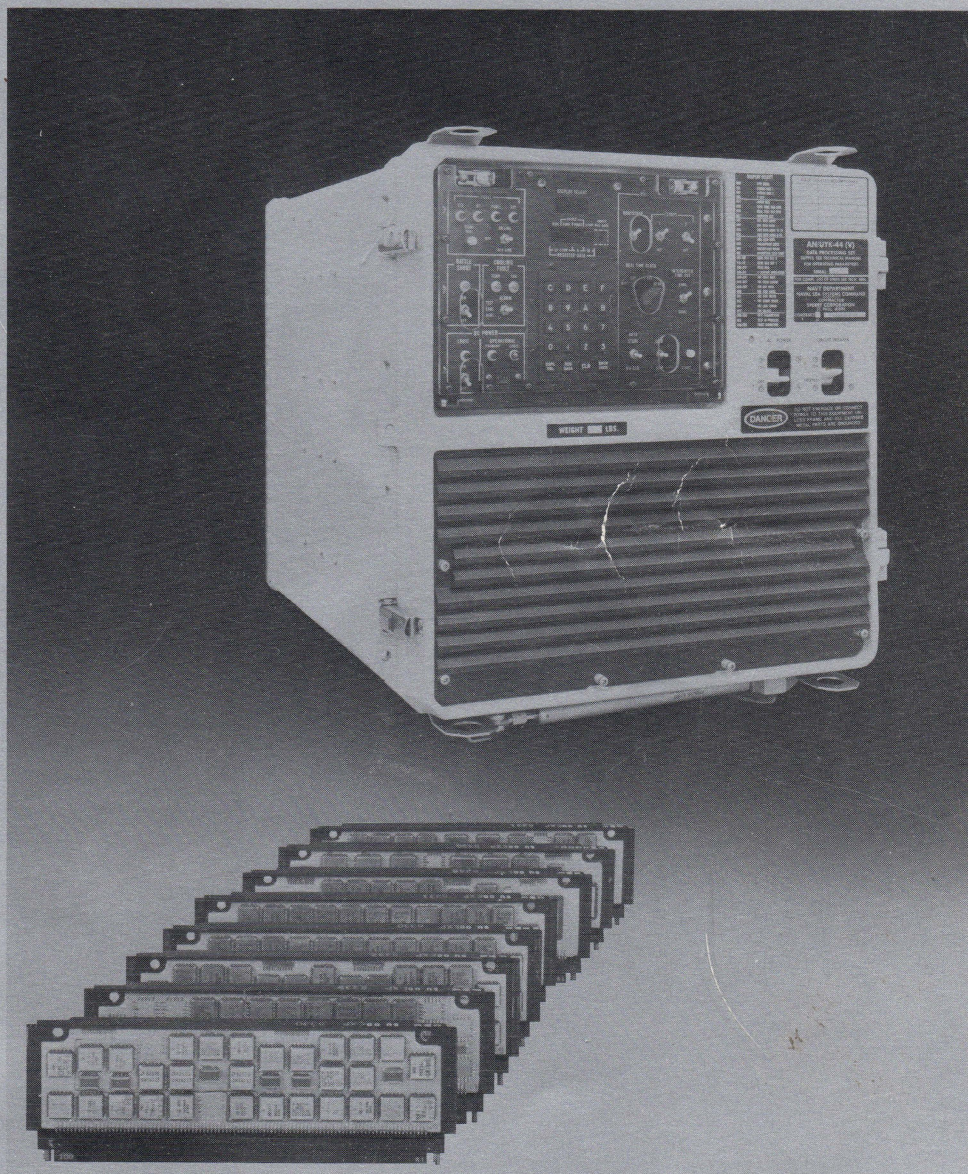
Technical Description

UNISYS



AN/UYK-44

Technical Description

November 1987

UNISYS

DISCLAIMER NOTICE

This technical description is intended to inform the reader of the operational characteristics, capabilities, features, and construction of the OL-335(V)/U Data Processing Group (Militarized Reconfigurable Processor (MRP)), the AN/UYK-44(V)/U Data Processing Set (Militarized Reconfigurable Computer (MRC)) and the AN/UYK-44(V) Microprocessor Development System (MDS).

This description is not an equipment specification nor is it warranted for accuracy or completeness for procurement purposes.

TABLE OF CONTENTS

Title	Page
THE AN/UYK-44 MILITARIZED RECONFIGURABLE PROCESSOR AND COMPUTER	1
INTRODUCTION	1
OL-335(V)/U DATA PROCESSING GROUP (MILITARIZED RECONFIGURABLE PROCESSOR) SPECIFICATIONS AND FEATURES	3
AN/UYK-44(V)/U DATA PROCESSING SET (MILITARIZED RECONFIGURABLE COMPUTER) SPECIFICATIONS AND FEATURES	7
AN/UYK-44(V) MICROPROCESSOR DEVELOPMENT SYSTEM SPECIFICATIONS AND FEATURES	11
AN EXCELLENT LONG-TERM INVESTMENT	12
REALTIME APPLICATIONS	12
Shipboard Defense Systems Applications	12
Communications Systems	15
Signal-Processing Systems	15
Control Systems	15
Other MRP/MRC Applications	15
MODULAR ARCHITECTURE - MRP AND MRC	15
FUNCTIONAL ARCHITECTURE - MRP AND MRC	15
Main Memory - MRC	16
Memory Interface - MRP and MRC	16
Semiconductor Memory on SEM - MRP	16
Direct Memory Access (DMA) - MRC	18
User Growth - MRP and MRC	18
Memory Address Allocations - MRP and MRC	18
Memory Addressing - MRP and MRC	18
Nondestructive Read-only (NDRO) Memory - MRP and MRC	18
Memory Address Expansion (MAE) - MRP and MRC	18
Input/Output - MRP and MRC	21
Memory Mapped I/O (MMIO) - MRP	25
CP Executive/Task Modes - MRP and MRC	27
CP General Registers - MRP and MRC	27
Status Registers - MRP and MRC	27
Realtime Clock (RTC) and Monitor Clock - MRP and MRC	27
Breakpoint - MRP and MRC	29
Control and Maintenance Interface (CMI) - MRP and MRC	29
Remote Control and Maintenance Interface (RCMI) - MRP and MRC	29
Memory Protection - MRP and MRC	29
Execute Lockout	29
Write Lockout	31
Read Lockout	31
Instruction Counter (P) History File - MRC	31
Power Failure Protection - MRC	31
Temperature Sensor - MRP and MRC Expansion Cabinet	31
CONSTRUCTION - MRP AND MRC	31

TABLE OF CONTENTS (continued)

Title	Page
MAINTAINABILITY – MRP AND MRC	32
Maintainability	32
FUNCTIONAL OPERATION – MRP AND MRC	38
Status Control	38
Instructions	39
Math Pac – MRP and MRC	40
Instruction Addressing	40
Instruction Word Formats	41
Single–Length Operands	41
Double–Length Operands	41
Floating–Point Operands	41
Operand Addressing	42
Interrupts	43
Interrupt Processing	44
I/O Operation	45
I/O Channel Status	45
MIL–STD–1397 Parallel Input Interface Communication	45
MIL–STD–1397 Parallel Output Interface Communication	45
MIL–STD–1397 Parallel Intercomputer Communication	45
MIL–STD–1397 Peripheral Input Channel	45
MIL–STD–188C, VACALES, and EIA–STD–RS–232–C Serial Channel Communication	47
MIL–STD–1397 Type D Serial Channel Communication	47
MIL–STD–1397 Type E Serial Channel Communication	47
NATO–STD–4156 (Modified) Serial Channel Communication	47
MIL–STD–1553B Serial Channel Communication	47
IOC Command Instruction	48
Program Chaining	48
Serial Channel Interrupts	49
Externally Specified Addressing (ESA)	49
Dual Channel	49
Master Clear	49
Control and Maintenance Panel (CMP) – MRC/Expansion Cabinet	50
MICROPROCESSOR DEVELOPMENT SYSTEM (MDS)	59
OL–335(V)/U DATA PROCESSING GROUP EMBEDDING	61
TRAINING	63
Navy Conducted Training	63
Unisys Conducted Training	63
SUPPORT SOFTWARE	64
MACHINE TRANSFERRABLE SUPPORT SOFTWARE (MTASS)	64
Concurrent Use	64
Transportability	64

TABLE OF CONTENTS (continued)

Title	Page
Adaptability	64
Project Control	64
Features	64
MACHINE TRANSFERRABLE SUPPORT SOFTWARE/MINICOMPUTER (MTASS/M)	65
Concurrent Use	65
Transportability	65
Adaptability	65
Project Control	65
Features	65
PROGRAMMER ORIENTED AN/UYK-44 LINK (PORTAL/44)	65
Compatibility	65
Efficiency	66
Realtime Checkout	66
Availability	66
Preserve Environment	66
Ease of Use	66
Symbolic Debug	66
Features	66
LEVEL 2	66
Features	66
Standard Executive/20 (SDEX/20)	66
Features	67
Standard Executive/Minicomputer (SDEX/M)	67
Features	68
ACRONYMS AND ABBREVIATIONS	69

APPENDIX A
REPERTOIRE OF INSTRUCTIONS

APPENDIX B
CP INSTRUCTION EXECUTION TIMES

APPENDIX C
APPLICABLE DOCUMENTS

LIST OF ILLUSTRATIONS

Figure	Title	Page
1	SEM, Format B, 2A Size	1
2	MRP Configuration Items	2
3	MRC Configuration Items	6
4	MDS Configuration Items	10
5	MRC Basic Cabinet, Front and Back Views	13
6	MRC Expansion Cabinet, Front and Back Views	14
7	MRP/MRC Functional Architecture Block Diagram	17
8	Direct Memory Access Interface	19
9	Memory Address Generation	19
10	Status Register 1 Format	22
11	MIL-STD-1397 Parallel Interface	23
12	MIL-STD-188C and VACALES Serial Interface	23
13	EIA-STD-RS-232-C Serial Interface	23
14	MIL-STD-1397 Type D Serial Interface	24
15	MIL-STD-1397 Type E Serial Interface	24
16	NATO-STD-4156 (Modified) Serial Interface	24
17	MIL-STD-1553B Interface	25
18	MMIO Control and Status Register	28
19	Status Register 2 Format	29
20	RCMI Serial Interface	30
21	Semiconductor Memory (6 x 9-inch) Status Register	37
22	SEM Semiconductor Memory Status Register	38
23	Instruction Word Format	40
24	Indirect Word Interpretation	42
25	Interrupt Entrance Address Index	45
26	I/O Status Word	46
27	Control and Maintenance Panels (CMPs)	51
28	Microprocessor Development System (MDS)	60

LIST OF TABLES

Table	Title	Page
1	External Connector/Cable Data	20
2	Assigned Memory Addresses (Memory Map)	21
3	I/O Physical Data And Channel Data Rate	26
4	Module Key Codes, Power, And Failure Rates Per Million Hours	33
5	Typical SEM Group Failure Rates And Power Requirements	34
6	Microcode BIT Tests	36
7	Macroexecutable BIT Subtests	37
8	Condition Code Indications From Status Register 1	38
9	Interrupt Priority	44
10	IOC Instruction Repertoire	46
11	CMP Controls And Indicators	52
12	Information Displayed By Display Select	56

THE AN/UYK-44 MILITARIZED RECONFIGURABLE PROCESSOR AND COMPUTER

INTRODUCTION

The AN/UYK-44 is available in three basic configurations:

- **The Militarized Reconfigurable Processor (MRP)** is a set of hardware modules consisting of a central processor (CP) with several options, input/output controller (IOC) with a choice of optional channel adapters and a memory interface (MI) option. The MRP performs the AN/UYK-20(V) and specified AN/AYK-14(V) processing and input/output (I/O) functions when the modules are interconnected, have power applied, and have memory available. The CP performs up to 900,000 instructions per second (900 KIPS). Processor speed and throughput is a function of memory cycle and access times. The military nomenclature for the MRP is OL-335(V)/U Data Processing Group.

- **The Militarized Reconfigurable Computer (MRC)** is a set of MRP hardware modules contained in an air-cooled cabinet with a control and maintenance panel, memory, and power supplies. This configuration is a self-contained computer. The MRC is expanded by interconnecting an expansion cabinet to the primary cabinet. The expansion cabinet houses additional I/O. The MRC performs the AN/UYK-20(V) and specified AN/AYK-14(V) processing and I/O functions without requiring additional hardware. The military nomenclature for the MRC is AN/UYK-44(V)/U data processing set (DPS). The military nomenclature for the MRC expansion cabinet is OF-174/UYK-44(V)/U Expansion Adapter Group.

- **The Microprocessor Development System (MDS)** is a set of MRP hardware modules housed, along with memory and power supplies, in commercial-quality cabinets, and a keyboard-display. This configuration is used for standalone operation, test, and software development.

The MRP and MRC are designed to meet the requirements of small- and medium-sized applications in shipboard, mobile shelter, or other severe environments. The MRP can be embedded in a variety of cabinet form factors including an air transportable rack (ATR) for airborne applications.

The basic MRP/MRC modules, shown in Figure 1, are standard electronic modules (SEMs), Format B, 2A size, developed in accordance with MIL-M-28787 (General Specification for Standard Electronic Modules Program), MIL-STD-1378 (Requirements for Employing Standard Electronic Modules), and MIL-STD-1389 (Design Requirements for Standard Electronic Modules).

The basic and optional configuration items for the MRP, the MRC, and the MDS are defined in Figures 2, 3, and 4.

The specifications and features of the MRC, MRP, and MDS are included after the MRC, MRP, and MDS configuration items (see Figures 2, 3, and 4).

Each SEM is 1.85 inches high by 5.74 inches wide by 0.29 inches thick and weighs less than 0.25 pound maximum. Each SEM has a cooling rib on each end, a cooling fin on the top, and a 100-pin plug connector on the bottom. Each SEM has three cooling modes: convection or direct air impingement; conduction cooled through the side ribs; and conduction cooled through the top rib.

The MRP SEM set is embedded or plugged into military system hardware to perform system processing and I/O functions using the AN/UYK-20(V) and specified AN/AYK-14(V) instruction set architecture (ISA).

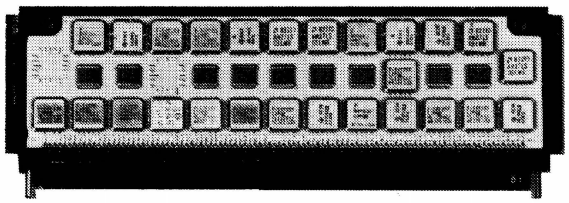
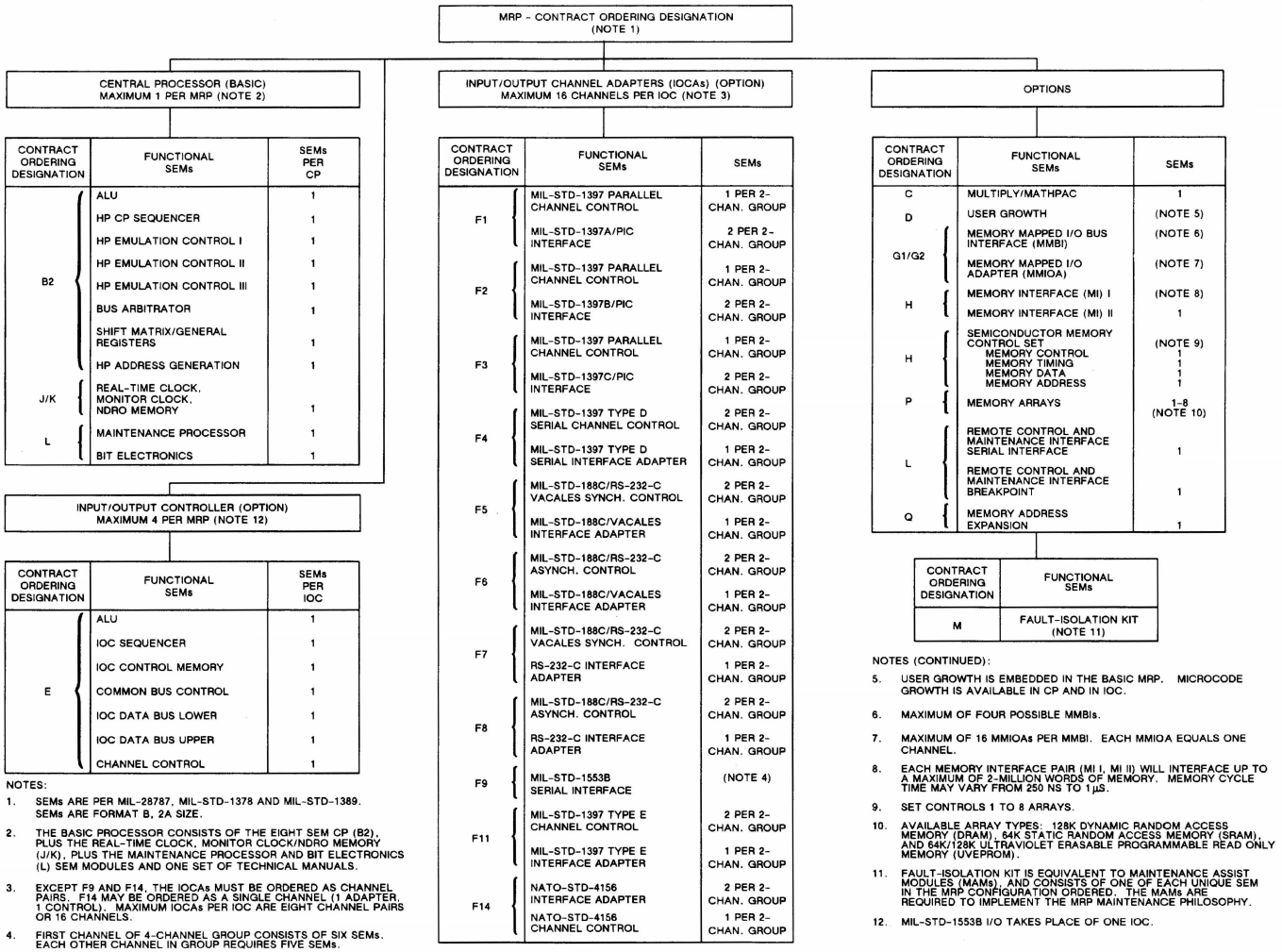


Figure 1. SEM, Format B, 2A Size



NOTES:

1. SEMs ARE PER MIL-28787, MIL-STD-1378 AND MIL-STD-1389. SEMs ARE FORMAT B, 2A SIZE.
2. THE BASIC PROCESSOR CONSISTS OF THE EIGHT SEM CP (B2), PLUS THE REAL-TIME CLOCK, MONITOR CLOCK/NDRO MEMORY (J/K), PLUS THE MAINTENANCE PROCESSOR AND BIT ELECTRONICS (L) SEM MODULES AND ONE SET OF TECHNICAL MANUALS.
3. EXCEPT F9 AND F14, THE IOCAs MUST BE ORDERED AS CHANNEL PAIRS. F14 MAY BE ORDERED AS A SINGLE CHANNEL (1 ADAPTER, 1 CONTROL). MAXIMUM IOCAs PER IOC ARE EIGHT CHANNEL PAIRS OR 16 CHANNELS.
4. FIRST CHANNEL OF 4-CHANNEL GROUP CONSISTS OF SIX SEMs. EACH OTHER CHANNEL IN GROUP REQUIRES FIVE SEMs.

NOTES (CONTINUED):

5. USER GROWTH IS EMBEDDED IN THE BASIC MRP. MICROCODE GROWTH IS AVAILABLE IN CP AND IN IOC.
6. MAXIMUM OF FOUR POSSIBLE MMBIs.
7. MAXIMUM OF 16 MMIOAs PER MMBI. EACH MMIOA EQUALS ONE CHANNEL.
8. EACH MEMORY INTERFACE PAIR (MI I, MI II) WILL INTERFACE UP TO A MAXIMUM OF 2-MILLION WORDS OF MEMORY. MEMORY CYCLE TIME MAY VARY FROM 250 NS TO 1 µS.
9. SET CONTROLS 1 TO 8 ARRAYS.
10. AVAILABLE ARRAY TYPES: 128K DYNAMIC RANDOM ACCESS MEMORY (DRAM), 64K STATIC RANDOM ACCESS MEMORY (SRAM), AND 64K/128K ULTRAVIOLET ERASABLE PROGRAMMABLE READ ONLY MEMORY (UVEPROM).
11. FAULT-ISOLATION KIT IS EQUIVALENT TO MAINTENANCE ASSIST MODULES (MAMs), AND CONSISTS OF ONE OF EACH UNIQUE SEM IN THE MRP CONFIGURATION ORDERED. THE MAMs ARE REQUIRED TO IMPLEMENT THE MRP MAINTENANCE PHILOSOPHY.
12. MIL-STD-1553B I/O TAKES PLACE OF ONE IOC.

Figure 2. MRP Configuration Items

OL-335(V)/U DATA PROCESSING GROUP (MILITARIZED RECONFIGURABLE PROCESSOR) SPECIFICATIONS AND FEATURES

GENERAL CHARACTERISTICS

Militarized construction: Format B, 2A, Standard Electronic Modules(SEMs) per MIL-M-28787, MIL-STD-1378, and MIL-STD-1389
General-purpose, 16-bit digital dual-state processor with options
Emulates AN/UYK-20(V) data processing sets and specified AN/AYK-14(V) standard airborne computer functions
Physically and functionally modular and expandable
Small Scale Integrated (SSI), Medium Scale Integrated (MSI), and Large Scale Integrated (LSI) components on Format B, 2A size SEMs, with 100-pin plug connector
Microprogrammed control structure
Plug-in options
Temperature classification: MIL-M-28787 and MIL-STD-1389; Module Class II
Operating and nonoperating temperature:
Fin/rib temperature of -55°C to +85°C
Convection and conduction cooled
Altitude: Operation at up to 21,335 meters (70,000 feet)
Humidity: 95 percent maximum
Voltages: +5, +15, -5.2, and -15 vdc
Power: 8 watts maximum per 2A SEM
5.15 watts average per 2A SEM
SEM size:
Height: 1.85 inches maximum
Width: 5.74 inches maximum
Thickness: 0.29 inch maximum
SEM weight: 0.25 pound maximum
3.25 ounces average per 2A SEM

CENTRAL PROCESSOR (CP)

Two's complement arithmetic
4-, 8-, 16-, and 32-bit operands
Two sets of 16 high-speed, general-purpose registers
Two execution modes: executive and task
Two program status registers (SRs)
3-level interrupt processing (hardware serviced)
16- and 32-bit instructions - five formats
Direct addressing to 65,536 (64K) words
Indexing by general registers
Cascaded indirect addressing
One 16-bit breakpoint register
8-word instruction counter (P) history file
Microcoded Built-In Test (BIT) firmware

Features of the MRP provide functional adaptability for many applications requirements. Available options increase its capacity, enhance its flexibility, and provide functions required by certain applications. The options defined below can be selected and added by plugging the required SEMs into the system.

USER GROWTH IN CP AND IOC

Customer-defined micromemory for CP and IOC

CP - Op codes: 670, 671, 672, octal; DC, DD, DE, hex, reserved for customer-defined microcode in CP
IOC - Op codes: 720, 721, octal; E8, E9, hex, reserved for customer-defined microcode in IOC

CONTROL AND MAINTENANCE INTERFACE (CMI)

Features microprocessor-driven BIT for self-test and provides defined interface to a user-defined operator/maintenance panel.

MEMORY ADDRESS EXPANSION (MAE)

Addressing to 4,194,304 (4096K) words
Four sets of 64 page registers and 1024 word pages
Execute, write, and read memory protection
Memory management instructions

MATH PAC

Square root
Trigonometric and hyperbolic vector and rotate
Fixed/floating-point conversion
Floating-point arithmetic, compare, normalize, square root, trigonometric, exponential, and natural logarithm
Double-precision multiply and divide
Algebraic left and right quadruple shifts

NDRO MEMORY AND CLOCKS

192-word bootstrap
32-bit realtime clock (RTC)
16-bit monitor clock
1 kHz, 32 kHz, or external clock rate
4K words of macro-BIT (I/O options)

I/O CONTROLLERS (IOC)

One to four I/O controllers (MIL-STD-1553B I/O takes place of one IOC)
Independent operation
Up to eight 2-channel groups (serial and parallel) per I/O controller (16 channels)
I/O instruction repertoire - same format as CP
Control memory for each I/O channel
Optional memory address expansion to 4,194,304 (4096K) words

I/O CHANNEL ADAPTERS (IOCAs)

Parallel Channels

Implemented as 2-channel group
MIL-STD-1397, Type A/PIC (NTDS Slow)
MIL-STD-1397, Type B/PIC (NTDS Fast)
MIL-STD-1397, Type C/PIC (ANEW)

Modes:

- 8-bit byte, 16-bit word, or 32-bit dual-channel transfers
- Dual-channel operations on two channels having the same type interface
- Normal transfers available on single- or dual-channels
- Externally specified addressing (ESA) operation on dual channels
- Intercomputer operation on single- or dual-channels

Serial Channels

MIL-STD-188C:

- Implemented as 2-channel group
- Synchronous - to 9600 baud
- Asynchronous - 50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 4800, or 9600 baud (program-controlled selection)
- Character size - 5-, 6-, 7-, or 8-bit (program-controlled selection)

EIA-STD-RS-232-C:

- Implemented as 2-channel group
- Synchronous - to 9600 baud
- Asynchronous - 50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 4800, or 9600 baud (program-controlled selection)
- Character size - 5-, 6-, 7-, or 8-bit (program-controlled selection)

MIL-STD-1397 Type D:

- Implemented as 2-channel group
- 8-, 16- or 32-bit transfers
- Asynchronous - 180,000 words per second
- Intercomputer or normal operation

MIL-STD-1397 Type E:

- Implemented as 2-channel group
- 8-, 16- or 32-bit transfers
- Asynchronous - 180,000 words per second
- Intercomputer or normal operation

VACALES (Variable Character Length Synchronous):

- Implemented as 2-channel group
- Synchronous - to 32,000 baud
- Character size -1 to 16 bits (program-controlled selection)
- MIL-STD-188C interface signals

NATO-STD-4156 (modified):

- Implemented as 1- or 2-channel group
- Synchronous - 1 to 3 megabits per second

MIL-STD-1553B:

- Compatible with MIL-STD-1553B
- Bus controller and remote terminal (RT) capability
- Bus controller mode supports up to 32 remote terminals
- Broadcast capability
- Dual redundancy
- Transformer-coupled to MIL-STD-1553B Bus
- Transfers data into or out of up to 4M words of addressable memory
- Interfaces using Data Device Corporation (DDC) SEMs:
  - Key Code UKS (67003 Bus Control)
  - Key Code UKT (67001 Bus Driver)
  - Key Code UKU (67002 Remote Terminal)
- No IOC required - interfaces directly to common bus

MEMORY INTERFACE (MI)

- Interfaces a maximum of 256K words of magnetic core or 2M words of semiconductor memory or any mix (up to 1M) to the common bus (not used with semiconductor memory on SEM)
- Cycle times may range from 250 ns To 1 μs
- Modularity of memory may be 64K (core), 128K, 256K or 512K (semiconductor)
- Maximum of four memory modules per MI

MEMORY MAPPED I/O (MMIO)

- Up to 64 channels per MRP
- Provides communication channel between CP and an external equipment under control of the external equipment
- Up to 16 Memory Mapped Input/Output Adapter (MMIOA) channels per Memory Mapped Bus Interface (MMBI)
- Up to four MMBIs per MRP

RCMI SERIAL INTERFACE

- Allows control of MRP by computer, terminal or maintenance panel
- Independent option of MRP
- Independent of other maintenance processor (MP) functions
- Synchronous or asynchronous RS-422/423 2-channel operation
- Selectable baud rate to 19.2 kilobaud

RCMI BREAKPOINT

- Contains four relative and four absolute breakpoints
- Up to four relative ranging breakpoints (any two relative breakpoints compose a ranging breakpoint)
- Up to four checkpoints available
- Checkpoint data queued in FIFO for access by MP
- Checkpoints indicated via RCMI auto status
- Checkpoint trace feature
- Four RCMI user-definable flags indicated via auto status
- Four hardware status discretes

SEM SEMICONDUCTOR MEMORY

AN/UYK-44 compatible - interfaces directly to common bus (MI not required)
Hardware refresh and scrub
Single-bit correct, double-error detect
Noncorrectable error detection
Correctable and noncorrectable error logs
Interface to NDRO memory
Expansion to 1M words provided
Supports access times of 140 to 490 ns
Control set supports eight array modules
BIT provided to achieve 95-percent testability
Supports simultaneous use of different memory types

Available types and characteristics are:

128K dynamic random access memory (DRAM):

(1) Processing speed (CP throughput) up to 900,000 instructions per second (900 KIPS)
Access time of 190 ns
Read/modify/write interface to MRP

64K static random access memory (SRAM):

(1) Processing speed (CP throughput) up to 930 KIPS
Access time of 140 ns
Read/modify/write interface to MRP
Can retain data with battery backup

64K/128K ultraviolet erasable programmable read-only memory (UVEPROM):

(1) Processing speed (CP throughput) up to 800 KIPS
Access time of 290 ns
Retains data without power applied
Easily field-programmable using adapter box
Stored operational program can eliminate slow external load device

NOTES: (1) These throughputs were measured using NOSC I/O Thru 4242, based on NOSC 424-LP-6222 (revised) instruction mix.

(2) CP throughput versus I/O throughput for various array access times is defined in Figure A of Appendix B.

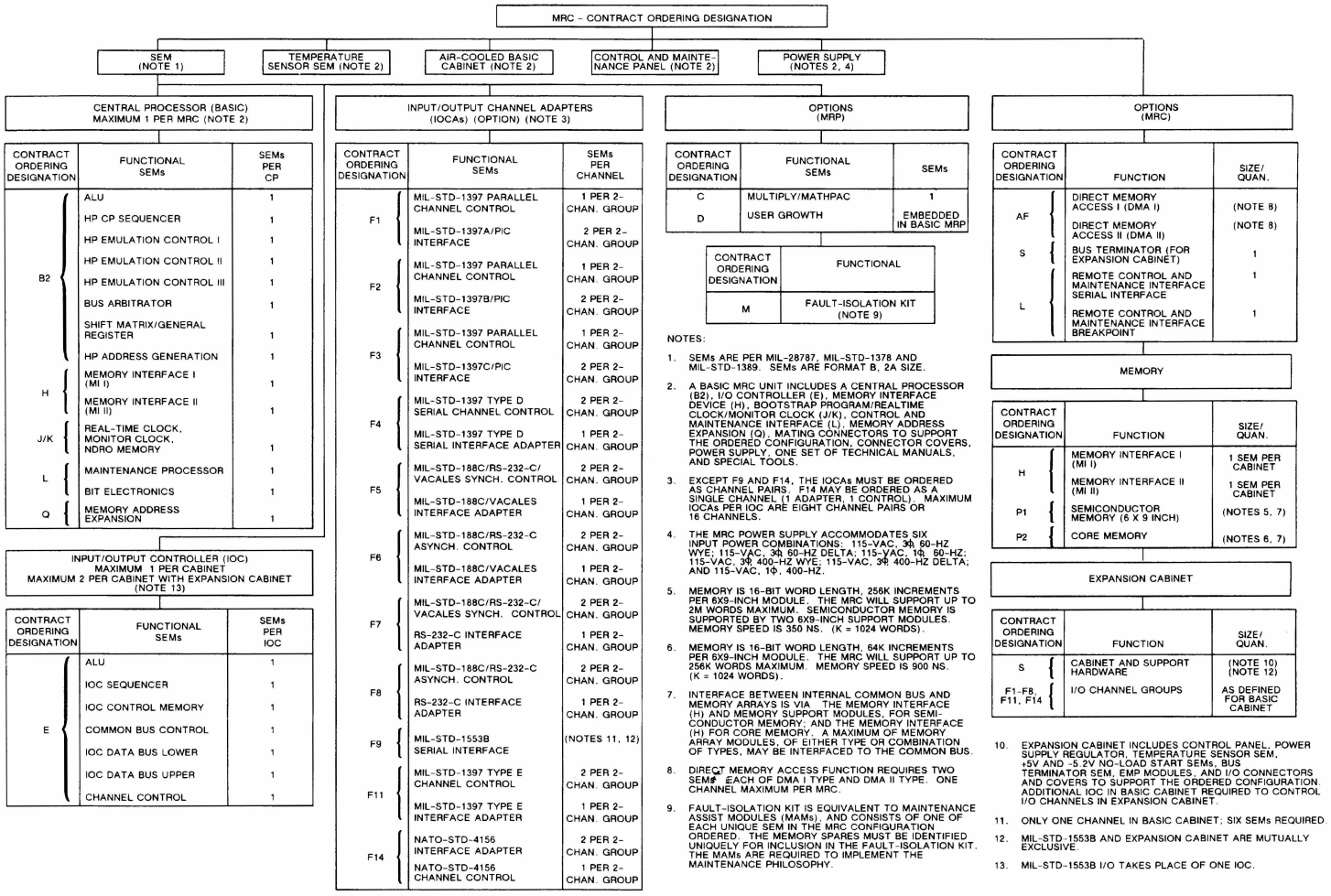


Figure 3. MRC Configuration Items

AN/UYK-44(V)/U DATA PROCESSING SET (MILITARIZED RECONFIGURABLE COMPUTER) SPECIFICATIONS AND FEATURES

GENERAL CHARACTERISTICS

Militarized construction: MIL-E-16400, MIL-STD-1378, and MIL-STD-1389
General-purpose, 16-bit dual-state digital computer
Emulates AN/UYK-20(V) and AN/UYK-20A(V) data processing sets and specified AN/AYK-14(V) standard airborne computer functions
Physically and functionally modular and expandable
One basic cabinet and one expansion cabinet
SSI, MSI, and LSI components on Format B, 2A size SEMs
Microprogrammed control structure
Plug-in options
Temperature range: MIL-E-16400 Range 1 (temperature only)
- Operating: -54°C to +65°C
- Nonoperating: -62°C to +71°C

Other environments: MIL-E-16400 Range 4
Air cooled
Power: 750 watts maximum per cabinet

BASIC CABINET

Free-standing base mount or 19-inch, rack-mountable
Front access for maintainability
Connectors on back
Control and maintenance panel on front
Integral blowers and power supplies
Overtemperature warning, shutdown, and override
Manufacturing test connector on rear panel
Capacity for (maximum):
- 1 central processor
- 1 or 2 I/O controllers
- 16 I/O channel adapters (16 channels)
- 256K words of magnetic core memory, or 2M words of semiconductor memory, or combination

Size, maximum:
- Height: 50.80 cm (20 in.)
- Width: 48.26 cm (19 in.)
- Depth: 60.96 cm (24 in.)

Weight, maximum module configuration: 190 lbs. maximum
Conforms to submarine installation requirements (pass through 25-inch hatch without disassembly)
Transient pulse protection on all external connectors

CENTRAL PROCESSOR

Two's complement arithmetic
4-, 8-, 16-, and 32-bit operands
Two sets of 16 high-speed, general-purpose registers
Two execution modes: executive and task
Two program status registers
3-level interrupt processing (hardware serviced)
16- and 32-bit instructions - five formats
Direct addressing to 65,536 (65K) words
Indexing by general registers
Cascaded indirect addressing
One 16-bit breakpoint register
8-word instruction counter (P) history file
Power fault/auto restart
Control and maintenance interface
Microcoded BIT firmware

POWER SUPPLIES

A single power conditioner to accommodate six input power options:
- 3-phase wye, 208 vac, 60 or 400 Hz
- 3-phase delta, 115 vac, 60 or 400 Hz
- 1-phase, 115 vac, 60 or 400 Hz

20-percent reserve for power demand increase (future growth)
Provides +5, +15, -5.2, -15, and -12 vdc

The MRC provides functional adaptability for many application requirements. Available options increase capacity, enhance flexibility, and provide functions required by certain applications. The options defined below can be selected and added without wiring or cabinet changes by plugging the required modules into the basic cabinet or expansion cabinet.

EXPANSION CABINET

Free-standing base mount or 19-inch, rack-mountable
Control panel (subset of basic cabinet controls/indicators)
Capacity for (maximum):
- 16 I/O channel adapters (16 I/O channels)
- Controlling IOC (IOC1) in MRC basic cabinet
- Power supply regulator
- Conditioned power supplied from basic cabinet

USER GROWTH IN CP AND IOC

Customer-defined micromemory for CP and IOC

- CP - Op codes: 670, 671, 672, octal; DC, DD, DE, hex, reserved for customer-defined microcode in CP
- IOC - Op codes: 720, 721, octal; E8, E9, hex, reserved for customer-defined microcode in IOC

MEMORY ADDRESS EXPANSION

Addressing to 4,194,304 (4096K) words
Four sets of 64 page registers and 1024 word pages
Execute, write, and read memory protection
Memory management instructions

MATH PAC

Square root
Trigonometric and hyperbolic vector and rotate
Fixed/floating-point conversion
Floating-point arithmetic, compare, normalize, square root, trigonometric exponential, and natural logarithm
Double-precision multiply and divide
Algebraic left and right quadruple shifts

NDRO MEMORY AND CLOCKS

192-word bootstrap
32-bit realtime clock
16-bit monitor clock
1 kHz, 32 kHz, or external clock rate
4K words of macro-BIT (I/O options)

CONTROL AND MAINTENANCE INTERFACE

Features microprocessor-driven BIT for self-test and provides defined interface to a control and maintenance panel.

I/O CONTROLLERS

One or two I/O controllers per basic cabinet without expansion cabinet, 2 per basic cabinet with expansion cabinet (MIL-STD-1553B I/O takes place of one IOC)
Independent operation
Up to eight 2-channel groups (serial and parallel) per I/O controller
Up to 32 program-initiated I/O chains per I/O controller
I/O instruction repertoire - same format as CP
Control memory for each I/O channel
Optional memory address expansion to 4,194,304 (4096K) words

I/O CHANNEL ADAPTERS (IOCAs)

Parallel Channels

Implemented as 2-channel group
MIL-STD-1397, Type A/PIC (NTDS Slow)
MIL-STD-1397, Type B/PIC (NTDS Fast)
MIL-STD-1397, Type C/PIC (ANEW)

Modes:

8-bit byte, 16-bit word, or 32-bit dual-channel transfers
Normal transfers available on single- or dual-channels
Externally specified addressing (ESA) operation on dual channels
Intercomputer operation on single- or dual-channels

Serial Channels

MIL-STD-188C:

Implemented as 2-channel group
Synchronous - to 9600 baud
Asynchronous - 50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 4800, or 9600 baud (program-controlled selection)
Character size - 5-, 6-, 7-, or 8-bit (program-controlled selection)

EIA-STD-RS-232-C:

Implemented as 2-channel group
Synchronous - to 9600 baud
Asynchronous - 50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 4800, or 9600 baud (program-controlled selection)
Character size - 5-, 6-, 7-, or 8-bit (program-controlled selection)

MIL-STD-1397 Type D:

Implemented as 2-channel group
8-, 16-, or 32-bit transfers
Asynchronous - 180,000 words per second
Intercomputer or normal operation

MIL-STD-1397 Type E:

Implemented as 2-channel group
8-, 16-, or 32-bit transfers
Asynchronous - 180,000 words per second
Intercomputer or normal operation

VACALES (Variable Character Length Synchronous):

Implemented as 2-channel group
Synchronous - to 32,000 baud
Character size - 1 to 16 bits (program controlled selection)
MIL-STD-188C interface signals

NATO-STD-4156 (modified):

Implemented as 1- or 2-channel groups
Synchronous - 1 to 3 megabits per second

MIL-STD-1553B:

Compatible with MIL-STD-1553B
Bus controller and RT capability
Bus controller mode supports up to 32 RTs
Broadcast capability
Dual redundancy
Transformer-coupled to MIL-STD-1553B Bus
Transfers data into or out of up to 4M words of addressable memory
Interfaces using Data Device Corporation (DDC) SEMs:
- Key code UKS (67003 Bus Control)
- Key code UKT (67001 Bus Driver)
- Key code UKU (67002 Remote Terminal)

No IOC required - interfaces directly to common bus
One channel per basic cabinet only

MEMORY INTERFACE (MI)

Interfaces a maximum of 256K words of magnetic core or 2M words of semiconductor memory to the common bus
Modularity of memory can be 64K (core), 128K, 256K or 512K (semiconductor)
Maximum of four memory modules per MI

RCMI SERIAL INTERFACE

Allows control of MRC by computer, terminal or maintenance panel
Independent option of MRC
Independent of other MP functions
Synchronous or asynchronous RS-422/423 2-channel operation
Selectable baud rate to 19.2 kilobaud

RCMI BREAKPOINT

Contains four relative and four absolute breakpoints
Up to four relative ranging breakpoints (any two relative breakpoints compose a ranging breakpoint)
Up to four checkpoints available
Checkpoint data queued in FIFO for access by MP
Checkpoints indicated via RCMI auto status
Checkpoint trace feature
Four RCMI user-definable flags indicated via auto status
Four hardware status discretes

MAIN MEMORY

Magnetic Core:

16-bit words
Asynchronous timing
64K modules of 256K words
900-ns read/write cycle

Semiconductor (6 x 9-inch module):

16-bit words
Asynchronous timing
256K word modules
Maximum of 2M (1M if mixed with core)
350-ns read cycle; 600-ns write cycle
Single-error correct/double-error detect

DIRECT MEMORY ACCESS (DMA)

External direct memory access
AN/UYK-20 compatible
Four SEM installation

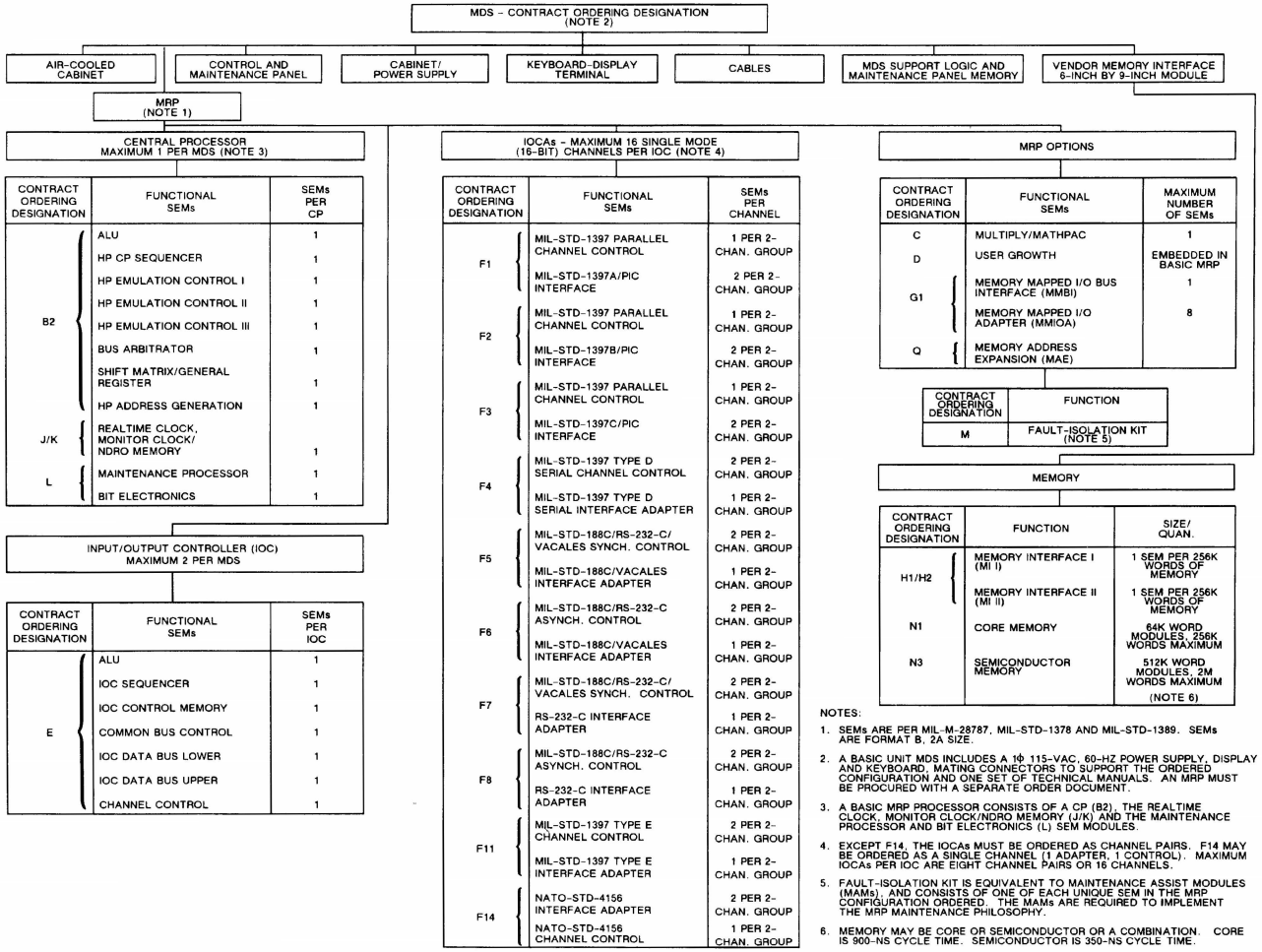


Figure 4. MDS Configuration Items

AN/UYK-44(V) MICROPROCESSOR DEVELOPMENT SYSTEM SPECIFICATIONS AND FEATURES

GENERAL CHARACTERISTICS

Supports embedded militarized reconfigurable processor (MRP) with options (two IOCs, 32 IOCAs, eight MMIOs maximum)
Two 19-inch wide commercial cabinets with control panel, SEM racks, memory wiring, power, and cooling provisions
Used for operation and test of the MRP and AN/UYK-44 software development
32 I/O connectors

MAIN STORAGE

Commercial memory
16-bit words
Asynchronous timing
Magnetic core memory:

256K words in 64K word modularity
900-nanosecond cycle time

Semiconductor memory:

2M words in 512K word modularity
350-nanosecond cycle time

POWER SUPPLIES

Commercial quality
115-120 vac, 60 Hz, 1 phase, 2 kilowatts maximum

KEYBOARD-DISPLAY TERMINAL

Unisys UTS 20 operator console
Alphanumeric display with 10 x 7-inch viewing area
Alphanumeric keyboard
25 lines of 80 characters each
Selectable instruction or constant format
32K character programmable erasable programmable read-only memory (EPROM) plus 16K character scratchpad random access memory (RAM)

SUPPORT LOGIC

6 x 9-inch circuit card assembly

Provides the following functions in support of the MRP:

9600 baud EIA-STD-RS-232-C serial interface between MRP MP and UTS 20 operator console
Four breakpoint registers with stop functions for CP
Control and indicator functional interface between MRP and MDS control panel
ROM and RAM for MDS functions
Access to an 8-address, P-jump history file in the CP
Discrete signals for MRP control and status indications

PHYSICAL CHARACTERISTICS

Temperature range:

Operating: +5°C to +35°C
Nonoperating: -10°C to +50°C

Size (nominal):

Cabinet assembly:
Height: 157.48 cm (62.00 in.)
Width: 115.27 cm (45.38 in.)
Depth: 64.77 cm (25.50 in.)

Keyboard display:
Height: 33 cm (13 in.)
Width: 50.8 cm (20 in.)
Depth: 71 cm (28 in.)

Table:
Height: 73.6 cm (29 in.)
Width: 111.7 cm (44 in.)
Depth: 76.1 cm (30 in.)

