes

Light-emitting diodes (LEDs) are used for maintenance panel displays and indicators. Most of the switches on the maintenance panel include an LED indicator. Pressing the switch toggles a latch in the control unit, and the latch output then feeds back to drive the indicator.

Time Multiplexing

The communication link between the control unit and the maintenance panel consists of two 6-bit, time multiplexed data paths. One path carries switch information from the panel to the control unit, and the other path carries display/indicator information from the control unit to the panel. The transmitting element of each path is a multiplexer, and the receiving element of each path is a shift register. A counter on pak CT10 generates mux select codes and shift register clocks so that switch and display information is refreshed every 10 milliseconds.

The 7 x 1 switch multiplexer is shown on diagram sheet PNL 3.3 in section 6. the 24 x 1 display multiplexer is distributed among five paks in the control unit. Backup text for diagram sheet CT1 3.6 contains a block diagram of the display multiplexer.

Maintenance panel and control unit shift registers are shown on diagram sheets PNL 3.4 and CT9 3.4, respectively.

DATA STORAGE

Diagram sheet MAP 1.3 shows the relationship of paks in a fully expanded, 256K data storage. Sections X and Y can each contain up to 96K, while section Z can contain up to 64K. The minimum size for a section is 8K.

8K Block

The basic unit of data storage is the 8K block, consisting of two STO paks. Each STO pak is an 18-bit (16 data bits, 1 parity bit, and 1 spare bit), 8192-word, semiconductor read/write memory. Diagram sheet STO 2.1 shows the internal organization of and addressing scheme used by the STO pak.

Data Storage Expansion

MAP data storage can be expanded from its minimum size of 24K by adding an expansion memory bay, a number of STO paks, and in some cases, additional power supplies. Whenever the size of data storage is changed, corresponding changes must also be made in the storage switch settings on pak CT7 and in the number and location of backpanel terminator assemblies. Section 5 contains information related to storage size switch settings and backpanel terminator assembly locations.

Data Storage Operation

A typical data storage related microinstruction contains two read requests, one write request, and information for generating three data storage addresses. The reads place operands for the current operation on the operand busses and the write transfers the result of the previous operation to data storage. The following sequence describes the complete data storage operation.

1. Access logic on pak CT6 generates three data storage addresses according to information in the access fields of the microinstruction. Each address is formed by adding, subtracting, or moving a source file register to/from a destination file register. For access C, the address can come from the C access adder or from a bit inversion/shift network.
2. Section address logic on pak CT7 accepts addresses from access logic, checks for section conflicts, and issues block addresses, block enables, and write enables to data storage. When two or three accesses request the same section of data storage during one microinstruction time, additional microinstruction activity is suspended until all requests have been satisfied.
3. A read request (write enable inactive) causes a section to return the contents of the selected cell to the bus paks for placement on the microinstruction selected operand bus.
4. A write request (write enable active) causes a section to store the quantity received from the bus paks in the selected cell.

ARITHMETIC UNITS

This subsection describes the number theory applicable to each of MAP's arithmetic units. Refer to the appropriate diagrams and associated backup text in section 6 for details concerning the hardware implementation of the theory presented here.

Add/Subtract Unit

Figure 3-22 shows operand flow through the add/ subtract unit for a typical add or subtract operation. Operand A is the augend or minuend, and operand B is the addend or subtrahend. Subinstruction bits select and load operands, determine the operating mode of the unit, and load the sum register.

Add/Subtract Initiation

The add/subtract operation begins with the selected A and B operands being clocked into input registers at T0. The A operand can come from operand bus 2 or from the multiply unit associated with the add/ subtract unit. The B operand can come from operand bus 1 or from the input to the sum register. When the sum register input is the B operand, four so-called double precision (DP) bits are appended to the right of the coefficient. These bits provide an additional four bits of coefficient accuracy when the sum is to be reused as an operand.

Sum Sign Formation

The add/subtract unit generates the sum sign bit according to the operation selected, the operand signs, and the magnitude of the operands.

Sum Exponent Generation

To determine the preliminary result exponent, the add/subtract unit simultaneously subtracts exponent A from exponent B and exponent B from exponent A. The results of the subtractions identify the larger of the two exponents (the larger exponent becomes the preliminary exponent) and also indicate the difference in power between the exponents. The coefficient associated with the smaller of the two exponents is then right-shifted (end-off) a number of bit positions equal to the difference in exponents. If the difference in exponents is more than 27, the shifted coefficient becomes zero. When the exponents are equal, coefficient B is gated to the right shifter, but is not shifted.

The add/subtract unit forms the final result exponent by subtracting the coefficient normalize count from the preliminary result exponent. The coefficient normalize count is the number of 0 bits between the radix point and the highest order 1 bit from the normalize mux.

Sum Coefficient Generation

When it has determined the difference in exponent magnitude, exponent compare logic enables the coefficient associated with the smaller exponent through a right shifter. The coefficient from this shifter and the coefficient associated with the larger exponent then become the operands applied to two identical adders.

The A and B inputs of adder 1 are the shifted and unshifted coefficients, respectively, while the A and B inputs of adder 2 are the unshifted and shifted coefficients, respectively. For an add operation, the outputs of both adders are the same, so it makes no difference which output is gated to the normalize mux. For a subtract operation, however, the outputs are usually different. The correct output is gated through the normalize mux if the adder 2 output is gated when adder 2 has a carry. This is because the adder 2 carry is related to the magnitudes of the operands from the A and B input registers. When the operand A exponent is greater than or equal to the operand B exponent, the B coefficient is shifted, adder 2 has no carry, and the correct A minus B result appears at adder 1. When the operand A exponent is less than the operand B exponent, the A coefficient is shifted, adder 2 has a carry, and the correct A minus B result appears at adder 2.

Sum Assembly

A shift encoder determines the number of leading zeros in the normalize mux output. This number is applied to the normalize left shifter to normalize the coefficient. The number is also subtracted from the preliminary exponent to yield the final exponent.