Weight (nominal):

Cabinet assembly: 454 kilograms (1,000 lbs.) (maximum configuration)
Keyboard display: 14.96 kilograms (33 lbs.)
Table: 40.9 kilograms (90 lbs.)

The MRC basic cabinet, as shown in Figure 5, can be mounted in a 19–inch rack, or deck mounted, and connected to other hardware equipment to perform system processing and I/O activity as a functional equivalent for the AN/UYK–20(V) DPS. The MRC expansion cabinet, as shown in Figure 6, is mounted in the same manner subject to cable length restrictions.

The versatility of the MRP/MRC/MDS design enables the CP and IOC to be interconnected to a variety of memories (various types and speeds). With the fast MRC memory (350 nanosecond semiconductor) the CP throughput is one and one–half times the throughput performance of the standard AN/UYK–20(V) DPS.

The MRP/MRC/MDS is reconfigurable and suitable for a variety of applications. A simple configuration can grow with optional functions that can increase efficiency and versatility. Options can be either added or changed in the field by incorporating or replacing modules to the basic unit. Hence, current and near–future applications can define the initial configuration based on known needs with minimum investment. Features to enhance processing and I/O capabilities, or to meet requirements of system growth, can be added in modular form later.

Modularity, versatility, and serviceability are design features that make the OL–335(V)/U and AN/UYK–44(V)/(U) adaptable to current and future applications in mission–critical systems. Each MRP and MRC accommodates current options and is designed to allow incorporation of future enhancements.

The MRP can operate with magnetic core memories and solid–state semiconductor memories. The CP can use memories with cycle times ranging from a low of 250 nanoseconds to a high of 1 microsecond. The MRC incorporates 900–nanosecond cycle time magnetic core memory and 350–nanosecond cycle time semiconductor memory. A flexible microprogrammable control section provides a fast computing capability while affording a basis for tailoring functional operations to specific or unique applications.

AN EXCELLENT LONG–TERM INVESTMENT

Expansion of functional capability is done by adding hardware options. Additional capability can be added by using the microcode growth features of the CP and IOC. The user can define a specific functional operation and have that operation incorporated as a new macroinstruction within the ISA.

A data processing system with a high performance/cost ratio is attainable when the MRP/MRC serves as a foundation. Simplicity and compatibility, combined with functional and physical flexibility, characterize the MRP/MRC in all of its available configurations. Simplicity, which is achieved using the capability and flexibility of the AN/UYK–44 ISA, provides simple and efficient program generation and implementation. The MRP/MRC provides the same dependable service expected in an MRP system as the field–proven AN/UYK–20(V) DPS. Reliability and maintainability, resulting in high operational availability, are incorporated in the design and development of the MRP/MRC. The I/O capabilities offer a wide interface potential that includes byte, word, or dual–word transfers, various types of parallel and serial interfaces, signal levels, and transfer speeds.

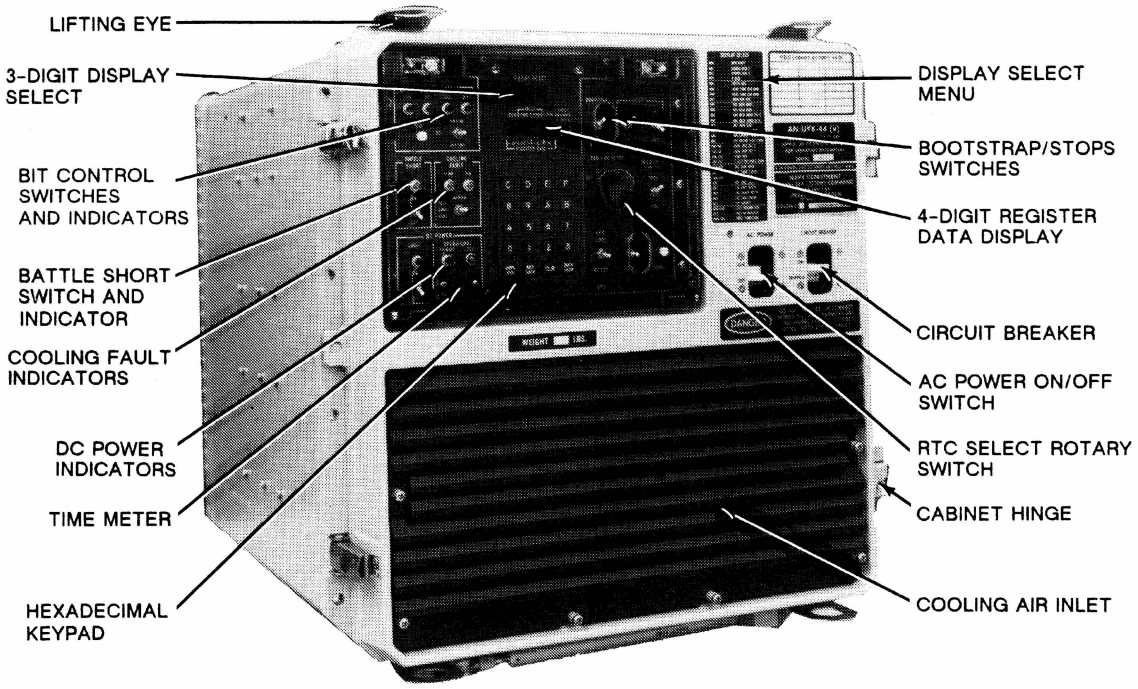
REALTIME APPLICATIONS

Functional characteristics of the MRP/MRC make it ideally suited to dedicated realtime applications such as standalone and distributed processing systems. A dual–state architecture, coupled with a hardware–initiated, multilevel, interrupt–processing capability, provides efficient and rapid parameter manipulation and preparation prior to the actual interrupt servicing. These overhead functions, normally performed by interrupt processing software routines, are thereby decreased and faster interrupt response time is achieved. The processing efficiency obtainable with the use of the general–purpose registers and related instructions provides the ability to meet the high data–rate environment encountered in time critical, realtime systems associated with fire control, telemetry, or online process–control applications, and realtime systems associated with communications, display control, or data systems.

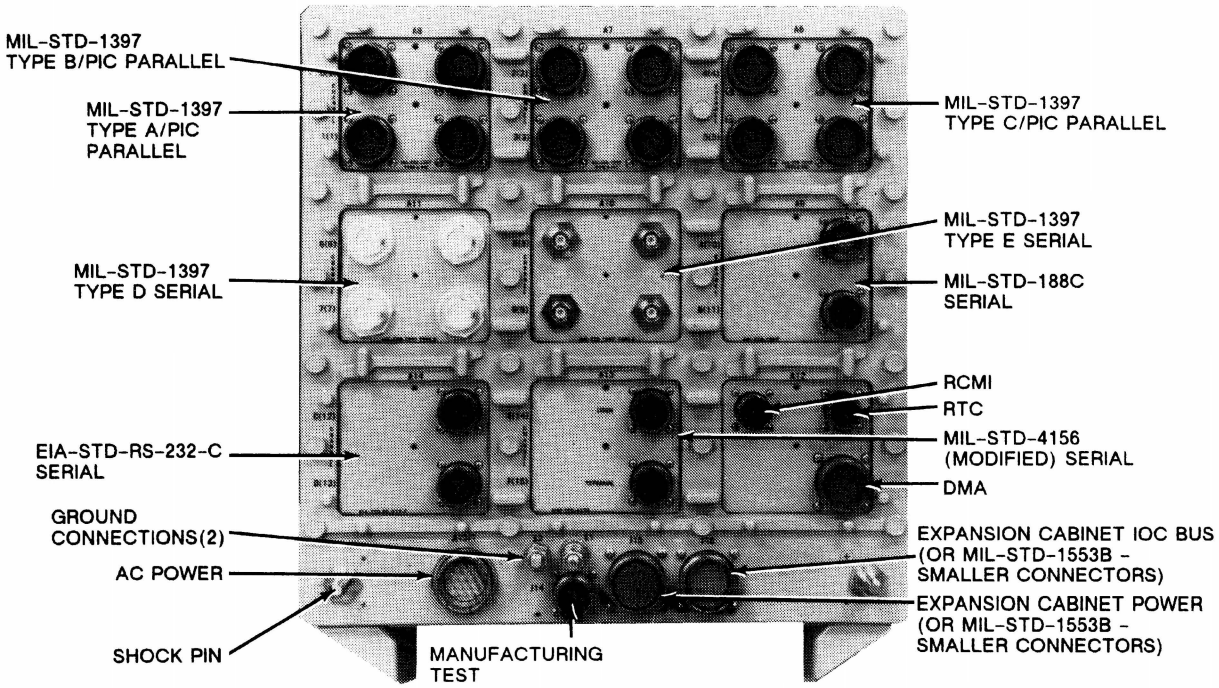
Shipboard Defense Systems Applications

Processing all raw data available from a task force and a ship's system is a huge assignment for a command, control and communications (C^3) system.

The MRP/MRC can be used effectively in reducing this burden by absorbing specific data reduction and related overhead tasks in the system. Functionally, a tactical data system coordinates the collection of data from many sources including sonar, radar, identification, friend or foe (IFF), and passive detection apparatus communication links. It coordinates all data with ship system status and navigation information, prepares a clear picture of the tactical situation to aid the decision–making process, and communicates the decisions to applicable and available action systems and personnel.



FRONT VIEW



This view shows nine EMP modules; for the MIL-STD-1397 interfaces, the input is on the left, the output is on the right.

BACK VIEW

Figure 5. MRC Basic Cabinet, Front and Back Views

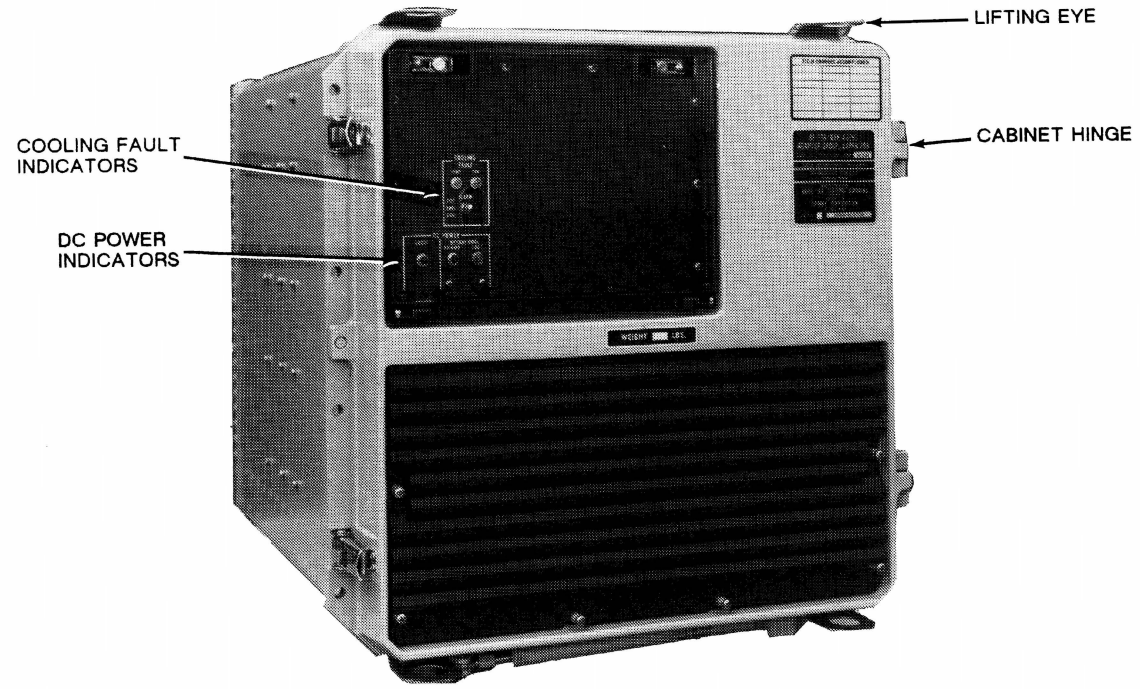


Figure 6. MRC Expansion Cabinet, Front and Back Views

The MRP/MRC implemented as a processor has the calculating speed and data handling characteristics required to reduce large volumes of raw data to usable values and to arrange them in a format acceptable to the C^3 computer for direct integration into the total system.

Communications Systems

The MRP/MRC with the independent IOC and its bit and byte manipulation instruction repertoire is ideally suited to communication applications. The many types of serial I/O channels provide great flexibility for handling both synchronous and asynchronous communication lines in a wide range of rates. Network control, store and forward, and line–concentration functions are readily implemented through the incorporation of the MRP/MRC in a communication system. Its inherent reliability ensures continuous, effective service in these applications.

Signal–Processing Systems

A major military application is in a signal–processing system where radar, sonar, and beacon signals are processed. In this application, the front–end signal processors provide a continuous input of data in a realtime environment to a system computer. The collection of processed signal data from a variety of sources at high data rates requires that a fast reaction time and shortened processing time processor be used in the system. This critical data processing task is handled easily by the MRP/MRC. Its comprehensive and flexible instruction set executed by the fast central processor; its programmable, realtime clock (RTC); and the high–speed, hardware initiated, interrupt structure perform the necessary data handling. Direct access to memory for realtime data input or output is done by the very fast, programmable I/O section, the externally controlled memory mapped I/O (MMIO) option in the MRP or the externally controlled direct memory access (DMA) option in the MRC.

Control Systems

The MRP/MRC can be used for a variety of control system applications including weapons, air traffic, sensor, electronic countermeasures, and navigation control systems. Complex control systems require a high level of computational capabilities. While the quantity of input data is lower than for signal processing, input is received from more than one source. Here again, the MRP/MRC qualifies for this application. The number of I/O channels can be expanded, as required, by plug–in modules. Complex computations required for commanding the system are accomplished with programs which use the fast general registers and the associated single– and double–precision arithmetic.

Other MRP/MRC Applications

- Message Handling – receiving, logging, and forwarding
- Fire Control
- Navigation
- Management Information
- Telemetry
- Communication Links
- Radar Processing
- Data Reduction
- Sensor Processing
- Range Tracking
- Logistics

MODULAR ARCHITECTURE – MRP AND MRC

The MRP/MRC is entirely modular. It is in complete compliance with the modularity requirements of MIL–E–16400 (General Specifications for Naval Electronic Equipment).

The basic physical format module of the MRP and MRC is the 2A size, Format B SEM, which complies with MIL–STD–1378 and MIL–STD–1389. The CP and options, the IOC and options, the input/output channel adapters (IOCAs), the memory bus interface, the control and maintenance interface, and the temperature sensors are contained on SEMs. The SEMs are intended to be embedded in other equipment, the MRC cabinet, or the MDS cabinet. A SEM is shown in Figure 1.

A fully configured MRC consists of an MRP and all its options, a temperature sensor, magnetic core memory or semiconductor memory, power regulators, a power conditioner, and a basic cabinet with a control and maintenance panel and I/O connectors. The MRC may also include an optional DMA feature. The DMA feature is functionally compatible with the DMA of the AN/UYK–20(V) Data Processing Set. An MRC cabinet, however, is not configured for the MMIO function. An additional IOC must be added to the basic MRC cabinet when an expansion cabinet is installed. IOCAs (up to 16 channels) for the additional IOC are contained in the expansion cabinet.

FUNCTIONAL ARCHITECTURE – MRP AND MRC

Functionally, the MRP/MRC is organized around a CP and an IOC, each having its own microprogrammed control structure. The CP and the

IOC functions are completely independent and asynchronous; they communicate with each other by a CP command instruction and an interrupt structure. All major elements of the MRP/MRC are interconnected by, and exchange data over, a dual–source, single–destination common bus structure. The IOC communicates with peripheral equipment and computers through parallel or serial IOCAs. The CP and IOC exchange data with memory through the MI modules that are connected to the common bus. In the MRC, display to and control from a control and maintenance panel are through the panel interface module. In the MRP, implementation of display and control functions is at the discretion of the user. The functional architecture is shown in Figure 7.

Main Memory – MRC

The MRC main memory is random access, 16–bit word, either magnetic core memory (nonvolatile), 6 x 9–inch semiconductor memory (volatile), or a combination of both types, up to a maximum of 2,097,152 words per cabinet.

Memory Interface – MRP and MRC

Memory interface SEMs provide the interface between the common bus structure and the 6 x 9–inch semiconductor or core memory modules.

Semiconductor Memory on SEM – MRP

The optional SEM semiconductor memory system is composed of a memory control set on four Format B, size 2A SEMs and several types and sizes of semiconductor memory arrays: the 128K dynamic random access memory (DRAM), the 64K static random access memory (SRAM), and the 64K or 128K ultraviolet erasable programmable read–only memory (UVEPROM). The control set, consisting of the address, data, timing, and control SEMs, interfaces directly to the common bus, eliminating the need for additional memory interface modules. Six error–detection check bits occupy data bit positions 21–16, and the data word occupies data bit positions 15–0. Up to eight memory array SEMs of different types and sizes can be installed for up to a total of 1 million words of storage.

The memory system provides a hardware–controlled refresh capability to keep data within the DRAM free from degradation caused by time and temperature. Refresh is performed once every 15.6 microseconds. Error detection and correction (EDC) circuits are provided in the memory system which increase memory system reliability and protect data integrity. EDC is implemented using a modified Hamming code which involves generating six check bits appended to the data word that contain enough redundant information to correct all single–bit errors in a data word and to detect all double–bit errors and some triple–bit errors. Error detection and correct logic is enabled and disabled using the operation code 00, m = 7, octal and 00, operation code 00, m = 8, octal macroinstructions.

Two basic types of memory errors can occur. Hard errors are permanent physical failures of either the whole memory array, a row, a column or a single bit and are caused by power shorts, open leads and physical damage. In the DRAM or SRAM arrays, soft errors occur which are nonrepeating, single–bit errors when there is no permanent damage. A soft error occurs when the charge state of a bit incorrectly shifts from 0 to 1 or 1 to 0. This can be caused by system noise, stored bit pattern sensitivity, power surges, or alpha particle bombardment. Memory error information is kept current by using a memory error log which keeps track of the type of error that occurred (hard or soft error) and the memory array module on which the error was detected. The error log is searched and read by the code 00, m = 9, octal macroinstruction. This macroinstruction searches for an indication of an error being detected or corrected in memory. The general register specified by the a–field (R_a) contains the value 0. The register specified by R_{a+1} contains the starting error log address. R_{a+2} contains error information. If an error is detected, R_{a-1} contains the error log address where the error indication was detected and R_{a+2} contains the next error log address after the error indication. Using this macroinstruction clears the memory status register error bits.

In addition to EDC and error logging, a hardware–controlled memory scrub operation is used to ensure double–bit errors do not occur in a RAM data word. Scrubbing out single–bit errors prevents error backlog so most double–bit errors are avoided. The scrub is performed by periodically reading out every word, checking it for errors, correcting any single–bit errors, and writing the word back into the RAM arrays. The scrub operation is performed as a background routine when the memory is not being used by requestors. A complement of eight DRAM array modules is completely scrubbed once every 69.8 minutes.

The memory system provides NDRO memory interfacing by performing the following functions:

- Initiating an NDRO memory cycle when requested

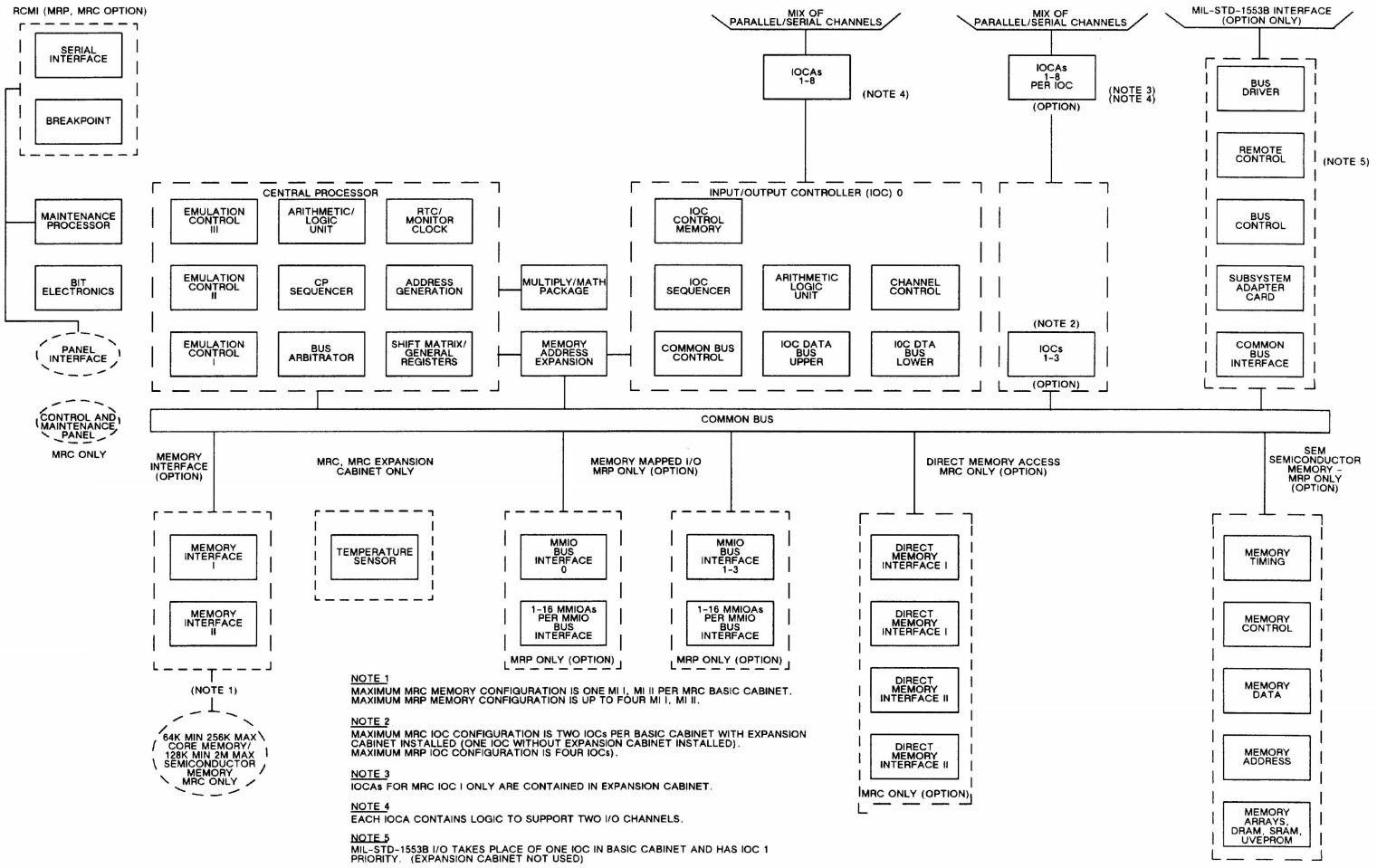


Figure 7. MRP/MRC Functional Architecture Block Diagram

- Disabling the memory array modules from responding
- Allowing usage of the memory system address and data paths

Direct Memory Access (DMA) - MRC

The MRC design permits the addition of a DMA capability. This option allows an external controller to read from and write into main memory (see Figure 8). Simultaneous requests for memory access by the DMA/IOC and the CP user are separated to give priority service to the CP/IOC. The connectors and cables are defined in Table 1.

User Growth - MRP and MRC

Optional microprogram read-only memory (ROM) is available for user-defined macroinstructions in both the CP and the IOC. Users can define the requirements for a new instruction. Operation codes 67 f = 0, 1 or 2, octal, and DC, DD, or DE, hex are available for CP instructions; operation codes 72 f = 0 or 1, octal, and E8 or E9, hex for IOC instructions.

Implementation of new instructions requires that the user define the operation in terms of input, output, registers to be used, error conditions, etc. The user should supply a detailed specification describing the specific operation to be performed. A specific operation code will be assigned when the instruction is implemented.

User growth is an ordering option and those users who wish to implement user growth should be prepared to discuss their specific requirements with the procuring government agency or manufacturers.

Since user growth is an optional add-on to the ISA of the MRP/MRC, users must support the generation of the appropriate op codes, etc., with the macro generation capability of the MTASS/minicomputer (MTASS/M) software.

Memory Address Allocations - MRP and MRC

To aid in the development of relocatable software, all memory locations are accessible to the programs at random and to all sections of the MRP/MRC.

Some memory locations are given special assignments, which programs must respect and provide for their contents.

Memory Addressing - MRP and MRC

Main memory is used for storage of programs, constants, and data. The CP and IOC provide expanded addressability to 4,194,304 words of memory through the addition of the optional memory address expansion and the use of the memory management instruction (CP operation codes 70 through 74, octal; codes f0 through f3, hex) by the executive program.

Addresses are specified using four sets of page registers, which select pages with 1,024 words per page. Status Register (SR) 1, bits 4 and 5, control the page register set selection for the CP. The IOC uses page sets 0, 2, and 3, with sets 2 and 3 used for data transfers only. Sets 2 and 3 are allocated to channels 0–7 and 8–15, respectively.

The lower-order 10 bits of the relative address specify the address of a word within a 1,024 word page of main memory. The most significant six bits (index) select one register from a group of 64-page address registers that contains the fixed-based address of a specific word page within main memory. Figure 9 illustrates the final address generation.

Nondestructive Read-only (NDRO) Memory - MRP and MRC

A block of 192 NDRO memory words is provided in the CP. The programs contained in the NDRO memory are fixed at the time of manufacture, based on the ordering document, and cannot be changed by read and write operations. The NDRO memory can be changed in the field by a simple and easy SEM replacement. Addresses assigned to NDRO memory, as defined in Table 2, parallel similarly numbered relative main memory addresses. Bit 12 in SR 1 (see Figure 10) controls the access to NDRO memory or to corresponding locations in main memory.

NDRO memory is convenient storage for programs available to the MRP/MRC. These might include an initial load routine which loads a program and checks the validity of the program load and an inspect and change routine. A portion of the BIT firmware is also included in the NDRO memory.

Memory Address Expansion (MAE) - MRP and MRC

The memory address range of the IOC in the MRP can be expanded with the addition of the MAE option. This capability is provided in a single SEM designed to interact with the MRP addressing scheme. With the addition of the MAE option, MRP addressing is extended to 4,194,304 16-bit words of memory, using four sets of 64-page registers, each page equal

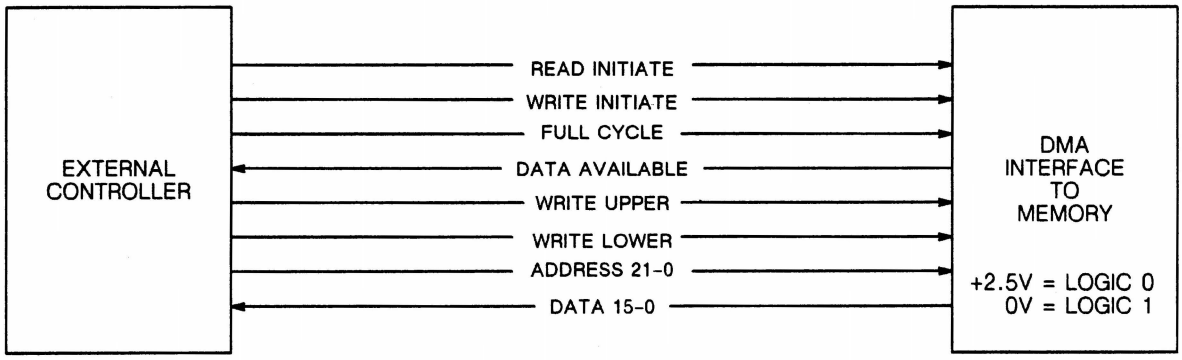


Figure 8. Direct Memory Access Interface

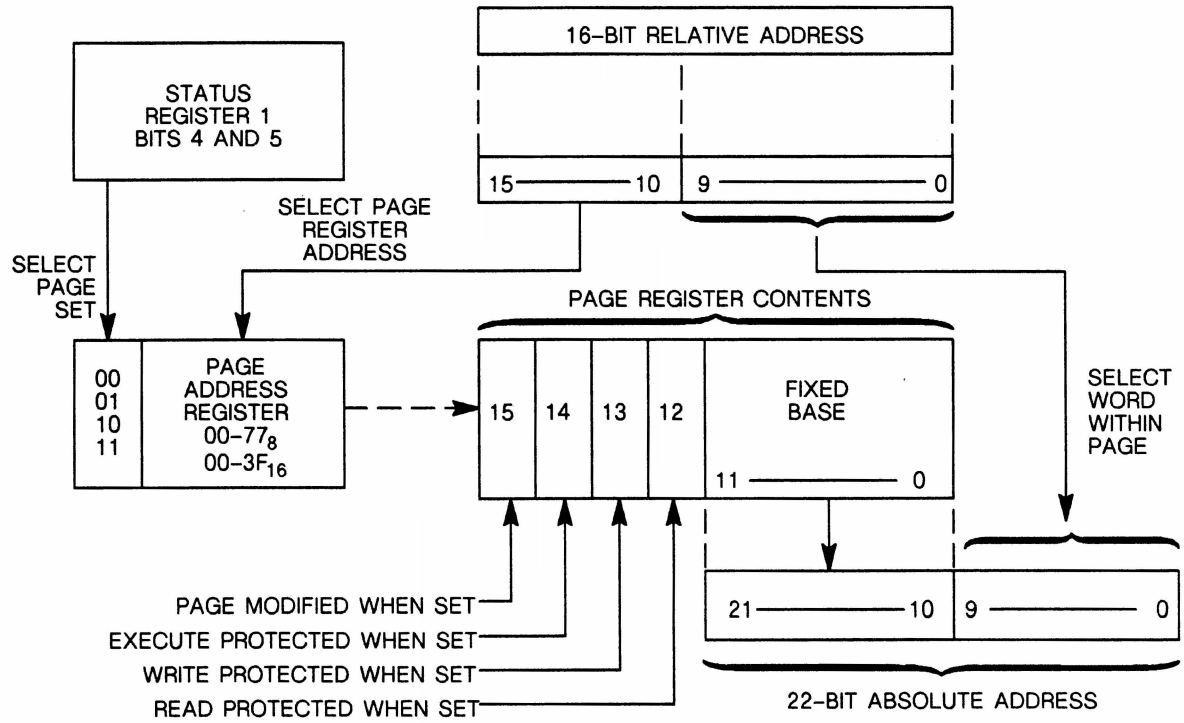


Figure 9. Memory Address Generation

TABLE 1. EXTERNAL CONNECTOR/CABLE DATA

Feature	Plug (Cable Connector) Part Number	Cable Connector Adapter Part Number	Cable Type
Direct Memory Access	D38999/26WH35SN	Glenair Inc. 390HS001NF2316H6	2U-45
External Realtime Clock	D38999/26WD35SN	Glenair Inc. 390HS001NF1508H6	M17/030-RG-62
Remote Control and Maintenance Interface (RCMI)	Unisys PN 7909696-00	Unisys PN 7908945-09	

TABLE 2. ASSIGNED MEMORY ADDRESSES (MEMORY MAP)

Assignment	Addresses	
	Octal	Hexadecimal
NDRO Memory	0-77	0-3F
Not Assigned	100-107	40-47
For Processing:		
Class III Interrupts	110-117	48-4F
Class II Interrupts	120-127	50-57
Class I Interrupts	130-137	58-5F
For IOC Operation:		
Command Cells IOC 0	140-141	60-61
Not Assigned	142-167	62-77
BIT Data	170-175	78-7D
Not Assigned	176	7E
Auto Start Entrance (Normal)	177	7F
External Interrupt Word Storage (IOC)	200-277	80-BF
NDRO Memory	300-477	C0-13F

to 1,024 words. Memory management instructions are also added to the ISA when the MAE module is included in the MRP.

Input/Output - MRP and MRC

IOCs relieve the CP of the peripheral communication burden and integrate the MRP/MRC into a system that has an I/O equipment complex established. When an input- or output-related function is required, the main program initiates an I/O chain that performs the input or output operations according to a stored program defined for the specified channel.

An MRP can be configured with up to four IOCs, each controlling up to eight 2-I/O channel groups, for a total of 64 I/O channels. A basic cabinet MRC is configured with one IOC for a total of 16 I/O channels. An MRC configured with an expansion cabinet has two IOCs for a total of 32 I/O channels. All IOCAs for the additional IOC are contained in the expansion cabinet.

The optional MIL-STD-1553B I/O channel replaces an IOC in either an MRP or in an MRC basic cabinet. The MIL-STD-1553B channel operates independent of an IOC and uses an optional backpanel.

Each IOC communicates with peripheral units in the system over the individual interfaces contained in the IOCAs (see Figures 11 through 17) and with memory on the common bus.

Each parallel I/O channel consists of one input cable and one output cable. The input cable carries data from the peripheral to the computer along with the appropriate control signal lines. The output cable carries data from the computer to the peripheral along with the appropriate control signal lines. Figure 11illustrates the various signal and data lines required for a parallel interface channel. Output channels are used to transmit data and commands to the peripheral equipment. Input channels are used to receive data or interrupt codes from the peripheral equipment.

An 8-bit byte, 16-bit word, or a 32-bit double-word parallel interface can be utilized for data transfers.

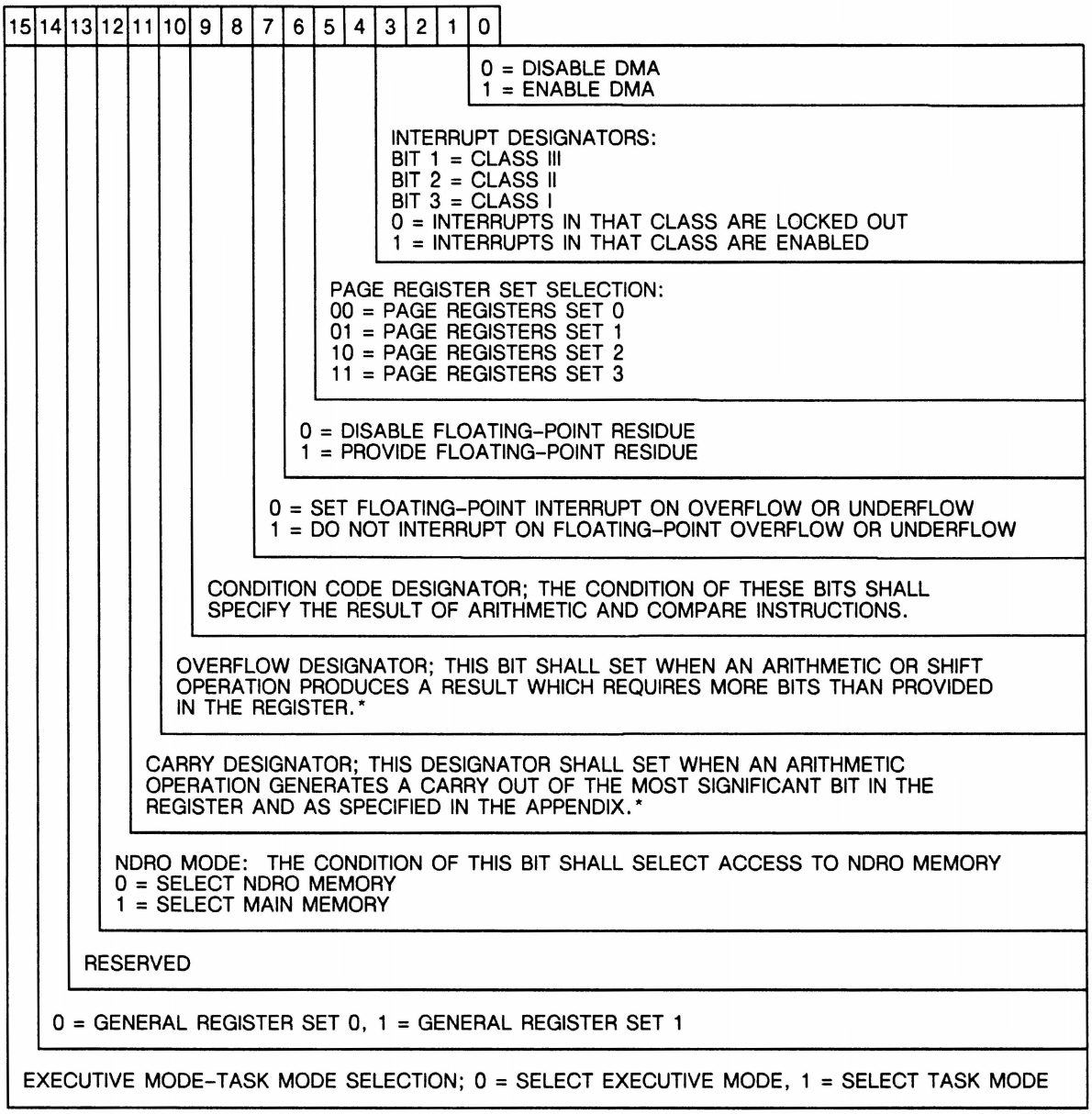
In dual-channel modes, the 32-bit parallel transfers use two 16-bit channels. Both single- and dual-output channels operate in the intercomputer or normal modes. Interface voltage levels on parallel channels can be the MIL-STD-1397 Type A (NTDS slow), Type B (NTDS fast), and Type C (ANEW) interfaces. Single- and dual-input channels have a peripheral input channel (PIC) option. All I/O activity is asynchronous, and the timing is dependent on the speed of the peripheral equipment compatible with MIL-STD-1397.

The optional serial interfaces available are:

- MIL-STD-1397 Type D
- MIL-STD-1397 Type E
- MIL-STD-188C synchronous or asynchronous
- MIL-STD-188C VACALES
- EIA-STD-RS-232-C synchronous or asynchronous
- NATO-STD-4156 (modified)
- MIL-STD-1553B

The IOCA performs the necessary serial-to-parallel and parallel-to-serial conversions. Table 3 defines the maximum I/O transfer rate per cable type and length for each type of interface.

The memory interface provides the IOC with an access to memory on a time-share basis with the CP.



*BIT 11 AND BIT 10 TOGETHER FORM THE FLOATING-POINT UNDERFLOW/OVERFLOW DESIGNATOR AS FOLLOWS:

01 = OVERFLOW, 11 = UNDERFLOW

Figure 10. Status Register 1 Format

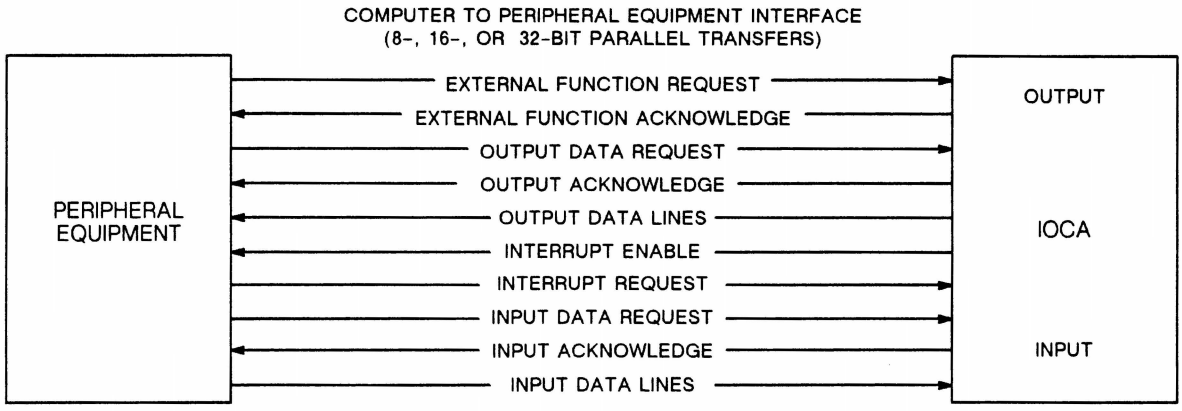
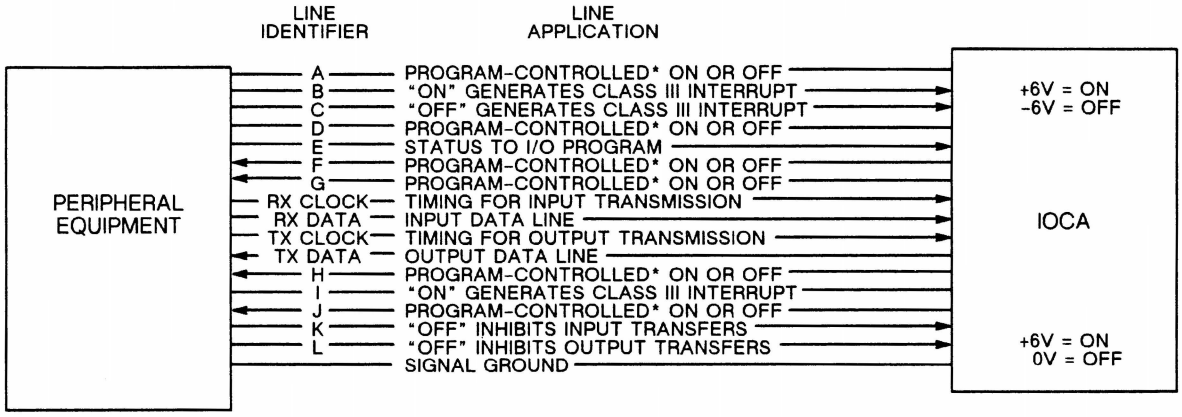
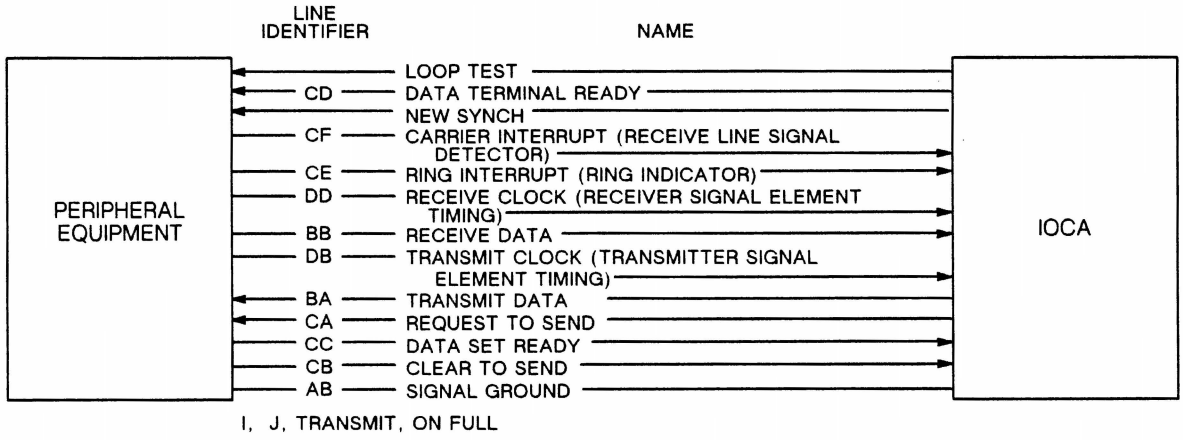


Figure 11. MIL-STD-1397 Parallel Interface



*Program controlled lines are assigned functions according to the need of the particular equipment connected to the channel.

Figure 12. MIL-STD-188C and VACALES Serial Interface



I, J, TRANSMIT, ON FULL

Figure 13. EIA-STD-RS-232-C Serial Interface

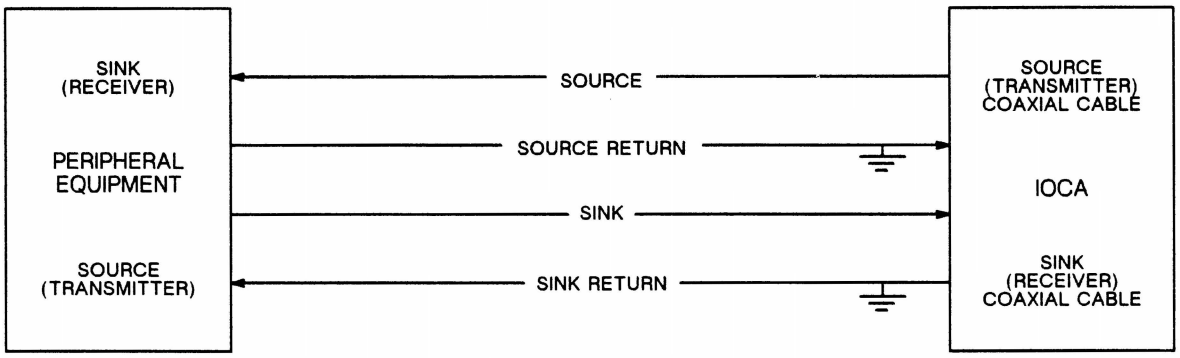


Figure 14. MIL-STD-1397 Type D Serial Interface

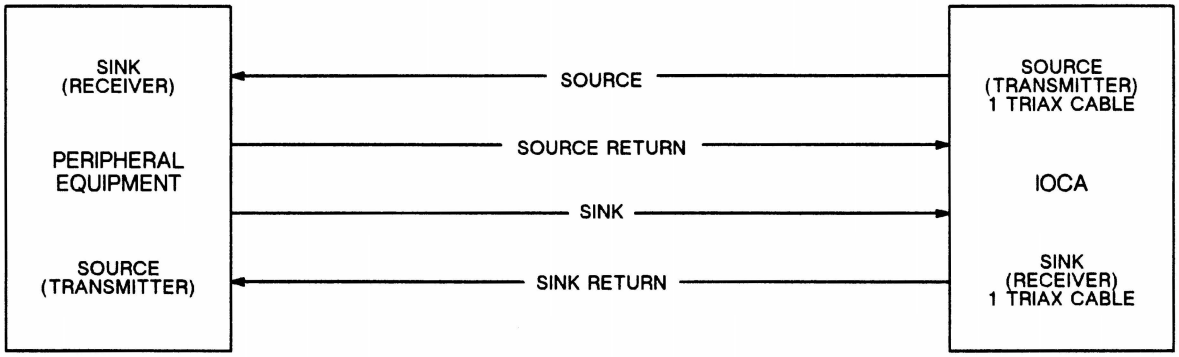


Figure 15. MIL-STD-1397 Type E Serial Interface

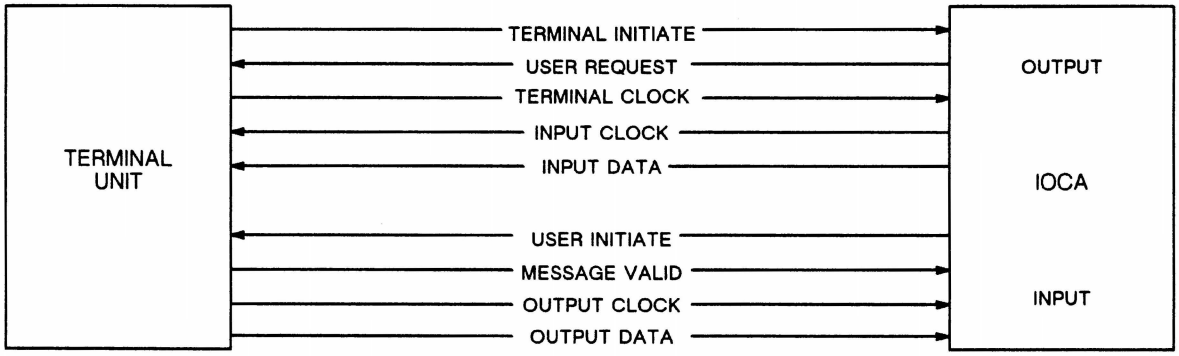


Figure 16. NATO-STD-4156 (Modified) Serial Interface

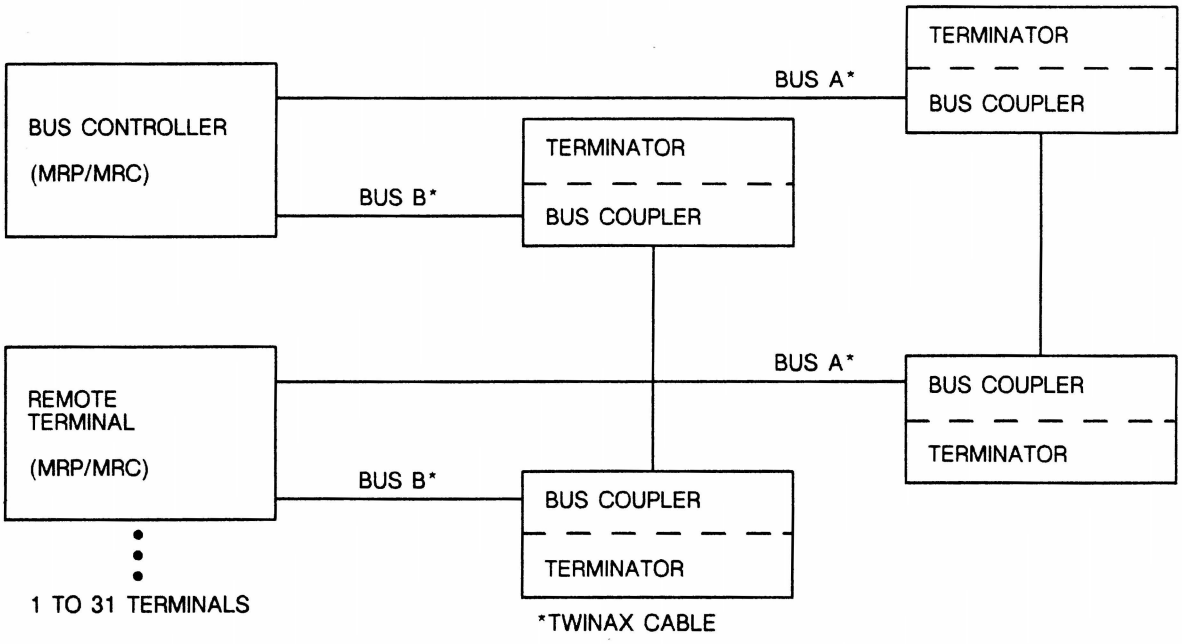


Figure 17. MIL–STD–1553B Interface

Priority is given to the IOC in case of simultaneous requests.