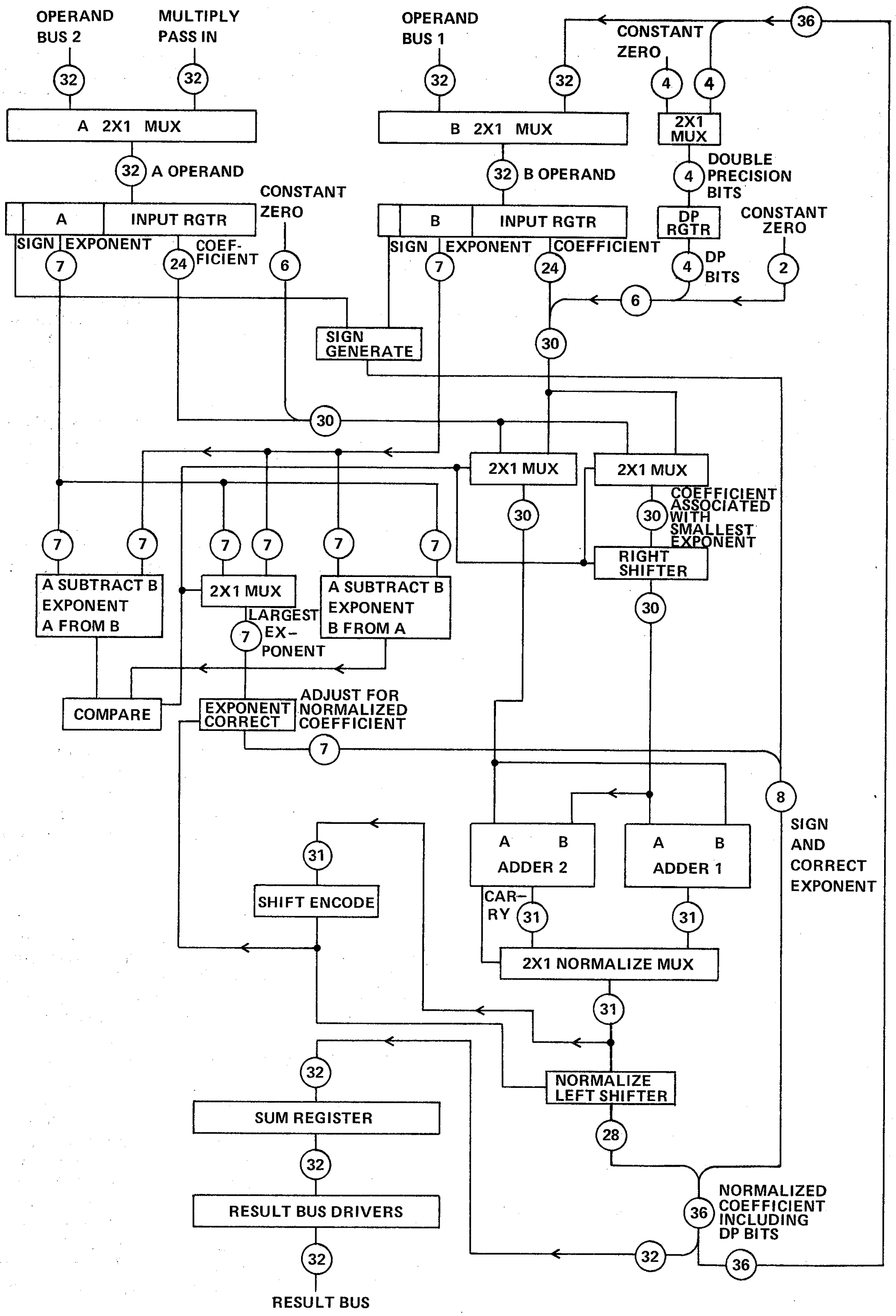


Figure 3-22. Add/Subtract Unit

The normalized coefficient (including double precision bits), the final exponent, and the sign bit combine to form a 36-bit sum which feeds back to the B operand input mux. The highest order 32 bits of the sum feed the sum register which in turn feeds result bus drivers. The sum register can be loaded one microinstruction time after the add/subtract operation begins.

Multiply Unit

Figure 3-23 shows operand flow through the multiply unit. Operand A is the multiplicand, and operand B is the multiplier. Subinstruction bits select operands, load them, and load results into the product register.