Memory Mapped I/O (MMIO) – MRP

To support the MRP as an embedded processor, a method of providing interface internal to the embedded processor and its host equipment is available. Termed memory mapped input/output (MMIO), this optional interface provides a transistor/transistor logic (TTL)–level connection between the memory of the MRP and the host device.

MMIO is physically partitioned as a single SEM memory mapped bus interface (MMBI) module supporting up to 16 input/output adapter (MMIOA) modules. The MRP can be enlarged by adding up to three additional MMBI modules, each supporting up to 16 MMIOA modules. Each input/output channel adapter (IOCA) module can receive data in a parallel or a serial mode, as defined by the external equipment to which it is connected.

Program operation of an MMIO channel is governed by execution of CP macroinstructions, bit set/clear conditions of the MMIO control and status register, and the CP response to Class III interrupts. Hardware operation requires that a level of logic be provided between the MMIO channel and the host equipment to latch TTL–logic level signals.

To the external equipment, the MMIO channel appears as a 4–word (16 bits each) data transfer and control area of computer memory. In reality, the 4–word area is a 4–register area on each MMIO module that is hardwired as a replacement for a specific 4–word location in main computer memory. As a design limitation, the user must specify that the 4–word location is within the first 8K area of main memory. The 4–word area is designed as:

- word n – External Interrupt Word Storage
- word n+1 – Input Data Word Storage
- word n+2 – Output Data Word Storage
- word n+3 – MMIO Channel Control/Status Word

An MMIO can be controlled by the CP in a polled or interrupt mode.

TABLE 3. I/O PHYSICAL DATA AND CHANNEL DATA RATE

Channel Type	Receptable (Jack) I/O Connector Part Number	Plug (Cable Connector) I/O Connector Part Number	Cable Connector Adapter Part Number	Cable Connector Contact Part Number	Cable Type	Cable Length (Ft)	Single-Channel Maximum Data Rate
MIL-STD-1397							
Type A/PIC (NTDS SLOW)	INPUT D38999/ 20WG35AN	INPUT D38999/ 26WG35BN	Glenair Inc. 390HS001NF2116H6	M39029/ 56-348	2U-45	300 200	41,600 Words/Second 41,600 Words/Second
Type B/PIC (NTDS FAST)						100	190,000 Words/Second
Type C/PIC (ANEW)	OUTPUT D38999/ 20WG35AA	OUTPUT D38999/ 26WG35BA	Glenair Inc. 390HS001NF2116H6	M39029/ 56-348	2U-45	300 200 100	190,000 Words/Second 190,000 Words/Second 190,000 Words/Second
MIL-STD-1397 Type D	COAXIAL AMPHENOL 34475-1050	COAXIAL AMPHENOL 53250-1000	Not applicable	Not applicable	RG-11 RG-12	1000 1000	10 Million Bits/Second 180,000 Words/Second
MIL-STD-1397 Type E	TRIAXIAL TROMPETER BJ-80	TRIAXIAL TROMPETER PL80-14A	Not applicable	Not applicable	TRF-58 TRF-8	400 1000	10 Million Bits/Second 180,000 Words/Second
MIL-STD-188C							
Synchronous						100	9,600 Baud
Asynchronous	D38999/ 20WE26AN	D38999/ 26WE26BN	Glenair Inc. 390HS001NF1710H6	M39029/ 56-351 Adapter-Bendix 10-407979-205	2U-19	100	9,600 Baud
VACALES						100	32,000 Baud
EIA-STD-RS-232-C							
Synchronous	D38999/ 20WE26PN	D38999/ 26WE26BN	Glenair Inc. 390HS001NF1710H6	M39029/ 56-351 Adapter-Bendix 10-407979-205	2U-19	50	9,600 Baud
Asynchronous						50	9,600 Baud
NATO-STD-4156 (modified)	User D38999/ 20WE26PA Terminal D38999/ 20WE26PB	User D38999/ 26WE26SA Terminal D38999/ 20WE26SB	Glenair Inc. 390HS001NF1710H6	M39029/ 56-351 Adapter-Bendix 10-407979-205	2U-10 (10 twisted pair with overall braid shield)	330 (at 1 MHz)	3 Million Bits/Second
MIL-STD-1553B	TRIAXIAL TROMPETER BJ-71	TRIAXIAL TROMPETER PL75-9	Not applicable	Not applicable	Twinaxial	–	1 Million Bits/Second 50,000 Words/Second

The MMIO data/interrupt transfer is controlled by the bit condition of the control and status register (word n+3) for each MMIO channel (see Figure 18). Bits 15, 14, and 13 are set by the program and indicate that a transfer can take place; bits 7, 6, and 5 indicate the status of the data in the external interrupt, and input and output registers. As the transfer is completed, the CP polls the condition of the status bits, if in the polled mode, or receives a Class III interrupt, if in the interrupt mode. When the CP has read (stored) the word from the MMIO register, the associated bit is cleared.

Because the MMIO transfer is accompanied by a Class III interrupt, the data transfers performed are generally low volume, that is, one word. The CP must move the data from the MMIO registers to an alternate storage area in main memory. The interrupt, in the interrupt mode, is Class III and is not differentiated from IOC/IOCA Class III interrupts.

The interrupt processing program must be able to differentiate between the two sources of interrupts.

Users who wish to employ the MMIO data transfer operations will find it useful for low-volume data transfers, as well as transfers from/to the embedded processor and the host device without the hardware/software overhead of an IOC/IOCA.

CP Executive/Task Modes - MRP and MRC

The CP operates in two modes: executive and task. The executive mode has a separate set of privileged instructions used for executive tasks. The task mode is for worker programs; programs in this mode cannot execute the privileged instructions reserved for the executive mode. Each mode can select a separate set of general registers; this feature precludes the need for storing and restoring register data when changing modes. Bit 15 of SR 1 controls the mode of operation of the CP. Appendix A identifies the executive mode privileged instructions.

CP General Registers - MRP and MRC

The CP has two sets of 16, 16-bit general-purpose registers, each set designated R_0 through R_{17} octal, and an instruction set tailored to their manipulation. The general registers provide for extremely rapid processing of parameters or data by decreasing the number of required main memory references. Contents of any number of registers in a set can be changed by one simple instruction, which saves program space and 50 percent of the time needed to execute the load and store process. With the availability of such registers, programs can be constructed with a greater proportion of single-word (RR Format) instructions, which decreases both program storage space and program execution time.

A general register can be used as: (1) an accumulator for arithmetic, shift, and logical functions, (2) an index register for address and operand modification, and (3) a temporary storage location for addresses, operands, etc.

The word format and operation code of an instruction requiring a general register reference defines the use of the register. One or both register designator fields (a,m) in that instruction select the register(s) in a set.

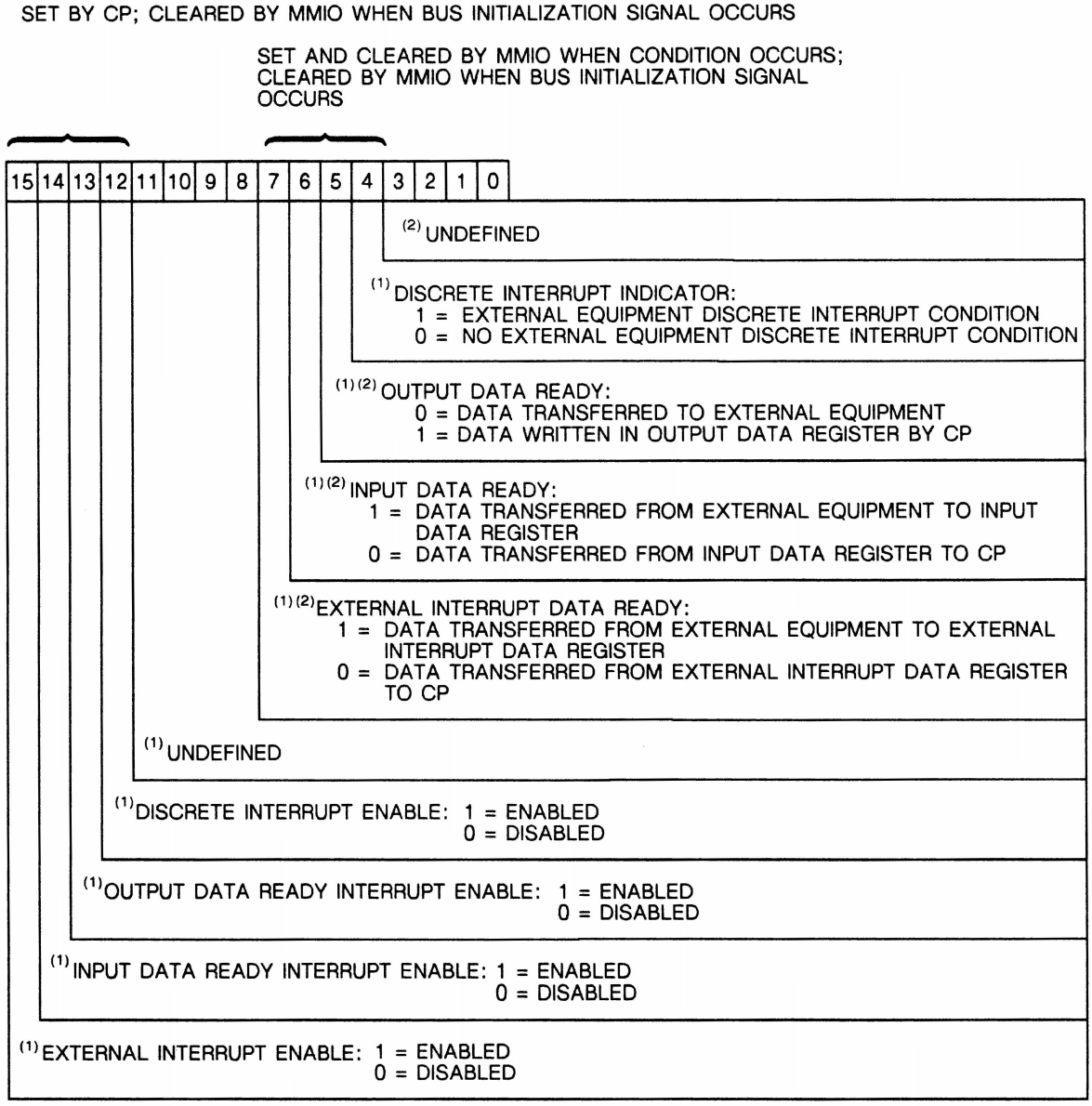
A program-controlled, 1-bit field in SR 1 selects the general register set used for processor operations. The dual general register set provides greater freedom and increased processing speeds in applications employing heavy interrupt processing and those using the multiprogramming technique. Programs repeatedly called for operations, and those in applications requiring rapid task changes (that is, switching from one to another), can be more independent when two sets are used. An executive program, for example, assigned a set for its own use need not store the contents of general registers used by worker programs every time it assumes control or when it is requested to process an interrupt.

Status Registers - MRP and MRC

The MRP and MRC provide a dynamic picture of their processing states using two 16-bit status registers. Fields in the status words can be modified by hardware and software or purely by hardware changes in the state of the processor. SR 1 is divided into fields that indicate CP status, such as interrupt status, and conditions resulting from arithmetic operations (see Figure 10). SR 2 (see Figure 19) controls indirect addressing and holds additional interrupt identification data.

Realtime Clock (RTC) and Monitor Clock - MRP and MRC

The optional RTC/monitor clock feature provides two program-controlled interrupts using two high-speed registers; one 32-bit register used as RTC count-up storage and the other a 16-bit register used as monitor clock count-down storage. This feature and associated controlling instructions are useful for program timing and for synchronous program segments with realtime events. An RTC oscillator, which has an accuracy of ± 2 counts in 10 seconds, runs continuously and controls the counting speed of the monitor clock and RTC registers. Selectable rates for the RTC



NOTES: (1) NOT MODIFIABLE BY EXTERNAL EQUIPMENT
(2) NOT MODIFIABLE BY CP

Figure 18. MMIO Control and Status Register

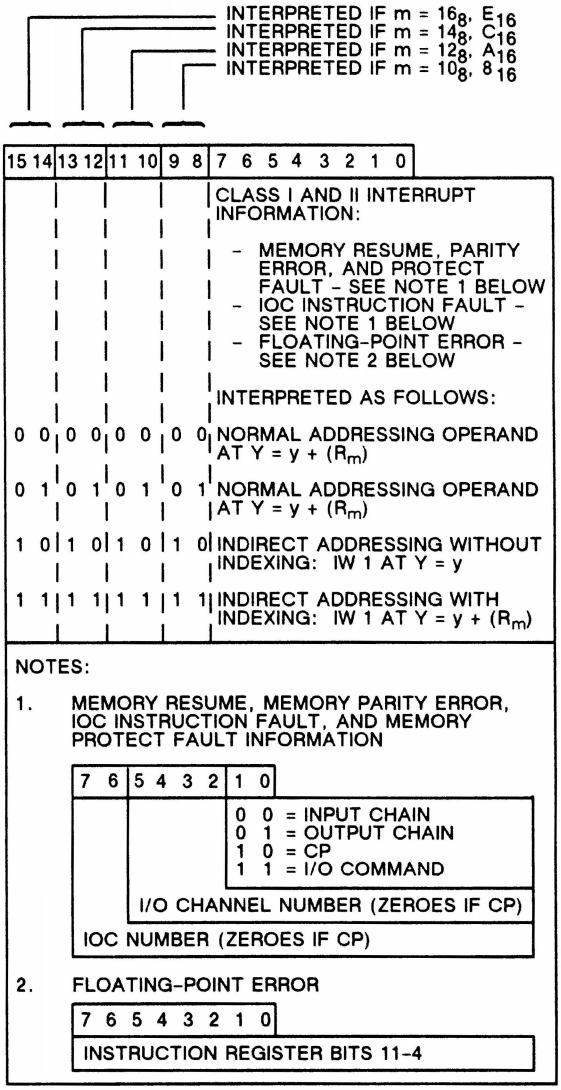


Figure 19. Status Register 2 Format

are 1 kHz and 32 kHz. An external clock oscillator with a frequency in the 0 to 50 kHz range can be used instead of the internal oscillator. The connectors and cables are defined in Table 1.

Breakpoint - MRP and MRC

The MRP/MRC has one 16-bit breakpoint register. The breakpoint function stops the MRP/MRC when a read or write reference is made, or when an instruction is executed at the address that matches the entry in the breakpoint register.

Control and Maintenance Interface (CMI) - MRP and MRC

The CMI consists of the MP SEM and the built-in test (BIT) electronics SEM and provides the microprocessor driven BIT and a TTL compatible interface to a control and maintenance panel (CMP).

Remote Control and Maintenance Interface (RCMI) - MRP and MRC

The optional RCMI provides the interface between the MP and user control and maintenance devices (see Figure 20). The RCMI is a set of two SEMs, one of which contains two channels that allow the processor to be controlled by a computer, a terminal or a panel, and provides status and BIT information to the device. All operations performed between the MP and the RCMI serial interface SEM are directed by firmware contained on the RCMI serial interface SEM and the MP SEM (that is, this firmware is required to establish channel configuration and transmission protocol, and handle data transfers).

The second SEM, the RCMI Breakpoint, contains eight macro breakpoints on any common bus function which can be:

- Eight relative (logical or virtual)
- Four relative and four absolute
- Up to four relative ranged

Additionally, the RCMI breakpoint contains checkpoints. Checkpoint data is queued in a first-in/first-out (FIFO) for later access by the MP. Checkpoints are indicated using RCMI auto status. The module also has a checkpoint trace feature, four RCMI user macro definable flags, and four hardware status discretes.

Memory Protection - MRP and MRC

A memory-protect fault interrupt is generated when an attempt is made to reference a protected page of memory. The memory-protect fault interrupt is a Class II interrupt. If Class II interrupts are locked out, the memory-protect fault interrupt is lost.

When the memory address expansion option is added to the MRP/MRC, a memory protection capability is provided for each page of memory. Three types of protection are implemented using bits stored in the page registers (see Figure 9). Memory protection is provided for instruction execution and memory read and write operations.

Execute Lockout - When bit 14 of the page register is set and instruction execution is attempted from the

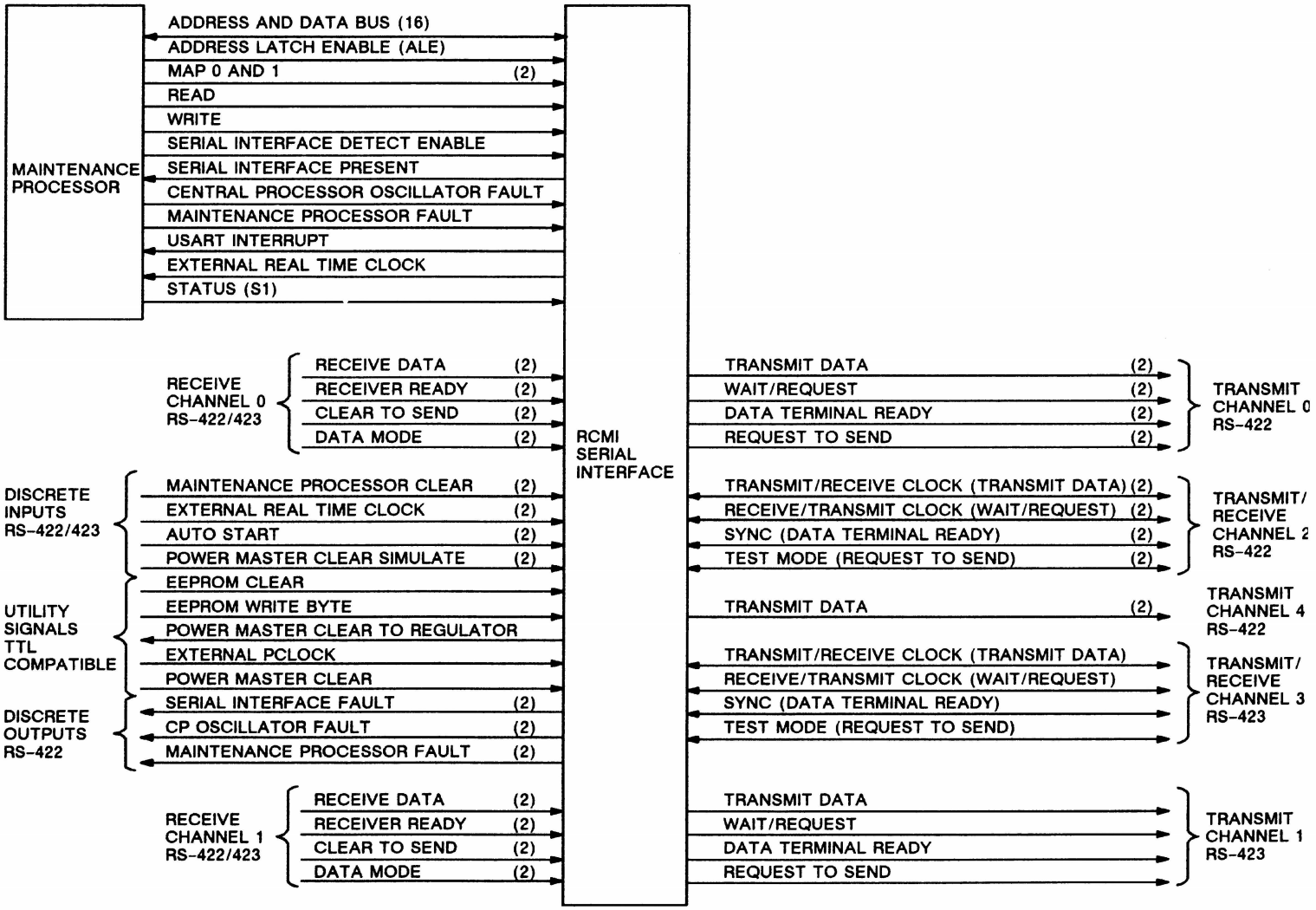


Figure 20. RCMI Serial Interface

associated page, the MRP aborts instruction execution and generates a Class II memory–protect fault interrupt. Indirect addressing words are exempt from execute lockout, but are subject to read and write lockout.

Write Lockout – When bit 13 of the page register is set and a write operation is attempted from the associated page, the MRP aborts the write operation and generates a memory–protect fault interrupt.

Read Lockout – When bit 12 of the page register is set and a read operation is attempted from the associated page, the MRP aborts the read operation and generates a memory–protect fault interrupt.

Read and write operations required as part of defined interrupt responses are exempt from read and write lockouts.

Instruction Counter (P) History File – MRC

The MRC has a 16–word instruction counter (P) history file that contains the address and source of the last eight events that caused a branch from the next sequential instruction to occur.

Power Failure Protection – MRC

The power fault and automatic recovery feature provides a systematic and safe shutdown and recovery capability in the event that any power to modular sections fall below an operable level. Sensors monitor the power supply voltage and, when an out–of–tolerance voltage is detected, a Voltage Out–of–Tolerance signal is generated. When the CP senses the Voltage Out–of–Tolerance signal, it generates a power fault interrupt (if enabled by the load PSW 07 1, octal; 1D, hex, instruction), which suspends the normal program sequence, transfers control to the user software power fault interrupt routine, and allows a minimum of 250 microseconds to arrive at an orderly termination. The power fault interrupt route stores the contents of all working registers and terminates in a jump instruction (operation code 82, a=6 and m = 0, hex, or 83, a = 6 and m = 0, hex) that jumps to itself as long as the Voltage Out–of–Tolerance signal is present. After the signal is removed (power returns to normal), the instruction allows the routine to continue to restore the working registers, and to return control to the interrupted program.

If the power continues to drop below operating limits, a CP master clears results in a shutdown. When power is reapplied and the auto start is selected, the CP generates an Auto Start signal that causes execution of the instruction located at address 000 of the NDRO memory. (This address usually contains a jump to the instruction located at the address specified by the contents of memory address 177, octal; 7F, hex.)

The automatic shutdown and recovery routine is a user software responsibility, but should normally perform as described and be compatible with the Class I interrupt codes assigned.

Temperature Sensor – MRC and MRC Expansion Cabinet

The temperature sensor modules contain the circuits required to advise the operator when an overtemperature condition exists, when a fan fault occurs, if the primary power is operational, and if the logic power is operational. It also has a power shutdown circuit that responds to a dc power logic switch and to an overtemperature condition (if the BATTLESHORT ON/OFF switch is in the OFF position). The MRC power supply will not function without the temperature sensor module in the SEM chassis.

CONSTRUCTION – MRP AND MRC

The physical size of each Format B SEM is 2A in accordance with MIL–STD–1389.

Most SEMs contain integrated circuits (chips) packaged in leadless chip carriers (LCCs) which are mounted on multilayer, thick–film, ceramic substrates. The substrate/LCC assembly is bonded in either a center frame or an offset frame, depending on the component area required. Some SEMs contain chips in dual in–line packages (DIPs) which are mounted on multilayer epoxy glass substrates. The substrate/DIP assembly is bonded to a T–Dip frame.

The MRC functional units are assembled in a cabinet constructed from an aluminum sheet and braced to provide structural rigidity. A cooling–air inlet and a filter are located on the front, and a cooling–air exhaust is located at the bottom of both the basic and the expansion cabinets. The fan is located at the bottom of each cabinet. Provision has been made in the base for free–standing mounting and for mounting within a standard 19–inch rack in accordance with MIL–STD–189. The front cover incorporates a rugged hinge and latch system that provides a uniform, high–clamping pressure against the cabinet opening. Gasketing around the periphery of the front cover eliminates electromagnetic interference (EMI) and provides moisture sealing. A control and maintenance

panel (CMP) is located on the outside of the front cover of the basic cabinet.

The rear of the basic cabinet contains the expansion cabinet connectors, I/O connectors, power connector, DMA connector, external RTC connector, and ground connectors.

The basic cabinet contains two module cages. A module cage at the front of the cabinet accommodates up to 72 SEMs and has a common multilayer back panel. The SEM cage swings out. A memory module cage is fixed and accommodates plug-in memory modules. The power regulator is mounted behind the SEM cage on top of the memory cage. A power conditioner is mounted behind the SEM cage to the right of the memory module cage.

The optional expansion cabinet contains a single SEM cage which holds up to 72 SEMs. The SEM cage swings out exposing a plate which must be removed to access the electromagnetic protection (EMP) modules and cabling. The power regulator is mounted behind the SEM cage at the top of the cabinet.

Optional features require only the removal, insertion, or substitution of plug-in SEMs. The MRC basic cabinet is wired for the maximum configuration including:

- CP with Math Pac
- RTC and monitor clock
- NDRO 192-word memory
- 2 IOCs (when used with expansion cabinet)
- Expanded IOC addressing
- 16 I/O channels, any combination parallel and serial channels
- Memory interface
- 256K words of core memory, or 2M words of semiconductor memory, or any combination that does not exceed four memory modules
- DMA

The expansion cabinet includes:

- Eight IOCAs (16 channels), any combination of parallel or serial
- Bus termination logic and power-up loading
- Cable and connector assemblies between the basic and expansion cabinets

Removal or addition of all options is permitted within the constraints of this maximum configuration.

SEMs are designed to operate with less than 8 watts of power. Table 4 lists the SEMs by key code, module name, power requirements, and failures per million hours of operation. Table 5 lists the wattage requirements for typical SEM groupings, as well as the failures per million hours of operation for those same module groupings.

The failure rate of the MRC (configuration per MIL-C-405, 25°C Naval unsheltered environment) is 198 failures per million hours.

Design and construction of the MRP and MRC, in accordance with MIL-STD-1378, MIL-STD-1389, and MIL-M-28787, is centered around a selection of high-quality components and precise manufacturing processes. Unisys experience in producing equipment for defense systems used in military environments provides the techniques for building exceptional quality into the manufactured product. Components selected under the Unisys quality assurance program are assembled into equipment that operates reliably in adverse environments. This same high quality is a characteristic of all modules of the MRP and MRC, thereby assuring high reliability for any configuration.

MAINTAINABILITY - MRP AND MRC

Accessibility to replaceable items, easy testing, and quick malfunction localization are essential to good and efficient maintenance. To meet these issues, Unisys has designed the MRP/MRC for easy operation with a minimum of operator and maintenance actions.

Maintainability

The failure detection and isolation system developed for the MRP and MRC ensures a mean-time-to-repair (MTTR) less than the 15-minute requirement. Implementation of the push-to-test MP and the incorporation of all detection and isolation diagnostics within the system in firmware provide a user-friendly system that simplifies training and operation, and requires no external loading device. Maintenance schemes requiring loading of the diagnostics force maintenance personnel to use time-consuming and difficult manual-isolation techniques whenever a failure prevents diagnostic loading from an external storage device. Typically, one-third of all system failures affect external load capability. The resident diagnostic concept is especially advantageous for MRP appli-

TABLE 4. MODULE KEY CODES, POWER, AND FAILURE RATES PER MILLION HOURS

Key Code	Module Name	Power (Watts)	Failure Per Million Hours
AEK	MIL-STD-188C/RS-232-C Async Serial Control	4.4	2.97
AFK	MIL-STD-188C/RS-232-C/VACALES Sync Serial Control	3.2	3.07
AGL	MIL-STD-1397 Type B/PIC Interface	4.5	1.55
AHL	MIL-STD-1397 Type C/PIC Interface	3.5	1.34
AJK	MIL-STD-1397 Type D/E Serial Control	3.3	2.20
AJL	MIL-STD-1397 Type D Serial Interface	4.2	3.00
AML	EIA-STD-RS-232-C	2.8	0.91
APC	MIL-STD-188C/VACALES Interface	3.3	1.17
ATC	MIL-STD-1397 Type E Serial Interface	4.2	2.20
ATK	NATO-STD-4156 Serial Control	6.6	2.30
ATL	NATO-STD-4156 Serial Interface	6.8	3.90
AWB	MIL-STD-1397 Parallel Channel Control	1.7	1.82
AWC	MIL-STD-1397 Type A/PIC Interface	4.2	1.71
AWF	IOC Control Memory	5.8	2.74
AWG	IOC Data Bus Lower	4.3	2.60
AWH	IOC Channel Control	2.2	2.54
† BWAA-XXX	IOC Sequencer	6.7	3.05
BWAC-XXX	CP Sequencer	7.6	3.06
BWD	Bus Arbitrator	3.8	2.91
BWE	Arithmetic Logic Unit (ALU)	7.4	2.48
BWF	IOC Common Bus Control	4.5	2.52
BWG	Memory Mapped IOA	7.5	3.29
BWAH-XXX	Maintenance Processor	6.2	3.10
CWA	MMIO Bus Interface	4.9	2.43
CWAF-XXX	Bootstrap/RTC	5.4	2.72
CWE	Shift Matrix/General Registers	5.7	2.80
CWG	Memory Interface 1	3.9	2.05
CWH	BIT Electronics	4.5	2.49
CWK	Address Generation	5.6	2.83
DWAE-XXX	Multiply/Math Pac	7.9	3.13
DWF	Memory Address Expansion	7.1	2.57
* DWG	Direct Memory Access I	1.6	1.67
* DWH	Direct Memory Access II	6.2	2.04
EWC	IOC Data Bus Upper	3.8	2.42
EWD	Memory Interface 2	3.1	2.27
EWAH-XXX	Emulation Control 1	7.6	3.66
FWA	Emulation Control 2	7.0	2.58
FWB	Emulation Control 3	5.0	2.60
* GTC	RCMI Breakpoint	7.8	4.14
* GWA	Subsystem Adapter, MIL-STD-1553B	7.8	2.35
* GWB	Common Bus Interface MIL-STD-1553B	7.0	2.47
* HRAA-XXX	RCMI Serial Interface	7.5	4.63
JEJ	128K Dynamic Random Access Memory Array	6.5	82.15
JEQ	Semiconductor Memory Timing	4.3	1.43
JML	Semiconductor Memory Data	6.5	2.14
JMM	Semiconductor Memory Address	6.4	1.94
JMN	Semiconductor Memory Control	6.1	1.69

NOTES: Unless otherwise specified, these failure rates were developed assuming a maximum configuration MRP in a typical environment (MRP embedded in an MRC cabinet which is subjected to a 25°C Naval sheltered environment). As an approximation to this environment, each module rib was assumed to be a temperature of +40°C (MIL-HDBK-217D).

* Same conditions as above, but Naval unsheltered environment.

† Key code dash number identifies specific PROM contents.

TABLE 5. TYPICAL SEM GROUP FAILURE RATES AND POWER REQUIREMENTS

SEM Groups	Nomenclature	Power (Watts)	Failures Per Million Hours
CENTRAL PROCESSOR			
CP without Multiply/Math Pac	CP-1607/U	51.5	23.39
CP with Multiply/Math Pac		59.4	26.52
Multiply/Math Pac	MU-805/U	7.9	3.13
Control Maintenance Interface	C-11599/U	10.7	5.59
Memory Interface	MK-2266/U	7.0	4.32
NDRO Memory/RTC/Monitor Clock		5.4	2.72
Memory Address Expansion	C-11598/U	7.0	2.57
Memory Mapped I/O			
Memory Mapped Bus Interface (MMBI)	J-4128/U	4.9	2.43
Memory Mapped Input/Output Adapter (MMIOA)	J-4119/U	7.5	3.29
1 Channel		12.4	5.72
16 Channels		124.9	55.07
INPUT/OUTPUT CONTROLLER	C-11354/U	34.7	18.35
I/O CHANNEL ADAPTERS			
MIL-STD-1397 Type A/PIC (NTDS Slow)			
2 Channels	MK-2262/U	10.1	5.24
16 Channels		80.8	41.92
MIL-STD-1397 Type B/PIC (NTDS Fast)			
2 Channels	MK-2263/U	10.7	4.92
16 Channels		85.6	39.36
MIL-STD-1397 Type C/PIC (ANEW)			
2 Channels	MK-2264/U	8.7	4.50
16 Channels		69.6	36.00
MIL-STD-188C Synchronous/VACALES			
2 Channels	MK-2265/U	9.7	7.31
16 Channels		77.6	58.48
MIL-STD-188C Asynchronous			
2 Channels	MK-2181/U	12.1	7.11
16 Channels		96.8	56.88
EIA-STD-RS-232-C Synchronous			
2 Channels	MK-2268/U	9.2	7.05
16 Channels		73.6	56.40
EIA-STD-RS-232-C Asynchronous			
2 Channels	MK-2180/U	11.6	6.85
16 Channels		92.8	54.80
MIL-STD-1397 Type D			
2 Channels	MK-2500/U	10.8	7.40
16 Channels		86.4	59.2
MIL-STD-1397 Type E			
2 Channels	MK-2501/U	10.8	6.60
16 Channels		86.4	52.80
NATO-STD-4156			
2 Channels	MK-2182/U	20.0	8.50
16 Channels		160.0	68.00

NOTE: These failure rates were developed assuming a maximum configuration MRP in a typical environment (MRP embedded in an MRC cabinet which is subjected to 25°C Naval sheltered environment). As an approximation to this environment, each module rib was assumed to be at a temperature of +40°C (MIL-HDBK-217D). Failure rates do not include MRC EMP modules.

cations, which often have limited maintenance controls. The diagnostics and BIT system detect 98 percent of all MRP malfunctions, exceeding the specification requirement of 95 percent. BIT isolates 80 percent of all MRP failures to one card, 95 percent of all MRP failures to two cards, and 98 percent of all MRP failures to three cards. The system configuration is determined without any user input. Execution of BIT within the operational runstream provides limited realtime equipment status and fault detection.

Accessibility to the line replaceable unit (LRU) is facilitated in the design of the MRC cabinet. With the exception of the rear–mounted I/O channel cable connectors, all replaceable elements of the MRC are accessible through the front cabinet doors. When the MRC cabinet front door is open, the SEM cage is exposed and the SEMs are accessible for removal or insertion. Any SEM is removable using the enclosed, simple extraction tool. Each SEM has its unique keyed position and is guided so that it is aligned properly to the chassis connector when inserted. When the SEM cage is swung out, the memory modules and the power supply modules are accessible. The power regulators, power conditioners, and memory modules can then be removed from the cabinet using standard tools.

Easy testing is provided through an easy–to–use operator panel (MRC only) or through the optional RCMI interface for remote execution and BIT microcode that obviates the need for a memory–space required macrodiagnostic software program. Testing requires that the operator only be acquainted with a few indicators and switches on the MRC and be able to interpret computer responses in the readout displays on the operator panel.

BIT can be activated as online macroinstructions, as a press–to–test operation enabled through the operator panel or the RCMI interface, and at initialization when the MRC/MRP is powered up. Selected subfunctions of BIT can be executed from BIT macroinstructions.

BIT is executed by the MP which is part of the CMI SEM set. BIT consists of a series of tests which determine the operational status of the MP, CP, IOC, and all options. BIT also contains tests for core and semiconductor memory and loopback tests for the I/O channel adapters. The 45 BIT subtests are executed when initiated using power–up or press–to–test operator initiated actions.

When executed from the macroinstructions, 13 of the BIT subtests can be executed. Two unique BIT macroinstructions are Initiate CP or IOC BIT (code 00, m = 1, octal; code 00, m = 1, hex; code 77 3, octal; FF, hex) and initiate MP BIT (code 00, m = 4, octal; code 00, m = 4 hex). In Initiate CP or IOC BIT, the BIT test sequencer in the CP or IOC saves SR 1 and MACRO P on entry and restores them on exit. No general registers are saved but page registers and main memory are left intact. It is not possible to interrupt a test and continue the test from the point of interrupt. If a macro level interrupt should occur, control is returned to the operational program within 5 microseconds by the interrupt return software.

If the test passes with no macro interrupt, bits 9–8 in SR 1 are zero and location 170 octal, 78 hex contains 177777 octal, FFFF hex. If the test has failed, there is will be a non–all–one error condition code in location 170, octal, 78, hex. While operating from macroinstruction, the data contained in locations 170–175, octal; 78–7D, hex are invalid for isolation purposes since only a fraction of the total test can be run in this interruptible mode. If a macrointerrupt stopped the test, bits 9–8 of SR 1 is 01 and the information in location 170, octal, 78, hex is invalid. (See Table 6 for a listing of the microcode BIT tests. Table 7 contains macroexecutable BIT tests.)

Initiate BIT macroinstructions require the interaction of the MP with the CP. These instructions are: code 00, m = 2, octal; code 00, m = 2, hex – Transfer MP RAM to main memory; code 00, m = 3, octal; code 00, m = 3, hex – Transfer main memory to MP RAM; code 00, m = 4, octal; code 00, m = 4, hex – Initiate MP BIT. The transfer instructions permit the program to save the contents of locations 170–175 octal, 78–7D hex and to preset those save locations from main memory.

On the Initiate MP macroinstruction, SR 1, SR 2, MACRO P, and switch discretes are saved. Everything but main memory is destroyed. The instruction is noninterruptible, and on exit to the CP, SR 1, SR 2, MACRO P, and the switch discretes are restored and the CP is placed in the Macro Run mode.

The macroinstruction repertoire also includes instructions which permit the user to obtain the results of the EDC features of the MRC semiconductor memory.

Operation code 00, m = 6, octal; code 00, m = 6, hex returns the semiconductor memory status register in R_a.

Operation code 00, m = 7, octal; code 00, m = 7, hex sets the error correct enable bit in the semiconductor memory status register.

TABLE 6. MICROCODE BIT TESTS

Test code (Hexadecimal) TT	Subtest Selected
01	MP PROM (MS0PRM)
02	MP RAM (MS0RAM)
03	MP NSC800 (MS0ALU)
04	BIT electronics SMI (MS0SMI)
05	(1) MDS PROM Checksum (MS0MAM)
06	(1) MDS RAM (MS0WPN)
07	(1) Panel (MS0WPN)
08	CP sequencer checksum (MS1CSQ)
09	CP sequencer (MS1QSQ)
0A	CPU ALU (MS1PAL/CS1PAL)
0B	CP address generation (MS1ADG/CS1ADG)
0C	CP status register (MS1RST/CS1RST)
0D	CP general register stack (MS1GRS/CS1GRS)
0E	CP ECW PROM (MS1SEC/CS1SEC)
0F	CP bus arbitrator (MS1BAR/CS1BAR)
10	CP jump/stop detect (MS1JSD/CS1JSD)
11	CP main memory loopback (MS1VML/CS1VML)
12	CP emulation control (MS1EMC/CS1EMC)
13	CP microbranch (MS1FBR/CS1FBR)
14	CP SMASH chip (MS1TSM/CS1TSM)
15	IOC sequencer (Part 1) (MS1ISQ)
16	IOC sequencer (Part 2) (MS1WSQ)
17	IOC ALU (MS1OAL/CS1OAL)
18	IOC control memory (MS1KCM/CS1KCM)
19	IOC channel control (MS1HCS/CS1HCS)
1A	IOC bus cycle (MS1DBC/CS1DBC)
1B	IOC common bus loopback (MS1LCB/CS1LCB)
1C	IOC interrupt stack (MS1YIS/CS1YIS)
1D	IOC data transfers (MS1NDT/CSINDT)
1E	CP interrupt generation (MS1MIG/CS1MIG)
1F	CP interrupt priority (MS1UIP/CS1UIP)
20	CP math pac (MS2PAC/CS2PAC)
21	CP page register (MS2RPG/CS2RPG)
22	CP memory address expansion (MS2EXP/CS2EXP)
23	CP array multiply (MS2ARM/CS2ARM)
24	CP bootstrap checksum (MS2NDR/CS2NDR)
25	CP RTC (MS2CLK/CS2CLK)
26	P history and breakpoint (MS2HPB/CS2HPB)
27	(2) CP memory interface (MS2BMI/CS2BMI)
28	(3) MMIO (MS2BMI/CS2BMI)
29	(2,4) Main Memory (MS2SCM/CS2SCM)
2A	(2,4) Error Detect/Correct (MS2EDC/CS2EDC)
2B	(4) Memory DMA (MS2DMA/CS2DMA)
2C	IOC interface adapter (MS2XIO/CS2XIO)
2D	IOC channel loopback-all channel types (MS2LPB/CS2LPB)
30	(5) SEM memory response (MS2MRS/CS2MRS)
31	(5) SEM memory data loopback (MS2DLB/CS2DLB)
32	(5) SEM memory status memory (MS2MSR/CS2MSR)
33	(5) SEM memory configuration register (MS2MCR/CS2MCR)
34	(5) SEM memory address loopback (MS2MAL/CS2MAL)
35	(5) SEM memory error logic RAM (MS2MEL/CS2MEL)
36	(5) SEM memory error detect/correct logic (MS2MED/CS2MED)
37	(5) SEM memory array (MS2MAR/CS2MAR)
38	(5) SEM scrub (MS2MSC/CS2MSC)
50	RCMI PROM checksum (MS0RCMI1)
51	RCMI control register (MS0RCMI0)
52	RCMI RAM (MS0RCMI2)

TABLE 6. MICROCODE BIT TESTS (continued)

Test Code (Hexadecimal) TT	Subtest Selected
53	RCMI loopback (MS0RCMI3)
54	RCMI block transfer (MS0RCMI4)
55	RCMI baud rate (MS0RCMI5)
60	Breakpoint RAM test (MS0BR)
61	Breakpoint FIFO test (MS0FIFO)
62	Breakpoint master write test (MS2BHW)
63	Breakpoint common bus buffer test (MS2BPCB)
64	Breakpoint stop logic test (MS2BPS)
NOTES: (1) MDS (2) 6 x 9-inch Semiconductor memory only (3) MRP/MDS only (4) MRC only (5) SEM Semiconductor memory only	

TABLE 7. MACROEXECUTABLE BITS SUBTESTS

Test No.	CP/IOC	Test Name
0B	CP	Address Generation
0C	CP	Status Register
0E	CP	ECW PROM
11	CP	Main Memory Loopback
15	CP	SMASH Chip
20	CP	Math Pac
23	CP	Array Multiply
24	CP	Bootstrap Checksum
1B	IOC	Channel Control
1C	IOC	Bus Cycle
1D	IOC	Common Bus Loopback
1E	IOC	Data Transfers
2C	IOC	Interface Adapter

Operation code 00, m = 10, octal; code 00, m = 8, hex clears the error correct enable bit in the semiconductor memory status register.

The 6 x 9-inch module semiconductor memory status register is formatted as shown in Figure 21. The SEM semiconductor memory status register is formatted as shown in Figure 22.

Error logging is also a feature of the MRC semiconductor memory. Operation code 00, m = 11, octal; code 00, m = 9, hex, permits a search of the error log for an indication of an error being detected/corrected in the semiconductor memory. The user supplies a beginning search address (R_a) and the instruction returns data in (R_a and R_{a+1}). If no errors are detected, R_a is unchanged and R_{a+1} is set to the error log address (right justified, zero filled) at which an

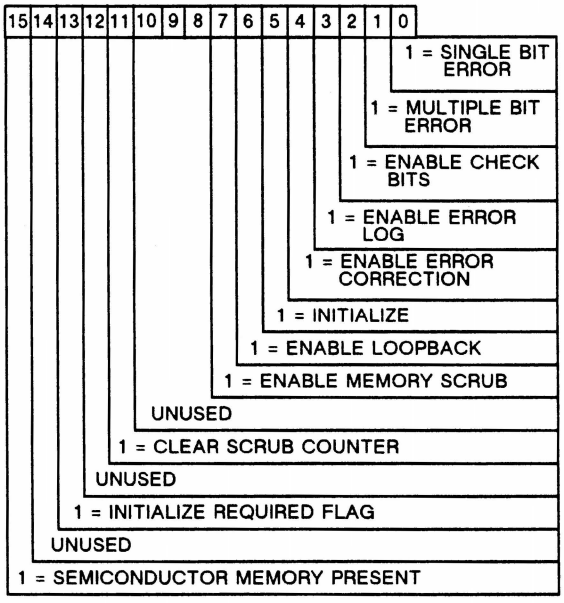
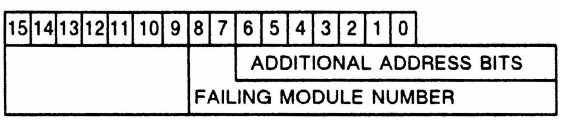


Figure 21. Semiconductor Memory (6 x 9-inch) Status Register

error indication was detected. The semiconductor memory status register error bits are cleared.

Error Log Address Bit Definition:



The user should be acquainted with the various registers, macroinstructions, BIT tests, etc., and, with judi-

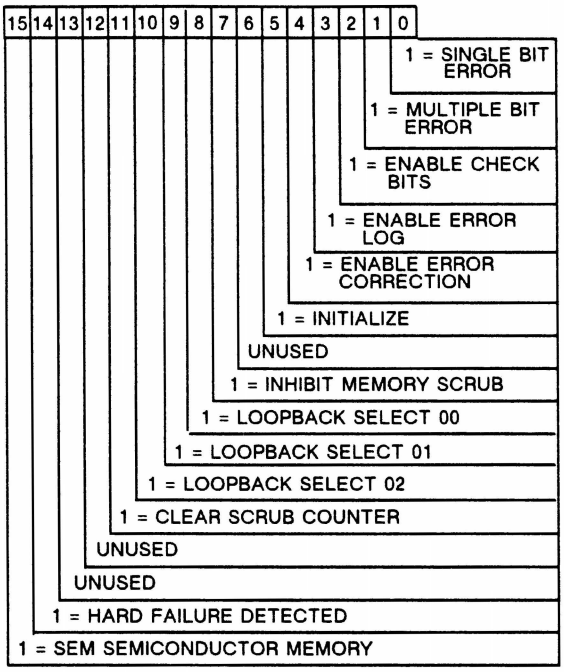


Figure 22. SEM Semiconductor Memory Status Register

cious programming, permit the operational software to perform validity checks of the computer as an aid to hardware error conditions. In this way, the user can detect errors before those errors manifest themselves in erroneous data generation or incorrect computer operation. The user may find that such program checks can be used to flag error conditions as they are detected rather than attempting to detect those error conditions during normal maintenance periods of operation.