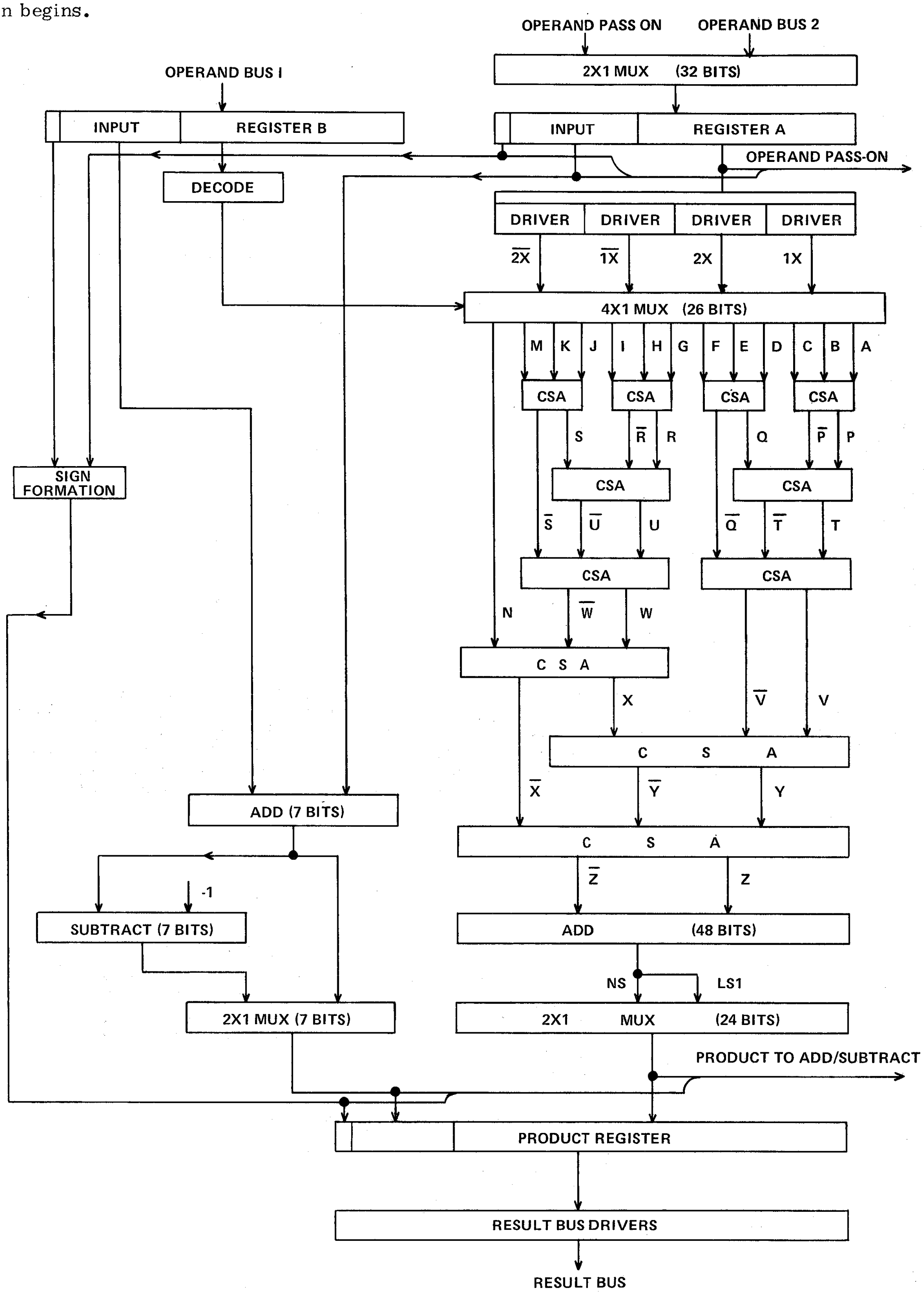


Figure 3-23. Multiply Unit

Multiply Initiation

The multiply operation begins with the selected A operand and the B operand being clocked into input registers at T0. The A operand can come from operand bus 2 or from the operand pass-on of the previous multiply unit (the first multiply unit has no pass-on input). The B operand comes from operand bus 1.

Product Sign Formation

The multiply unit generates the product sign bit by exclusive ORing the sign bits of the operands.

Product Exponent Generation

To determine the preliminary result exponent, the multiply unit adds the A and B exponents. If a 1-bit left-shift is required to normalize the result coefficient, the final result exponent is obtained by subtracting one from the preliminary result exponent. Otherwise, the preliminary result exponent becomes the final result exponent.

Product Coefficient Generation

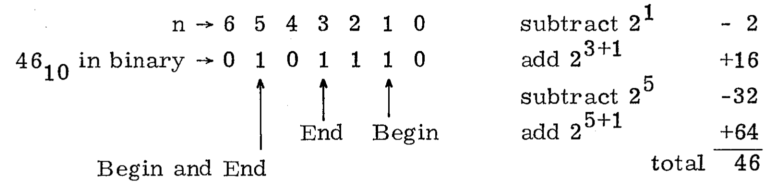
Figure 3-24 shows the algorithm which allows the multiplier coefficient to be expressed in a form which permits multiplication to be performed by carry save adds of -2, -1, 0, 1, and 2 times the multiplicand. The multiply unit places one imaginary 0 bit to the right and two imaginary 0 bits to the left of the multiplier coefficient to enable detection of the beginning and ending of strings of 1 bits.

The multiplier coefficient algorithm expresses a number according to strings of 1 bits in the number. The number is analyzed from right to left according to two rules.

- Subtract 2^n if bit n begins a string of 1 bits.
- Add 2^{n+1} if bit n ends a string of 1 bits.

Note that a string can contain a single 1 bit.

The number 46_{10} is analyzed according to the above rules as follows:



Decoders in the multiply unit examine overlapping 3-bit fields of the multiplier coefficient to detect beginnings, ends, and absences of strings. In order to permit string detection, one 0 bit is appended to the right of the number and two 0 bits are appended to the left of the number. By considering the three bits being examined at a time as being 2^{-1}, 2^0, and 2^1, a composite factor can be generated to represent the three bits. As the following shows, the possible values for this composite factor are -2, -1, 0, 1, and 2. These values can be rapidly generated by shift/complement operations in the multiply unit.

Analyzed Bits			String Status	Composite Factor
2^1	2^0	2^{-1}		
0	0	.0	No string	= 0
0	0	.1	End of string	$= 2^{-1+1} = 2^0 = 1$
0	1	.0	Beginning and end of string	$= -2^0 + 2^{0+1} = -1 +2 = 1$
0	1	.1	End of string	$= 2^{0+1} = 2^1 = 2$
1	0	.0	Beginning of string	$= -2^1 = -2$
1	0	.1	End and beginning of string	$= 2^{-1+1} -2^1 = 1-2_0 = -1$
1	1	.0	Beginning of string	$= -2^0 = -1$
1	1	.1	No string	= 0

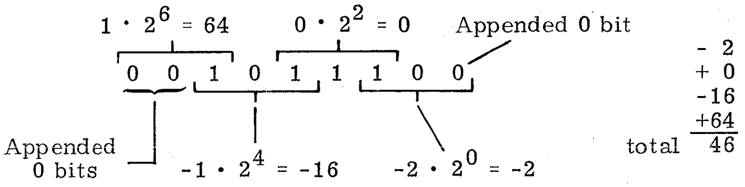


Figure 3-24. Multiplier Coefficient Algorithm

When the multiplier has been clocked into the B input register, the multiply unit develops 12 3-bit codes; one code for each two bits of the multiplier coefficient. These codes enable -2, -1, 0, 1, or 2 times the multiplicand through 12 4 x 1 muxes to the carry save adder ranks. Multiplicand multiples are left shifted to pass through the carry save adders according to their powers of two. An internal carry adder samples the last rank of the carry save adders to form the coefficient product. If bit 23 of the product coefficient is 0, the coefficient is left-shifted one place and the product exponent is decremented by 1 to normalize the product.