FUNCTIONAL OPERATION - MRP AND MRC

Status Control

Status control is provided by two 16-bit, high-speed registers: SRs 1 and 2.

The format for SR 1 is divided into fields that indicate MRP/MRC status, interrupt status, and conditions resulting from arithmetic operations (see Figure 10).

Details of field designators are:

a) Bit 0 = 0, disables DMA
Bit 0 = 1, enables DMA

b) Interrupt enable designator:
Bit 1 = Class III
Bit 2 = Class II
Bit 3 = Class I

When the bit is set, the respective class interrupt is enabled (can be honored).

c) Bits 4 and 5 select one of four page register sets; selected set number corresponds to 2-bit binary code.

d) Floating-point residue designator bit 6 = 0 specifies that the residue, in a floating-point arithmetic operation, is not saved. Bit 6 = 1 specifies that the residue is saved.

e) Floating-point overflow/underflow interrupt disable bit 7 = 0 enables the floating-point overflow/underflow interrupt when that result occurs, and bits 11-4 of the instruction register are stored into bits 7-0 of SR 2. If bit 7 = 1, no interrupt is generated on floating-point overflow or underflow.

f) The condition code (bits 9 and 8) indicates the results of arithmetic and compare instructions as shown in Table 8.

TABLE 8. CONDITION CODE INDICATIONS FROM STATUS REGISTER 1

Condition Code		Indicated Result of	
Bit 9	Bit 8	Arithmetic Operation	Compare Operations
0	0	Zero	$R_a = R_m$ or Y
0	0	Not Zero and Positive	$R_a > R_m$ or Y
1	1	Not Used	Not Used
1	1	Negative	$R_a < R_m$ or Y

g) The overflow designator (bit 10) is set when an arithmetic or a shift operation produces a result that requires more bits than provided in a register.

h) The carry designator (bit 11) is set when an arithmetic operation generates a carry beyond the most significant bit in the register.

i) Floating-point underflow/overflow designator (bits 11 and 10) is set when a floating-point operation causes a characteristic underflow or

overflow. Bit 11 = 1 and bit 10 = 1 indicate a floating-point underflow. Bit 11 = 0 and bit 10 = 1 indicate a floating-point overflow. Bits 9 and 8 are not used in either case.

j) The NDRO mode (bit 12) directs the CP to select memory as follows for addresses 00 through 77, octal, and 300, octal, through 477, octal:

Bits 4, 5 and 12 = 0, use NDRO memory
Bit 12 = 1, use main memory.

k) The general register designator (bit 14) specifies the set of 16 general registers selected by the a- and m-designators in the instruction word. Bit 14 = 0 selects set 0 and bit 14 = 1 selects set 1.

l) The CP mode designator, bit 15 = 0, CP is in the executive mode. Bit 15 = 1, CP is in the task mode. Instructions designated as executive mode instructions can only be executed when CP is in the executive mode.

The format for SR 2 is divided into fields which control MRP/MRC direct and indirect addressing processes and which hold memory resume interrupt and I/O instruction fault data. SR 2 is shown in Figure 19. Details of the field designators are:

a) Bit 15-8 direct and indirect addressing. Bits 15-14 are interpreted if m = 16, bits 13-12 if m = 14, bits 11-10 if m = 12, and bits 9-8 if m = 10. Each 2-bit field is interpreted as:

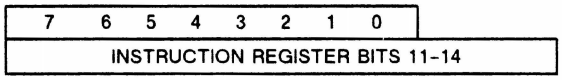
00 Normal Addressing, Operand at Y = y + (Rm)
01 Normal Addressing, Operand at Y = y + (Rm)
10 Indirect Addressing without Indexing IW 1 at Y = y
11 Indirect Addressing with Indexing IW 1 at Y = y + (Rm)

b) Bits 7-0 Class I and II interrupt information for memory resume, parity error, IOC Instruction Fault, and protect fault:

1) Bits 7-6 IOC Number (zeros if CP interrupt)
2) Bits 5-2 I/O Channel Number (zeros if CP interrupt)
3) Bits 1-0 as follows:

00 Input Chain
01 Output Chain
10 CP
11 I/O Command

c) Bits 7-0 Floating-Point Error:



Status register bit content can be set or cleared by executing the load status register instruction or the load program status word instruction, and can be initialized for a class interrupt processing subroutine by loading the SR 1 memory location assigned to that particular interrupt class.

Instructions

Instructions defining operations for the MRP/MRC are designed to maximize circuit effectiveness in attaining high-speed computer functions. The large set of flexible, comprehensive single- and double-word instructions give the MRP and MRC power equal to many large-scale computers, and capabilities beyond many minicomputers. Appendix A contains a repertoire of instructions; Appendix B lists the CP instructions and the execution time for each instruction in the applicable formats.

Within the repertoire are many instructions that speed up the capability of application programs and provide greater flexibility for programmers including:

a) *Biased fetch* which allows the CP to check on the performance of tasks it assigns to the I/O chaining program.

b) *Reverse register* feature which enables the reversing of a stream of data received from a communication system and which must be transmitted to another system in reverse order.

c) *Scale factor shift* instruction which provides a left-shift function that positions the word for greater significance and counts the number of bit positions shifted.

d) *Local jump* instructions which are storage space and time savers in all systems designed around the natural looping method of programming. These saving benefits are apparent in both the program-generated and job-processing phases.

e) *Jump and link* instructions which fill the requirement for linking to re-entrant routines. Because these routines cannot be changed internally, the linking is done externally either through general registers or main memory.

f) *Set bit, clear bit, and test bit* instructions which provide a fine-grain examination and change capability that is useful in realtime communica-

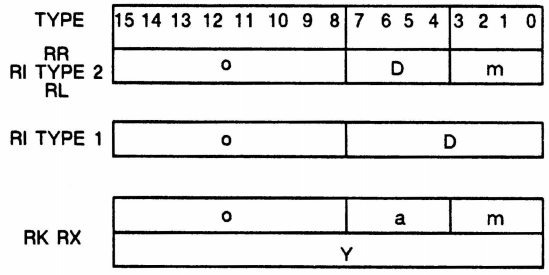
tions. Interacting tasks that communicate by flags and status words benefit highly by this flexible, bit–handling feature.

Figure 23 defines the 16–bit instruction word formats and the 32–bit, 2–word instruction formats. Single–word formats can be used when operands are manipulated in high–speed general registers. Double–word formats are used for operations requiring direct/indirect memory references with or without indexing and those that provide programmers the convenience of listing 16–bit constants in line with instructions. Programs that can be constructed with a high ratio of 1–word instructions to 2–word instructions greatly increase the computing speed and also occupy less memory space.

Math Pac – MRP and MRC

An additional set of arithmetic instructions that improves programming versatility and increases computational throughput is provided by the Math Pac hardware option. The following functions are included:

a) *Square root* instruction for scientific applications.



o OPERATION (FUNCTION) CODE AND FORMAT DESIGNATOR LOWER TWO BITS DEFINE FORMAT:
X0, X4, X8, XC = FORMAT RR OR RL-1
X1, X5, X9, XD = FORMAT RI OR RL-2
X2, X6, XA, XE = FORMAT RK OR RL-3
X3, X7, XB, XF = FORMAT RX OR RL-4

a GENERAL REGISTER OR SUBFUNCTION DESIGNATOR

m GENERAL REGISTER OR SUBFUNCTION DESIGNATOR 4-BIT UNSIGNED LITERAL CONSTANT IN RL FORMAT

D SIGNED DEVIATION VALUE (TWO'S COMPLEMENT)

Y ADDRESS OR ARITHMETIC CONSTANT

Figure 23. Instruction Word Format

b) *Hardware trigonometric and hyperbolic* functions are calculated by a linear approximation method that uses the general registers for parameter manipulation. Some functions provided by the one instruction include trigonometric and hyperbolic rotate, trigonometric and hyperbolic vector, $\log_e x$, exponential, polar–to–cartesian conversion, sin 0, and cos 0. Other functions related to these mathematical processes may be obtained by proper choice of input parameters.

c) *Floating–point* add, subtract, multiply, divide, compare, square root, normalize, conversion to/from fixed single, exponential, logarithm, and trigonometric instructions executed by the microprogrammed controller are much faster than ordinary floating–point subroutines in the system programs.

d) *Double–precision multiply and divide* instructions allow for more direct processing of double–length operands and for greater precision.

e) *Algebraic left and right quadruple–shift* instructions allow for faster shifting of quadruple–length operands.

Instruction Addressing

A program address (P) register is an incremental counter in the CP that specifies the address of the next instruction to be executed by the CP. As an instruction is executed, the register is advanced to the next sequential instruction address in memory. Executing a single–word instruction advances the register by 1; a double–word instruction advances it by 2. Any jump instruction executed with its jump condition satisfied changes the address in P and a new program sequence begins at that address. Local jump instructions, however, limit the change to the address in P to + 7F/–80, hex locations.

The P register is not used to read the next sequential instruction from memory. To gain a speed advantage, the MRP/MRC prefetches instructions addressed from a hardware/firmware–controlled instruction address register into a forward instruction queue. The P register merely keeps track of what instruction is being executed.

I/O operations use chain address pointer locations in control memory as instruction address counters. Each input and each output channel is assigned an address pointer that advances like a P–register as it executes its instructions in its program chain. If a jump is desired in the program, the chain address pointer is changed either by executing a load control memory instruction or the conditional jump instruction.

The a–, m–, and y–designator fields in the instructions define a variety of functional operations and parameters. Collectively, this variety offers a programmer much flexibility. For a computing system that interprets simple instruction formats with the variety of field assignments, speed is the reward. The general application of the a–, m–, and y–designator fields is described in the following paragraphs. Special assignments and uses are defined in the individual instruction descriptions.

Instruction Word Formats

RR format instructions perform operations involving general registers; no main memory references are made for operands. The a– and m–designators select the general registers designated R_a and R_m, respectively, that are used in the operation.

RL format instructions perform operations involving one or two general registers. The a–designator selects the general registers designated R_a and R_{a+1}. The m–designator is an unsigned literal that is used in the operation.

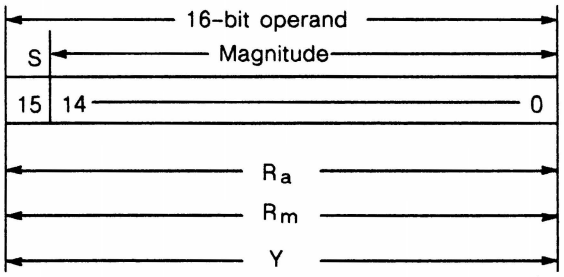
RI format, type 1 instructions are local jump operations that either increase or decrease the contents of P by the value D in the instruction. The effective jump address Y = (P) + D, where D is the two's complement deviation value.

RI format, type 2 instructions perform operations that involve general registers and a main memory reference. The a– and m–designators select general registers designated R_a and R_m, respectively. R_m, however, contains an address Y that is used for the main memory reference.

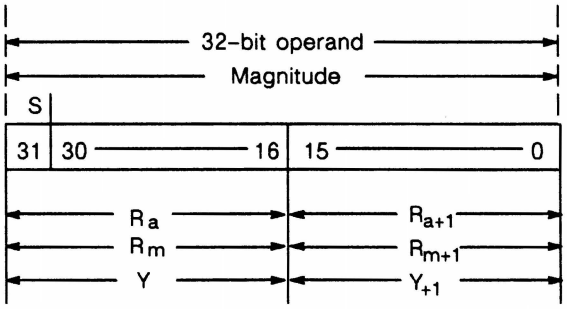
RK format instructions are 2–word instructions stored in two numerically adjacent memory locations. The first word contains the operation code and designator fields; the second word is a value y that can be used as a constant operand or address, or as a modified constant or address. The a–designator selects a general register designated R_a. When m = 0, the operand or address Y equals y and no R_m is selected; the operand Y equals y plus the contents of R_m, that is, y is indexed by the contents of R_m. (Operand Y is used as an address in RK format jump instructions and in the remote execute instructions.) For all I/O instructions, RK and RX formats, Y = y (no indexing or indirect addressing is performed).

RX format instructions are 2–word instructions stored in two numerically adjacent memory locations. The first word includes the a– and m–designators, and the second word contains the y–value. RX format instructions perform byte (8 bit), whole word (16 bit), and double–word (32 bit) operations with general registers and memory references. The a–designator selects a general register, designated R_a, for all three types of operands. For all I/O instructions, RK and RX formats, Y = y (no indexing or indirect addressing is performed).

Single–Length Operands. Instructions that perform operations with 16–bit words (single length) use one register R_a, R_m, or memory address Y.



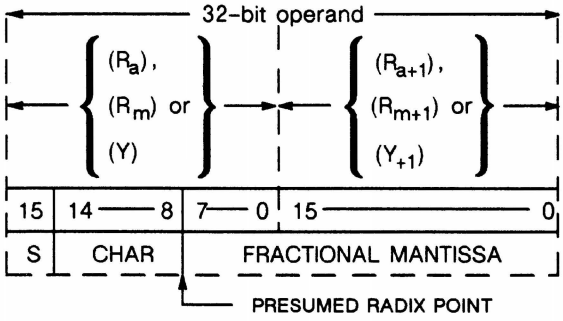
Double–Length Operands. Instructions that perform operations with 32–bit words (double length) use two adjacent registers and memory locations for each operand. The word in R_a, R_m, or in memory address Y, when selected by double length instructions, is the most significant half of the operand and contains the sign bit for both words (a, m, or Y must be even numbered). The word in R_{a+1}, R_{m+1}, or in memory address Y+1 is the least significant half of the operand. Double–length operands fetched from memory are required to have Y equal to an even address.



Floating–Point Operands. Floating–point addition, subtraction, multiplication, and division can be performed with or without a residue. The process uses a 2–word operand in the format shown below. For maximum accuracy, normalized operands should be used.

Word 1 of the operand, stored in R_a, R_m, or memory address Y, contains the algebraic sign (S) of the fractional mantissa, a biased characteristic in the range 0 $\leq$ C $\leq$ 177 (octal), 0 $\leq$ C $\leq$ 7F (hex) and the two

most significant hex digits of the fraction. Word 2 of the operand, stored in Ra+1, Rm+1, or memory address Y+1, contains the four least significant hex digits of the fractional mantissa. A normalized floating-point number has a nonzero hex digit in the most significant four bits of the fractional mantissa. When a residue is requested by the program, the CP stores the result in Ra, Ra+1 and stores the unused lower order digits (residue) in general registers Ra+2 and Ra+3 in floating-point data format.



a, m and address Y are even numbers

A change of one in the characteristics represents one hexadecimal digit position shift (4 bits). Therefore, the magnitude (M) of a floating-point number is approximately $5.4 \times 10^{-79} < M < 7.2 \times 10^{75}$. A zero quantity is represented by a positive sign (0), a zero characteristic, and a zero fractional mantissa.

Operand Addressing. The operand addressing process provides much programming flexibility. Direct addressing, indirect addressing, or cascaded indirect addressing or operands can be selected with the RX format instructions. When the instruction m-designator equals zero, direct addressing without indexing is selected (address Y = y). Direct addressing with indexing is specified when the m-values 1 through 7, 11, 13, 15 or 17, octal; 7, 9, B, D, or F, hex are used (address Y = y + (Rm)). The general registers specified by these m-values contain the indexing modifier. When the m- value 10, 12, 14, or 16, octal; 8, A, C, or E, hex is specified, corresponding indirect control fields in SR 2 are interpreted to generate the address of the operand or a pair of indirect words (IW 1 and IW 2) as illustrated in Figure 24.

If control bits in the field interpreted equal 00 or 01 (binary), direct addressing results as though m = 1 through 7, 11, 13, 15, or 17, octal; 7, 9, B, D, or F, hex; that is, Y = y + (Rm). But when m = 10, 12, 14, or 16, octal; 8 A, C, or E, hex, field interpreted equals 10 or 11 (binary), the addresses Y and Y+1

OCTAL J-VALUE	ADDRESS DETERMINATION
0	OPERAND AT ADDRESS SPECIFIED BY (IW 2)
1	OPERAND AT ADDRESS SPECIFIED BY (IW 2) + (R_x)
2	OPERAND AT ADDRESS SPECIFIED BY (IW 2) + (R_m)
3	OPERAND AT ADDRESS SPECIFIED BY (IW 2) + (R_{m+1})
4	CASCADED IW AT ADDRESS SPECIFIED BY (IW 2)
5	CASCADED IW AT ADDRESS SPECIFIED BY (IW 2) + (R_x)
6	CASCADED IW AT ADDRESS SPECIFIED BY (IW 2) + (R_m)
7	CASCADED IW AT ADDRESS SPECIFIED BY (IW 2) + (R_{m+1})
10-17	NOT ASSIGNED

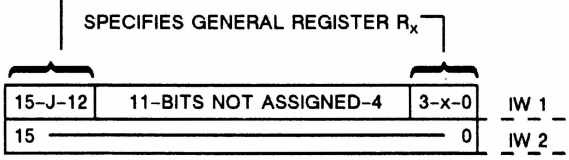


Figure 24. Indirect Word Interpretation

contain a pair of indirect words (IWs) interpreted according to Figure 24. The address Y must be even.

Double-length operations can be assigned to even-numbered addresses or registers for operand references. The user should familiarize himself with the requirements for addressing for those double-length operand instructions. The first or most significant half of an operand is in the even-numbered location. From the even-numbered address or register specified by the instruction (that is, Ra, Rm and Y as applicable), the CP logic selects the next sequentially numbered address or register for the second half of the operand. Memory addresses for the first half of the double-length operand are formed like those for whole-word operands.

Byte (8 bit, half word) operand addressing requires a byte identifier (B); that is, the upper byte or the lower byte in memory:

B = 0 designates the most significant half-word in address Y as the operand byte.

B = 1 designates the least significant half-word in address Y as the operand byte.

The least significant bit (LSB) in the indexing register is the byte identifier, and the value in the remaining

bits is used as the index to generate the effective address as follows:

m = 0, address Y = y and B = 0

m = 1–7, 9, B, D, or F (hex)

m = 1–7, 11, 13, 15, or 17 (octal)

$Y = y + \frac{(R_m)}{2}$ and B = LSB of (R_m).

When indirect addressing is used and

if j = 0, y = (IW 2) and B =0

if j = 1, $Y = (IW\ 2) + \frac{(R_x)}{2}$ and

B = LSB of (R_x)

if j = 2, $Y = (IW\ 2) + \frac{(R_m)}{2}$ and

B = LSB of (R_m)

if j = 3, $Y = (IW\ 2) + \frac{(R_{m+1})}{2}$ and

B = LSB of (R_{m+1})

Interrupts

The CP can be interrupted in its execution of programs. Some interrupts are generated by events within the CP, some within the IOC, and some as interrupt requests by peripheral equipment.

MRP/MRC system interrupts are classified in three priority levels. Interrupts within a class are assigned priority rank within the class, and an identifying code is used by CP logic to select the appropriate processing routine from memory. Table 9 lists the interrupts, their classification, and assigned identity codes. Higher priority is given to the class and the interrupt within the class that has the lower number. As each interrupt is honored, its class and all classes of interrupts can be locked out by the reloaded status register until released by the processing subroutine. Thus, an event in a higher priority class can interrupt a routine that is processing a lower priority class interrupt. The interrupt routine is suspended until the higher level is processed and then is allowed to continue.

RTC and monitor clock registers, when installed, can be loaded, read, enabled, or disabled under program control. When enabled by the appropriate instruction (code 03, m = 10, octal; code OC, m = 8, hex), the RTC register counts up at the rate of the RTC oscillator or external RTC input. As the register lower–order 16 bits overflow (change from all 1s to all 0s), the CP generates the RTC overflow interrupt, and control is transferred to the appropriate processing routine. The RTC register continues to count up until disabled by the disable RTC instruction (code 03, m = 11, octal; code OC, m = 9, hex). The RTC and monitor clock do not advance when the CP is stopped.

The monitor clock register countdown function is enabled by executing the load and enable monitor clock instruction (code 03, m = 12, octal; code OC, m = A, hex) which also loads the register with a starting point.

When the contents of the register decrements from all 0s to all 1s, a monitor clock interrupt is generated, the countdown function is disabled, and control is transferred to the appropriate processing routine. The countdown function can be disabled by programming a disable monitor clock instruction (code 03, m = 13, octal; code OC, m = B, hex).

A memory resume interrupt is generated when a memory module fails to acknowledge a request within a specified time. If Class I interrupts are not enabled when the event happens, the interrupt is lost.

A memory parity interrupt is generated when a noncorrectable error is detected in the semiconductor memory.

A CP instruction fault interrupt is generated when the CP attempts to execute an instruction that is illegal.

An IOC instruction fault interrupt is generated when the IOC attempts to execute an illegal instruction or an IOC does not respond to the CP's execution of the IOC command instruction within 192 microseconds.

The floating–point interrupt is generated whenever a floating–point overflow or underflow occurs and status register 1, bit 7 is set.

The executive return instruction interrupt is generated when the CP executes the code 03, m = 0, octal; code OC, m = 0, hex, instruction.

The executive mode fault interrupt is generated when the CP attempts to execute an instruction designated as an executive mode instruction when the CP is in the task mode.

The memory protect fault interrupt is generated when the CP or IOC (with memory address expansion) attempts to reference a protected page of memory.

Peripheral equipment units may attempt to interrupt the IOC by setting a coded message and an external interrupt request on the input cable. The IOC intercomputer timeout interrupt is generated when the IOC does not receive a resume from the receiving computer within the allotted time after sending an output word or external function (EF) word. Certain channels use this entrance and code to generate an error interrupt to the CP.

TABLE 9. INTERRUPT PRIORITY

Class	Priority Within Class	Interrupt	Binary Interrupt Code (LSBs)	Notes
Class I, Hardware Errors	1	Power Fault	0000	1
	2	IOC, CBI Memory Resume	0010	2
	3	IOC, CBI Memory Parity	0100	2
	4	CP Memory Resume	0010	2
	5	CP Memory Parity	0100	2
Class II, Software Interrupts	1	CP Instruction Fault	00000	1
	2	IOC Instruction Fault (35RR)	00010	3
	3	IOC, Common Bus Interface Instruction Fault	00010	3
	4	IOC Protect Fault	11000	2
	5	Floating Point Overflow or Underflow	00100	4
	6	Executive Return Instruction	00110	4
	7	Executive Mode Fault	10000	1
	8	CP Protect Fault	11000	2
	9	RTC Overflow	01000	5
	10	Monitor Clock	01010	5
Class III, IOC & MMIO Interrupts	1	IOC Intercomputer Timeout (SIF Errors)	110	6
	2	IOC External Interrupt/Discrete Interrupt	000	6, 7
	3	IOC Output Chain, CBI Exception/Mode Code Interrupt	100	6
	4	IOC Input Chain, CBI Programmable Interrupt	010	6
	5	MMIO Discrete Interrupt	110	8, 9
	6	MMIO External Interrupt	000	8, 9
	7	MMIO Output Data Ready Interrupt	100	8, 9
	8	MMIO Input Data Ready Interrupt	010	8, 9

NOTES:
1. Cannot be locked out.
2. Interrupt is lost if locked out.
3. Interrupt action is not locked out within the IOC, but the interrupt is lost if locked out by the CP.
4. No operation if locked out.
5. One level of queuing.
6. One level of queuing per channel.
7. Discrete interrupt for MIL-STD-188C, VACALES, or EIA-STD-RS-232-C
8. Bits 3 through 8 define the MMIO channel number.
9. MRP only.

The IOC external equipment interrupt is generated when the IOC has stored an external interrupt word (code) from a peripheral equipment in an assigned memory location (see Table 2) for the channel, and the external interrupt monitor is enabled by the program.

The IOC output chain interrupt is generated when the IOC executes the code 73, a = 1, octal; code EC, a = 1, hex, instruction from an output chain.

The IOC input chain interrupt is generated when the IOC executes the code 73 a = 1, octal; code EC, a = 1, hex, instruction from an input chain.

Interrupt Processing

When an interrupt is honored, the CP hardware enters the following interrupt processing sequence:

a) Terminates the current program sequence and locks out all interrupts.

b) Stores the contents of P, SR 1, SR 2, and the RTC register in assigned main memory as shown below.

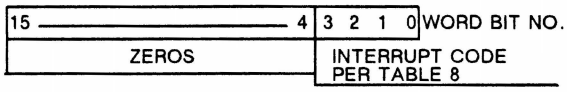
Function	Address Assignment to Class III	Address Assignment to Class II	Address Assignment to Class I
Stores the contents of P at address	110_8 48_{16}	120_8 50_{16}	130_8 58_{16}
Stores the contents of SR 1 at address	111_8 49_{16}	121_8 51_{16}	131_8 59_{16}
Stores the contents of SR 2 at address	112_8 $4A_{16}$	122_8 52_{16}	132_8 $5A_{16}$
Stores the contents of RTC lower at address	113_8 $4B_{16}$	123_8 53_{16}	133_8 $5B_{16}$
Stores the contents of RTC upper at address	117_8 $4F_{16}$	127_8 57_{16}	137_8 $5F_{16}$

c) Reloads the P, SR 1, and SR 2 from assigned memory locations as shown below. Interrupt lockouts and release is controlled by the program using the load status register instruction (code 03, m = 5, octal; code OC, m = 5, hex), or load PSW instruction (code 07 3, octal; code 1F, hex).

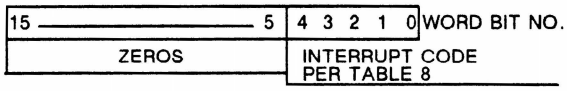
Function	Address Assignment to Class		
	III	II	I
Reloads P with index(1) plus the contents of address	114_8 $4C_{16}$	124_8 54_{16}	134_8 $5C_{16}$
Reloads SR 1(2) from address	115_8 $4D_{16}$	125_8 55_{16}	135_8 $5D_{16}$
Reloads SR 2 from address	116_8 $4E_{16}$	126_8 56_{16}	136_8 $5E_{16}$

(1) See Figure 25 for index values.
(2) SR 1 bits 3-1 control interrupt lockout/release.

CLASS I INDEX WORD



CLASS II INDEX WORD



CLASS III INDEX WORD

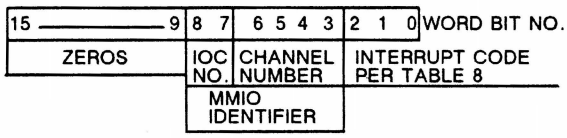


Figure 25. Interrupt Entrance Address Index

d) Resets SR 1 bit 15 to executive mode.

e) Enables honoring interrupts not locked out by new contents of SR 1.

f) Executes the instruction at address in P and continues the program sequence from that point.

I/O Operation

Two types of instructions in the IOC repertoire control I/O operations: (1) chaining instructions are executed under control of an active channel chain; and (2) command instructions are executed under direction of the main program. Table 10 defines the IOC instructions.

I/O Channel Status. Each channel has a status word, as shown in Figure 26, that can be read by the code 76 3, octal; code FB, hex, instruction.

MIL-STD-1397 Parallel Input Interface Communication. When a peripheral equipment is ready to transmit data or an interrupt code, it places the information on the data lines and raises the input data request line or the interrupt request line, respectively. The IOC, at its convenience, stores the word in memory and answers either request on the input acknowledge line.

MIL-STD-1397 Parallel Output Interface Communication. When a peripheral equipment is ready to accept a command, it raises the EF request line to the IOCA. At its convenience, the IOCA places a command code on the output data lines and sets the EF acknowledge line. In another method, the IOCA can force command words to external equipment, in which case the EF request line need not be set (refer to IOC instruction code 70 3, a = 3, octal; code E3, a = 3, hex). The IOCA places the command code on the data lines and sets the EF acknowledge. The peripheral equipment reads the code and performs as commanded. When the peripheral equipment is ready to receive data, it raises the output data request line and the IOCA responds, at its convenience, by placing a data word on the output lines and sets the output acknowledge line. EFs can be performed from either input or output chains. The parallel interface lines are shown in Figure 11.

MIL-STD-1397 Parallel Intercomputer Communication. Any parallel I/O channel can be used to communicate with another computer having a compatible interface. The logic of the intercomputer output channel provides a timeout interrupt if the receiving computer does not accept an output or EF word, or is too slow in responding.

MIL-STD-1397 Peripheral Input Channel. This is useful in communicating with a computer that does not have an available intercomputer channel. It allows the MRP/MRC equipped with a peripheral input channel to perform like an ordinary input channel of a peripheral. This channel is designed to capture data when presented rather than at its convenience, thus assuring the transmitting computer successful transfers. The peripheral input channel has characteristics of the MIL-STD-1397 Type A, Type B, and Type C interfaces.

TABLE 10. IOC INSTRUCTION REPERTOIRE

Code			Instruction, Command and Chaining	
Octal	Hexadecimal	Format		
70 0	E0	RR	Channel Control	
			Instruction and Type	
			Command	Chaining
70 2	E2	RK	Illegal	Illegal
70 3	E3	RX	Illegal	Initiate Transfer
71 2	E6	RK	Initiate Chain/ Load Control Memory	Load Control Memory
71 3	E7	RX	Load Control Memory	Load Control Memory
72 3	EB	RX	Store Control Memory	Store Control Memory
73 0	EC	RR	Illegal	Halt/Interrupt
73 3	EF	RX	Illegal	Control Condition Bit (Set/Clear)
74 2	F2	RK	Illegal	Conditional Jump
75 0	F4	RR	Illegal	Search for Sync/Set Monitor/Set Suppress
76 0	F8	RR	Set/Clear Discretes	Set/Clear Discretes
76 3	FB	RX	Store Status	Store Status
77 3	FF	RX	Built-In Test (BIT)	Illegal

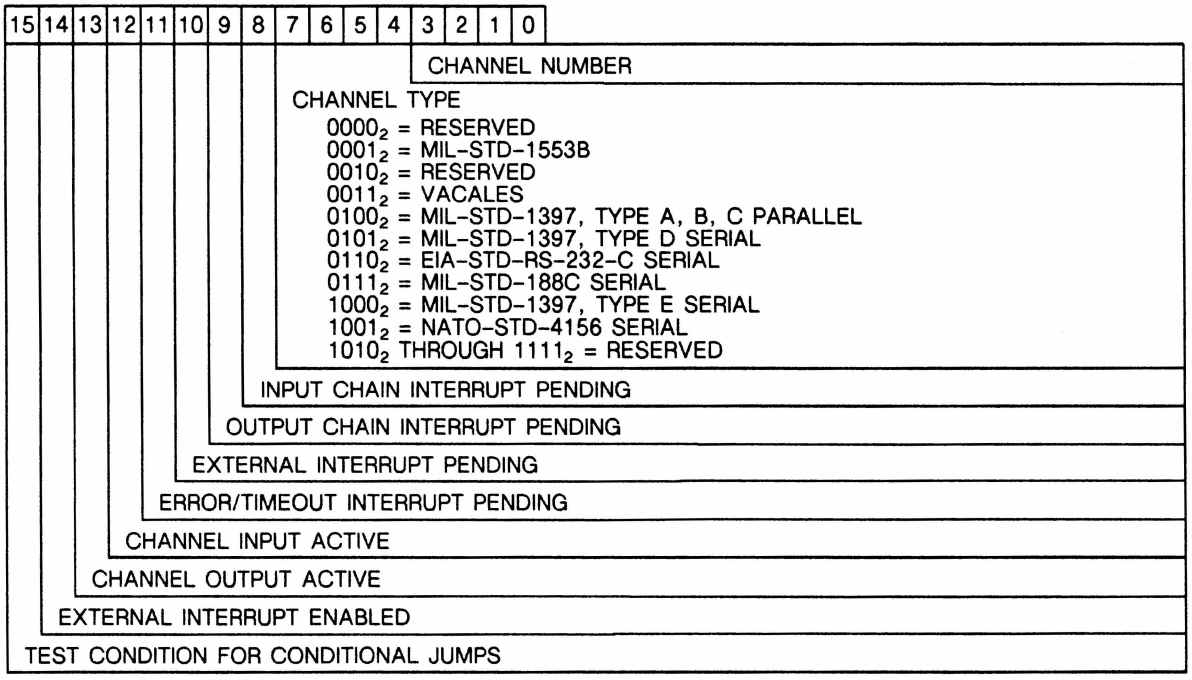


Figure 26. I/O Status Word

MIL-STD-188C, VACALES, and EIA-STD-RS-232-C Serial Channel Communication

MIL-STD-188C, VACALES, and EIA-STD-RS-232-C serial channels communicate over a serial interface that transfers data and control information in both directions. Discrete lines are turned on/off by command and chaining instructions to establish and hold a data link with external equipment. The peripheral equipment turns discrete lines on/off to interrupt the program, to furnish responses to program-controlled discretes, and to inform the IOC of ability to transmit (status). Asynchronous channels outline each character transmitted with start and stop signals. Synchronous data transmissions, however, are timed by the external equipment using the transmit and receive clock lines.

The MIL-STD-188C, VACALES, and EIA-STD-RS-232-C interface lines and the direction of signal transmission are shown in Figures 12 and 13.

MIL-STD-1397 Type D Serial Channel Communication

MIL-STD-1397 Type D serial channel communication transfers output data and EFs over a single coaxial cable, and receives input data and external interrupt codes over another coaxial cable.

Transmissions are initiated by the transfer of appropriate control frames between the IOCA and the peripheral equipment (see Figure 14). An interchange of compatible control frames is required for each word transmitted over the interface, with the exception of a forced EF transfer. In this case, the IOC transmits the EF word, even though the output request control frames do not specify the EF request or after a 20-microsecond timeout when no Output Request control frame is received.

Information frames contain a 32-bit information field, but only the data word determined by the transfer mode field of the buffer control word (see Figure B of Appendix A) contains valid data.

MIL-STD-1397 Type E Serial Channel Communication

MIL-STD-1397 Type E serial channels transfer output data and EFs over a single triaxial cable and receive input data and external interrupt codes over another triaxial cable.

Transmissions are initiated and carried out in a manner similar to MIL-STD-1397 Type D, except that a forced EF word transfer does not occur unless a sink status (SIS) request control frame is received (see Figure 15).

MIL-STD-1397 Type E serial interfaces contain special system integrity features (SIFs). SIFs include source parity generation, source/sink timeouts, sink parity error detection, sink timing error detection, sink illegal condition detection and provide status/error reporting. SIF can be selected and disabled under software control (load control memory macroinstruction using the format shown in Figure B of Appendix A).

NATO-STD-4156 (Modified) Serial Channel Communication

NATO-STD-4156 (modified) serial channels are used in a shipboard digital data communication environment where a number of user devices are required to exchange data with a terminal device over a common network. NATO-STD-4156 (modified) serial channels transfer data and control information over a non-return to zero (NRZ) digital data interface using two types of protocol (see Figure 16). Type A protocol is a basic protocol used to transfer all words between the terminal and a user device. Type B protocol is a complete control-word-plus-data protocol for use by more sophisticated user devices that contain an internal source/sink addressing capability. The NATO-STD-4156 (modified) serial mode (half-duplex) word is shown in Figure D of Appendix A.

MIL-STD-1553B Serial Channel Communication

The MRP can support multiple channels of MIL-STD-1553B in various configurations. These configurations include Bus Controller (BC) and remote terminal (RT) with nonredundant and dual-redundant transceivers. The 1553B channel interfaces directly to the MRP/MRC common bus and appears as an IOC to the CP and memory (see Figure 17) using the operation codes 74, E6, E7, EB, and FB, hex.

When the 1553B channel is acting as a bus controller, the CP is in complete control of all 1553B bus activity.

To initiate a 1553B bus transaction, a chain must be created in main memory. A chain consists of control words, command words, data words, and open locations for returned status and data words. When the chain is initiated, the 1553B channel fetches information from memory where the chain resides and carries out the required 1553B bus transaction.

Control words are divided into two types; bus instructions and jump instructions. Bus instructions cause

some type of 1553B bus transaction (BC → RT, etc.) to occur. Jump instructions test certain conditions (such as the busy bit set in the returned status word) and then jump to another location within the chain depending on whether or not the condition is satisfied.

Command words within the chain are the actual BC commands that appear on the 1553B bus. Specific bits within the command word determine the RT address, transmit or receive mode, RT subaddress, and word count or mode code.

Before the chain can be initiated, certain registers within the channel must be loaded. These registers determine the mode of operation and where the chain begins in main memory. Loading of these registers is done by having the 1553B channel fetch and execute load control memory instructions.

When in RT mode, only the 1553B system BC can initiate bus transactions. Unlike the BC mode, there is no chain associated with the RT mode. Once certain registers within the 1553B channel are loaded, the channel is ready to transfer data into or out of main memory. These transfers occur without any CP interaction. The location where data transfers take place depends on the subaddress and RT bit specified in the BC command and the contents of the base address register (located in the 1553B channel). As a result of this addressing scheme, each subaddress, both transmit and receive, has a separate main memory buffer. A busy state can be invoked in the RT mode by the CP to inhibit main memory transfers. Most mode code is handled directly by the 1553B channel without CP interaction. Mode codes 0, 1, 8, 17 are handled by the 1553B channel and can cause CP interrupts if enabled. Generally, RTs are true slaves to the BC and the BC moves data in and out of RT memory without the RT host intervention. The host is interrupted by the mode codes mentioned when its attention is needed. The MIL-STD-1553B serial mode word is shown in Figure E of Appendix A.

IOC Command Instruction

Input/output operations are initiated by the CP executing an I/O command instruction, operation code 35, octal; code 74, hex. The condition of the a field (a = 0, a ≠ 0) determines the location of the address of the instruction to be used by the IOC, and which IOC is to execute that instruction. If a = 0, IOC 0 executes the I/O command instruction stored in main memory address 140, octal, 141, octal, (60, hex, 61, hex) and clears bits 15 and 14 of address 140, octal, (60, hex). If a ≠ 0, the IOC specified by (R_a) executes the instruction at y + (R_m), y + (R_m) + 1.

After commanding the IOC to execute the I/O instruction, the CP halts further processing until the IOC completes processing the I/O instruction. The program has the option of checking the first command cell location (140, octal; 60, hex) before loading the command cell with a new instruction. Channel activity is established by the I/O instruction in the command cell. Execution of the command cell instruction will result in channel activity, that is, chaining. At the completion of the command cell instruction, one of three activities occurs: (1) the chain terminates, (2) the chain continues, or (3) the command cell is reloaded and the CP begins a new chain. If chaining activity is to be performed, chain addresses are derived from the chain address pointer location (word 2 plus channel number for input, word 6 plus channel number for output) in I/O control memory. Should the command cell be free, the program can reload the command cell for an activity related to another channel.

Program Chaining

Instructions that control the input and output activity of all IOCAs are executed from an active chain associated with each input and output channel. Words stored in I/O control memory, as specified in Appendix A, Table X, control the I/O operations.

The buffer control word (BCW) controls the type and the number of transfers. The transfer mode (TM) field of the BCW (bits 14 and 15) specifies the type of I/O transfer as follows:

TM = 0	abort the transfer (input only – peripheral handshaking continues until buffer transfer count = 0; however, the data is not written into memory)
TM = 01	transfer 8-bit bytes
TM = 10	transfer 16-bit words
TM = 11	transfer 32-bit (double) words
Bit 13 = 0	if MAE is present, use page set 0
Bit 13 = 1	if MAE is present, use page set 2 for channels 0-7 and page set 3 for channels 8-15

The BCW byte pointer (B) (bit 12) is used when performing 8-bit transfers, to specify the most or least significant byte in memory for the next transfer. As each byte is transferred, the B-bit changes state:

B = 0	specifies the most significant 8 bits
B = 1	specifies the least significant 8 bits

The buffer transfer count (BTC) field of the BCW (bits 0–11) specifies the number of bytes, single–length words, or double–length words to be transferred during the selected input data, output data, or EF buffer operation. As each byte or word is transferred, the buffer transfer count is decreased by one. When the count changes from 1 to 0, the buffer terminates. A beginning count of zero specifies the maximum number of transfers (4,096).

The buffer address pointer (BAP) specifies the memory address for the next transfer. The contents of this location are increased by one each time two bytes (parallel) or a single–length word is transferred. For double–length word operations, the contents are increased by two for each transfer.

Whenever the IOC executes the initiate chain instruction from the command cell, the chain address pointer (CAP) is loaded in the control memory for the specified channel and that chain is activated. A CAP specifies an address in main memory where the next chaining instruction is located. As the CAP is used, its value is advanced by 1 if a single–word instruction is read, and by 2 if a double–length instruction is read. Any time an executed instruction activates a buffer on a channel, the associated chain is deactivated until that buffer has terminated. Then the channel chain can proceed to the next instruction.

MIL–STD–188C, VACALES, and EIA–STD–RS–232–C channels use the monitor and suppress registers (see I/O operation code 75, octal; code F4, hex). The monitor register can be used to look for a specific character, such as end of the data in serial transmission, while the suppress register can be used to suppress specific characters in serial transmission. The serial mode information controls the mode, character size, and parity for serial transmission.

Serial Channel Interrupts

The CP can be informed of the status of serial channels by a Class III external interrupt that is generated when any of the applicable events occur (see Figure 26).

Externally Specified Addressing (ESA)

ESA provides peripheral equipment with a means of specifying a relative memory location for storage or retrieval of data. An active parallel, dual–channel mode of operation is required for IOC response to this function. If input is desired, the peripheral equipment presents an input data request with the address and data. The address is presented on the lower–order 16 data lines of the input pair and the data on the higher–order lines. The IOC stores all 32 bits from the channel data at the specified address and at the next sequential address. If output is desired, an output data request is presented on the output channel with the address on the lower–order input data lines. The IOC loads the contents of the specified address on the higher–order output data lines of the channel, the contents of the next sequential memory address on the lower–order output data lines, and raises the output acknowledge line.

EF and external interrupts on ESA channels operate as normal dual channels.

Dual Channel

Parallel and MIL–STD–1397D/E serial input/output channel adapters can operate in a dual–channel mode. In dual–channel operations, data transfers are performed in a 32–bit mode. The operating mode information (word A, hex, I/O control memory) determines if a channel is to operate in a dual–channel mode, test mode, etc. The operating mode word is a programmable feature of the IOC and IOCAs.

For dual–channel operations, two consecutive channels, an even and odd channel, n and n + 1, are paired together.

Master Clear

A hardware–initiated or operator–initiated master clear extinguishes the fault indicators on the MRC control and maintenance panel and places the MRP/MRC in an initial condition, which includes:

a) P–register cleared

b) SR 1 cleared

c) SR 2 cleared

d) RTC count–up and monitor clock countdown functions disabled (if installed)

e) Bits 0–11 of each page register set to its own register address and bits 12–15 cleared

f) All I/O channels cleared as follows:

   1) Status words cleared
   2) All data buffers and chains deactivated
   3) All interrupt data storage disabled, all external interrupt enables (EIEs) cleared
   4) Class III interrupts disabled
   5) Serial monitor and suppress flags cleared
   6) Intercomputer timeout function cleared

7) Operating mode register cleared. This register is not cleared by the I/O channel master clear.

g) For EIA–STD–RS–232–C Serial Channels:

1) Discrete line J on
2) All other discrete control lines off
3) Ring indicator interrupt disabled (on)
4) Internal loop test interrupt disabled
5) Transmitter in idle state (logical)

h) For MIL–STD–188C Serial Channels:

1) Discrete lines H and J on
2) All other discrete control lines off
3) Internal loop test disabled
4) Transmitter in idle state (logical)

i) For MIL–STD–188C VACALES Serial Channels:

1) Discrete lines Transmitter Preparation and J on
2) All other discrete control lines off
3) Internal loop test disabled
4) Transmitter in idle state (logical)

j) Normal display on control and maintenance panel set

k) Bus initialization signal generated

l) Run mode selected (only occurs on power–up master clear)

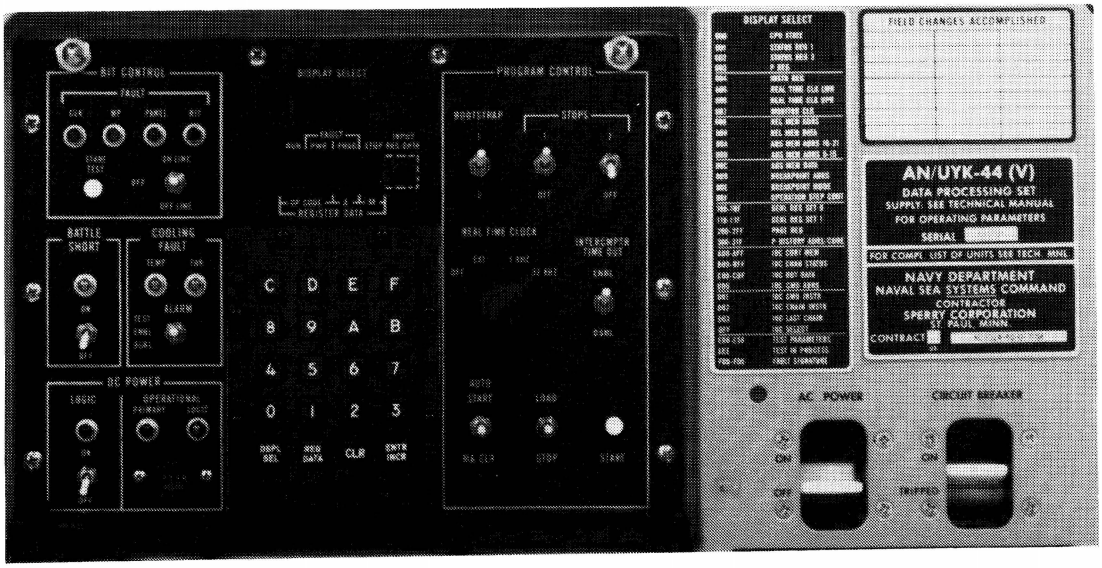
m) Program and power fault cleared

Control and Maintenance Panel (CMP) – MRC/ Expansion Cabinet

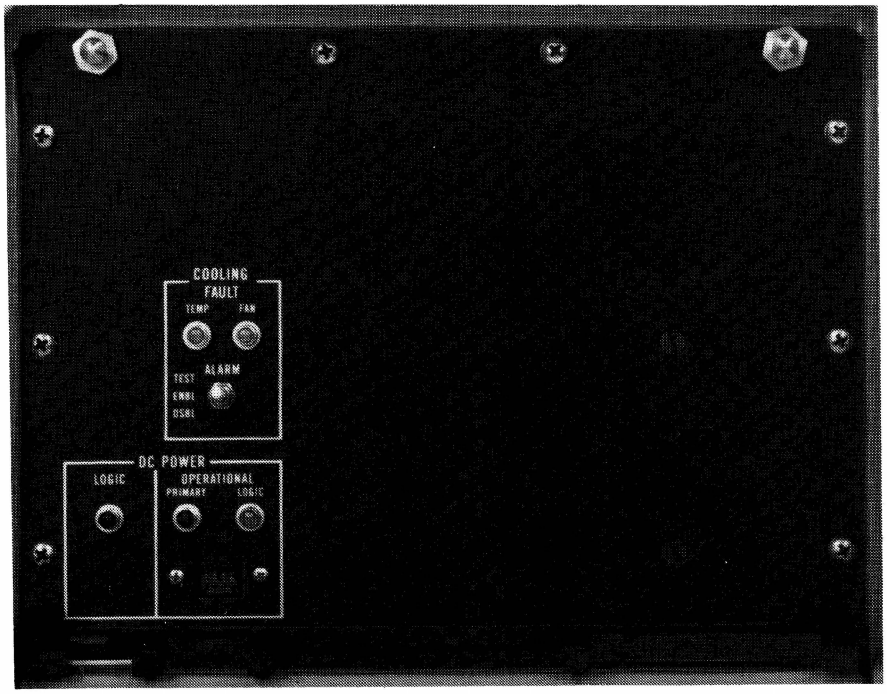
A complete set of controls and indicators is provided for both operator and maintenance personnel. The front of the MRC cabinet swing–out door contains the CMP (see Figure 27A). The CMP includes those items necessary for turning on the MRC, loading and running programs, and observing MRC operation. The expansion cabinet CMP is also located on the front of its swing–out door (see Figure 27B). This CMP contains only indicators and one control. These are a subset of the MRC controls and indicators.