Product Assembly

The multiply unit normalizes the product coefficient by left-shifting it one place if the highest order bit is 0. The product exponent is then reduced by 1 to compensate for the left-shift.

The sign bit, product exponent, and normalized product coefficient combine to form a 32-bit product which feeds the associated add/subtract unit and the product register. The product register can be loaded one microinstruction time after the multiply operation starts.

Divide Unit

Figure 3-25 shows operand flow through the divide unit. Operand A is the dividend, and operand B is the divisor. Subinstruction bits load operands into the input registers and the result into the quotient register. There is also a divide maintenance mode subinstruction bit which slows down the divide operation so the quotient can be sampled (by microinstruction) as it is being formed.

Divide Initiation

The divide operation begins with the selected A and B operands being clocked into input registers at T0. Operand A comes from operand bus 2, and operand B comes from operand bus 1.

Quotient Sign Formation

The divide unit generates the quotient sign bit by exclusive ORing the sign bits of the operands.

Quotient Exponent Generation

The divide unit forms the preliminary quotient exponent by subtracting the divisor (B) exponent from the dividend (A) exponent. When the last coefficient pass has completed, the divide unit reduces the preliminary quotient by 1 if a one-place left-shift is required to normalize the quotient coefficient.

Quotient Coefficient Generation

The divide unit forms the quotient coefficient by recording the number of times the divisor coefficient can be subtracted from the dividend coefficient. To reduce the number of passes required, the divide unit incorporates two sets of parallel subtracters and associated dividend coefficient left-shifters. The divisor coefficient is multiplied by the factors 0, 1, 2, and 3, and each of these divisor multiples is subtracted from the dividend coefficient. The divide unit then determines the largest divisor multiple, which when subtracted from the dividend coefficient, yields the smallest positive result. The divisor factor associated with this multiple is gated to the quotient catch register, the result is left-shifted two places, and another set of subtractions is performed.

The first pass results in a two-place shift, and succeeding passes result in four-place shifts (two two-place shifts per pass). Thus, a total of seven passes (five microinstruction times) is required to process the 24-bit dividend coefficient.

Quotient Assembly

The divide unit normalizes the quotient coefficient by left-shifting it one place if the highest order bit is 0. The quotient exponent is then reduced by 1 to compensate for the left-shift.

The sign bit, quotient exponent and quotient coefficient combine to form a 32-bit quantity which feeds the quotient register. The quotient register can be loaded five microinstruction times after the divide operation begins.

Square Root Unit

Figure 3-26 shows operand flow through the square root unit. Subinstruction bits load the operand into the input register, select a 12-bit or 24-bit root, and load the root register with the square root result.

Square Root Initiation

The square root operation begins with the operand being clocked from operand bus 2 into the operand input register at T0.

Root Sign Formation

The root sign is always positive. When the operand is a negative number, the square root unit provides the positive root and declares a square root error.

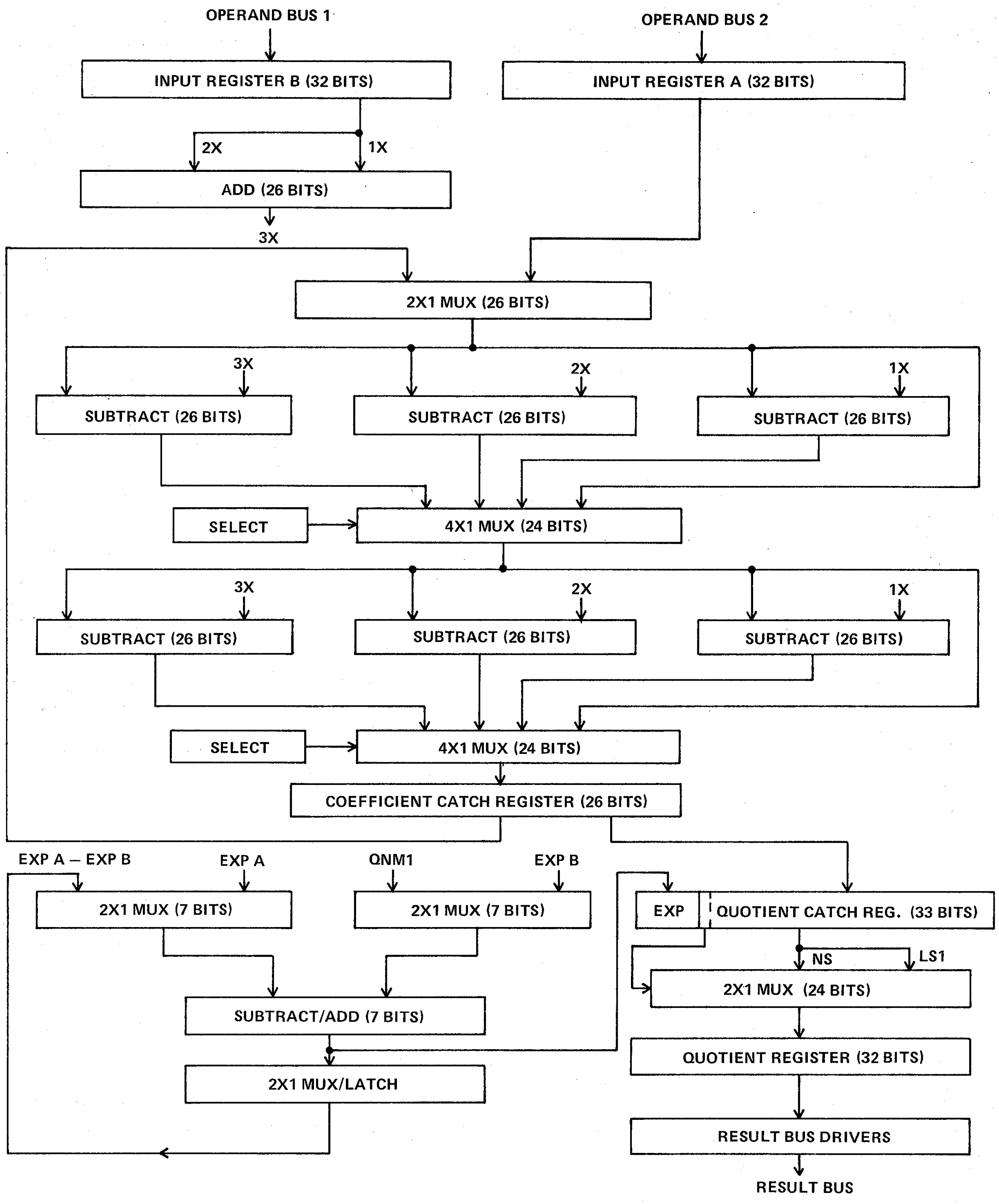


Figure 3-25. Divide Unit

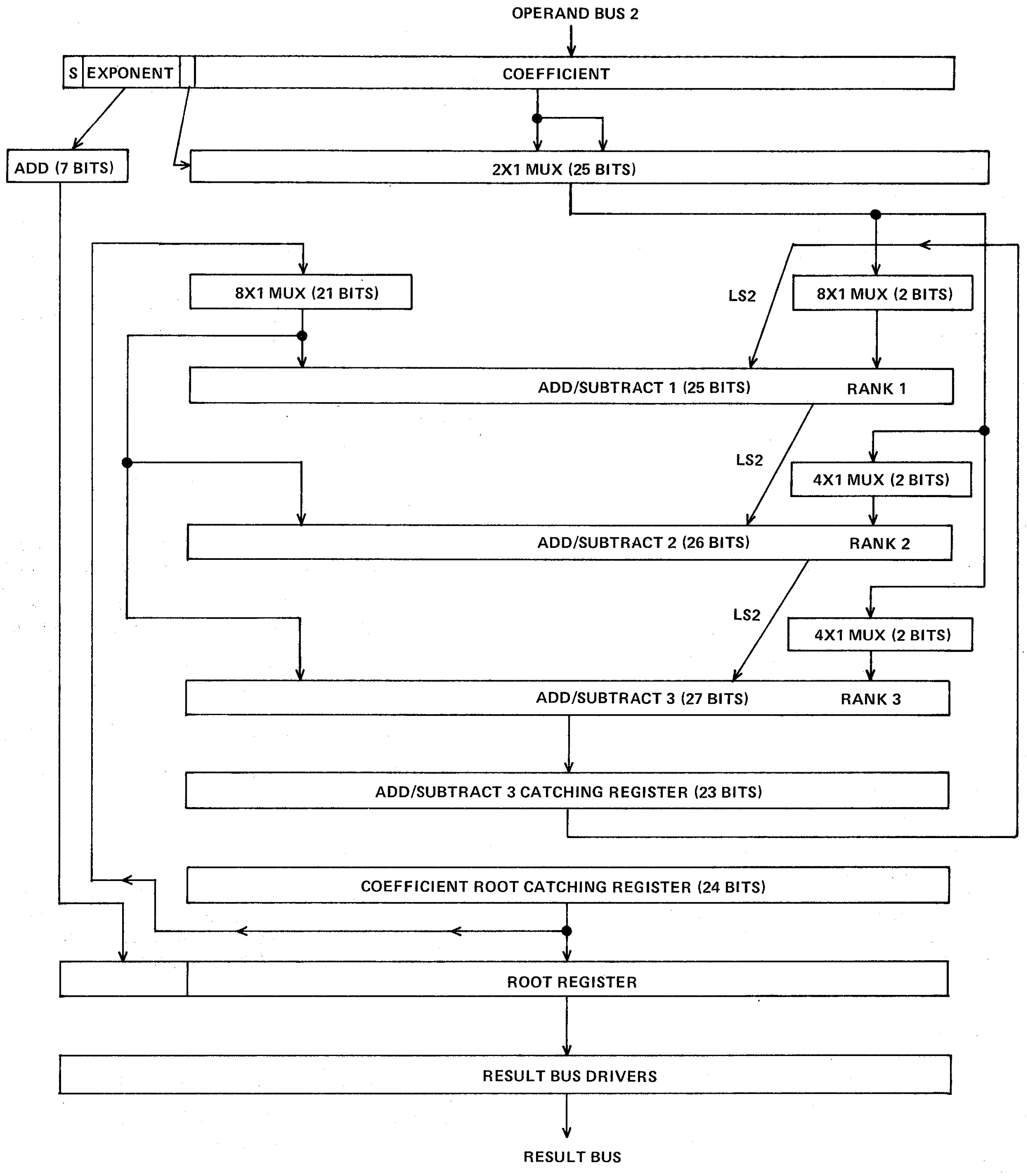


Figure 3-26. Square Root Unit

Root Exponent Generation

As figure 3-27 shows, the square root unit forms the root exponent by right-shifting the operand exponent one place (effectively dividing by 2) and extending the bias bit. To accommodate odd exponents, the lowest order bit of the exponent is added to the complete exponent after the right-shift. Thus, an operand exponent of 7_8 yields a preliminary root exponent of 4_8, 6_8 yields 3_8, 132_8 yields 155_8, and so forth. When the lowest order bit of the operand exponent is 1, the operand coefficient is right-shifted one place to accommodate the raising of the exponent after the exponent right-shift.

Root Coefficient Generation

The square root unit uses a self-restoring algorithm to generate a 12-bit root coefficient in four passes or a 24-bit root coefficient in eight passes. A subinstruction bit selects either the 12-bit mode or the 24-bit mode.

Figure 3-27 shows how the root coefficient is developed by recording carries from a series of twos complement adds and subtracts. Since the square root unit contains three add/subtract ranks, three root coefficient bits are produced for each pass and eight passes are required.

Note that in figure 3-27 the root is developed strictly by using add operations, with the addend being complemented when a subtract is required. The ALU chips used on the square root pak permit true numbers to be used throughout the operation because the chips have selectable add/subtract capability.