Table 11 lists the respective CMP controls, switches and indicators, and their functions.

Table 12 lists the information displayed during display select operations (MRC only).



A. MRC Basic Cabinet Control and Maintenance Panel (CMP)



B. MRC Expansion Cabinet Control and Maintenance Panel (CMP)

Figure 27. Control and Maintenance Panels (CMPs)

TABLE 11. CMP CONTROLS AND INDICATORS

Name	Type	Function
BIT CONTROL		
FAULT		
CLOCK	Indicator	When lit, indicates a clock failure.
MP	Indicator	When lit, indicates a maintenance processor failure.
PANEL	Indicator	When lit, indicates a CMP failure.
BIT	Indicator	When lit, indicates a BIT error has been found.
START TEST	Pushbutton switch	When pressed, starts BIT.
ON-LINE/ OFF/OFF-LINE	3-position toggle switch	ON-LINE position enables partial BIT (no main memory test) when initiated by a power-up master clear or by pressing the START TEST switch. OFF position disables BIT. OFF-LINE position enables full BIT when initiated by either a power-up master clear or by pressing START TEST switch.
BATTLE SHORT		
BATTLE SHORT	Indicator	When lit, BATTLE SHORT ON/OFF is in ON position.
BATTLE SHORT ON/OFF	2-position toggle switch (with guard)	ON position disables the overtemperature shutdown. OFF position enables the overtemperature shutdown.
COOLING FAULT (MRC and Expansion Cabinet)		Circuitry is powered when the MRC AC POWER switch is on.
TEMP	Indicator	When lit, indicates the MRC internal cabinet air temperature has risen to 65°C.
ALARM ENBL/ DSBL/TEST (Enable/Disable/ Test)	3-position toggle switch	When set to ENBL, an audible alarm sounds when the internal cabinet temperature is 65°C. This is a warning of a possible power shutdown. When set to DSBL, the alarm does not sound. When set to TEST, lights COOLING FAULT FAN and COOLING FAULT TEMP indicators and sounds audible alarm.
FAN	Indicator	Lights when fan RPM decreases by about 25 percent.
AC POWER	Indicator	When lit, indicates power is applied to the power conditioner, fan, and temperature sensor module.

TABLE 11. CMP CONTROLS AND INDICATORS (continued)

Name	Type	Function
AC POWER ON/OFF	2-position toggle switch	ON position enables power to MRC power conditioner, temperature sensor, and cooling fan and enables DC POWER LOGIC ON/OFF switch function. OFF position disables power to MRC power conditioner, temperature sensor, and cooling fan and disables DC POWER ON/OFF switch function.
CIRCUIT BREAKER ON/TRIPPED	Circuit breaker (Protective device)	Protective device which removes ac input power if current becomes too high (i.e., internal short circuit). When in ON position, causes ac input power to be applied to AC POWER ON/OFF switch. TRIPPED position removes ac input power from AC POWER ON/OFF switch.
DC POWER		
LOGIC ON/OFF	2-position toggle switch	ON position enables dc power by turning on the power regulator assembly. OFF position disables dc power by turning off the power regular assembly.
LOGIC (MRC and Expansion Cabinet)	Indicator	Lit when dc power is applied.
OPERATIONAL (MRC and Expansion Cabinet)		
PRIMARY	Indicator	Lit when the three MRC power conditioners each have normal 15 and 200 vdc output. Not lit if any one output is abnormal. Enabled by MRC AC POWER ON/OFF switch.
LOGIC	Indicator	Lights when all output from the MRC and expansion cabinet power regulators is in tolerance.
TIME METER (MRC and Expansion Cabinet)	4-digit, 0000-9999	Records the accumulated hours (in tenths) that MRC and expansion cabinet logic power has been applied.
DISPLAY SELECT	3-digit display	Displays address of selected computer resource.
REGISTER DATA OP CODE/A/M	4-digit display	Shows contents of the display selected by DISPLAY SELECT.
Hexadecimal keyboard F-O	20 pushbutton switches	Data entry device used to enter data and select address to be displayed by DISPLAY SELECT.
DSPL SEL	Pushbutton key	When pressed, enables DISPLAY SELECT display.
REG DATA	Pushbutton key	When pressed, enables REGISTER DATA display and lights INPUT/REG DATA indicator.
CLR (Clear)	Pushbutton key	When pressed, clears enabled display.

TABLE 11. CMP CONTROLS AND INDICATORS (continued)

Name	Type	Function
ENTR/INCR (Enter/Increment)	Pushbutton key	When pressed, enters data previously entered at keyboard. Also increments the DISPLAY SELECT value if data has not been entered.
INPUT REG DATA	Indicator	When lit, data can be entered into the REGISTER DATA display.
PROGRAM CONTROL		
BOOTSTRAP 1/2	2-position toggle switch	When BOOTSTRAP 1 is selected, the MRC executes the bootstrap loading program associated with this switch position. The bootstrap program assigned to BOOTSTRAP 1 depends on the MRC configuration. When BOOTSTRAP 2 is selected, the MRC executes the bootstrap loading program associated with this switch position. The bootstrap program assigned to BOOTSTRAP 2 depends on the MRC configuration.
STOPS 1/OFF	2-position toggle switch	In position 1, a program stop can be caused when this switch is tested by software. It is used by some programs to cause the MRC to stop executing macroinstructions when a condition is met. OFF position causes no program stop.
STOPS 2/OFF	2-position toggle	In position 2, a program stop can be caused when this switch is tested by software. It is used by some programs to cause the MRC to stop executing macroinstructions when a condition is met. OFF position causes no program stop.
REAL TIME CLOCK OFF/EXT/1 KHZ/32 KHZ	4-position rotary switch	OFF position prohibits incrementing of RTC and monitor clock registers. EXT (external) position causes the RTC and monitor clock registers to use the external clock source, if connected, for timing. 1 KHZ position causes the RTC and monitor clock registers to use the internal 1-KHZ clock for timing. 32 KHZ position causes the RTC and monitor clock registers to use the internal 32-KHZ clock for timing.
INTERCOMPTR TIMEOUT ENBL/DSBL	2-position toggle switch	ENBL position enables the occurrence of a Class III intercomputer timeout interrupt. DSBL position inhibits the occurrence of a Class III intercomputer timeout interrupt.
START	Pushbutton	When pressed, causes the MRC to begin executing instructions in the mode selected.

TABLE 11. CMP CONTROLS AND INDICATORS (continued)

Name	Type	Function
AUTO START/MA CLR	3-position toggle switch (momentary contact in down position - return to neutral).	In the AUTO START position, causes the MRC to begin executing instructions at NDRO address 0000 when power is applied or restored after a power failure. MA CLR (master clear) - When operated in the RUN mode, clears the power or program fault displays. When operated while the MRC is not in the RUN mode executing instructions, the MRC resets to a master-cleared state.
LOAD/STOP	3-position return-to-neutral switch	When operated to the STOP position in the RUN mode, causes the MRC to stop executing macroinstructions. When momentarily operated to the LOAD position, causes the MRC to execute a master clear, select RUN mode, and execute instructions beginning at NDRO address 0002.

TABLE 12. INFORMATION DISPLAYED BY DISPLAY SELECT

DISPLAY SELECT Value	Information
000	This data is read only. This number displays the current state of the CPU. The REGISTER/DATA display is interpreted according to the legend above the display:
RUN (Program Run)	AXX– Indicates MRC is executing instructions in run mode (– = Blank, X = Don't Care).
PWR (Power Fault)	XFXX When lit, indicates a power fault has occurred (X = Don't Care).
PROG (Program Fault)	XXFX When lit, indicates an instruction fault has occurred. This is caused by attempting to execute an illegal instruction (X = Don't Care).
STOP	–XXS Indicates MRC is not executing macroinstructions (– = Blank, X = Don't Care): S = STOP CONDITION (– = Blank). If the CP is stopped, the contents of the STOP indicator will define the reason for the stop as follows: 0 – Power–up or master clear 1 – Jump–stop 1 2 – Jump–stop 2 3 – Unconditional jump–stop 4 – Stop key pressed 5 – Breakpoint stop 6 – Opstep stop
001	Contents of Status Register 1 (STATUS REG 1) are contained in the REGISTER/DATA display. The value can be changed by the operator.
002	Contents of Status Register 2 (STATUS REG 2) are contained in the REGISTER/DATA display. The value can be changed by the operator.
003	Contents of the Program Address Register (P REG) are contained in the REGISTER/DATA display. The value can be changed by the operator.
004	Contents of the Instruction Register (INSTR REG) are contained in the REGISTER/DATA display. This value is read only.
005	The lower half of the RTC (REAL TIME CLK LWR) is contained in the REGISTER/DATA display. This value can be changed by the operator.
006	The upper half of the RTC (REAL TIME CLK UPR) is contained in the REGISTER/DATA display. This value can be changed by the operator.
007	Contents of the monitor clock are contained in the REGISTER/DATA display. This value is read only.
008	The last selected 16–bit relative memory address is contained in the REGISTER/DATA display. This address is used by the succeeding DISPLAY SELECT value (009) to do a relative memory read. This value can be changed by the operator.
009	A relative memory read (based on the 16–bit relative address specified by the preceding display select value (008)) is done. The result of this read is contained in the REGISTER/DATA display. This value can be changed by the operator.

TABLE 12. INFORMATION DISPLAYED BY DISPLAY SELECT (continued)

DISPLAY SELECT Value	Information
00A	The last selected absolute memory address upper (bits 21-16) is contained in the REGISTER/DATA display. This 6-bit value is combined with the 16-bit value (specified by the succeeding DISPLAY SELECT value (00B)) to do an absolute memory read when requested by DISPLAY SELECT value 00C. This 6-bit value can be changed by the operator.
00B	The last selected absolute memory address lower (bits 15-0) is contained in the REGISTER/DATA display. This 16-bit value is combined with the 6-bit value (specified by the preceding DISPLAY SELECT value (00A)) to do an absolute memory read when requested by DISPLAY SELECT value 00C. This 16-bit value can be changed by the operator.
00C	An absolute memory read, based on the 22-bit address specified by the preceding DISPLAY SELECT values (00A and 00B) is done. The result of this read is displayed in the REGISTER/DATA display. This value can be changed by the operator.
00D	The last selected relative breakpoint address is contained in the REGISTER/DATA display. This value can be changed by the operator.
00E	The 3-bit value of the current breakpoint control register (BREAKPOINT MODE) is contained in the REGISTER/DATA display. These bits are interpreted as follows: Bit 0 = 0 - disable instruction breakpoint 1 - enable instruction breakpoint Bit 1 = 0 - disable write breakpoint 1 - enable write breakpoint Bit 2 = 0 - disable read breakpoint 1 - enable read breakpoint This 3-bit value can be changed by the operator.
00F	The value of the current opstep control is contained in the REGISTER/DATA display. This value is interpreted as follows: 0 = RUN MODE - Start CP on Start key 1 = OPSTEP MODE - Step CP on Start key 2 = OPSTEP MODE - Step IOC on Start key This value can be changed by the operator.
100-10F	The contents of a general register from set 0 are contained in the REGISTER/DATA display. The general register is selected by the least significant digit of the DISPLAY SELECT value (0-F). This register value can be changed by the operator.
110-11F	The contents of a general register from set 1 are contained in the REGISTER/DATA display. The general register is selected by the least significant digit of the DISPLAY SELECT value (0-F). This register value can be changed by the operator.
200-2FF	The contents of a page register (PAGE REG) are contained in the REGISTER/DATA display. The page register set is selected by bits 6 and 7 of the DISPLAY SELECT value and the register within the set by the DISPLAY SELECT value bits 0-5. This register value can be changed by the operator.

TABLE 12. INFORMATION DISPLAYED BY DISPLAY SELECT (continued)

DISPLAY SELECT Value	Information
300-30F	The P-history address and jump code are contained in the REGISTER/DATA display. These values are interpreted as follows: Display — Register/Data 300 — Address of most recent instruction to alter P 301 — Type of instruction or class of interrupt that changed P (page set data included) 302 — Address of previous instruction to alter P 303 — Type of previous instruction or interrupt that changed P 30E — Address of oldest recorded instruction to alter P 30F — Type of oldest instruction or interrupt that changed P These values are read only.
A00-AFF	The contents of the IOC control memory location are contained in the REGISTER/DATA display. The IOC is selected by DISPLAY SELECT value DFF. The channel is selected by the middle digit (bits 7-4) of this DISPLAY SELECT value. The location is selected by the least significant digits (bits 3-0) of this DISPLAY SELECT value. This control memory can be changed by the operator.
B00-BFF	An IOC channel is contained in the REGISTER/DATA display. The IOC is selected by DISPLAY SELECT value DFF. The channel is selected by the middle digit (bits 7-4) of this DISPLAY SELECT value. The location is selected by the least significant digit (bits 3-0) of this DISPLAY SELECT value. This channel status is read only.
C00-COF	An IOC channel's output data is contained in the REGISTER/DATA display. The IOC is selected by DISPLAY SELECT value DFF. The channel is selected by the least significant digit (bits 3-0) of this DISPLAY SELECT value. The output data is read only.
D00	An IOC command address register's contents are contained in the REGISTER/DATA value. The IOC is selected by DISPLAY SELECT value DFF. This value is read only.
D01	An IOC command instruction register's contents are contained in the REGISTER/DATA display. The IOC is selected by DISPLAY SELECT value DFF. This value is read only.
D02	An IOC chain instruction register's contents are contained in the REGISTER/DATA display. The IOC is selected by DISPLAY SELECT value DFF. This value is read only.
D03	An IOC channel control register's contents are contained in the REGISTER/DATA display. The IOC is selected by DISPLAY SELECT value DFF. This value is read only.
DFF	The last selected IOC number is contained in the REGISTER/DATA display. This value is used in all the preceding IOC commands as an IOC select value. This value can be changed by the operator.
E00-E59	Test parameters are contained in the REGISTER/DATA display.
F00-F05	Fault signature values are contained in the REGISTER/DATA display.

MICROPROCESSOR DEVELOPMENT SYSTEM (MDS)

The MDS provides a vehicle into which the MRP, in any configuration, can be embedded to operate and test in a system environment and to develop the AN/UYK-44 software.

The MDS consists of two commercial (nonmilitary-environment protected), 19-inch wide cabinets with electronic chassis, and an operator console (alphanumeric cathode ray tube display and keyboard) placed on a table and connected to the MDS cabinets by a cable. The MDS is shown in Figure 28.

The MDS cabinets contain:

a) Power supply chassis

b) Memory chassis containing 32K-word modules with a capacity of up to 256K words of each type; magnetic core and 6 x 9-inch semiconductor

c) Two SEM chassis in which the Format B, 2A SEM MRP can be installed

d) One chassis in which the 6 x 9-inch support logic and memory interface modules are installed

e) Control panel

Each cabinet also has an expansion area where a chassis for the MMIO Format B, 2A SEMs and MMIO connector assemblies can be installed.

Each chassis has its own cooling fans that draw air through the front of the cabinet and exhaust air through the rear of the cabinet. Each cabinet has hinged doors, one on the front and one on the rear of the cabinet, a removable panel on the side, casters, leveling screws, air filters, power connectors, and 32 I/O connectors.

The support logic modules provide the interfaces between the MRP and the MDS operator console and control panel, micromemory for the MDS support functions, and four breakpoint registers.

Figure 28. Microprocessor Development System (MDS)

OL-335(V)/U DATA PROCESSING GROUP EMBEDDING

The OL-335(V)/U Data Processing Group or MRP is designed to be embedded in a host system or device. The SEM is the basic building block for an MRP.

The user can determine the packaging for the MRP while still maintaining the functionality of the computer.

The MRP is a user-determined configuration of SEMs. The user must also define the support structures needed to embed the MRP. Included in this definition are: enclosure, SEM backpanel and support structure, cooling, power supply and distribution system, memory, operator/maintenance panel, and documentation support. Each application is unique in its requirements and while the MRP can remain a constant, its configuration and the support hardware elements may vary from system to system.

A user who has determined that an MRP will be needed to satisfy his application requirements should use an engineering approach to the MRP design that:

- Makes a complete analysis of all system requirements
- Defines the MRP module complement
- Identifies the supporting architecture
- Familiarizes the user with the available engineering documentation
- Provides specifications for the final design

The analysis of the system requirements should include the physical structure, thermal analysis, reliability analysis, system modeling, simulation, and program timing analysis.

The definition of the MRP module complement should include the present SEM complement, as well as the projected growth requirements. As an adjunct, the user should consider any new design activity currently in process for application to the system.

A complete engineering investigation should include the identification of supporting architectures needed including operator/maintenance panels, volatile and nonvolatile memories, and special-purpose functional modules such as A-to-D and D-to-A converters. The user should identify those elements of the computer system that are not part of the MRP SEM complement. When the end system becomes operational, it must be supportable from a logistic and training standpoint; therefore, the user needs to identify the system requirements to those concerned and provide for each as the system is implemented.

The user must plan to reproduce the system. To ensure that this can be achieved, a complete set of specifications and supporting documentation, that is, technical manuals, installation control drawings, and program development specifications must be provided.

The MRP Organizational Level maintenance manual provides the generic maintenance information which the system user can incorporate in the system maintenance manual. The system maintenance philosophy and the MRP configuration dictates what MRP information the user incorporates in the system manual. The MRP design data manual and the following documents provide the technical information necessary to embed the MRP in the user's system:

a) *Allocated Provisioning Lists (APLs)* – APLs have been assigned for the MRP, MRC, and ordering options. They are available from:

Navy Ships Parts Control Center (SPCC)
Mechanicsburg, PA
Code 0522

b) *Installation Control Drawings (ICDs)* – ICDs have been submitted to PMS-408. Document numbers have not yet been assigned.

c) *Maintenance Assistance Modules (MAMs)* – The general contents of a MAM set are listed in the Organizational Level maintenance manuals. The MAMs are used to support the BIT and contain one of each SEM or LRU which can be detected by BIT as having a failure. The MAM configuration is dependent on the MRP or MRC configuration. No document numbers are assigned.

d) *Fault Isolation Kit (FIK)* – The general contents of an FIK are one of each SEM, LRU or module in an MRP or MRC. The FIK configuration is dependent on the MRP or MRC configuration. FIKs do not have document numbers.

To ensure that the user fully understands the requirements for properly embedding the MRP, total familiarity with all existing documentation is recommended. Appendix C defines applicable documentation available from Naval Sea Systems Command, PMS–408 and Unisys. Course training is also recommended to become familiar with the design and operations of the MRP.

The embedding activity presented here is introductory in content. Users desiring further information should contact:

Naval Sea Systems Command
PMS–408
Department of the Navy
Washington, DC 20362
Telephone: (202) 692–8204

or

Unisys Corporation
Defense Systems
P.O. Box 64525
St. Paul, MN 55164–0525
Telephone: (800) 328–0204

TRAINING

Navy Conducted Training

Training materials for MRP and MRC Organizational Level maintenance are available from PMS–408. The training materials include:

Course/Curriculum Outline
Instructor Guide
Student Study Guide
Audiovisual Aids
Post Course Test

MRP – The training materials incorporate the embedded system's maintenance level training. Standalone maintenance training is not planned for the MRP by the Navy. All MRP maintenance training will be provided as part of the system level training.

MRC – The training materials are used by the Navy for standalone MRC maintenance training.

For more information regarding Navy conducted training contact:

Naval Sea Systems Command
PMS–408
Department of the Navy
Washington, DC 20362
Telephone: (202) 692–8204

Unisys Conducted Training

Unisys has developed several courses to meet the needs of the military and industry. These courses are available on a regularly scheduled basis at the Unisys St. Paul Training Center. They can also be conducted at the customer's facility.

The following is a brief description of the training courses:

Management Familiarization – This 2.5–day course familiarizes managers and supervisory personnel with the AN/UYK–44 program, and the computer capabilities, features, and options.

Technical Familiarization – This 5–day course acquaints technical personnel with the computer instruction set, software and hardware features, operation, and maintenance concepts. It is designed for a wide range of students including software, hardware, and systems personnel.

Engineering Design and Architecture – This 10–day course provides the engineering detail required to embed the MRP into a host system. Students attending this course are normally experienced hardware engineers involved in the embedding process.

Organizational Level Maintenance – This 5–day course covers the operator and maintenance actions required of the organizational level operator/technician. Students attending the course do not need an electronics background.

Training Materials – Materials for the Unisys conducted courses have been delivered to PMS–408 and include:

Course/Curriculum Outline
Instructor Guide
Student Study Guide

Unisys can tailor existing curricula to more specifically address customer OL–335(V)/U and AN/UYK–44(V)/U based systems.

For more information regarding training, including scheduling and prices, contact the AN/UYK–44 Training Manager at:

Unisys Corporation
Defense Systems
P.O. Box 64525, M.S. E2D15
St. Paul, MN 55164–0525
Telephone: (612) 456–6557

SUPPORT SOFTWARE

Six major standard software development and maintenance tools are available to users of the MRP/MRC/MDS. The Navy (NAVSEA PMS–408) maintains these software tools consisting of the following.

MACHINE TRANSFERABLE SUPPORT SOFTWARE (MTASS)

The MTASS system is an integrated set of Navy standard software development and maintenance tools created specifically for the following target computers:

AN/UYK–20(V)
AN/UYK–20A(V)
AN/AYK– 14(V)
OL–335(V)/U
AN/UYK–44(V)/U
AN/UYK–7(V)
AN/UYK–43(V
MIL–STD–1750A

The MTASS system provides one CMS–2 high–order language (HOL), one MACRO assembler, one linkage editor (LE) (loader), and tape builders/simulators for all of the above computers. By providing this integrated programming environment, MTASS enhances the ability to reuse software and provides projects with the ability to generate object programs for many different target computers. MTASS is hosted on large–scale and medium–scale commercial host computers. Thus MTASS can take advantage of the latest hardware and software development tools available on the host computers.

The existing host computers and operating systems include Unisys 1100 Series with Unisys 1100 TIME SHARING EXEC OS 1100 (level 36 or later); IBM 370/4341 Series with IBM Operating System OS/VS2 (MVS) R3.7 or VM/370–CMS Release 3; DEC VAX Series with VAX/VMS Operating System (version 4.2 or later).

Concurrent Use – Queuing for target computer time blocks is eliminated by using time–shared host computer systems.

Transportability – A common data exchange format facilitates the distribution of project development and maintenance among different host computer operating systems.

Adaptability – MTASS contains a full set of general–purpose development tools making it ideal for initial design analysis, program compile or assembly, system build, developmental testing, and follow–on maintenance.

Project Control – Host computer system tools facilitate generation of project status reports detailing software changes, status, costs, etc.

Features – Primary MTASS components are:

- CMS–2 – CMS–2 HOL Compiler
- FORTRAN – FORTRAN/77 Compiler
- MACRO – MACRO Assembler
- LE – Linkage Editor
- SIM/M – AN/UYK–20(V), AN/UYK–20A(V) AN/AYK–14(V), AN/UYK–44(V) Simulator
- SIM/14 – AN/AYK–14(V) Simulator including dual SCP Simulation
- SIM/A – MIL–STD–1750A Computer Simulator
- SIM/L – AN/UYK–7(V) and AN/UYK–43(V) Computer Simulator
- TB/M – SDEX/20 and SDEX/M Tape Builder
- TB/A – MIL–STD–1750A Tape Builder
- TB/L – SDEX/7 and SDEX/43 Tape Builder
- CIR – Host Operating System Common Interface Routines
- FXS – File Exchange System

Primary MTASS features are:

Program Compile – The MTASS system has the following language processors: CMS–2 Compiler, FORTRAN Compiler, and MACRO Assembler.

Program Load – The MTASS Linkage Editor (LE) loads and links the relocatable object code produced by CMS–2, FORTRAN, and MACRO language processors. LE allows for implicit library retrieval and incremental system building.

System Build – The Tape Builders reformat the LE output to produce a target computer executable image. The Tape Builders generate executive tables and create a target load file on a magnetic tape or disk file for loading into the target computer.

Program Checkout – The system has a simulator for each target computer. This allows program checkout

to be performed on the host computer. Also, the target load files produced by the Tape Builders can be loaded into the target computer for program checkout.

MACHINE TRANSFERABLE SUPPORT SOFTWARE/MINICOMPUTER (MTASS/M)

The MTASS/M system is a set of Navy standard software development and maintenance tools created specifically for the following Navy standard minicomputers:

AN/UYK-20(V)
AN/UYK-20A(V)
AN/AYK-14(V)
OL-335(V)/U
AN/UYK-44(V)/U)

By providing a standard minicomputer programming environment on large-scale and medium-scale host systems, MTASS/M combines the advantages of generating software on the target computer with the advantages of using the latest hardware and software development tools available on the host computers.

Active host computers and operating systems include:

Unisys 1100 with OS 1100
CDC CYBER Series with NOS 2.4
DEC VAX Series with UNIX
BSD Version 4.1 (Berkley)
AN/UYK-7 with SHARE/7
IBM 370/4341 Series with IBM
OS/VS2 (MVS) R3.7
or VM/370-CMS Release 3
DEC VAX Series with VAX/VMS 4.4
DEC SYSTEM 20 Series with TOPS-20
Monitor 5
DEC VAX Series with UNIX
System V(AT&T)

Concurrent Use – Host systems are time-shared, thus queues for computer time are minimized.

Transportability – With a common data exchange format, the development and maintenance of a project can be distributed among several host systems and target computers.

Adaptability – With a full set of general-purpose development tools, MTASS/M is ideal for initial design analysis, program compile or assembly, system build, development testing, and follow-on maintenance.

Project Control – With host system tools, project status reports (which detail software changes, status, costs) can be generated.

Features – Primary MTASS/M features are:

Program Compile – The system has the following language processors: CMS-2M Compiler, FORTRAN/M Compiler, and MACRO/M Assembler.

Program Load – The system has a versatile LE and system tape (file) generation program (SYSGEN/M Loader/System Generator) for linking relocatable object code and system building.

Program Checkout – The system has a 16-bit computer simulation program (SIM/M or SIM/14 Program Interpreters) that provides facilities for checkout execution of the generated program on the host system.

Utility Package – The system contains (as a part of the Loader/System Generator) a set of translator programs to perform required conversions among the various data formats used (that is, convert source code files in target computer format to corresponding host system format, prepare headers for library files, etc.) as well as utilities to move source and relocatable object code from one host computer operating system to another.

PROGRAMMER-ORIENTED AN/UYK-44 LINK (PORTAL/44)

PORTAL/44 is a software gateway which allows software developers to easily transfer programs from the commercial host environment into the AN/UYK-44 military environment (download) and back to the host system (upload). PORTAL/44 is currently operational as a Navy standard software product on a DEC VAX computer running either the BELL/UNIX or VAX/VMS operating systems. In the future, PORTAL/44 may be available on other host systems.

PORTAL/44 interfaces with the Navy standard MTASS/M or MTASS software development and maintenance tools. This interface allows users to compile, assemble, link and download/upload directly to and from the OL-335(V)/U or AN/UYK-44(V)/U target computer. Through this interface, PORTAL/44 facilitates system checkout and provides an upward step from the MTASS simulators.

Compatibility – PORTAL/44 has been designed to work with either the MTASS/M or MTASS software systems.

Efficiency – Valuable MRP/MRC resources are efficiently managed with queuing and fast download/upload capabilities.

Realtime Checkout – System checkout of time-dependent code can be performed directly on the OL-335(V)/U or AN/UYK-44(V)/U target computer.

Availability – AN/UYK-44 program checkout can be performed at the programmer's terminal, thus time and money traveling to a hardware site may be reduced from project budgets.

Preserve Environment – PORTAL/44 can save the AN/UYK-44 environment in a checkpoint file format and pass it to the MTASS/M or MTASS simulator for continued checkout.

Ease of Use – The PORTAL/44 command language is based upon the MTASS/M and MTASS simulator command language, thus minimizing PORTAL/44 training time.

Symbolic Debug – PORTAL/44 provides a symbolic debug capability and thus physical memory addresses need not be computed.

PORTAL/44 executes under control of the host computer operating system (OS) in either a batch or interactive mode. In order to establish communication between the host computer and the OL-335(V)/U or AN/UYK-44(V)/U, PORTAL/44 requires two interfaces. The first interface is a support channel which is used to control, access registers, and access switches on the MRP/MRC. The second interface is an auxiliary I/O channel which is used for MRP or MRC memory uploading/downloading. This auxiliary I/O channel also provides programmer terminal communication with the application program running in the MRP/MRC.

Features – Primary PORTAL/44 features available to the programmer consist of the ability to:

- Inspect/change AN/UYK-44 memory
- Inspect/change AN/UYK-44 registers
- Run/stop the AN/UYK-44
- Op step the AN/UYK-44
- Set breakpoints on the AN/UYK-44
- Snap locations in the AN/UYK-44
- Master clear the AN/UYK-44
- Set keys on the AN/UYK-44
- Keep a session log
- Examine P history
- Use symbolic debug on the AN/UYK-44
- Receive fault reporting from AN/UYK-44
- Application program terminal communication interface
- Download/upload AN/UYK-44 memory and registers
- AN/UYK-44 queuing
- MTASS simulator command language
- Execute lists of command sequences

LEVEL 2

The LEVEL 2 support software system is a set of software development and maintenance tools created specifically for the Navy standard minicomputers (AN/UYK-20(V), OL-335(V)/U, and AN/UYK-44(V)/U). LEVEL 2 support software is hosted on the target computer but can generate object codes for the AN/UYK-20(V), AN/UYK-20A(V), OL-335(V)/U, or AN/UYK-44(V)/U targets regardless of the host computer.

Features – Primary LEVEL 2 features are:

Program Compile – The system has the following language processors: CMS-2M Compiler, FORTRAN/M Compiler, and ULTRA/16 Macro Assembler.

Program Load – The system has a versatile LE and system tape generation program (Loader Generator).

Resident Monitor – The system contains special-purpose programs to perform those functions necessary for management of data, resources, and generation of object and source code files:

- Disk File Management
- I/O System
- Command Language Processor
- Expanded Memory Management

System Utilities – The system provides utility programs for source updating, data manipulation, program patching, and program checkout:

- Text Editor
- File Processor
- Debugger (Expanded Memory only)

Standard Executive/20 (SDEX/20)

SDEX/20 is the Navy standard executive system which fully supports the OL-335(V)/U, AN/UYK-44(V)/U, AN/UYK-20(V), and AN/UYK-20A(V) 16-bit computers.

The SDEX/20 system is comprised of four components:

- SDEX/20 – Kernel realtime executive
- PHM – Peripheral Handler Module

- DM – Debug Module
- EPM – Error Processing Module

SDEX/20 forms the basic kernel for a realtime system. A complete system can be formed by the addition of site–specific system functions and user modules. The functions provided by SDEX/20 are as follows:

- Initialization
- Scheduling
- Interrupt management
- Input/output management
- Error management
- Executive service request (ESR)
- Memory management

Features – Primary SDEX/20 features are:

Initialization – The initialization function provides the means to load and set SDEX/20 and user modules to their initial states during computer startup or restart. A subset of the AN/UYK–44 BITs can be run by SDEX/20.

Scheduling – The scheduling function provides the means for allocating processor resources among the user modules. Both the executive and task modes of the MRP/MRC are supported for user modules. The scheduling function provides for scheduling four different types of user module tasks as follows:

- Successor Tasks – Tasks requiring processing in response to any module's request. They are prioritized and synchronized.
- Message Tasks – Tasks requiring processing to receive a message from another module.
- Time–dependent Tasks – Tasks requiring processing on a time–related basis such that the interval of time between executions is not less than a specified increment.
- Background Tasks – Tasks requiring execution on a time–available basis.

Interrupt Management – The interrupt management function receives and initially decodes all interrupts. If the interrupt is associated with executive processing, SDEX/20 performs the required processing. Otherwise, SDEX/20 releases processor control to the module which has registered responsibility for processing the interrupt.

Input/Output Management – The I/O management function is peripheral device–independent and provides the means whereby user modules can initiate and control computer I/O operations.

Error Management – The error management function identifies hardware and software errors upon occurrence and takes action as directed by the user modules.

Executive Service Request (ESR) – The ESR feature provides additional ESRs to allow users access to the architectural features of the MRP/MRC including multiple IOCs, MMIO, status register, memory protection, and page address registers.

Memory Management – The memory management feature includes changes to provide management with four sets of page address registers and the additional memory available in the MRP/MRC. Runtime allocation from a memory pool is also supported.

Peripheral Handler Module (PHM) – The PHM provides realtime I/O control for user selected peripherals. PHM manages queueing and dequeueing of user I/O requests and provides a standard format for user I/O requests. A library of 37 device handlers exists and additional handlers are being developed.

Debug Module (DM) – The DM provides debugging tools to the user: software breakpoints, SDEX/20 table dumps, resource monitoring table dumps, snapshots, patching, and capture of unregistered errors. A library of 16 device handlers exists and additional handlers are being incorporated.

Error Processing Module (EPM) – The EPM registers all errors recognized by SDEX/20. EPM maintains a round–robin table of all errors that have occurred. It defaults error processing if users have not registered with EPM for specific error handling.

Standard Executive/Minicomputer (SDEX/M)

The SDEX/M is a set of Navy standard software elements created specifically for the Navy standard minicomputer (AN/UYK–20(V), AN/AYK–14(V), OL–335(V)/U, and AN/UYK– 44(V)/U).

SDEX/M forms the basic kernel of Runtime Support Software/Microcomputer (RSS/M). A complete system can be formed by the addition of site–specific system functions and user modules.

SDEX/M is the basic common control mechanism for application software operating in the computer. The functions provided by SDEX/M are:

- Initialization
- Task management

- Task synchronization
- Event management
- General services

SDEX/M is a functionally independent kernel providing executive services with common interfaces for a target computer's application. It is independent of application restraints and computer configuration.

Features – Primary SDEX/M features are:

Initialization – This function initializes SDEX/M for the computer system bootstrap load, cold start, and auto start conditions. This function controls all initialization and does not release control for general processing until initialization of SDEX/M and the computer is complete.

Task Management – This function controls task processing on a priority basis. The function consists of task scheduling, task dispatching, task state control, and memory management subfunctions.

Task Synchronization – This function coordinates task processing using signal and wait operations performed on binary semaphores. It also provides a mechanism for mutual exclusion over short sections of code.

Event Management – This function performs all event processing within SDEX/M and handles interrupts and process errors. The event management function processes all event registration and cancellation requests. It also processes all nonuser–captured interrupts. When an event occurs, the event management function causes action to be taken as directed by previous event registration.

General Services – This function provides clock handling and miscellaneous processing services, some of which require the execution of privileged instructions. The general services function provides the means by which information about specified tasks is obtained and by which parameters associated with specified tasks are changed. It provides for initiation of computer I/O operations, modification of time values for timed events and for acquisition and alteration of the RTC value.

The descriptions provided here are intended to be introductory in content. For those users desiring further information on the available standard software, they should contact:

Naval Sea Systems Command
PMS–408
Department of the Navy
Washington, DC 20362
Telephone: (202) 692–8204

Acronyms and Abbreviations

ac	alternating currrent
APL	allocated provisioning list
ATR	air transportable rack
B	byte pointer
BAP	buffer address pointer
BC	bus controller
BCW	buffer control word
BIT	built–in test
BTC	buffer transfer count
C^3	command, control, and communication
CAP	chain address pointer
CMI	control and maintenance interface
CMP	control and maintenance panel
CP	central processor
dc	direct current
DDC	Data Device Corporation
DIP	dual in–line package
DM	debug module
DMA	direct memory access
DPS	data processing set
DRAM	dynamic random access memory
EDC	error detection and correction
EF	external function
EIE	external interrupt enable (line)
EMI	electromagnetic interference
EMP	electromagnetic protection
EPM	error processing module
EPROM	erasable programmable read–only memory
ESA	externally specified addressing
ESR	executive service request
FIK	fault isolation kit
FIFO	first in/first out
HOL	high–order language
Hz	hertz
ICD	installation control drawing
IFF	identification friend or foe
I/O	input/output
IOC	input/output controller
IOCA	input/output channel adapter
ISA	instruction set architecture
IW	indirect word
KIPS	instructions per second (in thousands)
LCC	leadless chip carrier
LE	linkage editor
LRU	line replaceable unit
LSB	least significant bit
MAE	memory address expansion
MAM	maintenance assist module
MDS	microprocessor development system
MI	memory interface
MMBI	memory mapped bus interface
MMIO	memory mapped input/output
MMIOA	memory mapped input/output adapter
MTTR	mean time to repair
MP	maintenance processor
MRC	militarized reconfigurable computer
MRP	militarized reconfigurable processor
MTASS	machine transferable support software
MTASS/M	MTASS/minicomputer
NDRO	nondestructive read–only (memory)
NRZ	non–return to zero
ns	nanosecond
OS	operating system
ϕ	phase
P	program address (register)
PHM	peripheral handler module
PIC	peripheral input channel
PORTAL/44	programmer oriented AN/UYK–44 link
PROM	programmable read–only memory
RAM	random access memory
RCMI	remote control and maintenance interface
ROM	read–only memory
RPM	revolutions per minute
RSS/M	runtime support software/ microcomputer
RT	remote terminal
RTC	realtime clock
SDEX/M	standard executive/minicomputer
SDEX/20	standard executive/20
SEM	standard electronic module
SIF	system integrity feature
SIS	sink status
SPCC	ships parts control list
SR	status register
SRAM	static random access memory
TM	transfer mode
TTL	transistor/transistor logic
UVEPROM	ultraviolet erasable programmable read–only memory
VACALES	variable character length synchronous
vac	volts alternating current
vdc	volts direct current

APPENDIX A
REPERTOIRE OF INSTRUCTIONS

Instruction defined in this list include the basic instruction set and those required for optional features in the AN/UYK–44. Users of configurations that do not include certain optional instructions must place those respective instructions in the "Not assigned" category and assemble programs accordingly. "Not assigned" codes generate indeterminate results, which may or may not result in instruction fault interrupts when executed.

The instructions are described in the following format:

(Operation Code in octal and hexadecimal notation)

(ULTRA symbol) (instruction format), (instruction name)

(Detailed descriptive text that includes special designator interpretations when applicable)

When the a– or m–designator is used as a sub–function code, the information is presented in table form.

When the instruction also sets the Condition Code as a result of its function the symbol "(CC)" appears after the description.

Privileged instructions are indicated by †.

Double instruction using R_a, R_m or Y require that a, m or Y be even (except for 37, octal, 7C, hex instructions).

SYMBOLS USED IN INSTRUCTIONS

Symbol	Description
a	The a–designator from instruction words (a–values are expressed in octal).
D	The two's complement deviation value in a local jump instruction.
R_a	The register designated by a.
m	The m–designator from instruction words.
R_m	The register designated by m.
Y	The operand or operand address generated in the execution of an instruction.
y	The contents of the second word of an RK or RX instruction.
P	The Program Address register.
( )	The contents of the location specified within the parenthesis.

CENTRAL PROCESSOR INSTRUCTIONS

OPERATION CODE 00 00 OCTAL 00 HEX

– † Built In Test (BIT)
Perform the operation specified for the m–value in Table I.

OPERATION CODE 00 2 OCTAL 02 HEX

SPT RK Format – STACK PUT TOP
Store the contents of memory address Y at memory address defined by (R_a), then store the contents of R_a at memory address Y.

OPERATION CODE 00 3 OCTAL 03 HEX

BL RX Format BYTE LOAD
Load the selected byte from address Y in bits 7 through 0 of R_a and clear bits 15 through 8. (CC)

OPERATION CODE 01 0 OCTAL 04 HEX

LR RR Format – LOAD
Load (R_m) in R_a. (CC)

TABLE I. BUILT-IN TEST (BIT) m-VALUES

ULTRA Symbol	m-Value	a-Value	Operation
ICP	01	-	Execute the CP BIT Subtest specified by $(R_a)_{7-0}$
SRM	02	0	RAM BIT Signature to CP Control Memory $0170–0175_8$, $78–7D_{16}$
LRM	03	0	CP Control Memory $0170–0175_8$, $78–7D_{16}$ to RAM BIT Signature
IMP	04	0	Execute Maintenance Processor BIT
RSCS	06	-	Semiconductor Memory Status Register to R_a
EEC *	07	-	Enable Error Correct Logic or Memory Scrub
DEC*	10	-	Disable Error Correct Logic or Memory Scrub
SEL	11	-	Search Error Log
m = 00, 05, $12–17_8$, $A–F_{16}$ Illegal, causes CP Instruction Fault Interrupt when executed.			

* R_a = Address Bits 21–16 of the Memory Controller being referenced.
$R_{a+1} \neq 0$ indicates error correct function is selected.
$R_{a+1} = 0$ indicates scrub function is selected.

OPERATION CODE 01 1 OCTAL 05 HEX

LI — RI Format Type 2 – LOAD
Load the contents of memory address Y in R_a. Y = (R_m). (CC)

OPERATION CODE 01 2 OCTAL 06 HEX

LK — RK Format – LOAD
Load the Operand Y in R_a. (CC)

OPERATION CODE 01 3 OCTAL 07 HEX

L — RX Format – LOAD
Load the contents of memory address Y in R_a. (CC)

OPERATION CODE 02 0 OCTAL 08 HEX

– — RR Format – UNARY ARITHMETIC
Perform the operation specified for the m–value in Table II.

OPERATION CODE 02 1 OCTAL 09 HEX

LDI — RI Format, Type 2 – LOAD DOUBLE
Load the contents of addresses Y and Y+1 in R_a and R_{a+1} respectively. Y = (R_m). Y must be an even address. (CC)

OPERATION CODE 02 3 OCTAL 0B HEX

LD — RX Format – LOAD DOUBLE
Load the contents of memory addresses Y and Y+1 in R_a and R_{a+1} respectively. Y must be an even address. (CC)

OPERATION CODE 03 0 OCTAL 0C HEX

– † RR Format – UNARY–CONTROL
Perform the operation specified in Table III for the m–value.

OPERATION CODE 03 3 OCTAL 0F HEX

LM — RX Format – LOAD MULTIPLE
Load the contents of sequential memory addresses beginning at Y, in sequential registers beginning at R_a and ending at R_m. If a is greater than m, load registers in the order R_a, R_{a+1}, ..., R_{17}, R_o, ..., R_m. Address Y is equal to y; no indexing or indirect addressing is performed.

TABLE II. UNARY ARITHMETIC INSTRUCTION m–VALUES

ULTRA Symbol	Operation	Description
PR	MAKE POSITIVE Octal 02 0 a 00 Hex 08 a 0	If (R_a) are negative, perform the two's complement of (R_a) and store the result in R_a. When the maximum negative number* is complemented, set the overflow designator. (CC) If (R_a) are positive, do not change R_a. (CC)
NR	MAKE NEGATIVE Octal 02 0 a 01 Hex 08 a 1	If (R_a) are positive and not zero, perform the two's complement of (R_a) and store the result in R_a. (CC) If (R_a) are negative or zero, do not change R_a. (CC)
RR	ROUND R_a Octal 02 0 a 02 Hex 08 a 2	Add bit 15 of R_{a+1} to (R_a) and store the result in R_a. "a" must be even.
TCR	TWO'S COMPLEMENT, SINGLE Octal 02 0 a 04 Hex 08 a 4	Perform the two's complement of (R_a) and store the result in R_a. When the maximum negative number* is complemented, set the overflow designator. (CC)
TCDR	TWO'S COMPLEMENT, DOUBLE Octal 02 0 a 05 Hex 08 a 5	Perform the two's complement of double length (R_a, R_{a+1}) and store the result in R_a, R_{a+1}. When the maximum negative number is complemented, set the overflow designator. (CC)
OCR	ONE'S COMPLEMENT, SINGLE Octal 02 0 a 06 Hex 08 a 6	Perform the one's complement of (R_a) and store the result in R_a. (CC)
IROR	INCREASE R_a BY 1 Octal 02 0 a 10 Hex 08 a 8	Increase (R_a) by 1 and store the result in R_a. (CC)
DROR	DECREASE R_a BY 1 Octal 02 0 a 11 Hex 08 a 9	Decrease (R_a) by 1 and store the result in R_a. (CC)
IRTR	INCREASE R_a BY 2 Octal 02 0 a 12 Hex 08 a A	Increase (R_a) by 2 and store the result in R_a. (CC)
DRTR	DECREASE R_a BY 2 Octal 02 0 a 13 Hex 08 a B	Decrease (R_a) by 2 and store the result in R_a. (CC)

*(1,000,000,000,000,000) binary

TABLE III. UNARY–CONTROL INSTRUCTION M–VALUES

ULTRA Symbol	Operation	Description
ER	EXECUTIVE RETURN Octal 03 0 a 00 Hex 0C a 0	If Class II interrupts are enabled, generate Class II priority 4 Interrupt, store (P)+1 in R_a; No–op if Class II is not enabled. (CC)
SSOR	STORE STATUS REGISTER # 1 Octal 03 0 a 01 Hex 0C a 1	Store the contents of Status Register 1 in R_a. (CC)
SSTR	STORE STATUS REGISTER # 2 Octal 03 0 a 02 Hex 0C a 2	Store the contents of Status Register 2 in R_a. (CC)
SCR	STORE RTC LOWER Octal 03 0 a 03 Hex 0C a 3	Store the contents of the Real Time Clock Register, lower order half, in R_a. (CC)
LPR	LOAD P Octal 03 0 a 04 Hex 0C a 4	Load (R_a) in P
LSOR	LOAD STATUS REGISTER # 1 Octal 03 0 a 05 Hex 0C a 5	Load (R_a) in Status Register 1.
LSTR	LOAD STATUS REGISTER # 2 Octal 03 0 a 06 Hex 0C a 6	Load (R_a) in Status Register 2.
LCR	LOAD RTC LOWER Octal 03 0 a 07 Hex 0C a 7	Load (R_a) in the lower order half of the Real Time Clock Register
ECR	ENABLE RTC Octal 03 0 00 10 Hex 0C a 8	Enable the Real Time Clock Register to increase by one for each cycle of the RTC oscillator and enable generation of the RTC Interrupt when the lower half of the RTC register overflows.
DCR	DISABLE RTC Octal 03 0 00 11 Hex 0C a 9	Disable the Real Time Clock Register from advancing. The RTC oscillator continues to operate. Disable further RTC interrupts but process any currently queued RTC interrupt.
LEM	LOAD AND ENABLE MONITOR CLOCK Octal 03 0 a 12 Hex 0C a A	Load (R_a) in the Monitor Clock Register, enable the register to decrease by one for each cycle of the RTC oscillator and enable generation of the monitor clock interrupt when the register contents decrements from all zeros to all ones.

TABLE III. UNARY–CONTROL INSTRUCTION M–VALUES (continued)

ULTRA Symbol	Operation	Description
DM	DISABLE MONITOR CLOCK Octal 03 0 00 13 Hex 0C 0 B	Disable the Monitor Clock Register from counting down. Disable further Monitor Clock Interrupts but process any currently queued Mon. Clk. Int.
LCRD	LOAD RTC DOUBLE Octal 03 0 a 14 Hex 0C a C	Load (R_a, R_{a+1}) in the Real Time Clock Register and enable it to advance by one for each cycle of the RTC oscillator.
SCRD	STORE RTC DOUBLE Octal 03 0 a 15 Hex 0C a D	Store the contents of the Real Time Clock Register in R_a and R_{a+1}. (CC on R_{a+1} only.)
ECIR	ENABLE RTC INTERRUPT Octal 03 0 00 16 Hex 0C 0 E	Enable the generation of an RTC Interrupt when the lower half of the RTC register overflows.
DCIR	DISABLE RTC INTERRUPT Octal 03 00 017 Hex 0C 0 F	Disable the generation of the RTC Interrupt.

OPERATION CODE 04 0 OCTAL 10 HEX

\–

RR Format – UNARY–SHIFT
Perform the operation specified in Table IV for the m–value.

OPERATION CODE 04 2 OCTAL 12 HEX

QPT

RK Format – QUEUE PUT TOP
Store contents of memory address Y at memory address specified by (R_a). If the contents of memory address Y $\neq$ 0 store (R_a) at memory address Y. If the contents of memory address Y = 0, store (R_a) at memory addresses Y and Y+1.

OPERATION CODE 04 3 OCTAL 13 HEX

BLX

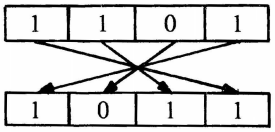
RX Format – BYTE LOAD AND INDEX BY 1
Generate memory address Y, increase (R_m) by 1, load the selected byte from memory address Y in bits 7 through 0 of R_a, clear bits 8 through 15. a$\neq$m. (CC)

OPERATION CODE 05 0 OCTAL 14 HEX

SBR

RR Format – SET BIT
Set the bit in R_a corresponding to the value of m. (CC)

TABLE IV. UNARY–SHIFT INSTRUCTION m–VALUE

ULTRA Symbol	Operation	Description
SQR	SQUARE ROOT* Octal 04 0 a 00 Hex 10 a 0	Perform the square root of the double length (R_a, R_{a+1}) and store the result in R_{a+1} with the remainder in R_a. The remainder is a 16–bit magnitude quantity and is the difference between the original number and the root squared. (CC)
RVR	REVERSE REGISTER Octal 04 0 a 01 Hex 10 a 1	Change (R_a) to the reverse order according to the 4–bit example. (CC)  Initial Final
CNT	COUNT ONES Octal 04 0 a 02 Hex 10 a 2	Count the number of one bits in (R_a), and store the count in R_{a+1}. "a" may be even or odd.
SFR	SCALE FACTOR Octal 04 0 a 03 Hex 10 a 3	Shift the double length (R_a, R_{a+1}) to the left with zeros extended to fill, until bits 15 and 14 of R_a are not equal and store the shift count in R_{a+1+1}. (If the registers contain all zeros or all ones, the shift count is 31.)
SMC	STORE MONITOR CLOCK Octal 04 0 a 04 Hex 10 a 4	Store the contents of the monitor clock in R_a.
SQRT	FLOATING POINT SQUARE ROOT* Octal 04 0 a 05 Hex 10 a 5	Perform the square root of floating point number (R_a, R_{a+1}) and store normalized result in R_a, R_{a+1}. (CC)
LCEP †	LOAD CLOCK, ENABLE PERIODIC Octal 04 0 a 06 Hex 10 a 6	Load the real-time clock lower register with (R_a), save (R_{a+1}), enable the real-time clock and interrupt. Reload the real-time clock lower register with the saved value of (R_{a+1}) upon occurrence of the real-time clock interrupt. Execution of any RTC instruction other than a store clock instruction disables the pending RTC reload.
IS †	INITIALIZE SYSTEM Octal 04 0 a 10 Hex 10 a 8	Initialize all registers and buses.
IB †	INITIALIZE BUS Octal 04 0 a 11 Hex 10 a 9	Initialize the common bus and all non–CP modules connected to bus.

* Optional Math Pac instruction. The square root of a positive number larger than 07777777777_8 or of a negative number sets the overflow designator, bit 10 in status register 1.

OPERATION CODE 05 1 OCTAL 15 HEX

LXI — RI Format, Type 2 – LOAD AND INDEX BY 1
Generate memory address Y, increase (R_m) by 1 and then load the contents of memory address Y in R_a. Y = (R_m). a ≠ m. (CC)

OPERATION CODE 05 2 OCTAL 16 HEX

QPB — RK Format – QUEUE PUT BOTTOM
Store (R_a) at the memory address defined by the contents of memory address Y+1, store (R_a) at memory address Y+1, then clear memory address defined by (R_a).

OPERATION CODE 05 3 OCTAL 17 HEX

LX — RX Format – LOAD AND INDEX BY 1
Generate memory address Y, increase (R_m) by 1 and then load the contents of memory address Y in R_a. a ≠ m. (CC)

OPERATION CODE 06 0 OCTAL 18 HEX

ZBR — RR Format – ZERO BIT (Clear Bit)
Clear the bit in R_a corresponding to the value of m. (CC)

OPERATION CODE 06 1 OCTAL 19 HEX

LDXI — RI Format, Type 2 – LOAD DOUBLE AND INDEX BY 2
Generate memory address Y, increase (R_m) by 1, load the contents of memory addresses Y+1 in R_{a+1}. Increase (R_m) by 1, load the contents of memory address Y in R_a. Y = (R_m). a, a+1 ≠ m. Y must be an even address. (CC)

OPERATION CODE 06 2 OCTAL 1A HEX

SGT — RK Format – STACK GET TOP
Load contents of memory address Y into R_a. If the contents of Y = 0, execute the next instruction. If the contents of Y ≠ 0, store the contents of the memory address specified by the contents of memory address Y at a memory address Y and skip the next instruction.

OPERATION CODE 06 3 OCTAL 1B HEX

LDX — RX Format – LOAD DOUBLE AND INDEX BY 2
Generate memory address Y, increase (R_m) by 1, load the contents of memory address Y+1 in R_{a+1}. Increase (R_m) by 1, load the contents of memory address Y into R_a. a, a+1 ≠ m. Y must be an even address. (CC)

OPERATION CODE 07 0 OCTAL 1C HEX

CBR — RR Format – COMPARE BIT
Compare the bit in R_a corresponding to the m–value with zero. (CC)

OPERATION CODE 07 1 OCTAL 1D HEX

LPI † RI Format, Type 2 – LOAD PSW
Load the contents of memory addresses Y, Y+1 and Y+2 in Program Address Register, Status Register 1 and Status Register 2, respectively. Y = (R_m). Enable Power Fault Interrupt generation.

OPERATION CODE 07 2 OCTAL 1E HEX

QGT RK Format – QUEUE GET TOP
Load the contents of Y into R_a. If the contents of Y = 0, execute the next instruction. If the contents of Y $\neq$ 0, store the contents of the memory address defined by the contents of memory address Y, ((Y)), at memory address Y and skip the next instruction and if ((Y)) = 0, store Y at memory address Y+1.

OPERATION CODE 07 3 OCTAL 1F HEX

LP † RX Format – LOAD PSW
Load the contents of memory address Y, Y+1 and Y+2 in Program Address Register, Status Register 1 and Status Register 2, respectively. Enable Power Fault Interrupt generation.

OPERATION CODE 10 0 OCTAL 20 HEX

LRSR RR Format – LOGICAL RIGHT SINGLE SHIFT
Shift (R_a) to the right n–places with zeros extended to fill. n is the value in bits 5–0 of R_m. (CC)

OPERATION CODE 10 2 OCTAL 22 HEX

LRS RK Format – LOGICAL RIGHT SINGLE SHIFT
Shift (R_a) to the right n–places with zeros extended to fill. n is the value in bits 5–0 of operand Y. (CC)

OPERATION CODE 10 3 OCTAL 23 HEX

BS RX Format – BYTE STORE
Store bits 7–0 of (R_a) in the selected byte of memory address Y.

OPERATION CODE 11 0 OCTAL 24 HEX

ARSR RR Format – ALGEBRAIC RIGHT SINGLE SHIFT
Shift (R_a) to the right n–places with sign extended to fill. n is the value in bits 5–0 of R_m. (CC)

OPERATION CODE 11 1 OCTAL 25 HEX

SI RI Format, Type 2 – STORE
Store (R_a) at memory address Y. Y = (R_m).

OPERATION CODE 11 2 OCTAL 26 HEX

ARS — Format RK – ALGEBRAIC RIGHT SINGLE SHIFT
Shift (R_a) to the right n–places with sign extended to fill. n is the value in bits 5–0 of operand Y. (CC)

OPERATION CODE 11 3 OCTAL 27 HEX

S — RX Format – STORE
Store (R_a) at memory address Y.

OPERATION CODE 12 0 OCTAL 28 HEX

LRDR — RR Format – LOGICAL RIGHT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) to the right n–places with zeros extended to fill. n is the value in bits 5–0 of R_m. (CC)

OPERATION CODE 12 1 OCTAL 29 HEX

SDI — RI Format, Type 2 – STORE DOUBLE
Store (R_a) and (R_{a+1}) at memory addresses Y and Y+1 respectively. Y = (R_m).

OPERATION CODE 12 2 OCTAL 2A HEX

LRD — RK Format – LOGICAL RIGHT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) to the right n–places with zeros extended to fill. n is the value in bits 5–0 of operand Y. (CC)

OPERATION CODE 12 3 OCTAL 2B HEX

SD — RX Format – STORE DOUBLE
Store (R_a) and (R_{a+1}) at memory addresses Y and Y+1 respectively.

OPERATION CODE 13 0 OCTAL 2C HEX

ARDR — RR Format – ALGEBRAIC RIGHT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) to the right n–places with the sign extended to fill. n is the value in bits 5–0 of R_m. (CC)

OPERATION CODE 13 2 OCTAL 2E HEX

ARD — RK Format – ALGEBRAIC RIGHT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) to the right n–places with the R_a sign extended to fill. n is the value in bits 5–0 of operand Y. (CC)

OPERATION CODE 13 3 OCTAL 2F HEX

SM — RX Format – STORE MULTIPLE
Store in sequential memory addresses beginning at Y, the contents of sequential registers beginning at R_a and ending at R_m. If a is greater than m store registers in the order R_a, R_{a+1}, ..., R_{17}, R_0, ..., R_m. Y equals y; no indexing or indirect addressing is performed.

OPERATION CODE 14 0 OCTAL 30 HEX

ALSR — RR Format – ALGEBRAIC LEFT SINGLE SHIFT
Shift (R_a) to the left n–places with zeros extended to fill. n is the value in bits 5–0 of R_m. (CC)

OPERATION CODE 14 2 OCTAL 32 HEX

ALS — RK Format – ALGEBRAIC LEFT SINGLE SHIFT
Shift (R_a) to the left n–places with zeros extended to fill. n is the value in bits 5–0 of operand Y. (CC)

OPERATION CODE 14 3 OCTAL 33 HEX

BSX — RX Format – BYTE STORE AND INDEX BY 1
Store bits 7–0 of R_a in the selected byte memory address Y; and then increase (R_m) by 1.

OPERATION CODE 15 0 OCTAL 34 HEX

CLSR — RR Format – CIRCULAR LEFT SINGLE SHIFT
Shift (R_a) circularly to the left n–places. n is the value in bits 5–0 of R_m. (CC)

OPERATION CODE 15 1 OCTAL 35 HEX

SXI — RI Format, Type 2 – STORE AND INDEX BY 1
Store (R_a) at memory address Y; and then increase (R_m) by 1. Y = (R_m).

OPERATION CODE 15 2 OCTAL 36 HEX

CLS — RK Format – CIRCULAR LEFT SINGLE SHIFT
Shift (R_a) circularly to the left n–places. n is the value in bits 5–0 of operand Y. (CC)

OPERATION CODE 15 3 OCTAL 37 HEX

SX — RX Format – STORE AND INDEX BY 1
Store (R_a) at memory address Y; and then increase (R_m) by 1. Y = (R_m).

OPERATION CODE 16 0 OCTAL 38 HEX

ALDR — RR Format – ALGEBRAIC LEFT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) to the left n–places with zeros extended to fill. n is the value in bits 5–0 of R_m. (CC)

OPERATION CODE 16 1 OCTAL 39 HEX

SDXI — RI Format, Type 2 – STORE DOUBLE AND INDEX BY 2
Generate memory address Y. Store (R_{a+1}) in memory address Y+1 increase (R_m) by 1. Store (R_a) in memory address Y, increase (R_m) by 1. Y = (R_m). $a \neq m$.

OPERATION CODE 16 2 OCTAL 3A HEX

ALD — RK Format – ALGEBRAIC LEFT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) to the left n–places with zeros extended to fill. n is the value in bits 5–0 of operand Y. (CC)

OPERATION CODE 16 3 OCTAL 3B HEX

SDX — RX Format – STORE DOUBLE AND INDEX BY 2
Generate memory address Y. Store (R_{a+1}) in memory address Y+1, increase (R_m) by 1. Store (R_a) in memory address Y, increase (R_m) by 1. $a \neq m$.

OPERATION CODE 17 0 OCTAL 3C HEX

CLDR — RR Format – CIRCULAR LEFT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) circularly to the left n–places with bit 15 of R_a transferred to bit 0 of R_{a+1} in each shift. n is the value in bits 5–0 of R_m. (CC)

OPERATION CODE 17 1 OCTAL 3D HEX

SZI — RI Format, Type 2 – STORE ZEROS
Store all zeros at memory address Y. Y = (R_m).

OPERATION CODE 17 2 OCTAL 3E HEX

CLD — RK Format – CIRCULAR LEFT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) circularly to the left n–places with bit 15 of R_a transferred to bit 0 of R_{a+1} in each shift. n is the value in bits 5–0 of operand Y. (CC)

OPERATION CODE 17 3 OCTAL 3F HEX

SZ — RX Format – STORE ZEROS
Store all zeros at memory address Y.

OPERATION CODE 20 0 OCTAL 40 HEX

SUR — RR Format – SUBTRACT
Subtract (R_m) from (R_a) and store the result in R_a. (CC)

OPERATION CODE 21 1 OCTAL 41 HEX

SUI — RI Format, Type 2 – SUBTRACT
Subtract the contents of memory address Y from (R_a) and store the result in R_a. Y = (R_m). (CC)

OPERATION CODE 21 2 OCTAL 42 HEX

SUK — RK Format – SUBTRACT
Subtract Y from (R_a) and store the result in R_a. (CC)

OPERATION CODE 21 3 OCTAL 43 HEX

SU — RX Format – SUBTRACT
Subtract the contents of memory address Y from (R_a) and store the result in R_a. (CC)

OPERATION CODE 21 0 OCTAL 44 HEX

SUDR — RR Format – SUBTRACT DOUBLE
Subtract the double length (R_m, R_{m+1}) from the double length (R_a, R_{a+1}) and store the result in R_a and R_{a+1}. (CC)

OPERATION CODE 21 1 OCTAL 45 HEX

SUDI — RI Format, Type 2 – SUBTRACT DOUBLE
Subtract the double length contents of memory addresses Y, Y+1 from the double length (R_a, R_{a+1}) and store the result in R_a and R_{a+1}. Y = (R_m). Y must be an even address. (CC)

OPERATION CODE 21 3 OCTAL 47 HEX

SUD — RX Format – SUBTRACT DOUBLE
Subtract the double length contents of memory addresses Y, Y+1from the double length (R_a, R_{a+1}) and store the result in R_{a+1}. Y must be an even address. (CC)

OPERATION CODE 22 0 OCTAL 48 HEX

AR — RR Format – ADD
ADD (R_m) to (R_a) and store the result in R_a. (CC)

OPERATION CODE 22 1 OCTAL 49 HEX

AI — RI Format, Type 2 – ADD
Add the contents of memory address Y to (R_a) and store the result in R_a. Y = (R_m). (CC)

OPERATION CODE 22 2 OCTAL 4A HEX

AK — RK Format – ADD
Add operand Y to (R_a) and store the result in R_a. (CC)

OPERATION CODE 22 3 OCTAL 4B HEX

A — RX Format – ADD
Add the contents of memory address Y to (R_a) and store the result in R_a. (CC)

OPERATION CODE 23 0 OCTAL 4C HEX

ADR — RR Format – ADD DOUBLE
Add the double length (R_m, R_{m+1}) to the double length (R_a, R_{a+1}) and store the result in R_a and R_{a+1}. Y = (R_m). Y must be an even address. (CC)

OPERATION CODE 23 1 OCTAL 4D HEX

ADI — RI Format, Type 2 – ADD DOUBLE
Add the double length contents of memory addresses Y, Y+1 to the double length (R_a, R_{a+1}) and store the result in R_a and R_{a+1}. Y = (R_m. Y must be an even address. (CC)

– — RK Format – Illegal

OPERATION CODE 23 3 OCTAL 4F HEX

AD — RX Format – ADD DOUBLE
Add the double length contents of memory addresses Y, Y+1 to the double length (R_a, R_{a+1}) and store the result in R_a and R_{a+1}. Y must be an even address. (CC)

OPERATION CODE 24 0 OCTAL 50 HEX

CR — RR Format – COMPARE
Arithmetically compare (R_a) to (R_m). (CC)

OPERATION CODE 24 1 OCTAL 51 HEX

CI — RI Format, Type 2 – COMPARE
Arithmetically compare (R_a) to the content of memory address Y = (R_m). (CC)

OPERATION CODE 24 2 OCTAL 52 HEX

CK — RK Format – COMPARE
Arithmetically compare (R_a) to operand Y. (CC)

OPERATION CODE 24 3 OCTAL 53 HEX

C — RX Format – COMPARE
Arithmetically compare (R_a) to the content of memory address Y. (CC)

OPERATION CODE 25 0 OCTAL 54 HEX

CDR — RR Format – COMPARE DOUBLE
Arithmetically compare the double length (R_a, R_{a+1}) to the double length (R_m, R_{m+1}) (CC)

OPERATION CODE 25 1 OCTAL 55 HEX

CDI — RI Format, Type 2 – COMPARE DOUBLE
Arithmetically compare the double length (R_a, R_{a+1}) to the double length contents of memory addresses Y, Y+1 Y = (R_m). Y must be an even address. (CC)

OPERATION CODE 25 3 OCTAL 57 HEX

CD — RX Format – COMPARE DOUBLE
Arithmetically compare the double length (R_a, R_{a+1}) to the double length contents of memory addresses Y+1. Y must be an even address. (CC)

OPERATION CODE 26 0 OCTAL 58 HEX

MR — RR Format – MULTIPLY
Multiply (R_{a+1}) by (R_m) and store the double length result in R_a, R_{a+1}. (CC)

OPERATION CODE 26 1 OCTAL 59 HEX

MI — RI Format, Type 2 – MULTIPLY
Multiply (R_{a+1}) by the contents of memory address Y and store the double length result in R_a, R_{a+1}. Y = (R_m). (CC)

OPERATION CODE 26 2 OCTAL 5A HEX

MK — RK Format – Multiply
Multiply (R_{a+1}) by operand Y and store the double length result in R_a, R_{a+1}. (CC)

OPERATION CODE 26 3 OCTAL 5B HEX

M — RX Format – MULTIPLY
Multiply (R_{a+1}) by the contents of memory address Y and store the double length result in R_a, R_{a+1}. (CC)

OPERATION CODE 27 0 OCTAL 5C HEX

Note: For all divide operations, the remainder has the same sign as the dividend and the absolute value of the remainder is less than the absolute value of the divisor. Generation of the maximum negative number as the quotient will cause overflow.

DR — RR Format – DIVIDE
Divide the double length (R_a, R_{a+1}) by (R_m), store the quotient in R_{a+1} and the remainder in R_a. (CC)

OPERATION CODE 27 1 OCTAL 5D HEX

DI — RI Format, Type 2 – DIVIDE
Divide the double length (R_a, R_{a+1}) by the contents of memory address Y, store the quotient in R_{a+1} and the remainder in R_a. Y = (R_m). (CC)

OPERATION CODE 27 2 OCTAL 5E HEX

DK — RK Format – DIVIDE
Divide the double length (R_a, R_{a+1}) by operand Y, store the quotient in R_{a+1} and the remainder in R_a. (CC)

OPERATION CODE 27 3 OCTAL 5F HEX

D — RX Format – DIVIDE
Divide the double length (R_a, R_{a+1}) by the contents of memory address Y, store the quotient in R_{a+1} and the remainder in R_a. (CC)

OPERATION CODE 30 0 OCTAL 60 HEX

ANDR — RR Format – AND
Perform the logical AND of (R_a) and (R_m), and store the result in R_a. (CC)

OPERATION CODE 30 1 OCTAL 61 HEX

ANDI — RI Format, Type 2 – AND
Perform the logical AND of (R_a) and the contents of memory address Y and store the result in R_a. Y = (R_m). (CC)

OPERATION CODE 30 2 OCTAL 62 HEX

ANDK — RK Format – AND
Perform the logical AND of (R_a) and operand Y, and store the result in R_a. (CC)

OPERATION CODE 30 3 OCTAL 63 HEX

AND — RX Format – AND
Perform the logical AND of (R_a) and the contents of memory address Y, and store the result in R_a. (CC)

OPERATION CODE 31 0 OCTAL 64 HEX

ORR — RR Format – OR
Perform the logical OR of (R_a) and (R_m), and store the result in R_a. (CC)

OPERATION CODE 31 1 OCTAL 65 HEX

ORI — RI Format, Type 2 – OR
Perform the logical OR of (R_a) and the contents of memory address Y, and store the result in R_a. Y = (R_m). (CC)

OPERATION CODE 31 2 OCTAL 66 HEX

ORK — RK Format – OR
Perform the logical OR of (R_a) and operand Y, and store the result in R_a. (CC)

OPERATION CODE 31 3 OCTAL 67 HEX

OR — RX Format – OR
Perform the logical OR of (R_a) and the contents of memory address Y, and store the result in R_a. (CC)

OPERATION CODE 32 0 OCTAL 68 HEX

XORR — RR Format – EXCLUSIVE OR
Perform the exclusive OR of (R_a) and (R_m) and store the result in R_a. (CC)

OPERATION CODE 32 1 OCTAL 69 HEX

XORI — RI Format, Type 2 – EXCLUSIVE OR
Perform the exclusive OR of (R_a) and the contents of memory address Y, and store the result in R_a. Y = (R_m). (CC)

OPERATION CODE 32 2 OCTAL 6A HEX

XORK — RK Format – EXCLUSIVE OR
Perform the exclusive OR of (R_a) and operand Y, and store the result in R_a. (CC)

OPERATION CODE 32 3 OCTAL 6B HEX

XOR — RX Format – EXCLUSIVE OR
Perform the exclusive OR of (R_a) and the contents of memory address Y, and store the result in R_a. (CC)

OPERATION CODE 33 0 OCTAL 6C HEX

MSR — RR Format – MASKED SUBSTITUTE
For each bit set in (R_{a+1}), transfer the corresponding bit of (R_m) to the corresponding bit in R_a and leave the remaining bits in R_a unchanged. "a" must be even. (CC)

OPERATION CODE 31 1 OCTAL 6D HEX

MSI — RI Format, Type 2 – MASKED SUBSTITUTE
For each bit set in (R_{a+1}), transfer the corresponding bit of the contents of memory address Y to the corresponding bit in R_a and leave the remaining bits in R_a unchanged. Y = (R_m). "a" must be even. (CC)

OPERATION CODE 31 2 OCTAL 6E HEX

MSK — RK Format – MASKED SUBSTITUTE
For each bit set in (R_{a+1}), transfer the corresponding bit in operand Y to the corresponding bit in R_a and leave the remaining bits of R_a unchanged. "a" must be even. (CC)

OPERATION CODE 31 3 OCTAL 6F HEX

MS — RX Format – MASKED SUBSTITUTE
For each bit set in (R_{a+1}), transfer the corresponding bit in the contents of memory address Y to the corresponding bit in R_a and leave the remaining bits of R_a unchanged. "a" must be even. (CC)

OPERATION CODE 34 0 OCTAL 70 HEX

Note: This instruction with a positive mask will give results per Table V; with a negative mask (Bit 2^{15} of R_{a+1} set) a resulting CC = 00_2 indicates equality and a CC $\neq 00_2$ indicates inequality.

CMR

RR Format – COMPARE MASKED
Compare (bit by bit) the result of the logical AND of (R_a) and (R_{a+1}) to the result of the logical AND of (R_m) and (R_{a+1}). "a" must be even. (CC)

OPERATION CODE 34 1 OCTAL 71 HEX

CMI

RI Format, Type 2 – COMPARE MASKED
Compare (bit by bit) the logical AND of (R_a) and (R_{a+1}) to the logical AND of contents of memory address Y and (R_{a+1}). Y = (R_m). "a" must be even. (CC)

OPERATION CODE 34 2 OCTAL 72 HEX

CMK

RK Format – COMPARE MASKED
Compare (bit by bit) the logical AND of (R_a) and (R_{a+1}) to the logical AND of operand Y and (R_{a+1}). "a" must be even. (CC)

OPERATION CODE 34 3 OCTAL 73 HEX

CM

RX Format – COMPARE MASKED
Compare (bit by bit) the logical AND of (R_a) and (R_{a+1}) to the logical AND of contents of memory address Y and (R_{a+1}). "a" must be even. (CC)

OPERATION CODE 35 0 OCTAL 74 HEX

IOCR (If a = 0)
IOC (If a $\neq$ 0)

RR Format –I/O COMMAND START
If a = 0, IOC_0 executes the I/O command instructions stored in main memory addresses 000140_8, 000141_8 and clears bits 15 and 14 of address 000140_8. If a $\neq$ 0, the IOC specified by (R_a) executes the instruction at y + (R_m) and y + (R_m) + 1 and does not clear bits 14 and 15 of the contents of y + R_m. The lower two bits of the value in R_a select 1 of 4 IOCs. The upper fourteen bits are ignored.

OPERATION CODE 35 1 OCTAL 75 HEX

BFI

RI Format, Type 2 – BIASED FETCH
Set the Condition Code on the contents of memory address Y and then set the two most significant bits at that memory location leaving the remaining bits unchanged.

OPERATION CODE 35 2 OCTAL 76 HEX

REX

RK Format – REMOTE EXECUTE
Execute the instruction as specified by the contents of memory address Y; do not change (P) when reading this instruction. Then continue with the next sequential instruction unless the remote instruction changes (P).

OPERATION CODE 35 3 OCTAL 77 HEX

BF — RX Format – BIASED FETCH
Set the Condition Code on the contents of memory address Y and then set the two most significant bits at that memory location leaving the remaining bits unchanged.

OPERATION CODE 36 0 OCTAL 78 HEX

CLR — RR Format – COMPARE LOGICAL REGISTER
Perform the unsigned comparison between (R_a) and (R_m). Set the overflow and carry designators according to the results. (CC)

OPERATION CODE 36 1 OCTAL 79 HEX

CLI — RI Format, Type 2 – COMPARE LOGICAL INDIRECT
Perform the unsigned comparison between (R_a) and the contents of memory address Y. Set the overflow and carry designators according to the results. (CC)

OPERATION CODE 36 2 OCTAL 7A HEX

CLK — RK Format – COMPARE LOGICAL CONSTANT
Perform the unsigned comparison between (R_a) and Y. Set the overflow and carry designators according to the results. (CC)

OPERATION CODE 36 3 OCTAL 7B HEX

CL — RX Format – COMPARE LOGICAL
Perform the unsigned comparison between (R_a) and the contents of memory address Y. Set the overflow and carry designators according to the results. (CC)

OPERATION CODE 37 0 m = 0 THRU 7 OCTAL 7C a m = 0 THRU 7 HEX

(Optional Math Pac instructions)

RR Format – CORDIC (m = 0 – 7), floating point (m = 10 – 13), quadruple shift (m = 16, 17). Perform the arithmetic function specified by the m–designator on the initial contents of three general registers specified by the a–designator and leave the results in the same respective general registers. See Table V for input parameters and output results when m = 0 – 7.

TABLE V. TRIGONOMETRIC AND HYPERBOLIC FUNCTIONS

(Operation Code 37, Octal, 7C Hex)

$37_8/7C_{16}$ m	NAME OF FUNCTION	INPUT PARAMETERS			OUTPUT RESULTS		
		R_a	R_a+1	R_a+2	R_a (Y)	R_a+1 (X)	R_a+2 (W)
0	Trigonometric vector without correction	y	x	0	0	$X = \frac{R}{K} = \sqrt{\frac{x^2 + y^2}{K}}$	$W = \Theta = \tan^{-1}\frac{y}{x}$
1	Trigonometric rotate without correction	y	x	Θ	$Y = \frac{y\cos\Theta + x\sin\Theta}{K}$	$X = \frac{x\cos\Theta - y\sin\Theta}{K}$	0
2	Trigonometric vector with correction	y	x	0	0	$X = R = \sqrt{x^2 + y^2}$	$W = \Theta = \tan^{-1}\frac{y}{x}$
3	Trigonometric rotate with correction	y	x	Θ	$Y = y\cos\Theta + x\sin\Theta$	$X = x\cos\Theta - y\sin\Theta$	0
4	Hyperbolic vector without correction	y	x	0	0	$X = \sqrt{\frac{x^2 - y^2}{K_1}}$	$W = v = \tanh^{-1}\frac{y}{x}$
5	Hyperbolic rotate without correction	y	x	v	$Y = \frac{y\cosh v + x\sinh v}{K_1}$	$X = \frac{x\cosh v + y\sinh v}{K_1}$	0
6	Hyperbolic vector with correction	y	x	0	0	$X = \sqrt{x^2 - y^2}$	$W = v = \tanh^{-1}\frac{y}{x}$
7	Hyperbolic rotate with correction	y	x	v	$Y = y\cosh v + x\sinh v$	$X = x\cosh v + y\sinh v$	0
1	Sin Θ, cos Θ (used for higher speed)	0	0.46672_8	Θ	$Y = \sin\Theta$	$X = \cos\Theta$	0
3	Sin Θ, cos Θ (used for more accuracy)	0	1	Θ	$Y = \sin\Theta$	$X = \cos\Theta$	0
1	Polar to Cartesian	0	R	Θ	$Y = \frac{R\sin\Theta}{K}$	$X = \frac{R\cos\Theta}{K}$	0
3	Polar to Cartesian with prescale	0	R	Θ	$Y = R\sin\Theta$	$X = R\cos\Theta$	0
6	Log_e x	x–1	x+1	0	0	$2\sqrt{x}$	$W = 1/2\ \text{log}_e\ x$ $= \tanh^{-1}\frac{x-1}{x+1}$
7	Exponential	1	1	v	$Y = e^v = \cosh v + \sinh v$	$X = e^v = \cosh v + \sinh v$	0

NOTES

X & Y	Cartesian coordinates
Θ	Angle of rotation Trigonometric mode
v	Angle of rotation Hyperbolic mode
K	0.46672_8, $0.4DBA_{16}$
K_1	1.15217_8, $1.1A8F_{16}$

Bit 15 of all input parameters indicates sign: 0 = positive, 1 = negative.

Two's Complement notation is used for negative values.

The radix point for Registers R_a and R_a+1 must be the same.

The radix point for W = Constant in hyperbolic mode is between bit 2^{15} and bit 2^{14}.

The maximum value for positive trigonometric coordinates x and y is 333668 for m = 0, 1 and 552028 for m = 2, 3, ($36F6_{16}$, $5A82_{16}$).

The maximum value for positive hyperbolic coordinates x and y is 327008 for m = 5 and 265748 for m = 7, ($35CD_{16}$, $2D7C_{16}$).

Angle Θ is represented in Binary Angular Measurement (BAMS), Bit 2^{15} represents 180°. Each successive bit equal to one represents an angle one-half as large as its adjoining higher order bit. Least significant bit = .0054931° = 19.7".

y/x .75 for m $\leq$ 4, 6 and x $\leq$ 756468 for m = 6, ($76A6_{16}$).

The a–designator specifies R_a, R_{a+1} and R_{a+2}; the m–designator specifies function as follows:

	m–value	Function
VF	0	Vector function trigonometric mode (without correction)
RF	1	Rotate function trigonometric mode (without correction)
VFP	2	Vector function trigonometric mode (with correction)
RFP	3	Rotate function trigonometric mode (with correction)
VH	4	Vector function hyperbolic mode (without correction)
RH	5	Rotate function hyperbolic mode (without correction)
VHP	6	Vector function hyperbolic mode (with correction)
RHP	7	Rotate function hyperbolic mode (with correction)

OPERATION CODE 37 0 OCTAL 7C HEX

FC — RR Format – FLOATING POINT COMPARE (m = 010)
Compare a floating point number in R_a and R_{a+1} to a floating point number in memory address Y and Y+1 where Y is the contents of P+1 and set the condition code. This instruction requires both operands to be normalized and cannot be executed via an Execute Remote Instruction. "a" and "Y" may be even or odd.

OPERATION CODE 37 0 OCTAL 7C HEX

FXC — RR Format – FIXED TO FLOATING POINT CONVERSION (m = 011)
Form a normalized floating point in R_a and R_{a+1} from the binary exponent (unbiased characteristic) in R_a and an integer mantissa in R_{a+1} and set the condition code. Two's complement is used in the fixed point format to represent both negative exponents and negative mantissas. "a" may be even or odd.

OPERATION CODE 37 0 OCTAL 7C HEX

FLC — RR Format – FLOATING POINT TO FIXED CONVERSION (m = 012)
Unpack a single precision floating point number in R_a and R_{a+1} into a binary exponent (unbiased characteristic) in R_a and an integer mantissa in R_{a+1} and set condition code upon (R_{a+1}). Two's complement is used in the fixed point format to represent both negative exponents and negative mantissa. "a" may be even or odd.

OPERATION CODE 37 0 OCTAL 7C HEX

NF — RR Format – FLOATING POINT NORMALIZE (m = 013)
Normalize the floating point number in R_a and R_{a+1} and set the condition code. If underflow occurs, a floating point interrupt will occur if enabled. "a" may be even or odd.

– — RR Format (m = 014 and m = 015) – Illegal

OPERATION CODE 37 0 OCTAL 7C HEX

QAL

RR Format – ALGEBRAIC LEFT QUADRUPLE SHIFT (m = 016)
Shift the contents of R_a, R_{a+1}, R_{a+2}, and R_{a+3} left with zero fill by the shift count in bits 5 – 0 of Y, where Y is the contents of P+1. This instruction cannot be executed via an Execute Remote Instruction. "a" may be even or odd.

OPERATION CODE 37 0 OCTAL 7C HEX

QAR

RR Format – ALGEBRAIC RIGHT QUADRUPLE SHIFT (m = 017)
Shift the contents of R_a, R_{a+1}, R_{a+2}, and R_{a+3} right with sign fill by the shift count in bits 5 – 0 of Y, where Y is the contents of P+1. This instruction cannot be executed via an Execute Remote Instruction. "a" may be even or odd.

OPERATION CODE 37 1 m = 0 THRU 7 OCTAL 7D a m = 0 THRU 7 HEX

RI Format, Type 2 – MATH FLOATING POINT
Perform the operation specified for the m–value in Table VI.

OPERATION CODE 40 0 a = 0 THRU 13 OCTAL 80 a = 0 THRU B HEX

RR Format – CONDITIONAL JUMP

Test for the condition specified in Table VII for the a–value and perform one of the following:

(1) If the specified condition is met, jump to the instruction located at the address specified by (R_m). If the condition is not met, execute the next instruction.

(2) If a specified Stop, or a Stop Key condition is met stop the CP. When the CP is started after a stop or the condition is not met, load (R_m) in P and execute the instruction at that address – (unconditional jump).

OPERATION CODE 40 1 OCTAL 81 HEX

LJ

RI Format, Type 1 – LOCAL JUMP
Jump to the instruction located at memory address Y. Y = (P)+D.

OPERATION CODE 40 2 a = 0 THRU 13 OCTAL 82 a = 0 THRU 8 HEX

RK Format – CONDITIONAL JUMP

Test for the condition specified in Table VII for the a–value and perform one of the following:

(1) If the specified condition is met, jump to the instruction located at the address specified by operand Y. If the condition is not met execute the next instruction.

TABLE VI. MATH PAC FLOATING POINT INSTRUCTION m–VALUES

ULTRA Symbol	m Value	Operation	Description
SIN	0	FLOATING POINT SINE	Perform the sine X operation on the floating point value in R_a, R_{a+1} and return the result to R_a, R_{a+1}. The input value, X, is required to be expressed in radian units.
COS	1	FLOATING POINT COSINE	Perform the cosine X operation on the floating point value in R_a, R_{a+1} and return the result to R_a, R_{a+1}. The input value, X, is required to be expressed in radian units.
TAN	2	FLOATING POINT TANGENT	Perform the tangent X operation on the floating point value in R_a, R_{a+1}, and return the result to R_a, R_{a+1}. The input value, X, is required to be expressed in radian units.
ASIN	3	FLOATING POINT ARCSINE	Perform the arcsine X operation on the floating point X value in R_a, R_{a+1}, and return the result to R_a, R_{a+1}. Input values of magnitude greater than 1 result in overflow. The output value is expressed in radian units.
ACOS	4	FLOATING POINT ARCCOSINE	Perform the arccosine X operation on the floating point X value in R_a, R_{a+1}, and return the result to R_a, R_{a+1}. Input values of magnitude greater than 1 result in overflow. The output value is expressed in radian units.
ATAN	5	FLOATING POINT ARCTANGENT	Perform the arctangent X operation on the floating point X value in R_a, R_{a+1}, and return the result to R_a, R_{a+1}. The output value is expressed in radian units.
EXP	6	FLOATING POINT EXPONENT	Perform the exponential (e^X) operation on the floating point X value in R_a, R_{a+1} and return the result to R_a, R_{a+1}.
ALOG	7	FLOATING POINT LOG_eX	Perform the natural logarithm ($\log_e X$) operation on the floating point X value in R_a, R_{a+1} and return the result to R_a, R_{a+1}.

TABLE VII. CONDITIONS FOR a–VALUE IN JUMP INSTRUCTIONS

ULTRA Symbol for Format			a– Value	Jump Condition	
40 0 Octal 80 Hex	40 2 Octal 82 Hex	40 3 Octal 83 Hex		Condition code for Compare Operation Indicates	Condition code for Compare Operation Indicates
JER	JE	JE	0	Zero	Equal
JNER	JNE	JNE	1	Not Zero	Not Equal
JGER	JGE	JGE	2	Positive	Greater Than or Equal
JLSR	JLS	JLS	3	Negative	Less Than
JOR	JO	JO	4	Overflow designator set	
JCR	JC	JC	5	Carry Designator is set	
JPTR	JPT	JPT	6	Power is out of tolerance	
JBR	JB	JB	7	Bootstrap 2 is selected	
JR	J	J	10	Unconditional jump	
JSR	JS	JS	11 †	Unconditional Stop; jump on restart	
JKSR	JKS	JKS	12 †	Stop if program stop key 1 is selected, then jump on restart; otherwise, unconditional jump	
JKSR	JKS	JKS	13 †	Stop if program stop key 2 is selected, then jump on restart; otherwise, unconditional jump	
–	–	–	14–17	Unconditional jump	

(2) If a specified Stop, or a Stop Key condition is met, stop the CP. When the CP is started after a stop or the condition is not met, load the operand Y in P and execute the instruction at that address (unconditional jump).

OPERATION CODE 40 3 a = 0 THRU 13 OCTAL 83 a = 0 THRU 8 HEX

RX Format – CONDITIONAL JUMP

Test for the condition specified in Table VII for the a–value and perform one of the following:

(1) If the specified condition is met, jump to the instruction located at the address specified by the contents of memory address Y. If the condition is not met execute the next instruction.

(2) If a specified Stop, or a Stop Key condition is met, stop the computer. When the CP is started after a stop or the condition is not met, load (Y) in P and execute the instruction at that address (unconditional jump).

OPERATION CODE 41 0 OCTAL 84 HEX

XJR — RR Format – INDEX JUMP

Test (R_a) and perform one of the following:

(1) If (R_a) does not equal zero, decrease (R_a) by 1, jump to the instruction located at the address stored in R_m.

(2) If (R_a) equals zero, execute the next instruction.

OPERATION CODE 41 1 OCTAL 85 HEX

LJI — RI Format, Type 1 – LOCAL JUMP INDIRECT
Jump unconditionally to the address specified by the contents of memory address Y. Y = (P)+D.

OPERATION CODE 41 2 OCTAL 86 HEX

XJ — RK Format – INDEX JUMP

Test (R_a) and perform one of the following:

(1) If (R_a) does not equal zero, decrease (R_a) by 1 and jump to the instruction located at address Y.

(2) If (R_a) equals zero, execute the next instruction.

OPERATION CODE 41 3 OCTAL 87 HEX

XJ — RX Format – INDEX JUMP

Test (R_a) and perform one of the following:

(1) If (R_a) does not equal zero, decrease (R_a) by 1 and jump to the instruction located at the address specified by the contents of memory address Y.

(2) If (R_a) equals zero, execute the next instruction.

OPERATION CODE 42 0 OCTAL 88 HEX

JLRR — RR Format – JUMP AND LINK REGISTERS
Store (P)+1 in R_a, and jump to the instruction located at the address stored in R_m.

OPERATION CODE 42 2 OCTAL 8A HEX

JLR — RK Format – JUMP AND LINK REGISTER
Store (P)+2 in R_a, and jump to the instruction located at the address specified by operand Y.

OPERATION CODE 42 3 OCTAL 8B HEX

JLR — RX Format – JUMP AND LINK REGISTER

Store (P)+2 in R_a, and jump to the instruction located at the address specified by the contents of address Y.

OPERATION CODE 43 1 OCTAL 8D HEX

LJLM — RI Format, Type 1 – LOCAL JUMP AND LINK MEMORY

Store (P)+1 at memory address Y, and jump to the instruction located at memory address Y+1. Y = (P)+D.

OPERATION CODE 43 2 OCTAL 8E HEX

JLM — RK Format – JUMP AND LINK MEMORY

Store (P)+2 at memory address Y, and jump to the instruction located at memory address Y+1.

OPERATION CODE 43 3 OCTAL 8F HEX

JLM — RX Format – JUMP AND LINK MEMORY

Store (P)+2 at the address specified by the contents of address Y, and jump to the instruction located at the address specified by (Y)+1.

OPERATION CODE 44 0 OCTAL 90 HEX

JXR — RR Format – JUMP REGISTER = 0

Test (R_a) and perform one of the following:

(1) If (R_a) equals zero, jump to the instruction located at the address stored in R_m.

(2) If (R_a) does not equal zero, execute the next instruction.

OPERATION CODE 44 1 OCTAL 91 HEX

LJE — RI Format, Type 1 – LOCAL JUMP EQUAL

Test the Condition Code in the Status Register and perform one of the following:

(1) If bit 8 of the Condition Code is "zero", jump to the instruction located at memory address Y. Y = (P)+D.

(2) If bit 8 of the Condition code is "one", execute the next instruction.

OPERATION CODE 44 2 OCTAL 92 HEX

JZ — RK Format – JUMP REGISTER = 0

Test (R_a) and perform one of the following:

(1) If (R_a) equals zero, jump to the instruction located at the address specified by operand Y.

(2) If (R_a) does not equal zero, execute the next instruction.

OPERATION CODE 44 3 OCTAL 93 HEX

JZ RX Format – JUMP REGISTER = 0

Test (R_a) and perform one of the following:

(1) If (R_a) equals zero, jump to the instruction located at address specified by the contents of memory address Y.

(2) If (R_a) does not equal zero, execute the next instruction.

OPERATION CODE 45 0 OCTAL 94 HEX

JNZR RR Format – JUMP REGISTER $\neq$ 0

Test (R_a) and perform one of the following:

(1) If (R_a) does not equal zero, jump to the instruction located at the address specified by R_m.

(2) If (R_a) equals zero, execute the next instruction.

OPERATION CODE 45 1 OCTAL 95 HEX

LJNE RI Format, Type 1 – LOCAL JUMP NOT EQUAL

Test the Condition Code and perform one of the following:

(1) If bit 8 of the Condition Code is "one", jump to the instruction located at memory address Y. Y = (P)+D.

(2) If bit 8 of the Condition Code is "zero", execute the next instruction.

OPERATION CODE 45 2 OCTAL 96 HEX

JNZ RK Format – JUMP REGISTER $\neq$ 0

Test (R_a) and perform one of the following:

(1) If (R_a) does not equal zero, jump to the instruction located at the address specified by operand Y.

(2) If (R_a) equals zero, execute the next instruction.

OPERATION CODE 45 3 OCTAL 97 HEX

JNZ — RX Format – JUMP REGISTER $\neq 0$

Test (R_a) and perform one of the following:

(1) If (R_a) does not equal zero, jump to the instruction located at the address specified by the contents of the memory address y.

(2) If (R_a) equals zero, execute the next instruction.

OPERATION CODE 46 0 OCTAL 98 HEX

JPR — RR Format – JUMP REGISTER POSITIVE

Test (R_a) and perform one of the following:

(1) If (R_a) is equal to or greater than zero, jump to the instruction located at the address specified by R_m.

(2) If (R_a) is less than zero, execute the next instruction.

OPERATION CODE 46 1 OCTAL 99 HEX

LJGE — RI Format, Type 1 – LOCAL JUMP GREATER THAN OR EQUAL

Test the Condition Code and perform one of the following:

(1) If bit 9 of the Condition Code is "zero", jump to the instruction located at memory address Y. Y = (P)+D.

(2) If bit 9 of the Condition Code is "one", execute the next instruction.

OPERATION CODE 46 2 OCTAL 9A HEX

JP — RK Format – JUMP REGISTER POSITIVE

Test (R_a) and perform one of the following:

(1) If (R_a) is equal or greater than zero, jump to the instruction located at the address specified by operand y.

(2) If (R_a) is less than zero, execute the next instruction.

OPERATION CODE 46 3 OCTAL 9B HEX

JP — RX Format – JUMP REGISTER POSITIVE

Test (R_a) and perform one of the following:

(1) If (R_a) is equal to or greater than zero, jump to the instruction located at address specified by the contents of memory address Y.

(2) If (R_a) is less than zero, execute the next instruction.

OPERATION CODE 47 0 OCTAL 9C HEX

JNR RR Format – JUMP REGISTER NEGATIVE

Test (R_a) and perform one of the following:

(1) If (R_a) is less than zero, jump to the instruction located at the address specified by R_m.

(2) If (R_a) is equal to or greater than zero, execute the next instruction.

OPERATION CODE 47 1 OCTAL 9D HEX

LJLS RI Format, Type 1 – LOCAL JUMP LESS THAN

Test the Condition Code and perform one of the following:

(1) If bit 9 of the Condition Code is "one", jump to the instruction located at memory address Y. Y = (P)+D.

(2) If bit 9 of the Condition Code is "zero", execute the next instruction.