The self-restoring algorithm produces the square root of a coefficient as follows:

1. Partition the coefficient into bit pairs starting at the most significant bit of the coefficient. Because the coefficient is normalized, the most significant bit pair is either $1X_2$ (even exponent) or 01_2 (odd exponent caused one-place right-shift of coefficient).
2. Subtract 01_2 (add twos complement 11_2) from (to) the first bit pair and record the carry. This carry, which is the most significant bit of the root coefficient, is always present, so the root is automatically normalized.
3. Bring down the next bit pair and append it to the result of the previous operation.
4. Take the current partial root, append 01_2 (11_2 twos complement) to it, subtract (add twos complement) it from (to) the result of the previous operation, and record the carry.
5. If the carry is 1, go to step 3. If the carry is 0, continue.
6. Take the current partial root, append 11_2 to it, add it to the result of the previous operation, record the carry, and go to step 5.

Root Assembly

The root coefficient, root exponent, and a 0 sign bit combine to form a 32-bit number which feeds the root register. The root register can be loaded four microinstruction times (12-bit root coefficient) or eight microinstruction times (24-bit root coefficient) after the square root operation begins.

DATA BUSSES

MAP contains four data storage busses, two operand busses, and a result bus. Figure 3-28 shows how these busses interconnect data storage with numerical conversion and arithmetic units. Subinstruction bit fields control the quantities applied to the operand busses and the result bus.

Bus Select Logic

This logic is contained on the two bus paks and performs the following functions.

- Gates data from any data storage section to any of the three access catching registers.
- Gates data from the result bus or any access catching register to either operand bus.
- Gates data from the result bus to all three data storage sections.
- Checks parity of data from data storage.
- Generates parity for data to data storage.

Data Storage Busses

Each of these 34-bit busses transfers 32 data bits and 2 parity bits between bus select logic and data storage. There are three busses from data storage and a common bus to data storage.

Operand Busses

Each of these busses provides 32-bit operands to all arithmetic units except the square root units. Square root units and the numerical conversion unit accept operands only from operand bus 2.

Result Bus

This bus transfers 32-bit quantities from arithmetic units or the numerical conversion unit to bus select logic. From bus select logic, result bus data can be sent to data storage or gated to one or both of the operand busses under subinstruction control.