OPERATION CODE 47 2 OCTAL 9E HEX

JN RK Format – JUMP REGISTER NEGATIVE

Test (R_a) and perform one of the following:

(1) If (R_a) is less than zero, jump to the instruction located at the address specified by operand Y.

(2) If (R_a) is equal to or greater than zero, execute the next instruction.

OPERATION CODE 47 3 OCTAL 9F HEX

JN RX Format – JUMP REGISTER NEGATIVE

Test (R_a) and perform one of the following:

(1) If (R_a) is less than zero, jump to the instruction located at the address specified by the contents of memory address Y.

(2) If (R_a) is equal to or greater than zero, execute the next instruction.

OPERATION CODE 50 0 OCTAL A0 HEX

(Optional Math Pac instructions)

FSUR — RR Format – FLOATING POINT SUBTRACT
Subtract the floating point number (R_m, R_{m+1}) from the floating point number (R_a, R_{a+1}); store the floating point difference in R_a, R_{a+1} and then set the Condition Code. If residue is specified, store the residue in R_{a+2} and R_{a+3} in floating point format.

OPERATION CODE 50 1 OCTAL A1 HEX

FSUI — RI Format, Type 2 – FLOATING POINT SUBTRACT
Subtract the floating point number at memory address Y, Y+1 from the floating point number (R_a, R_{a+1}); store the floating point difference in R_a, R_{a+1} and then set the Condition Code. If residue is specified, store the residue in R_{a+2} and R_{a+3} in floating point format. Y = (R_m). Y must be an even address.

OPERATION CODE 50 3 OCTAL A3 HEX

FSU — RX Format – FLOATING POINT SUBTRACT
Subtract the floating point number at memory addresses Y, Y+1 from the floating point number (R_a, R_{a+1}); store the floating point difference in R_a, R_{a+1} and then set the Condition Code. If residue is specified, store the residue in R_{a+2} and R_{a+3} in floating point format. Y must be an even address.

OPERATION CODE 51 0 OCTAL A4 HEX

(Optional Math Pac instructions)

FAR — RR Format – FLOATING POINT ADD
Add the floating point number (R_m, R_{m+1}), to the floating point number (R_a, R_{a+1}); store the floating point sum in R_a, R_{a+1} and then set the Condition Code. If residue is specified, store the residue in R_{a+2} and R_{a+3} in floating point format.

OPERATION CODE 51 1 OCTAL A5 HEX

FAI — RI Format, Type 2 – FLOATING POINT ADD
Add the floating point number at memory addresses Y, Y+1 to the floating point number (R_a, R_{a+1}); store the floating point sum in R_a, R_{a+1} and then set the Condition Code. If residue is specified, store the residue in R_{a+2} and R_{a+3} in floating point format. Y = (R_m). Y must be an even address.

OPERATION CODE 51 3 OCTAL A7 HEX

FA — RX Format – FLOATING POINT ADD
Add the floating point number at memory addresses Y, Y+1 to the floating point number (R_a, R_{a+1}); store the floating point sum in R_a, R_{a+1} and then set the Condition Code. If residue is specified, store the residue in R_{a+2} and R_{a+3} in floating point format. Y must be an even address.

OPERATION CODE 52 0 OCTAL A8 HEX

(Optional Math Pac instructions)

FMR — RR Format – FLOATING POINT MULTIPLY
Multiply the floating point number (R_m, R_{m+1}) by the floating point number (R_a, R_{a+1}); store the floating point product in R_a, R_{a+1} and then set the Condition Code. (R_a, R_{a+1}) will be a floating point number representing the most significant digits of the product. If residue is specified, R_{a+2} and R_{a+3} will contain a floating point number representing the least significant portion of the product.

OPERATION CODE 52 1 OCTAL A9 HEX

FMI — RI Format, Type 2 – FLOATING POINT MULTIPLY
Multiply the floating point number at memory addresses Y, Y+1 by the floating point number (R_a, R_{a+1}); store the floating point product in R_a, R_{a+1} and then set the Condition Code. (R_a, R_{a+1}) will be a floating point number representing the most significant digits of the product. If residue is specified, R_{a+2} and R_{a+3} will contain a floating point number representing the least significant portion of the product. Y = (R_m). Y must be an even address.

OPERATION CODE 52 3 OCTAL AB HEX

FM — RX Format – FLOATING POINT MULTIPLY
Multiply the floating point number at memory addresses Y, Y+1 by the floating point number (R_a, R_{a+1}); store the floating point product in R_a, R_{a+1} and then set the Condition Code. (R_a, R_{a+1}) will be a floating point representing the most significant digits of the product. If residue is specified, R_{a+2} and R_{a+3} will contain a floating point number representing the least significant portion of the product. Y must be an even address.

OPERATION CODE 53 0 OCTAL AC HEX

(Optional Math Pac instruction)

FDR — RR Format – FLOATING POINT DIVIDE
Divide the floating point number (R_a, R_{a+1}) by the floating point number (R_m, R_{m+1}); store the floating point quotient in R_a, R_{a+1} and then set the Condition Code. If residue is specified, R_{a+2} and R_{a+3} will contain the remainder in floating point format.

OPERATION CODE 53 1 OCTAL AD HEX

FDI — RI Format, Type 2 – FLOATING POINT DIVIDE
Divide the floating point number (R_a, R_{a+1}) by the floating point number at memory addresses Y, Y+1; store the floating point quotient in R_a, R_{a+1} and then set the Condition Code. If residue is specified, $R_a + 2$ and $R_a + 3$ will contain the remainder in floating point format. Y = (R_m). Y must be an even address.

OPERATION CODE 53 3 OCTAL AF HEX

FD — RX Format – FLOATING POINT DIVIDE

Divide the floating point number (R_a, R_{a+1}) by the floating point number at memory addresses Y, Y+1; store the floating point quotient in R_a, R_{a+1} and then set the Condition Code. If residue is specified, R_{a+2} and R_{a+3} will contain the remainder in floating point format. Y must be an even address.

OPERATION CODE 54 0 OCTAL B0 HEX

LARR † RR Format – LOAD PAGE ADDRESS REGISTER

Load (R_m) in the page register specified by (R_a) 5–0 within the page register set specified by (R_a) 7, 6.

OPERATION CODE 54 1 OCTAL B1 HEX

LARI † RI Format, Type 2 – LOAD PAGE ADDRESS REGISTER

Load the page register specified by (R_a) 5–0 within the page register set specified by (R_a) 7, 6 with the contents of the memory address specified by (R_m).

OPERATION CODE 54 3 OCTAL B3 HEX

LARM † RX Format – LOAD PAGE ADDRESS REGISTER MULTIPLE

Load the contents of sequential memory addresses beginning at Y, in sequential page registers beginning at the address defined by (R_a) 5–0 within the page register set specified by (R_a) 7, 6 until the number of executions equals one plus the count in (R_a) 15–8. A count of "zero" loads one page register.

OPERATION CODE 55 0 OCTAL B4 HEX

SARR † RR Format – STORE PAGE ADDRESS REGISTER

Store the page register by (R_a) 5–0 within the page register set specified by (R_a) 7, 6 in R_m.

OPERATION CODE 55 1 OCTAL B5 HEX

SARI † RI Format, Type 2 – STORE PAGE ADDRESS REGISTER

Store the page register specified by (R_a) 5–0 within the page register set specified by (R_a) 7, 6 at the memory address specified by (R_m).

OPERATION CODE 55 3 OCTAL B7 HEX

SARM † RX Format – STORE PAGE ADDRESS REGISTER MULTIPLE

Store the contents of sequential page registers beginning at the register number defined by (R_a) 5–0 within the page register set specified by (R_a) 7, 6 in sequential memory addresses beginning at Y until the number of executions equals one plus the count in (R_a) 13–8. A count of "zero" stores one page register.

OPERATION CODE 56 0 OCTAL B8 HEX

(Optional Math Pac instructions)

MDR — RR Format – DOUBLE PRECISION MULTIPLY
Multiply the double length (R_m, R_{m+1}) by the double length (R_a, R_{a+1}) store the result in R_a*, R_{a+1}, R_{a+2}, R_{a+3}. (CC)

OPERATION CODE 56 1 OCTAL B9 HEX

MDI — RI Format, Type 2 – DOUBLE PRECISION MULTIPLY
Multiply the double length contents of memory addresses Y, Y+1, by the double length (R_a, R_{a+1}) store the results in R_a*, R_{a+1}, R_{a+2}, R_{a+3}. Y = (R_m). Y must be an even address. (CC)

OPERATION CODE 56 3 OCTAL BB HEX

MD — RX Format – DOUBLE PRECISION MULTIPLY
Multiply the double length contents of memory addresses Y, Y+1 by the double length (R_a, R_{a+1}) store the results in R_a*, R_{a+1}, R_{a+2}, R_{a+3}. Y must be an even address. (CC)

OPERATION CODE 57 0 OCTAL BC HEX

See Note under Operation Code 27

(Optional Math Pac instructions)

DDR — RR Format – DOUBLE PRECISION DIVIDE
Divide the number (R_a*, R_{a+1}, R_{a+2}, R_{a+3}) by the double length number (R_m, R_{m+1}), store the double length quotient in R_{a+2}, R_{a+3} and the double length remainder in R_a, R_{a+1}. Generation of the maximum negative number as the quotient shall cause overflow. (CC)

OPERATION CODE 57 1 OCTAL BD HEX

DDI — RI Format, Type 2 – DOUBLE PRECISION DIVIDE
Divide the number (R_a*, R_{a+1}, R_{a+2}, R_{a+3}) by the double length contents of memory addresses Y, Y+1, store the double length quotient in R_{a+2}, R_{a+3} and the double length remainder in R_a, R_{a+1}. Generation of the maximum negative number as the quotient shall cause overflow. Y = (R_m). (CC)

*Bit 15 of (R_a) is the sign bit for the 64–bit operand.

OPERATION CODE 57 3 OCTAL BF HEX

DD — RX Format – DOUBLE PRECISION DIVIDE
Divide the number (R_a*, R_{a+1}, R_{a+2}, R_{a+3}) by the double length contents of memory addresses Y, Y+1 store the double length quotient in R_{a+2}, R_{a+3} and the double length remainder in R_a, R_{a+1}. Generation of the maximum negative number as the quotient shall cause overflow. Y must be an even address. (CC)

OPERATION CODE 60 0 OCTAL C0 HEX

LLRS — RL–1 Format – LOGICAL RIGHT SINGLE SHIFT
Shift (R_a) right n–places with zeros extended to fill. n is the value of the m–designator. (CC)

OPERATION CODE 60 1 OCTAL C1 HEX

LARS — RL–2 Format – ALGEBRAIC RIGHT SINGLE SHIFT
Shift (R_a) right n–places with the sign extended to fill. n is the value of the m–designator. (CC)

OPERATION CODE 60 2 OCTAL C2 HEX

LLRD — RL–3 Format – LOGICAL RIGHT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) right n–places with zeros extended to fill. n is the value of the m–designator. (CC)

OPERATION CODE 60 3 OCTAL C3 HEX

LARD — RL–4 Format – ALGEBRAIC RIGHT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) right n–places with sign extended to fill. n is the value of the m–designator. (CC)

OPERATION CODE 61 0 OCTAL C4 HEX

LALS — RL–1 Format – ALGEBRAIC LEFT SINGLE SHIFT
Shift (R_a) left n–places with zeros extended to fill. n is the value of the m–designator. (CC)

OPERATION CODE 61 1 OCTAL C5 HEX

LCLS — RL–2 Format – CIRCULAR LEFT SINGLE SHIFT
Shift (R_a) left circular n–places. n is the value of the m–designator. (CC)

OPERATION CODE 61 2 OCTAL C6 HEX

LALD — RL–3 Format – ALGEBRAIC LEFT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) left n–places with zeros extended to fill. n is the value of the m–designator. (CC)

OPERATION CODE 61 3 OCTAL C7 HEX

LCLD — RL–4 Format – CIRCULAR LEFT DOUBLE SHIFT
Shift the double length (R_a, R_{a+1}) left circular n–places. n is the value of the m–designator. (CC)

OPERATION CODE 62 0 OCTAL C8 HEX

LSU — RL–1 Format – SUBTRACT
Subtract the 4–bit m–value from (R_a), store the result in R_a. (CC)

OPERATION CODE 62 1 OCTAL C9 HEX

LSUD — RL–2 Format – SUBTRACT DOUBLE
Subtract the 4–bit m–value from the double length (R_a, R_{a+1}), store the result in R_a, R_{a+1}. (CC)

OPERATION CODE 62 2 OCTAL CA HEX

LA — RL–3 Format – ADD
Add the 4–bit m–value to (R_a), store the result in R_a. (CC)

OPERATION CODE 62 3 OCTAL CB HEX

LAD — RL–4 Format – ADD DOUBLE
Add the 4–bit m–value to the double length (R_a, R_{a+1}), store the result in R_a, R_{a+1}. (CC)

OPERATION CODE 63 0 OCTAL CC HEX

LL — RL–1 Format – LOAD
Load the 4–bit m–value into R_a. (CC)

OPERATION CODE 63 1 OCTAL CD HEX

LC — RL–2 Format – COMPARE
Arithmetically compare the 4–bit m–value with (R_a). (CC)

OPERATION CODE 63 2 OCTAL CE HEX

LMUL — RL–3 Format – MULTIPLY
Multiply the 4–bit m–value by (R_{a+1}) store the double length result in R_a, R_{a+1}. (CC)

OPERATION CODE 63 3 OCTAL CF HEX

LDIV — RL–4 Format – DIVIDE
Divide the double length (R_a, R_{a+1}) by the 4–bit m–value, store the quotient in R_{a+1}, the remainder in R_a. Generation of the max. neg. number as the quotient will cause overflow. (CC)

OPERATION CODE 64 0 OCTAL D0 HEX

LIR † — RR Format – LOAD INTER–REGISTER
Load (R_m) into R_a. R_a is in register set selected by status register 1, bit 14 and R_m is in other register set.

OPERATION CODE 64 2 OCTAL D2 HEX

LMR — RK Format – LOAD MULTIPLE REGISTER
Load contents of sequential memory addresses beginning at Y and continuing through memory address Y + 17_8 – a into R_a, R_{a+1}, ..., R_{17}.

OPERATION CODE 64 3 OCTAL D3 HEX

BSU — RX Format – BYTE SUBTRACT
Subtract the selected byte of the contents of memory address Y from (R_a), store the result in R_a. (CC)

OPERATION CODE 65 0 OCTAL D4 HEX

SIR † RR Format - STORE INTER-REGISTER
Store (R_a) into R_m. R_a is in register set selected by status register 1, bit 14 and R_m is in other register set.

OPERATION CODE 65 2 OCTAL D6 HEX

SMR — RK Format – STORE MULTIPLE REGISTER
Store (R_a, R_{a+1}, ..., R_{17}) into sequential memory addresses beginning at Y and continuing through memory address Y + 17 – a.

OPERATION CODE 65 3 OCTAL D7 HEX

BA — RX Format – BYTE ADD
Add the selected byte from the contents of memory address Y to (R_a), store the sum in R_a. (CC)

OPERATION CODE 66 3 OCTAL DB HEX

BC — RX Format – BYTE COMPARE
Arithmetically compare (R_a) to the selected byte of contents of memory address Y and set the Condition Code.

OPERATION CODE 67 0 OCTAL DC HEX

– RR Format – Reserved for "user" defined CP macroinstructions; otherwise illegal.

– RI Format – (Same as RR)

– RK Format – (Same as RR)

OPERATION CODE 67 3 OCTAL DF HEX

BCX — RX Format – BYTE COMPARE AND INDEX BY 1
Arithmetically compare (R_a) to the selected byte of contents of memory address Y, set the Condition Code and increase (R_m) by 1.

OPERATION CODE 70 0 OCTAL E0 HEX

(Optional Memory Management instructions)

LPAR † RR Format – LOAD PHYSICAL ADDRESS (REGISTER)
Load the physical address (double word) associated with relative address Y = (R_m) into R_a, R_{a+1} using the page register set specified by (R_{m+1}).

OPERATION CODE 70 1 OCTAL E1 HEX

LPAI † RI Format, Type 2 – LOAD PHYSICAL ADDRESS (INDIRECT)
Read the contents of memory address Y = (R_m), using the page register set selected by status register 1. Load the physical address (double word) associated with the memory contents of Y into R_a, R_{a+1} using the page register set specified by (R_{m+1}).

OPERATION CODE 70 2 OCTAL E2 HEX

LPAK † RK Format – LOAD PHYSICAL ADDRESS (CONSTANT)
Load the physical address associated with relative address Y = y + (R_m) into R_a, R_{a+1} using the page register set selected by (R_{m+1}).

OPERATION CODE 70 3 OCTAL E3 HEX

LPA † RX Format – LOAD PHYSICAL ADDRESS
Read the contents of memory address Y = y+(R_m) using the page register set selected by status register 1. Load the physical address (double word) associated with the memory contents of Y into R_a, R_{a+1} using the page register set specified by (R_{m+1}).

OPERATION CODE 71 3 OCTAL E7 HEX

(Optional Memory Management instructions)

LMAP † RX Format – LOAD MAPPED
Load the contents of memory address Y using the page register set specified by (R_{m+1}) in R_a. (CC)

OPERATION CODE 72 3 OCTAL EB HEX

(Optional Memory Management instructions)

SMAP † RX Format – STORE MAPPED
Store (R_a) at memory address Y using the page register set specified by (R_{m+1}).

OPERATION CODE 73 1 OCTAL ED HEX

(Optional Memory Management instructions)

LPLI † RI Format, Type 2 – LOAD PHYSICAL LOCATION (INDIRECT)
Load the contents of the non–paged memory address (R_m, R_{m+1}) in R_a. (CC)

OPERATION CODE 73 3 OCTAL EF HEX

LPL † RX Format – LOAD PHYSICAL LOCATION
Load the contents of the non–paged memory address y + (R_m, R_{m+1}) in R_a. (CC)

OPERATION CODE 74 1 OCTAL F1 HEX

(Optional Memory Management instructions)

SPLI † RI Format, Type 2 – STORE PHYSICAL LOCATION (INDIRECT)
Store (R_a) at the non–paged memory address (R_m, R_{m+1}).

OPERATION CODE 74 3 OCTAL F3 HEX

SPL † RX Format – STORE PHYSICAL LOCATION
Store (R_a) at the non–paged memory address y + (R_m, R_{m+1}).

INPUT/OUTPUT INSTRUCTIONS

OPERATION CODE 70 0 OCTAL E0 HEX

ACR,
CCR

RR Format – CHANNEL CONTROL (Command/Chaining)

Perform the operation specified for the m–value and a–value in Table VIII.

OPERATION CODE 70 3 OCTAL E3 HEX

IO

RX Format – INITIATE TRANSFER (Chaining)
Load the contents of memory addresses Y and Y + 1 in control memory BCW and BAP locations, respectively, and enable input/output transfers in accordance with Table IX on the channel corresponding to the chain executing the instruction. Disable the chain until the buffer terminates and then enable the chain corresponding to buffer terminated (the m–designator is not used). Y may be even or odd.

OPERATION CODE 71 2 OCTAL E6 HEX

Input/Output Instruction
Perform the operation specified for m–value. a = channel number.

ICK

m–value = 2, a–value = channel number
Initiate Input chain, Y = chain start address

OCK

m–value = 6, a–value = channel number
Initiate Output chain, Y = chain start address

OPERATION CODE 71 2 OCTAL E6 HEX

LCMK
WCMK
WIMK

RK Format – LOAD CONTROL MEMORY (Chaining)
Load operand Y in the control memory location specified in Table X for the m–designator (a–designator values are not interpreted.) (See ICK/OCK for m values of 2 or 6.)

OPERATION CODE 71 3 OCTAL E7 HEX

WIM

RX Format – WRITE CONTROL MEMORY (Command)
Load the contents of memory address Y in the control memory location specified in Table X for the combined am–designator.

OPERATION CODE 71 3 OCTAL E7 HEX

LCM

RX Format – LOAD CONTROL MEMORY (Chaining)
Load the contents of memory address Y in the control memory location specified in Table X for the m–designator (all a–designator values are not interpreted.)

OPERATION CODE 72 3 OCTAL EB HEX

–	RR Format – Reserved for "user" defined IOC micro–instructions; otherwise illegal.
–	RI Format – Reserved for "user" defined IOC macro–instructions; otherwise illegal.
RCM RIM	RX Format – STORE CONTROL MEMORY (Command) Store in memory address Y the contents of control memory location specified in Table X for the combined am–designator.

OPERATION CODE 72 3 OCTAL EB HEX

SCM	RX Format – STORE CONTROL MEMORY (Chaining) Store in memory address Y the contents of control memory location specified in Table X for the m–designator (a–designator values are not interpreted).

TABLE VIII. CHANNEL CONTROL INSTRUCTION m–DESIGNATOR AND a–DESIGNATOR

	ULTRA Symbol	m– Value	a– Value	Operation
	ACR	0*	0	Master Clear – all channels – deactivate all chains – terminate all I/O transfers – disable all external interrupt data storage and clear all EIE lines – clear all pending Class III interrupts, disable further generation of Class III Priority 2,3, and 4 interrupts – clear intercomputer time–out function – clear serial monitor and suppress flags.
		1	–	Illegal
*		2	–	Illegal
		3	–	Illegal
		4*	–	Enable all channels' external interrupt data storage; set all EIE lines.
		5*	–	Disable all channels' external interrupt data storage; clear all EIE lines.
			0	Enable all channels' external interrupt monitors to allow Class III, priority 2, 3, and 4 interrupt generation. If external interrupt data were stored while monitors were disabled, generate the Class III, priority 2 interrupt.
		6*	1–17	Enable external interrupt monitors for all channels with priority lower than the channel defined by a to allow Class III, priority 2, 3, and 4 interrupt

TABLE VIII. CHANNEL CONTROL INSTRUCTION m–DESIGNATOR AND a–DESIGNATOR
(continued)

ULTRA Symbol	m– Value	a– Value	Operation
			generation. If external interrupt data was stored while monitors were disabled, generate the Class III, priority 2 interrupt.
	7*	0	Disable priority 2, 3, and 4 interrupt generation for all channels.
	7*	1–17	Disable priority 2, 3, and 4 interrupt generation by channels with priority lower than the channel defined by a.
CCR	10*	0–17	Master Clear the channel defined by a (see m = 0 above)
	11*	0–17	Master Clear input on channel defined by a. Master clear input on channel a. Clear the buffers and chain associated with the input half of channel a in the 16–channel group when executed out of the command cell, or the channel associated with the active chain when executed out of an I/O chain. The Class III Interrupt Enable state, the EIE state, and the state of any pending Class III interrupts for the specified channels shall remain unaffected by execution of this instruction. If the channel is a synchronous serial channel, sync shall be lost and the "search for sync" function shall be disabled by the execution of this instruction.
	12	0–17	Master Clear output on channel defined by a. Master clear output on channel a. Clear buffers and chain associated with the output half of channel a in the 16–channel group when executed out of the command cell, or the channel associated with the active chain when executed out of an I/O chain. If the specified channel is an intercomputer channel (IC), the execution of this instruction shall free the Output Data Register and clear any potential Class III IC timeout interrupt. The state of all other pending Class III interrupts, the Class III Interrupt Enable state, and the EIE state for the specified channel shall remain unaffected by execution of this instruction.
	13	–	Illegal
**	14*	0–17	Enable the channel external interrupt data storage; set EIE line or send command.
	15*	0–17	Disable the channel external interrupt data storage; clear EIE line.
	16*	0–17	Enable the channel Class III priority 2, 3, and 4 interrupt generation (see m = 0 above).
	17*	0–17	Disable the channel Class III priority 2, 3, and 4 interrupt generation.

*Operations affecting all channels collectively (command or chaining)
**Operations affecting only the channel specified by the a–designator (command) or the associated channel (chaining)
For all I/O instructions, RK and RX formats, Y = y (indexing and indirect addressing cannot be used).

TABLE IX. INITIATE TRANSFER INSTRUCTION a–DESIGNATOR FUNCTIONS

a–Value	Type of Transfer	Notes
0	Input data (input BCW and BAP loaded)	1
1	Output data (output BCW and BAP loaded)	2
2	External function (results in a no operation for VACALES, RS–232–C or MIL–STD–188C serial I/O)	3
3	External function with force (results in a no operation for VACALES, RS–232–C, or MIL–STD–188C serial I/O)	4
4 through 17	(Interpreted as though a = 0–3)	

NOTES:

1 When executed from an output chain, the output chain is terminated. The input chain will be reactivated after the buffer is terminated using the old output chain pointer.

2 When executed from an input chain, the input chain is terminated. The output chain will be reactivated after the buffer is terminated, using the old input chain pointer.

3 When executed from an input chain, the input BCW and BAP is loaded; otherwise the output BCW and BAP is loaded.

4 When executed from an input chain, the input BCW and BAP is loaded; otherwise the output BCW and BAP is loaded.

TABLE X. I/O CONTROL MEMORY ADDRESS SELECTION

a–Value	Word	m–Value	MIL–STD–1397 A, B, C, PIC	MIL–STD–1397 Type D/E Serial	RS–232–C MIL–STD–188C, VACALES	NATO–STD–4156
Channel designator for command instructions, not used for chaining instructions.	0	0	Input BCW	Input BCW	Input BCW	Input BCW
	1	1	Input BAP	Input BAP	Input BAP	Input BAP
	2	2	Input CAP	Input CAP	Input CAP	Input CAP
	3	3	Reserved*	Reserved*	Reserved*	Reserved*
	4	4	Output BCW	Output BCW	Output BCW	Output BCW
	5	5	Output BAP	Output BAP	Output BAP	Output BAP
	6	6	Output CAP	Output CAP	Output CAP	Output CAP
	7	7	Reserved*	Reserved*	Reserved*	Reserved*
	10	10	Reserved*	Reserved*	Monitor Word	Reserved*
	11	11	Reserved*	Reserved*	Suppress Word	Reserved
	12	12	Operating Mode (Figure A)	Operating Mode (Figure B)	Serial Mode (Figure C)	Operating Mode (Figure D)
	13–17	13–17	Reserved*	Reserved*	Reserved*	Reserved*

* Reserved addresses may be used for future enhancements.

MODE REGISTER						MODE OF OPERATION
15–5	4	3	2	1	0	
	0	0	0	0	0	COMPUTER TO PERIPHERAL 16 BIT
	0	0	0	0	1	
	0	0	0	1	0	
	0	0	0	1	1	
	0	0	1	0	0	
	0	0	1	0	1	
	0	0	1	1	0	
	0	0	1	1	1	
	0	1	0	0	0	COMPUTER TO PERIPHERAL – 16 BIT
	0	1	0	0	1	COMPUTER TO COMPUTER – 16 BIT
	0	1	0	1	0	UNDEFINED
	0	1	0	1	1	TEST MODE, 16 BIT
	0	1	1	0	0	COMPUTER TO PERIPHERAL – 32 BIT
	0	1	1	0	1	COMPUTER TO COMPUTER – 32 BIT
	0	1	1	1	0	EXTERNALLY SPECIFIED ADDRESSING
	0	1	1	1	1	TEST MODE, 32 BIT
	1	1	0	0	0	PERIPHERAL INPUT CHANNEL (PIC) (16 – BIT)
	1	1	1	0	0	PERIPHERAL INPUT CHANNEL (PIC) (32 – BIT)
RESERVED						

Figure A. Parallel Operating Modes

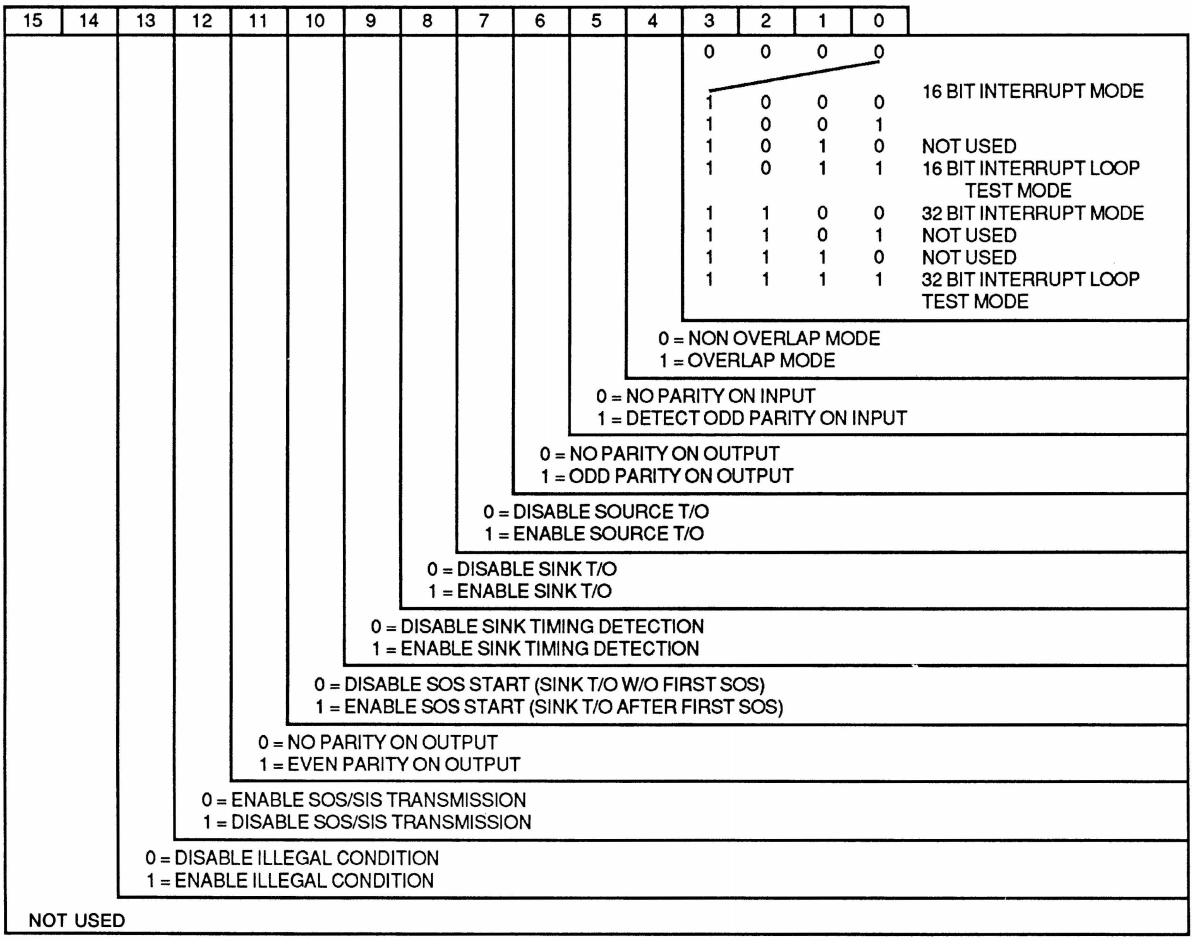
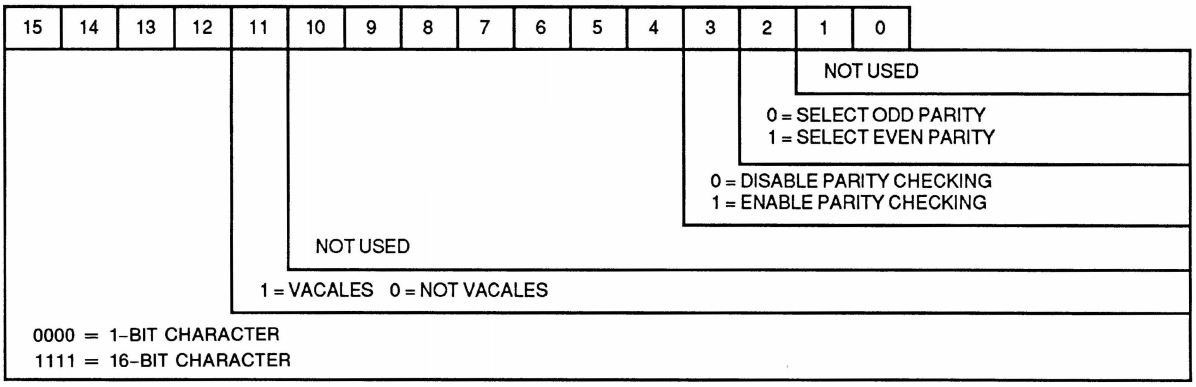


Figure B. MIL-STD-1397 Type D/E Operating Modes

VACALES INTERFACE



MIL–STD–188C AND EIA–STD–RS–232–C INTERFACES

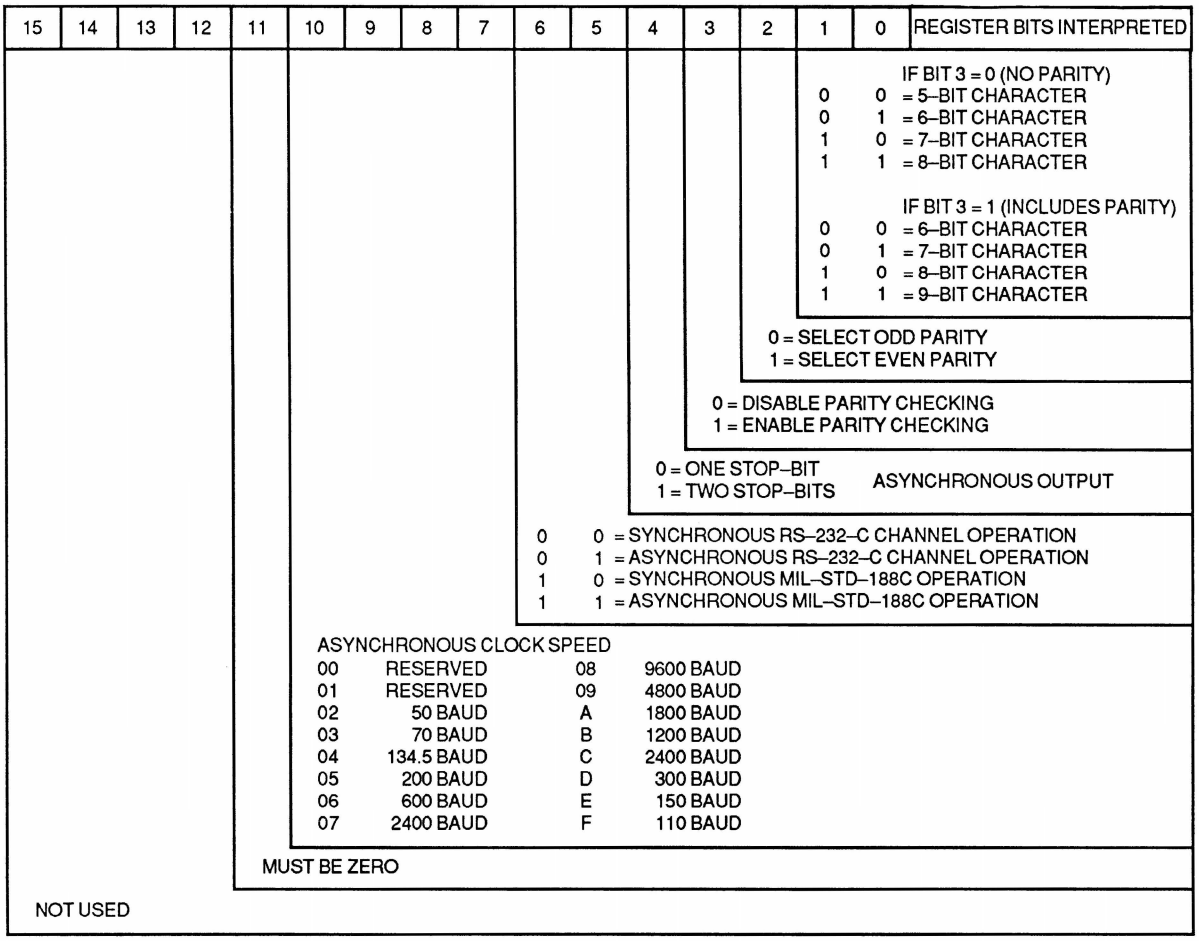


Figure C. Serial Operating Mode Information Interpretation

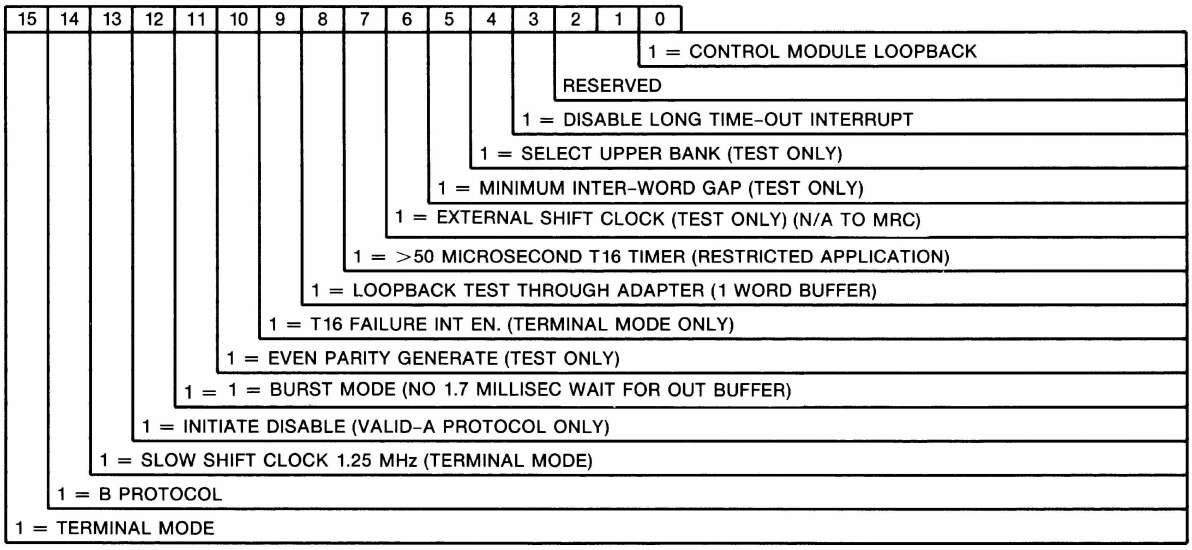


Figure D. NATO–STD–4156 Serial Mode Information Interpretation

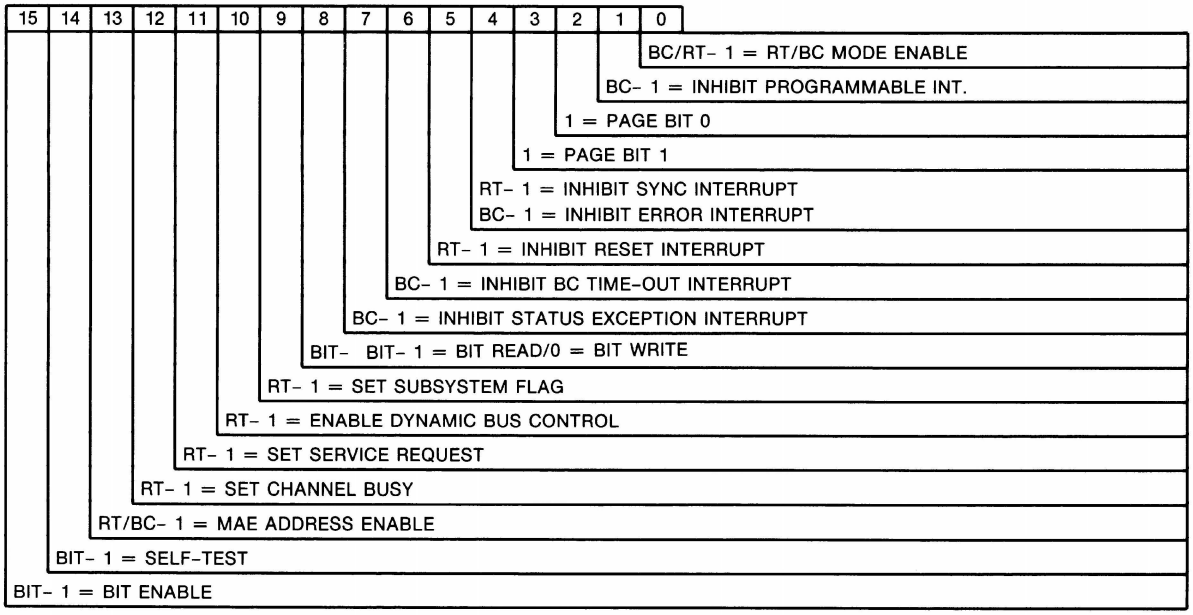


Figure E. MIL–STD–1553B Configuration Register Interpretation

OPERATION CODE 73 0 OCTAL EC HEX

– RR Format – HALT/INTERRUPT (Chaining)
Perform the operation specified as follows:

HCR If a = 0, halt the chaining action on the associated channel.

IPR If a = 1, generate the chain interrupt to the central processor. (Class III, Priority 3 when executed from an output chain; Priority 4 when executed from input chain).
The m–designator and a–values $2–17_8$ are not interpreted.

OPERATION CODE 73 3 OCTAL EF HEX

– RX Format – SET/CLEAR FLAG (Chaining)

ZF If a = 0, Clear the most significant two bits (flag) in memory location Y.

SF If a = 1, Set the most significant two bits (flag) in memory location Y.

TF If a = 2, Test and set flag (bits 15 and 14) and set condition bit in the channel status word if the flag was clear, else clear condition bit.

ZB If a = 4, Zero bit m at Y.

SB If a = 5, Set bit m at Y.

CB If a = 7, Compare bit m at Y to 0. Set condition code in the channel status word if the bit was 0, else clear condition bit.

a = $3,6,10–17_8$ are illegal.

OPERATION CODE 74 2 OCTAL F2 HEX

– RK Format – CONDITIONAL JUMP (Chaining)
If the condition specified for the a–value in Table XI is satisfied, load the chain address pointer location with Y; otherwise execute the next sequential instruction in the chain.

OPERATION CODE 75 0 OCTAL F4 HEX

SFSC RR Format – SEARCH FOR SYNC/SET MONITOR/SET SUPPRESS (Chaining)

Condition the MIL–STD–188C, RS–232–C, and VACALES serial channel for the next activated input buffer to perform the operation specified by bits 3, 2, 1 and 0 of the m–designator in accordance with Table XII and as follows:

2^0–bit set (Set synchronous serial active)

TABLE XI. a–DESIGNATOR JUMP CONDITIONS

ULTRA Symbol	a–Value	Jump Condition
SJC	0	Unconditional jump.
	1	Serial jump if suppress flag not set (for MIL–STD–188C and RS–232–C, flag is cleared during next character time; for VACALES, flag is cleared when next character is transferred to memory). (No jump for MIL–STD–1397.)
SJMC	2	Serial jump is monitor flag set (for MIL–STD–188C and RS–232–C, flag is cleared during next character time; for VACALES, flag is cleared when next character is transferred to memory). (No jump for MIL–STD–1397.)
	4	Jump if condition bit (bit 15) in the I/O status word is set.
	10	Jump if input buffer is active.
	11	Jump if output buffer is active.
	12	Jump if external function buffer is active. (No jump for MIL–STD–188C, RS–232–C, or VACALES.)
	$3,5–7, 13–17_8$	Illegal.

TABLE XII. OPERATION CODE 75, 0 OCTAL F4 HEX, FORMAT RR, SET SUPPRESS, SET MONITOR AND SEARCH FOR SYNC m–DESIGNATOR FUNCTIONS

m–Value	Interface	Function
0000	Sync	Disable sync, disable monitor, disable set suppress, and enable next chain instruction.
0000	Async	Disable set monitor, disable set suppress, and enable next chain instruction.
0001	Sync	Hold sync active, disable set monitor, disable set suppress, and enable next chain instruction.
0010	Async	Enable set suppress, disable set monitor, and enable next chain instruction.
0011	Sync	Hold sync active, enable set suppress, disable set monitor, and enable next chain instruction.
0100	Async	Enable set monitor, disable set suppress, and enable next chain instruction.
0101	Sync	Hold sync active, enable set monitor, disable set suppress, and enable next chain instruction.
0110	Async	Enable set monitor, enable set suppress, enable next chain instruction.
0111	Sync	Hold sync active, enable set monitor, enable set suppress, enable next chain instruction.
1000	–	No operation and enable next chain instruction.
1001	Sync,	Enable search for sync, disable set suppress after sync is established, and disable

TABLE XII. OPERATION CODE 75, 0 OCTAL F4 HEX, FORMAT RR, SET SUPPRESS, SET MONITOR AND SEARCH FOR SYNC m–DESIGNATOR FUNCTIONS (continued)

m–Value	Interface	Function
	MIL–STD–188C, RS–232–C	set monitor; disable chaining until function is complete.
1001	Sync, VACALES	Enable search for sync bit–by–bit, disable set suppress after sync is established, and disable set monitor; disable chaining until function is complete.
1010	–	No operation and enable next chain instruction.
1011	Sync, MIL–STD–188C, RS–232–C	Enable search for sync, enable set suppress, and disable set monitor; disable chaining until function is complete.
1011	Sync, VACALES	Enable search for sync bit–by–bit, enable set suppress, and disable set monitor; disable chaining until function is complete.
1100	–	No operation and enable the next chain instruction.
1101	Sync, MIL–STD–188C, RS–232–C	Enable search for sync, disable set suppress after sync is established, and enable set monitor; disable chaining until function is complete.
1101	Sync, VACALES	Enable search for sync character compare, set suppress flag if the next character compares, disable set monitor and disable set suppress on subsequent characters; enable next chain instruction after the character compare.
1110	–	No operation and enable the next chain instruction.
1111	Sync, MIL–STD–188C, RS–232–C	Enable search for sync, enable set monitor and enable set suppress; disable chaining until function is complete.
1111	Sync, VACALES	Enable search for sync character compare, set suppress flag if the next character compares, enable set monitor, and enable set suppress on subsequent characters; enable next chain instruction after the character compare.

Enable synchronization on the serial channel. If 2^0–bit is clear, disable synchronization.

2^1–bit set (Set Suppress on synchronous and asynchronous channel)

Compare each character of the input stream to the character in the suppress register and store it in memory. If equality is not detected, then suppress that character and continue comparing.

2^2–bit set (Set Monitor on synchronous or asynchronous channel)

Compare each character of the input stream to the character in the monitor register and store it in memory. When equality is detected set the monitor designator, store the character and enable in memory. When equality is detected set the monitor designator, and enable the channel chain (terminate the buffer). If 2^2–bit is cleared, disable monitor character compare.

2^3–bit and 2^0–bit set (Search for Sync on synchronous channels)

At each bit–time of the incoming data stream, compare the value of a character length word to the character in the suppress register. When equality is detected, compare the next character to the suppress register and enable the channel chain; if equality is detected again set the suppress designator. If 2^3–bit is clear, disable the search for sync function.

2^3–bit set 2^2–bit clear and 2^0–bit set (Search for Sync, Bit by Bit, on VACALES Channels)

At each bit time of incoming data stream, compare the value of a character length to the contents of the suppress register. When a match occurs start assembling the next input data and enable the next chain instruction. If 2^3–bit is cleared, disable search for sync function.

2^3–bit, 2^2–bit set and 2^0–bit set (Search for Sync, Character, on VACALES Channels)

Compare each character of the input stream to the contents of the suppress register. When a compare results in a match, set the suppress flag and enable the next chain instruction. If compare does not result in a match, the suppress flag is not set and the next chain instruction is enabled.

OPERATION CODE 76 0 OCTAL F8 HEX

–

RR Format – SET/CLEAR DISCRETE (Command and chaining)
Set or clear the discrete associated with the MIL–STD–188C, VACALES or EIA–STD–RS232–C serial interface according to the m–value in Table XIII.

SICR

Command instruction – "a" specifies the channel.

CSIR

Chaining instruction – "a" is not used.

OPERATION CODE 76 3 OCTAL FB HEX

–

RX Format – STORE STATUS (Command and chaining)
Store the channel status word specified in Table XIV for the m–designator at memory address Y.

SST

Command Instruction – "a" specifies the channel.

CSST

Chaining Instruction – "a" is not used.

OPERATION CODE 77 3 OCTAL FF HEX

IIC

RX Format – Built In Test (BIT).
Execute the IOC BIT subtest specified by (Y).

TABLE XIII. DISCRETE SET/CLEAR FUNCTIONS

		MIL-STD-188C/VACALES		EIA-STD-RS-232-C	
m-Value	Function	Line Designator MIL-STD-188C	Line Designator VACALES	Discrete	Line Designator
0	Enable Loop Test (internal)	*	*	*	–
1	Disable Loop Test (internal)	*	*	*	–
2	Illegal	Not Used	Not Used	Not Used	–
3	Illegal	Not Used	Not Used	Not Used	–
4	On	J	J	J (non-Std.)	J
5	Off	J	J	J (non-Std.)	J
6	On	H	Transmitter Prep.	Dsble. Ring Indicator Intrpt.*	–
7	Off	H	Transmitter Prep.	Enble. Ring Indicator Intrpt.*	–
10	Off	G	G	Request to Send	CA
11	On	G	G	Request to Send	CA
12	Off	F	F	New Sync	CH
13	On	F	F	New Sync	CH
14	Off	D	D	Data Terminal Ready	CD
15	On	D	D	Data Terminal Ready	CD
16	Off	A	Loop Back (external)	Modem Loop Test (external)	–
17	On	A	Loop Back (external)	Modem Loop Test (external)	–

* Internal function – no interface line affected.

TABLE XIV. STORE STATUS

m-Value	MIL-STD-1397 A, B, C, PIC	MIL-STD-1397 Type D/E Serial	RS-232-C	MIL-STD-188C	VACALES	NATO-STD-4156
0	*	Status Word 1 (Table XVI)	Table XV	Table XV	Table XV	*
1						
2		Status Word 2 (Table XVII)				
3						
4	*	*	*	*	*	*
5						
6						
7						
10	Status Word 0 per Figure 22; MIL-STD-1397 Type D/E serial definitions per Table XVIII.					
11						
12						
13						
14	*	*	*	*	*	*
15						
16						
17						

* Zeros shall be stored in memory.

TABLE XV. STATUS WORD INTERPRETATION

Word Bit	MIL–STD–188C Function	EIA–STD–RS–232–C Function	MIL–STD–188C and EIA–STD–RS–232–C Description
2^0	Parity Error	Parity Error	Serial I/O detects a parity error on an input word.
2^1	Overrun	Overrun	Serial I/O does not store an input word before another is transmitted.
2^2	Break	Break	Serial I/O does not detect a STOP–bit. (Used in asynchronous mode only.)
2^3	E Active	Clear to Send	Line is set "active" by a peripheral equipment.

Word Bit	VACALES Function	VACALES Description
2^1	Overrun	The serial I/O did not transfer to memory before another I/O word was received.
2^2	Parity Error	The serial I/O detected a parity error on an input data word.
2^3	Sync Error	The inbound discrete control line, Sync Error, was set by a peripheral equipment.

TABLE XVI. MIL-STD-1397 TYPE D/E STATUS REGISTER 1 DEFINITIONS

BIT	DEFINITION
0	SOURCE TIMEOUT ERROR—When set indicates that a source timeout error has been detected. A source timeout is detected if output active is set and no output word is transferred within 150–200 ms (9 to 12 ms for Type D) or when an external function active is set and no EF word is transferred within 150–200 ms (9 to 12 ms for Type D). The bit is cleared when the error register is read and by a master clear.
1	SINK TIMEOUT ERROR—When set, indicates that a sink timeout error was detected. This error is detected if EI active or input active is set and no source response occurs, either a control frame or information frame, for 1.5 ms. The bit is cleared when the error register is read and by a master clear.
2	PARITY ERROR—When set, indicates that a parity error was detected. The bit is cleared when the error register is read and by a master clear.
3	SINK TIMING ERROR—When set, indicates that a timing error was detected. A timing error occurs when the information frame contains too many or too few bits. The bit is cleared when the error register is read and by a master clear.
4	ILLEGAL CONDITION ERROR—When set, indicates that an illegal condition error was detected. An illegal condition is detected when the sink is not ready for input data but an input word is received or when the sink is not ready for external interrupts but an interrupt word is received. The bit is cleared when the error register is read and by a master clear.
5	WORD IDENTIFIER—Identifies the word type ("1" = EI, "0" = INPUT) of the most recent information frame received. When an error is detected (parity, timing, illegal condition) this bit identifies the type of transfer which caused the error.
11–0	BWC—Displays the input buffer word count (BWC) register contained on the MIL-STD-1397 Type D/E IOCAs. This word counter is necessary to ensure proper input buffer termination in the overlap mode.
12, 13	Not used.
14	INPUT TM0—Displays bit 0 of the input transfer mode field.
15	INPUT TM1—Displays bit 1 of the input transfer mode field.

TABLE XVII. MIL–STD–1397 TYPE D/E STATUS REGISTER 2 DEFINITIONS

BIT	DEFINITION
11–0	BCW – Displays the input buffer word count (BWC) register contained on the MIL–STD–1397 Type D/E IOCAs. This word counter is necessary to insure proper input buffer termination in the overlap mode.
12, 13	Not used.
14	INPUT TMO – Displays bit 0 of the input transfer mode field.
15	INPUT TM1 – Displays bit 1 of the input transfer mode field.

TABLE XVIII. MIL–STD–1397 TYPE D/E STATUS REGISTER 0 DEFINITIONS

BIT	DEFINITION
3–0	CHANNEL NUMBER – Indicates the logical channel number of the interface adapter.
7–4	CHANNEL TYPE – Identifies the type of interface installed at the respective channel location.
8	INPUT CHAIN INTERRUPT PENDING – Indicates that an IOC interrupt processor instruction (CODE IPR) has been executed on this channel from an input chain and results in a Class III interrupt being generated. Bit 8 will remain set pending servicing of the Class III interrupt by the CP.
9	OUTPUT CHAIN INTERRUPT PENDING – Indicates that an interrupt processor instruction (CODE IPR) has been executed on this channel from an output chain and results in a Class III interrupt being generated. Bit 9 will remain set pending servicing of the Class III interrupt by the CP.
10	EXTERNAL INTERRUPT – Indicates that an external interrupt word has been received on the channel and results in a Class III interrupt being generated. Bit 10 will remain set pending servicing of the Class III interrupt by the CP.
11	ERROR/TIME–OUT – Indicates that a SIF error (parity, timing, illegal condition, source time–out, or sink time–out) has been detected on the channel and results in a Class III interrupt being generated. Bit 11 will remain set pending servicing of the Class III interrupt by the CP.
12	CHANNEL INPUT ACTIVE – Indicates that either an input or external function transfer is enabled on the input channel.
13	CHANNEL OUTPUT ACTIVE – Indicates that either an output or external function transfer is enabled on the output channel.
14	EXTERNAL INTERRUPT – Indicates that an external interrupt transfer is enabled on the channel.
15	TEST CONDITION FOR CONDITIONAL JUMPS – Indicates the state of the condition bit for the conditional jump instruction (CODE SJMC 4,y). Bit 15 sets and clears via IOC instructions TFy and CBy,m.

APPENDIX B
AN/UYK–44 CP MACROINSTRUCTION EXECUTION TIMES

The CP macroinstruction execution times contained in this appendix were measured on an AN/UYK–44(V) Microprocessor Development System (MDS) and the AN/UYK–44(V)/U Data Processing Set (DPS). Table B–1 contains the CP macroinstruction times measured. Memories used were 900 ns. cycle time core, 350 ns. cycle time semiconductor and 250 ns. (simulated using a VMS/64 Variable Speed Memory) semiconductor.

The timing data presented herein is valid with only the CP functioning. If there are other activities which require memory access, such as I/O or customer–designed devices interfacing to the common bus, the apparent execution time for macroinstructions will increase due to bus contention. Therefore, no IOC macroinstruction execution times are contained herein.

The definition of macroinstruction execution time is the time measured from the beginning of one macroinstruction to the beginning of the next macroinstruction. Macroinstruction execution time is not constant for all CP instructions. The time from the beginning of the macroinstruction to the beginning of the next macroinstruction can vary. This appendix does not present a calculated average of these times, but provides actual minimum and maximum times.

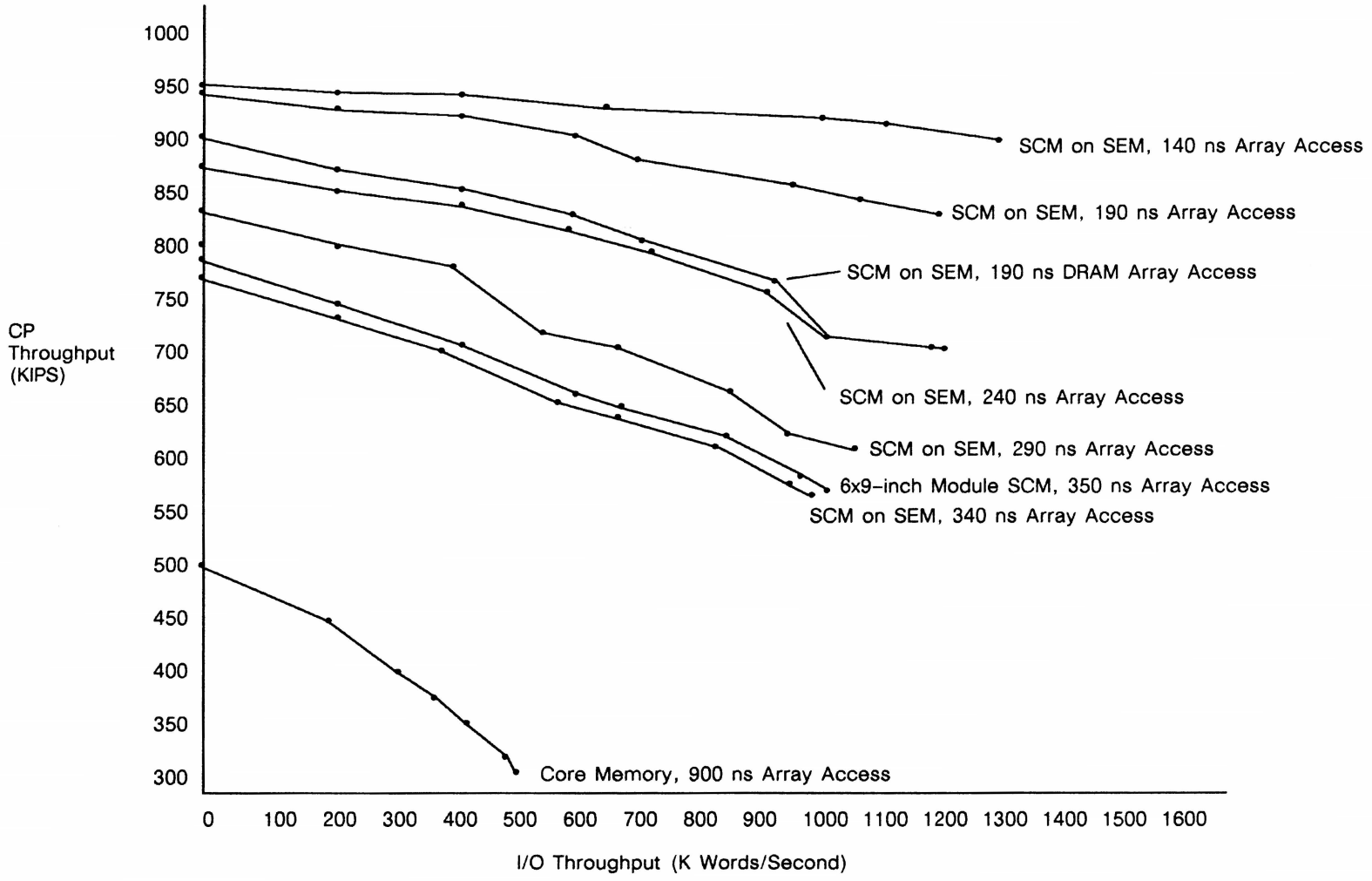
Minimum time is the time required to execute the macroinstruction only; no fetch time is included. For those macroinstructions which do not require another memory reference, macroinstruction execution time is the sum of the execution times of the microinstructions required to execute the macroinstruction. Execution time for macroinstructions which require an operand memory reference may be affected by having to wait for memory. In this case minimum time includes the time for memory to respond.

Maximum time includes fetch time for the macroinstruction following the macroinstruction being measured. If additional memory references are needed, extra time is required for the fetch operations. The maximum time also varies with instruction execution from odd or even memory addresses. All times are given for the worst case.

Measurement of macroinstruction execution time was done at the microinstruction level. The time was measured from the start of the first microinstruction of the macroinstruction to the first microinstruction of the next macroinstruction. Measurements made with the instruction stack (ISTACK) full, produced the minimum times. The maximum times were obtained from simulating an ISTACK empty condition. This condition was created by making the instruction to be timed the first instruction in a loop. This makes sure that an instruction fetch is required before the instruction following (the instruction to be timed) can be executed.

The minimum program execution time may be determined by adding the minimum times, and adding the maximum times will determine the maximum program execution time. These limits are the extremes, actual program execution times will fall somewhere between. Precise timing will most likely only be found by measuring the actual execution time of the program segment.

Figure A defines CP macroinstruction throughput with I/O activity.



NOTE: These throughputs were measured using NOSC I/O thru 4242, based on NOSC 424-LP-6222 (revised) instruction mix.

Figure A. MRP/MRC CP Throughput Versus I/O Throughput

Table B–1. CP Instruction Times

Code (Hex) OP	a	m	Mnemonic	Execution Time (Microseconds) 250 ns Cycle Semiconductor Min.	Max.	350 ns Cycle Semiconductor Min.	Max.	900 ns Cycle Core Min.	Max.	
00	0	1	ICP	*						
00	0	2	SRM	*						
00	0	3	LRM	*						
00	0	4	IMP	*						
00	0	5	IDS	*						
02			SPT	*						
03			BL	1.0	1.0	1.25	1.45	1.3	3.45	
04			LR	0.5	0.5	0.5	0.5	0.5	0.9	
05			LI	1.0	1.0	1.05	1.2	1.05	3.45	
06			LK	0.5	0.5	0.5	0.7	0.5	1.75	
07			L	1.0	1.1	1.25	1.45	1.3	3.45	
08		0	PR	0.75	0.75	0.75	0.75	0.75	0.9	
08		1	NR	0.75	0.75	0.75	0.75	0.75	0.9	
08		2	RR	0.95	0.95	0.95	0.95	0.95	0.95	
08		4	TCR	0.25	0.25	0.25	0.45	0.25	0.9	
08		5	TCRD	0.7	0.7	0.7	0.7	0.7	0.9	
08		6	OCR	0.25	0.25	0.25	0.45	0.25	0.9	
08		8	IROR	0.25	0.25	0.25	0.45	0.25	0.9	
08		9	DROR	0.25	0.25	0.25	0.45	0.25	0.9	
08		A	IRTR	0.25	0.25	0.25	0.45	0.25	0.9	
08		B	DRTR	0.25	0.25	0.25	0.45	0.25	0.9	
09			LDI	1.2	1.2	1.45	1.6	2.1	4.3	
0B			LD	1.25	1.35	1.5	1.7	2.15	4.3	
0C		0	ER	4.85		6.3	6.55	8.7	11.2	
0C		1	SSOR	0.25	0.25	0.25	0.45	0.25	0.9	
0C		2	SSTR	0.25	0.25	0.25	0.45	0.25	0.9	
0C		3	SCR	0.25	0.25	0.25	0.45	0.25	0.9	
0C		4	LPR	1.45	1.45	1.25	1.55	1.3	5.2	
0C		5	LSOR	1.8	1.8	2.0	2.0	2.05	5.2	
0C		6	LSTR	1.25	1.25	1.25	1.65	1.3	5.2	
0C		7	LCR	0.5	0.5	0.5	0.5	0.5	0.9	
0C		8	ECR	0.45	0.45	0.45	0.45	0.45	0.9	
0C		9	DCR	0.45	0.45	0.45	0.45	0.45	0.9	
0C		A	LEM	0.5	0.5	0.5	0.5	0.5	0.9	
0C		B	DM	0.45	0.45	0.45	0.45	0.45	0.9	
0C		C	LCRD	0.75	0.75	0.75	0.75	0.75	0.9	
0C		D	SCRD	0.5	0.5	0.5	0.5	0.5	0.9	
0C		E	ECIR	0.45	0.45	0.45	0.45	0.45	0.9	
0C		F	DCIR	0.45	0.45	0.45	0.45	0.45	0.9	
0F			LM	1.05	1.25	1.25	11.3	1.3	18.1	Note 1

Table B–1. CP Instruction Times (continued)

Code (Hex) OP	a	m	Mnemonic	Execution Time (Microseconds) 250 ns Cycle Semiconductor Min.	250 ns Cycle Semiconductor Max.	350 ns Cycle Semiconductor Min.	350 ns Cycle Semiconductor Max.	900 ns Cycle Core Min.	900 ns Cycle Core Max.	
10		0	SQR	7.9	7.9	7.9	7.9	7.9	7.9	
10		1	RVR	0.7	0.7	0.7	0.7	0.7	0.9	
10		2	CNT	1.6	1.6	1.6	1.6	1.6	1.6	
10		3	SFR	1.75	1.75	1.75	1.75	1.75	1.75	
10		4	SMC	0.25	0.25	0.25	0.45	0.25	0.9	
10		5	SQRT	19.5	19.5	19.5	19.5	19.5	19.5	
10		6	LCEP	0.5	0.5	0.5	0.5	0.5	0.9	
10		8	IS	*						
10		9	IB	43.85		44.2	44.6	44.3	48.15	
12			QPT	2.45	2.65	2.75	4.05	3.2	7.05	
13			BLX	1.0	1.1	1.25	1.5	1.3	3.45	
14			SBR	0.45	0.45	0.45	0.45	0.45	0.9	
15			LXI	1.2	1.2	1.2	1.2	1.2	3.45	
16			QPB	3.15	3.15	3.35	4.1	4.1	6.9	
17			LX	1.25	1.25	1.25	1.45	1.25	3.05	
18			ZBR	0.45	0.45	0.45	0.45	0.45	0.9	
19			LDXI	1.7	1.7	1.7	1.7	2.1	4.3	
1A			SGT	3.6	3.6	3.85	4.25	6.45	7.15	
1B			LDX	1.75	1.75	1.75	1.95	2.15	4.3	
1C			CBR	0.45	0.45	0.45	0.45	0.45	0.9	
1D			LPI	2.95	2.95	3.4	3.95	4.35	5.9	
1E			QGT	3.6	3.6	4.1	4.5	6.7	8.0	
1F			LP	2.95	2.95	3.45	3.95	4.35	6.5	
20			LRSR	0.7	0.7	0.7	0.95	0.7	0.9	
22			LRS	0.75	0.75	0.75	0.95	0.75	1.8	
23			BS	1.15	1.35	1.25	1.65	1.3	5.2	
24			ARSR	0.9	0.9	0.9	0.9	0.9	0.9	
25			SI	1.15	1.25	1.25	1.7	1.3	3.45	
26			ARS	0.95	0.95	0.95	1.15	0.95	1.8	
27			S	1.15	1.3	1.25	1.65	1.3	5.2	
28			LRDR	1.2	1.2	1.2	1.2	1.2	1.2	
29			SDI	1.65	1.73	1.95	2.35	2.15	4.3	
2A			LRD	1.25	1.25	1.25	1.45	1.25	1.9	
2B			SD	1.8	1.82	1.95	2.35	2.15	6.05	
2C			ARDR	1.2	1.2	1.2	1.2	1.2	1.2	
2E			ARD	1.25	1.25	1.25	1.45	1.25	1.9	
2F			SM	1.35	1.35	1.5	12.4	1.55	18.1	Note 1

Table B–1. CP Instruction Times (continued)

Code (Hex) OP a m	Mnemonic	Execution Time (Microseconds)					
		250 ns Cycle Semiconductor		350 ns Cycle Semiconductor		900 ns Cycle Core	
		Min.	Max.	Min.	Max.	Min.	Max.
30	ALSR	2.9	2.9	2.9	2.9	2.9	2.9
32	ALS	2.9	2.9	2.9	3.1	2.9	3.5
33	BSX	1.1	1.3	1.25	1.65	1.3	5.2
34	CLSR	0.7	0.7	0.7	0.7	0.7	0.9
35	SXI	1.0	1.0	1.25	1.65	1.3	3.45
36	CLS	0.75	0.75	0.75	0.95	0.75	1.8
37	SX	1.2	1.3	1.25	1.65	1.3	5.2
38	ALDR	3.3	3.3	3.3	3.3	3.3	3.3
39	SDXI	1.85	1.85	2.0	2.35	2.15	4.3
3A	ALD	3.3	3.3	3.3	3.5	3.3	3.9
3B	SDX	1.85	1.85	2.0	2.35	2.15	6.0
3C	CLDR	1.2	1.2	1.2	1.2	1.2	1.2
3D	SZI	1.15	1.25	1.25	1.65	1.3	3.45
3E	CLD	1.25	1.25	1.25	1.45	1.3	1.85
3F	SZ	1.15	1.35	1.25	1.65	1.3	5.2
40	SUR	0.5	0.5	0.5	0.5	0.5	0.9
41	SUI	1.0	1.0	1.2	1.25	1.25	3.45
42	SUK	0.5	0.5	0.5	0.7	0.5	1.75
43	SU	1.1	1.1	1.25	1.45	1.3	3.45
44	SUDR	1.0	1.0	1.0	1.0	1.0	1.0
45	SUDI	1.2	1.25	1.45	1.45	2.1	4.35
47	SUD	1.25	1.35	1.5	1.7	2.15	4.35
48	AR	0.5	0.5	0.5	0.5	0.5	0.9
49	AI	0.95	1.0	1.2	1.2	1.25	3.45
4A	AK	0.5	0.5	0.5	0.7	0.5	1.7
4B	A	1.1	1.1	1.25	1.45	1.3	3.45
4C	ADR	1.0	1.0	1.0	1.0	1.0	1.0
4D	ADI	1.25	1.25	1.45	1.45	2.1	4.35
4F	AD	1.25	1.25	1.5	1.7	2.15	4.35
50	CR	0.5	0.5	0.5	0.5	0.5	0.9
51	CI	0.95	1.0	1.2	1.2	1.25	3.45
52	CK	0.5	0.5	0.5	0.7	0.5	1.75
53	C	1.0	1.0	1.25	1.45	1.3	3.45
54	CDR	1.0	1.0	1.0	1.0	1.0	1.0
55	CDI	1.25	1.25	1.45	1.45	2.1	4.3
57	CD	1.25	1.25	1.5	1.7	2.15	4.35

Table B–1. CP Instruction Times (continued)

Code (Hex) OP a m	Mnemonic	Execution Time (Microseconds)					
		250 ns Cycle Semiconductor		350 ns Cycle Serhiconductor		900 ns Cycle Core	
		Min.	Max.	Min.	Max.	Min.	Max.
58	MR	1.45	1.45	1.45	1.45	1.45	1.45
59	MI	1.7	1.7	1.9	1.9	1.95	4.15
5A	MK	1.45	1.45	1.45	1.65	1.45	2.1
5B	M	1.7	1.8	1.95	2.15	2.0	4.15
5C	DR	**8.3		**8.3		**8.45	
5D	DI	**8.75		**8.75		**8.85	
5E	DK	**8.3		**8.3		**8.45	
5F	D	**8.75		**8.9		**8.85	
60	ANDR	0.5	0.5	0.5	0.5	0.5	0.9
61	ANDI	1.0	1.0	1.2	1.25	1.25	3.45
62	ANDK	0.5	0.5	0.5	0.7	0.5	1.75
63	AND	1.1	1.1	1.25	1.45	1.3	3.45
64	ORR	0.5	0.5	0.5	0.5	0.5	0.9
65	ORI	1.0	1.0	1.2	1.2	1.25	3.45
66	ORK	0.5	0.5	0.5	0.7	0.5	1.75
67	OR	1.0	1.0	1.25	1.45	1.3	3.45
68	XORR	0.5	0.5	0.5	0.7	0.5	0.9
69	XORI	1.0	1.0	1.2	1.2	1.25	3.45
6A	XORK	0.5	0.5	0.5	0.7	0.5	1.75
6B	XOR	1.1	1.1	1.25	1.45	1.3	3.45
6C	MSR	1.25	1.25	1.25	1.25	1.25	1.25
6D	MSI	1.3	1.3	1.5	1.5	1.55	3.7
6E	MSK	1.25	1.25	1.25	1.45	1.25	1.85
6F	MS	1.35	1.35	1.5	1.7	1.55	3.7
70	CMR	1.25	1.25	1.25	1.25	1.25	1.25
71	CMI	1.3	1.3	1.5	1.5	1.55	3.7
72	CMK	1.25	1.25	1.25	1.45	1.25	1.85
73	CM	1.35	1.35	1.5	1.7	1.55	3.75
74	IOCR	**9.75		**9.25		**9.0	
75	BFI	1.3	1.4	1.45	1.9	1.5	3.3
76	REX	1.75	2.0	1.75	2.15	1.8	5.65
77	BF	1.45	1.45	1.45	1.9	1.55	5.4
78	CLR	1.0	1.0	1.0	1.0	1.0	1.0
79	CLI	1.45	1.45	1.45	1.45	1.45	3.7
7A	CLK	1.0	1.0	1.0	1.2	1.0	1.7
7B	CL	1.5	1.5	1.5	1.7	1.5	3.7

Table B–1. CP Instruction Times (continued)

Code (Hex) OP	a	m	Mnemonic	Execution Time (Microseconds) 250 ns Cycle Semiconductor Min.	Max.	350 ns Cycle Semiconductor Min.	Max.	900 ns Cycle Core Min.	Max.
7C		0	VF	**25.05		**25.05		**25.05	
7C		1	RF	**18.45		**15.2		**18.45	
7C		2	VFP	**23.85		**23.35		**23.85	
7C		3	RFP	**15.85		**15.6		**15.85	
7C		4	VH	**24.65		**24.65		**24.65	
7C		5	RH	**14.55		**14.55		**14.55	
7C		6	VHP	**24.2		**24.2		**24.2	
7C		7	RHP	**12.85		**12.8		**12.85	
7C		8	FC	3.4	3.4	3.4	4.45	3.85	5.5
7C		9	FXC	**8.35		**8.6		**8.6	
7C		A	FLC	**3.75		**3.75		**3.75	
7C		B	NR	**2.85		**2.85		**5.25	
7C		E	QAL	**5.4		**5.4		**5.4	
7C		F	QAR	**4.75		**4.75		**4.75	
7D		0	SIN	**25.55		**25.55		**25.55	
7D		1	COS	**26.4		**26.4		**26.4	
7D		2	TAN	**26.35		**26.35		**26.35	
7D		3	ASIN	**19.8		**19.8		**19.8	
7D		4	ACOS	**20.8		**20.8		**20.8	
7D		5	ATAN	**19.3		**19.3		**19.3	
7D		6	EXP	**24.75		**24.75		**24.75	
7D		7	ALOG	**25.25		**25.3		**25.3	
80	0		JER	1.0	1.1	1.25	1.55	1.3	4.3
80	1		JNER	1.0	1.1	1.25	1.55	1.3	4.3
80	2		JGER	1.0	1.1	1.25	1.55	1.3	4.3
80	3		JLSR	1.0	1.1	1.25	1.55	1.3	4.3
80	4		JOR	1.0	1.1	1.25	1.55	1.3	4.3
80	5		JCR	1.0	1.1	1.25	1.55	1.3	4.3
80	6		JPTR	*					
80	7		JBR	1.0	1.1	1.25	1.55	1.3	4.3
80	8		JR	1.0	1.1	1.25	1.55	1.3	4.3
80	9		JSR	*					
80	A		JKSR1	1.0	1.1	1.25	1.55	1.3	4.3
80	B		JKSR2	1.0	1.1	1.25	1.55	1.3	4.3
81			LJ	1.0	1.1	1.25	1.55	1.3	4.3
82	0		JE	1.0	1.1	1.25	1.55	1.3	4.3
82	1		JNE	1.0	1.1	1.25	1.55	1.3	4.3
82	2		JGE	1.0	1.1	1.25	1.55	1.3	4.3
82	3		JLS	1.0	1.1	1.25	1.55	1.3	4.3
82	4		JO	1.0	1.1	1.25	1.55	1.3	4.3
82	5		JC	1.0	1.1	1.25	1.55	1.3	4.3

Table B–1. CP Instruction Times (continued)

Code (Hex) OP a m	Mnemonic	Execution Time (Microseconds)					
		250 ns Cycle Semiconductor		350 ns Cycle Semiconductor		900 ns Cycle Core	
		Min.	Max.	Min.	Max.	Min.	Max.
82 6	JPT	*					
82 7	JB	1.0	1.1	1.25	1.55	1.3	4.3
82 8	J	1.0	1.1	1.25	1.55	1.3	4.3
82 9	JS	*					
82 A	JKS1	1.0	1.1	1.25	1.55	1.3	4.3
82 B	JKS2	1.0	1.1	1.25	1.55	1.3	4.3
83 0	JE	1.85	1.85	2.25	2.55	2.35	5.35
83 1	JNE	1.95	1.95	2.25	2.55	2.35	5.35
83 2	JGE	1.85	1.85	2.25	2.55	2.35	5.35
83 3	JLS	1.85	1.85	2.25	2.55	2.35	5.35
83 4	JO	1.9	1.9	2.25	2.55	2.35	5.35
83 5	JC	1.95	1.95	2.25	2.55	2.35	5.35
83 6	JPT	*					
83 7	JB	1.85	1.85	2.25	2.55	2.35	5.35
83 8	J	1.9	1.9	2.25	2.55	2.35	5.35
83 9	JS						
83 A	JSK1	1.85	1.85	2.25	2.55	2.35	5.35
83 B	JSK2	1.9	1.9	2.25	2.55	2.35	5.35
84	XJR	1.25	1.25	1.5	1.55	1.55	4.3
85	LJI	1.85	1.85	2.25	2.55	2.35	5.35
86	XJ	1.25	1.25	1.5	1.55	1.55	4.3
87	XJ	1.85	1.85	2.25	2.55	2.35	5.35
88	JLRR	1.1	1.1	1.25	1.55	1.3	4.3
8A	JLR	1.0	1.1	1.25	1.55	1.3	4.3
8B	JLR	1.9	1.9	2.25	1.55	2.35	5.35
8D	LJLM	1.65	1.65	1.85	2.15	2.15	5.2
8E	JLM	1.65	1.65	1.85	2.0	2.15	5.2
8F	JLM	2.5	2.5	2.7	3.0	3.2	7.1
90	JZR	1.25	1.25	1.5	1.55	1.55	4.3
91	LJE	1.0	1.0	1.25	1.55	1.2	4.3
92	JZ	1.25	1.25	1.5	1.55	1.55	4.3
93	JZ	1.9	1.9	2.25	2.55	2.35	5.35
94	JNZR	1.35	1.35	1.5	1.55	1.45	4.3
95	LJNE	1.1	1.1	1.25	1.55	1.35	4.3
96	JNZ	1.25	1.25	1.5	1.55	1.55	4.3
97	JNZ	1.9	1.9	2.25	2.55	2.35	5.35

Table B–1. CP Instruction Times (continued)

Code (Hex) OP a m	Mnemonic	Execution Time (Microseconds)					
		250 ns Cycle Semiconductor		350 ns Cycle Semiconductor		900 ns Cycle Core	
		Min.	Max.	Min.	Max.	Min.	Max.
98	JPR	1.25	1.25	1.5	1.55	1.55	4.3
99	LJGE	1.1	1.1	1.25	1.55	1.3	4.3
9A	JP	1.35	1.35	1.5	1.55	1.55	4.3
9B	JP	1.85	1.85	2.25	2.55	2.35	5.35
9C	JNR	1.35	1.35	1.5	1.55	1.55	4.3
9D	LJLS	1.0	1.0	1.25	1.55	1.3	4.3
9E	JN	1.35	1.35	1.5	1.55	1.55	4.3
9F	JN	1.85	1.85	2.25	2.55	2.35	5.35
A0	FSUR	**8.5		**8.5		**8.5	
A1	FSUI	**8.75		**7.85		**8.95	
A3	DSU	**8.75		**6.85		**8.95	
A4	FAR	**8.25		**8.25		**8.25	
A5	FAI	**8.5		**8.5		**8.7	
A7	FA	**8.5		**8.5		**8.7	
A8	FMR	**9.4		**9.4		**9.4	
A9	FMI	**9.65		**9.9		**10.55	
AB	FM	**9.65		**9.9		**10.55	
AC	FDR	**23.1		**23.4		**23.1	
AD	FDI	**23.45		**23.9		**24.25	
AF	FD	**23.45		**23.9		**24.25	
B0	LARR	1.8	1.8	2.0	2.0	2.05	3.45
B1	LARI	2.35	2.4	2.75	2.75	2.85	5.0
B3	LARM	3.25	3.3	3.5	3.7	3.9	6.9
B4	SARR	1.45	1.5	1.45	1.45	1.45	4.55
B5	SARI	2.0	2.0	2.0	2.0	2.0	5.2
B7	SARM	3.1	3.1	3.1	3.3	3.1	5.2
B8	MDR	**3.45		**3.45		**3.45	
B9	MDI	**3.95		**4.2		**4.85	
BB	MD	**3.95		**4.2		**4.85	
BC	DDR	**22.75		**22.75		**22.75	
BD	DDI	**23.0		**23.0		**23.65	
BF	DD	**23.0		**23.0		**23.65	

Table B–1. CP Instruction Times (continued)

Code (Hex) OP a m	Mnemonic	Execution Time (Microseconds)						
		250 ns Cycle Semiconductor		350 ns Cycle Semiconductor		900 ns Cycle Core		
		Min.	Max.	Min.	Max.	Min.	Max.	
C0	LLRS	0.7	0.7	0.7	0.7	0.7	0.9	
C1	LARS	0.9	0.9	0.9	0.9	0.9	0.9	
C2	LLRD	1.2	1.2	1.2	1.2	1.2	1.2	
C3	LARD	1.2	1.2	1.2	1.2	1.2	1.2	
C4	LALS	2.9	2.9	2.9	2.9	2.9	2.9	
C5	LCLS	0.7	0.7	0.7	0.7	0.7	0.9	
C6	LALD	3.3	3.3	3.3	3.3	3.3	3.3	
C7	LCLD	1.2	1.2	1.2	1.2	1.2	1.2	
C8	LSU	0.25	0.25	0.25	0.45	0.25	0.9	
C9	LSUD	0.7	0.7	0.7	0.7	0.7	0.9	
CA	LA	0.25	0.25	0.25	0.45	0.25	0.9	
CB	LAD	0.5	0.5	0.5	0.5	0.5	0.9	
CC	LL	0.25	0.25	0.25	0.45	0.25	0.9	
CD	LC	0.25	0.25	0.25	0.45	0.25	0.9	
CE	LMUL	1.2	1.2	1.2	1.2	1.2	1.2	
CF	LDIV	8.3	8.3	8.3	8.3	8.3	8.3	
D0	LIR	0.5	0.5	0.5	0.5	0.5	0.9	
D2	LMR	1.0	1.0	1.25	11.35	1.25	18.1	Note 1
D3	BSU	1.05	1.1	1.25	1.45	1.3	3.45	
D4	SIR	0.5	0.5	0.5	0.5	0.5	0.9	
D6	SMR	1.55	1.55	2.0	12.45	2.0	18.1	Note 1
D7	BA	1.0	1.1	1.25	1.45	1.3	3.45	
DB	BC	1.0	1.1	1.25	1.45	1.3	3.45	
DF	BCX	1.35	1.35	1.45	1.45	1.55	3.75	
E0	LPAR	2.8	2.8	***		***		
E1	LPAI	2.8	2.8	***		***		
E2	LPAK	2.8	2.8	***		***		
E3	LPA	2.8	2.8	***		***		
E7	LMAP	1.6	1.6	***		***		
E8	SMAP	2.4	2.4	***		***		
ED	LPLI	1.15	1.2	***		***		
EF	LPL	1.25	1.25	***		***		

Table B–1. CP Instruction Times (continued)

Code (Hex) OP a m	Mnemonic	Execution Time (Microseconds)					
		250 ns Cycle Semiconductor		350 ns Cycle Semiconductor		900 ns Cycle Core	
		Min.	Max.	Min.	Max.	Min.	Max.
F1	SPLI	1.35	1.35	***		***	
F3	SPL	1.4	1.4	***		***	

* Unable to measure with these procedures.

** The times shown for these instructions are the times measured with a legal set of operands; no attempt was made to tailor the operands for execution time. The execution time of these instructions is operand–dependent so the times shown should not be interpreted as minimum, maximum, or average. Execution time can vary significantly; for example, the time for a floating–point subtract (FSUR) was observed varying from 5.95 microseconds to 10.2 microseconds with different operands.

*** Optional memory management instructions could not be tested.

Note 1. The maximum times are for 16 registers, while the minimum times are for one register. The minimum and maximum times for the 250 ns memory are for one register.

APPENDIX C
APPLICABLE DOCUMENTS

Militarized Reconfigurable Computer

Specification ELEX–C–405 governs the MRC and specifies the following specifications, standards, handbooks and publications.

SPECIFICATIONS

MILITARY

MIL–S–901	Shock Tests, H.I. (High–Impact); Shipboard Machinery, Equipment And Systems, Requirements For
MIL–P–15024	Plates, Tags and Bands for Identification of Equipment
MIL–E–16400	Electronic, Interior Communication And Navigation Equipment, Naval Ship and Shore: General Specification For
MIL–E–17555	Electronic And Electrical Equipment, Accessories, And Repair Parts, Packaging And Packing Of
MIL–M–28787	Modules, Standard Electronic, General Specification For
MIL–P–28847	Processing Sets, Data, AN/UYK–20 (V) and AN/UYK–20X (V)

NAVAL ELECTRONIC SYSTEMS COMMAND

ELEX–P–351	Processor, Reconfigurable, Militarized

STANDARDS

MILITARY

MIL–STD–109	Quality Assurance Terms And Definitions
MIL–STD–130	Identification Marking Of U.S. Military Property
MIL–STD–167–1	Mechanical Vibrations Of Shipboard Equipment (Type I — Environmental And Type II — Internally Excited)
MIL–STD–189	Racks, Electrical Equipment, 19–Inch and Associated Panels
MIL–STD–454	Standard General Requirements For Electronic Equipment
MIL–STD–461	Electromagnetic Interference Characteristics Requirements For Equipment
MIL–STD–462	Electromagnetic Interference Characteristics, Measurement Of

MIL–STD–471	Maintainability Demonstration
MIL–STD–740	Airborne And Structureborne Noise Measurements and Acceptance Criteria of Shipboard Equipment
MIL–STD–781	Reliability Design Qualification And Production Acceptance Tests: Exponential Distribution Parts Control Program
MIL–STD–965	Parts Control Program
MIL–STD–1285	Marking Of Electrical And Electronic Parts
MIL–STD–1364	Standard General Purpose Electronic Test Equipment
MIL–STD–1378	Requirements For Employing Standard Electronic Modules
MIL–STD–1389	Design Requirements For Standard Hardware Program Electronic Modules
MIL–STD–1399, Section 300	Interface Standard For Shipboard Systems, Electric Power, Alternating Current (Metric)
MIL–STD–1399, Section 532	Interface Standard For Shipboard Systems, Cooling Water For Support Of Electronic Equipment (Metric)
MIL–STD–1472	Human Engineering Design Criteria For Military Systems, Equipment And Facilities

HANDBOOKS

MILITARY

MIL–HDBK–217	Reliability Prediction Of Electronic Equipment MRC–1
MIL–HDBK–237	Electromagnetic Compatibility/Interference Program Requirements
MIL–HDBK–241	Design Guide For EMI Reduction In Power Supplies

PUBLICATIONS

MILITARY

SE–610–PV–MMO–010/UYK–44(V)	Organizational Level Maintenance Manual, Data Processing Set, AN/UYK–44(V)
SE–610–PV–MMM–010/UYK–44(V)	Design Data Manual, Data Processing Set, AN/UYK–44(V), (Volume 1), Design Data
SE–610–PV–MMM–020/UYK–44(V)	Design Data Manual, Data Processing Set, AN/UYK–44(V), (Volume 2) Equipment Diagrams, Part 1

SE–610–PV–MMM–030/UYK–44(V)	Design Data Manual, Data Processing Set, AN/UYK–44(V), (Volume 3) Equipment Diagrams, Part 2
SE–610–PV–MMM–040/UYK–44(V)	Design Data Manual, Data Processing Set, AN/UYK–44(V), Device Descriptions and PAL Listings
SE–610–PV–MMM–050/UYK–44(V)	Design Data Manual, Data Processing Set, AN/UYK–44(V), (Volume 5), CP Microprogram Listing
SE–610–PV–MMM–060/UYK–44(V)	Design Data Manual, Data Processing Set, AN/UYK–44(V), (Volume 6) Miscellaneous Listings (ECW PROM, Sync and Async Serial Control PROM, IOC Microprogram, Bootstrap Bit, Math Table PROM)
SE–610–PV–MMM–070/UYK–44(V)	Design Data Manual, Data Processing Set, AN/UYK–44(V), (Volume 7) Maintenance Processor Listings
SE–610–PV–MMM–080/UYK–44(V)	Design Data Manual, Data Processing Set, AN/UYK–44(V), (Volume 8) Interconnect Diagrams (DPS)
SE–610–PV–MMM–090/UYK–44(V)	Design Data Manual, Data Processing Set, AN/UYK–44(V), (Volume 9) Interconnect Diagrams, Expansion Group

UNISYS

PX15156–1–1	Direct Fleet Support Maintenance Manual, Data Processing Set, AN/UYK–44(V) DFS Maintenance
PX15156–1–2	Direct Fleet Support Maintenance Manual, Data Processing Set, AN/UYK–44(V) OPT Program Listings, Part 1
PX15156–1–3	Direct Fleet Support Maintenance Manual, Data Processing Set, AN/UYK–44(V) OPT Program Listings, Part 2

Militarized Reconfigurable Processor

Specification ELEX–P–351A governs the MRP and specifies the following specifications, standards, handbooks and publications.

SPECIFICATIONS

MILITARY

MIL–E–17555	Electronic And Electrical Equipment, Accessories, And Repair Parts, Packaging And Packing Of
MIL–C–28754	Connector, Electrical, Modular, And Component Parts, General Specification For
MIL–M–28787	Modules, Standard Electronic, General Specification For
MIL–A–85232	Advanced Signal Processor (ASP) Analyzer — Detecting Set AN/UYS–1(V)

STANDARDS

MILITARY

MIL–STD–109	Quality Assurance Terms And Definitions
MIL–STD–188	Military Communication System Technical Standards
MIL–STD–202	Test Methods For Electronic And Electrical Component Parts
MIL–STD–781	Reliability Design Qualification And Production Acceptance Tests: Exponential Distribution
MIL–STD–1378	Requirements For Employing Standard Electronic Modules
MIL–STD–1389	Design Requirements For Standard Electronic Modules
MIL–STD–1397	Input/Output Interfaces, Standard Digital Data, Navy Systems
MIL–STD–1553	Aircraft Internal Time Division Command/Response Multiplex Data Bus

INTERNATIONAL

MILITARY

NATO–STD–4153 NATO STANAG	Serial Point–To–Point Input/Output Interface For NATO Naval Systems
NATO–STD–4156 NATO STANAG	Standard Specification For A Serial Data Interface For Synchronous Connections To A Data Network

DRAWINGS

MILITARY

NAVAL ELECTRONIC SYSTEMS COMMAND

0102–691	Connector, Electrical, Module Interface And Component Parts (Connector, Type IV, 100 Pin, Contact Tails On .050 Centers)
0102–694	Connector, Electrical, Module Interface And Component Parts (Connector, Type IV, 100 Pin, Contact Tails On .100 Centers)
0102–702	Connector, Type IV, 100 Pin, Right Angle
0102–704	Module Frame, 2A Center ISEM
0102–705	Connector Assembly, 2A Dip ISEM
0102–706	Module Frame, 2A T–Dip ISEM
0102–707	Connector Assembly, 2A T–Dip ISEM
0102–708	Connector Assembly, 2A Center ISEM
0102–709	Module Frame, 2A Offset ISEM
0102–710	Connector Assembly, 2A Offset ISEM

PUBLICATIONS

MILITARY

SE600–AD–MMO–010/OL–335(V)	Organizational Level Maintenance Manual, Data Processing Group, OL–335(V)/U
SE600–AD–MMM–010/OL–335(V)	Design Data Manual, Data Processing Group, OL–335(V)/U, (Volume 1), Design Data
SE600–AD–MMM–020/OL–335(V)	Design Data Manual, Data Processing Group, OL–335(V)/U, (Volume 2) Installation
SE600–AD–MMM–030/OL–335(V)	Design Data Manual, Data Processing Group, OL–335(V)/U, (Volume 3) Equipment Diagrams, Part 1
SE600–AD–MMM–040/OL–335(V)	Design Data Manual, Data Processing Group, OL–335(V)/U, (Volume 4) Equipment Diagrams, Part 2
SE600–AD–MMM–050/OL–335(V)	Design Data Manual, Data Processing Group, OL–335(V)/U, (Volume 5), Device Descriptions and PAL Listings
SE600–AD–MMM–060/OL–335(V)	Design Data Manual, Data Processing Group, OL–335(V)/U, (Volume 6), CP Microprogram Listing

SE600-AD-MMM-070/OL-335(V)	Design Data Manual, Data Processing Group, OL-335(V)/U, (Volume 7) Miscellaneous Listings (ECW PROM, Sync and Async Serial Control PROM, IOC Microprogram,Bootstrap Bit, Math Table PROM)
SE600-AD-MMM-080/OL-335(V)	Design Data Manual, Data Processing Group, OL-335(V)/U, (Volume 8) Maintenance Processor Listings

UNISYS

PX14874-0-2	Organizational Level Maintenance Manual, Expansion Adapter Group (EAG) OF-174(V)
PX15126-0-1	Organizational Level Maintenance Manual, Semiconductor Memory on SEM.

INDUSTRY

ELECTRONIC INDUSTRY ASSOCIATION

RS-232-C	Interface Between Data Terminal Equipment And Data Communication Equipment Employing Serial Binary Data Interchange

HANDBOOKS

MILITARY

MIL-HDBK-217	Reliability Prediction Of Electronic Equipment
MIL-HDBK-237	EMC Management Guide
MIL-HDBK-241	Design Guide For EMI Reduction In Power Supplies

Unisys Defense Systems is in business to satisfy customers' needs through on–time delivery and support of real–time, high–technology, low–cost, reliable and ruggedized information processing systems and products; to provide employees an environment in which they can use and develop their talents; to help make our communities a better place for everyone to live and work; to promote responsible government at all levels; and to generate growth and a fair return on company investment.

HEADQUARTERS OFFICE

Unisys Corporation
Defense Systems
P.O. Box 64525
St. Paul Minnesota 55164–0525
(800) 328–0204

UNISYS

Printed in U S America 11/87 PX 14309 Rev. A