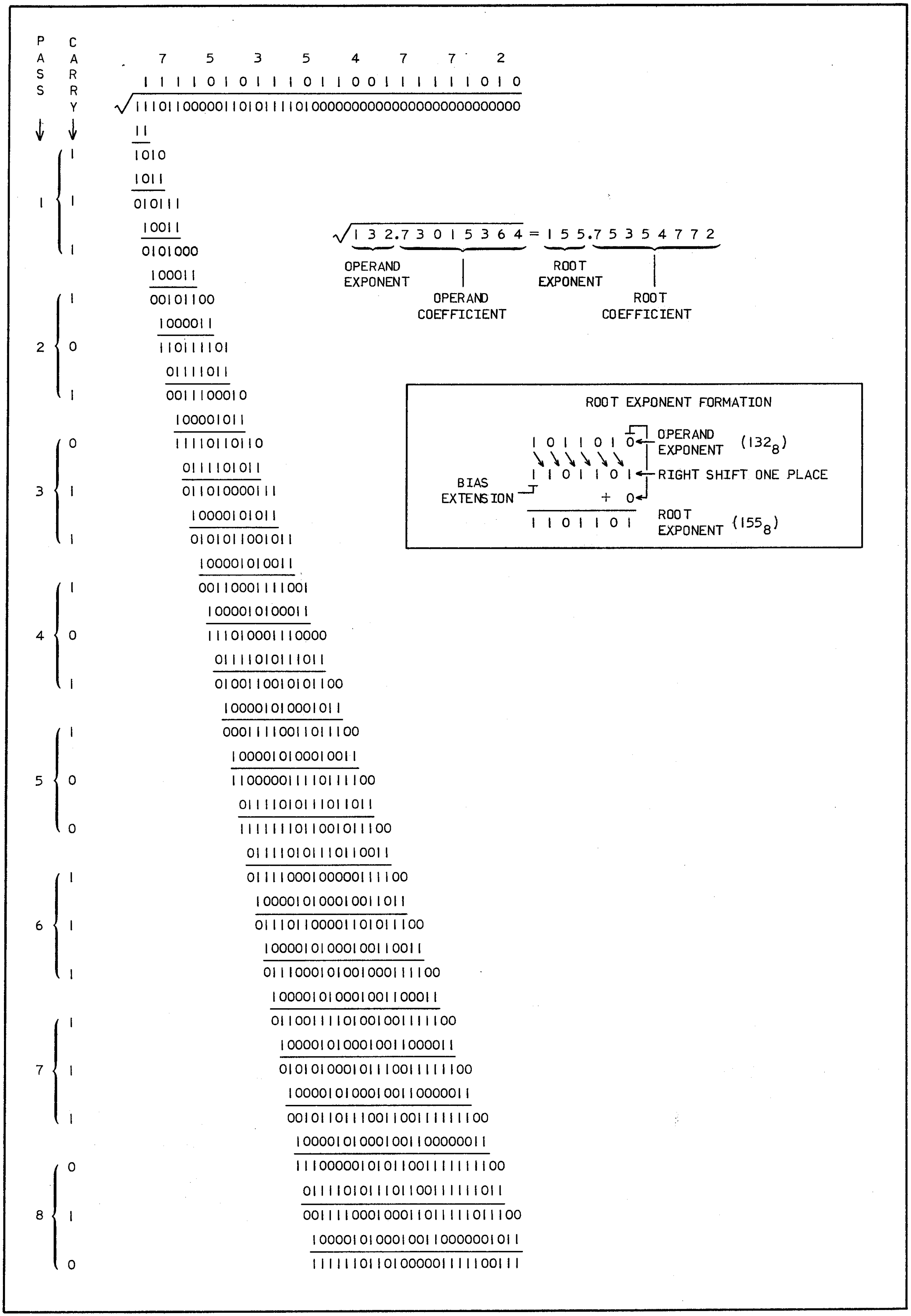


Figure 3-27. Sample Square Root Calculation

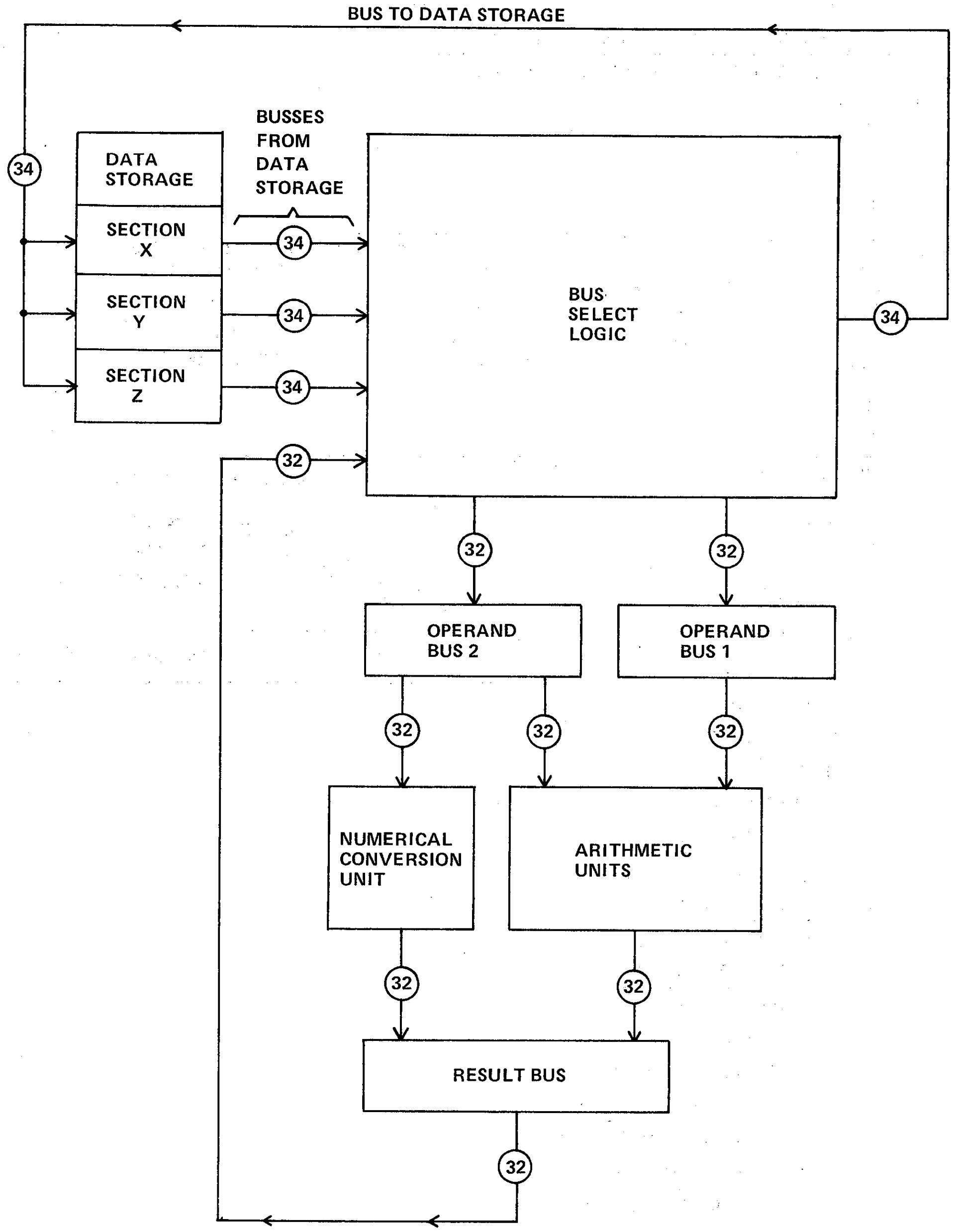


Figure 3-28. Data Busses

POWER SUPPLY/DISTRIBUTION

This subsection describes MAP power components and also describes the signal transmission methods used in MAP. Diagram sheet PWR in section 6 shows power distribution from the AC inputs to the DC bus bars. Diagram sheet REF shows the locations of power components.

Power Requirements

MAP requires both 120-volt 60-Hz, single-phase and 120/208-volt 400-Hz, 3-phase power. Table 1-2 lists the amperages required for two of the many possible MAP configurations. The 60-Hz power feeds blowers and the maintenance panel power section while 400-Hz power feeds ECL power supplies and protection circuitry.

Power Supplies

ECL Power Supplies

These high amperage supplies provide the -5.2-volt and -2.2-volt power required by ECL 10,000 series logic. When MAP has 96K or less of data storage, one 750-ampere, -5.2-volt power supply and one 400-ampere, -2.2-volt power supply occupy the lower two power supply positions in the power bay. When more data storage is added, another 750-ampere, -5.2-volt power supply and another 400-ampere, -2.2-volt power supply must be installed in the upper two power supply positions.

Each ECL power supply consists of a circuit breaker, a variable ratio transformer, a fixed transformer, and 12 diodes. Power supply A7A3 contains a meter circuit in addition to the above. The meter circuit is connected to each power bus to permit monitoring the voltage output of each ECL power supply.

Maintenance Panel Power Section

This low amperage assembly provides the -12-volt, +5-volt, and +12-volt power required by the maintenance panel and cassette transport. The maintenance panel power section consists of a multiple tap 60-Hz transformer and three modular power supplies. The transformer supplies 16-volt alternating current to the -12-volt and +12-volt supplies and 10-volt alternating current to the +5-volt supply. A cable assembly carries power from the maintenance panel power section to the maintenance panel printed circuit board.

Protection Circuitry

MAP protection circuitry detects the following abnormal conditions.

Overcurrent

Each of the ECL power supplies is protected by its own circuit breaker. When the power supply draws current in excess of the circuit breaker's rating, the breaker trips to remove 400-Hz power from the supply. The circuit breaker must be reset to the ON position to restore power to the supply.

Overtemperature

A thermostat mounted on the power supply heat sink continually monitors the temperature inside the MAP cabinet. When the temperature exceeds 130°F (55°C), the thermostat opens to drop the 400-Hz power entering the cabinet.

Airflow Loss

A vane switch monitors the airflow through each logic and memory blower. When one of these blowers fails, its vane switch opens to drop the 400-Hz power entering the cabinet.

When the 60HZ DISCONNECT switch is turned to the ON position, 400-Hz power comes up only after the last vane switch closes.

Power Distribution

Copper bus bars carry power from the ECL power supplies to the backplanes of the logic and memory bays. Individual wires carry -5.2 volts and -2.2 volts to the 180-pak power pins listed in table 3-2. The ground pins listed in table 3-2 connect to the backpanel ground plane.

TABLE 3-2. 180-PAK POWER/GROUND PINS

-5.2-Volt Pins		-2.2-Volt Pins	Ground Pins			
B004	B077	B01	B003	B022	B042	B062
B025	B097	B02	B007	B027	B047	B067
B049	B121	B143	B012	B032	B052	B071
B073	B141	B144	B017	B037	B057	B074
			B078	B098	B118	B138
			B083	B103	B123	B142
			B088	B108	B128	
			B093	B113	B133	

Figure 3-29 shows power and ground pins for a typical 180-pak chip location. When a terminator chip is to be placed in a chip location, a wire on the board carries -2.2 volts from the pin 0 position to the pin 1 position.

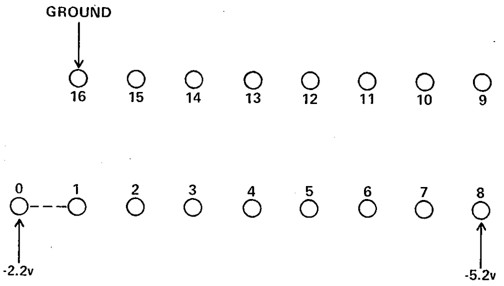


Figure 3-29. Chip Power/Ground Pins

Signal Transmission Methods

Five signal transmission methods are used in MAP. ECL single-ended and ECL bus methods are used extensively both on and between paks. Clock signals and signals between the control unit and the maintenance panel are transmitted differentially to increase noise immunity. The TTL single-ended method is used to interconnect maintenance panel components. Finally, a coaxial method is used to transmit signals between MAP and the host computer.

Table 3-3 provides nominal logic voltages for each signal transmission method.

TABLE 3-3. LOGIC VOLTAGE LEVELS

Signal Transmission Method	Nominal High Voltage	Nominal Low Voltage
ECL single-ended	-0.92 volts	-1.75 volts
ECL bus	-0.92 volts	-2.10 volts
Differential	-0.92 volts	-1.75 volts
TTL single-ended	+3.3 volts	+0.2 volts
Coaxial	+1.8 volts	-1.8 volts

ECL Single-Ended Method

Figure 3-30 shows a typical ECL single-ended transmission line. Note that the terminating resistor is always placed at the end of the transmission line. Approximately nine ECL loads can be driven off an ECL single-ended transmission line without a significant increase in propagation delay time.

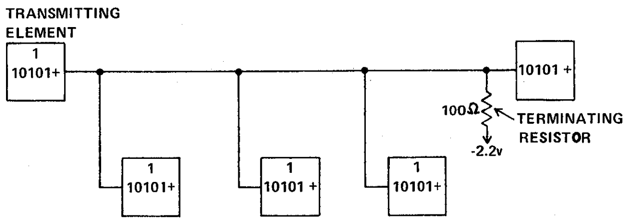


Figure 3-30. ECL Single-Ended Transmission Method

ECL Bus Method

Figure 3-31 shows a typical ECS bus. Note that terminating resistors are placed at both ends of the bus transmission line to eliminate reflections. The high output impedance of the bus drivers causes the bus to approach closely the -2.2-volt terminating voltage when all bus driver outputs are low. When a bus driver goes high, the bus assumes the nominal ECL high voltage of- -0.92 volt.

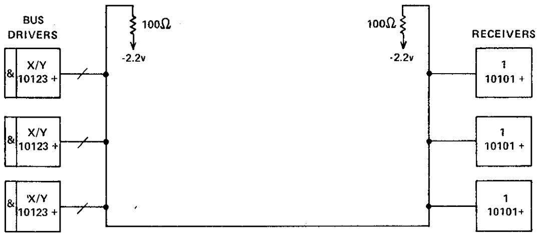


Figure 3-31. ECL Bus Transmission Method

Differential Methods

Figure 3-32 shows the three differential transmission methods used in MAP. All of the differential methods require terminating resistors at the end of the transmission line.

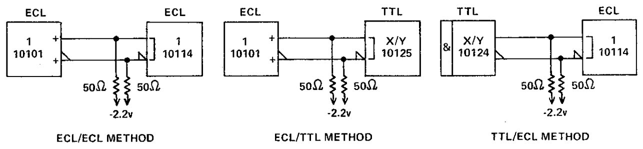


Figure 3-32. Differential Transmission Methods

TTL Single-Ended Method

This method is similar to the ECL single-ended method except that the collector resistor is often included in the transmitting element. Figure 3-33 shows both normal and open collector TTL configurations.

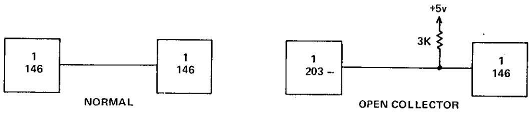


Figure 3-33. TTL Single-Ended Transmission Method

Coaxial Method

This method is used to transmit 25-nanosecond pulses between MAP and a PPU or MAP and ECS. Refer to the input/output specifications manual listed in the preface for information pertaining to the coaxial transmission method.

COMMENT SHEET

MANUAL TITLE CDC AE121-B/AE125-B Matrix Algorithm Processor
Volume 1 Hardware Maintenance Manual

PUBLICATION NO. 60429103 REVISION D

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 Revision Record page at the front of the manual. Customer engineers are urged to use the TAR.

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

FOLD ON DOTTED LINES AND STAPLE

CUT ALONG LINE

PRINTED IN U.S.A.

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Publications and Graphics Division
ARH219
4201 North Lexington Avenue
Saint Paul, Minnesota 55112

CUT ALONG LINE

FOLD

FOLD